GeoTORA Optimization

By

MAULIK TRIVEDI

10MCEC19

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2012

GeoTORA Optimization

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Maulik Trivedi

(10MCEC19)

Guided By

Prof. Gaurang Raval

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2012

DECLARATION

I, **Maulik Trivedi**, **10MCEC19**, give undertaking that the Major Project entitled **"GeoTORA Optimization"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Institute of Technology of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Maulik Trivedi

Certificate

This is to certify that the Major Project entitled "GeoTORA Optimization" submitted by Maulik Trivedi (10MCEC19), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Gaurang Raval
Guide, Associate Professor, PG-Coordinator(ICT),
Department of C.S.E.,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.S.N.Pradhan
Professor and PG-Coordinator(CSE),
Department of C.S.E,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. D. J. Patel
Professor and Head,
Department of C.S.E,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K. Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Abstract

Geocasting is a variation on the notion of multicasting. Geocasting is useful for sending messages to nodes in a specified geographical region. This region is called the geocast region. This work tries to improve a protocol, named Geo-TORA, for geocasting in mobile ad hoc networks. The modified GeoTORA protocol combines routing with clustering to implement geocasting. Thus, GeoTORA requires two phases for routing. First, it divides the group of nodes into regions and then anycasting is performed which is done by modifying the Temporally-Ordered Routing Algorithm (TORA)(unicast) routing protocol. Subsequently clusters are formed within the geocast region to deliver the messages to nodes within the region. This integration of TORA and Clustering can significantly reduce the overhead of control packet overhead, while maintaining reasonable accuracy. Also this Protocol works on large MANET with accuracy and delivers optimized output.

Acknowledgements

My deepest thanks to **Prof. Gaurang Raval**, Associate Professor, PG-Coordinator (ICT), Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad the Guide of the project that I undertook for giving his valuable inputs and correcting various documents of mine with attention and care. He has taken the pain to go through the project and make necessary amendments as and when needed.

My deep sense of gratitude to **Dr.S.N.Pradhan**, Professor and PG-Coordinator of Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for an exceptional support and continual encouragement throughout part one of the Major project.

I would like to thanks **Dr.Ketan Kotecha,** Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for his unmentionable support, providing basic infrastructure and healthy research environment.

I would also thank my Institution, all my faculty members in Department of Computer Science.

I would like to thank my colleagues Dhwani, Nikunj, Vinit and especially Maitrik and Tushar for being with me in any condition. And I am thanking them to keep me calm and cool every time.

I wish to thank Hemang Kothari, my friend cum senior student from IT- NU, for giving me all support and tremendous guidance for understanding fundamentals thing related to thesis.

I wish to thank Bhavesh, Nirav, Vishal and Darshan for providing me some unreachable re- sources and support.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom able to complete my dissertation work successfully.

- Maulik Trivedi
10MCEC19

Contents

List of Tables

List of Figures

Abbreviation Notation and Nomenclature

MANET .... Mobile Ad-hoc NETwork
DSR .... Dynamic source routing
DSDV .... Destination-sequenced distance vector
TORA .... Temporally ordered routing algorithm
AODV .... Ad hoc On demand Distance Vector
E2E .... End-to-End
NAM .... Network AniMator
NS .... Network Simulator
PDR .... Packet Delivery Ratio
TCL .... Tool Command Language

Chapter 1

Introduction

1.1 General

In MANET the set of wireless mobile nodes connected together to form temporary network in which the nodes are communicating with each other without centralized control. The nodes are free to move randomly and organize themselves arbitrarily. Hence the networks topology may change rapidly and unpredictably. The nodes that are within each others radio range can communicate directly, while remote nodes rely on their neighboring nodes to forward packets as a router. Routing is a core problem in networks for sending data from one node to another. Routing protocols works well in wired networks does not show the same performance in mobile ad hoc networks due to the rapid change of topology.

Efficiency of any adhoc network also depends on the management of its moving nodes. The basic idea for topology management is to elect the leader and all the moving nodes' information collected by it. This is basically referred as clustering in adhoc network. Also self organized and distributed approach is applicable for this. A highly dynamic topology is a distinguishing feature and challenge of a mobile ad hoc net- work. Links between nodes are created and broken, as the nodes move within the network. This node mobility affects not only the source and/or destination, as in a conventional wireless network, but also intermediate nodes, due to the networks' multihop nature. The resulting routes can be extremely volatile, making successful ad hoc routing dependent on efficiently reacting to these topology changes. In order to better understand this environment, a number of characteristics have been studied concerning the links and routes that make up an ad hoc network. Several network parameters are examined, including number of nodes, network dimensions, and radio transmission range, as well as mobility parameters for maximum speed and wait times. Here different clustering techniques are reviewed and summarized to manage the topology of the network which I have proposed as an enhancement to the optimize scheme which provides appropriate accuracy in routing with TORA protocol by managing the history about how many times each node has served as cluster head in the network. To manage and update the topology, Simulations are carried out using network simulator

In MANET, the topology of the network may change rapidly and unpredictably over the time(fig-1.1). The network in MANET is decentralized. A MANET includes many challenges and issues such as Dynamic topologies, Frequency of updates or network overhead, speed, routing. The routing protocol is required whenever the source needs to transmit and delivers the packets to the destination. Many routing protocols have been proposed for the mobile ad hoc network and classified as Proactive or Table Driven routing Protocol, Reactive or On Demand Routing Protocol.

A. Proactive or table-driven routing protocols: In proactive protocols, each node maintains individual routing table containing routing information for every node in the network. Each node maintains consistent and current up-to-date routing information by sending control messages periodically between the nodes which update their routing tables. The proactive routing protocols use link-state routing algorithms which frequently flood the link information about its neighbors. The drawback of proactive routing protocol is that all the nodes in the network always maintain an updated table. Some of the existing proactive routing protocols are DSDV and OLSR.

B. Reactive or On Demand Routing Protocol: In Reactive routing protocols, when a source wants to send packets to a destination, it invokes the route discovery mechanisms to find the route to the destination. The route remains valid till the destination is reachable or until the route is no longer needed. Unlike table driven protocols, all nodes need not maintain up-to-date routing information. Some of the most used on demand routing protocols are DSR and AODV.

C. Hybrid Routing Protocol: Hybrid routing protocol combines the advantages of both proactive and reactive routing protocols. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. Some of the existing hybrid protocols are ZRP and TORA.

Ad hoc formation of groups of nodes is necessary in the areas of military battlefield networks, sensor networks. These clusters must be formed rapidly and must scale in size while minimizing non-mission-critical messaging overhead. Each cluster has a node that behaves as a cluster head and is the primary point of communication with nodes in other clusters. Cluster heads therefore have the responsibility of routing messages with other cluster heads. The configuration must be stable but it must be able to rapidly reconfigure when the topology changes significantly. For example, when groups of nodes join or leave, or when the network is partitioned. There should be proper balance between the number of clusters and the size of each cluster. If there are too many clusters, then the routing overhead will be high. But if there are very large clusters, then the messaging overhead for cluster maintenance is likely to be high. This report contains a proposed GeoTORA protocol combines routing with clustering to implement geocasting and description of the simulator used to gather the results.

1.2 Motivation

The purpose of this project is to address the need of optimization in routing of wireless network with TORA Protocol.

Ad hoc Network has it's own challenging properties like limited dynamic topology, fast topology changes battery power, and all. There are many proactive routing protocols which dynamically finds the routes which may help to survive in dynamic environment. If the structure of the network is maintained in synchronization with the changes in it's topology, then it can be useful for the routing, broadcasting,throughput and delay faced in routing and load balancing in the Ad hoc network. Clustering is useful to maintain the topology of the network where the connections of the network are reduced by selecting proper nodes from the network which connects remaining nodes with it and thus provides the complete connectivity of the network.

1.3 Scope of Project

In this thesis, performance, reliability, and scalability of the adhoc mobile wireless network is improved by optimizing the GeoTORA protocol. In this, analysis is performed on the current trends. And after analysis, we have tried to eliminate the problems. And tested it by designing and then simulating the optimized TORA routing protocol.

1.4 Thesis Organization

The Thesis on " GeoTORA Optimization " has been divided in chapters as follows:

Chapter 2, Literature Survey and Important observations, presents the literature review. It provides overview of Temporary Ordered Routing Protocol, Geocasting, Clustering schemes.

Chapter 3, GeoTORA Protocol , includes the route creation and route maintenance by TORA routing protocol.

Chapter 4, Proposed Solution , includes solutions that by using Geocasting and Clustering Technique to achieved optimized protocol.

Chapter 5, Tools and Technique, provides the installation and mobility model, how to generate that model in NS.

Chapter 6, Simulation and Result Analysis, covers simulation of adhoc wireless network with different parameter using ns2 simulator. The chapter includes the analysis and discussion of the simulation and its graph.

Finally, In **Chapter 7** concludes this project with a summary, and provides possible directions for relevant future research.

Chapter 2

Literature Survey and Important Observations

2.1 Temporally Ordered Routing Algorithm

TORA [1] is a distributed routing protocol for mobile, multihop wireless networks. Its intended use is for the routing of IP datagrams within an autonomous system. The basic, underlying algorithm is neither a distance vector nor a link state; it is one of a family of algorithms referred to as ”link-reversal” algorithms. The protocol's reaction is structured as a temporally ordered sequence of diffusing computations, each computation consisting of a sequence of directed link reversals. The protocol is highly adaptive, efficient, and scalable, and is well suited for use in large, dense, mobile networks. In these networks, the protocol's reaction to link failures typically involves only a localized ”single pass” of the distributed algorithm. This desirable behavior is achieved through the use of a physical or logical clock to establish the ”temporal order” of topological change events. The established temporal ordering is subsequently used to structure (or order) the algorithm's reaction to topological changes.

TORA's design is predicated on the notion that a routing algorithm that is well suited for operation in this environment should possess the following properties:

- Executes distributedly.
- Provides loop-free routes.

- Provides multiple routes (i.e., to reduce the frequency of reactions to topologicalchanges, and potentially to alleviate congestion).
- Establishes routes quickly (i.e., so they may be used before the topology changes).
- Minimizes communication overhead by localizing algorithmic reaction to topological changes when possible (i.e., to conserve available bandwidth and increase scalability).

Routing optimality (i.e., determination of the shortest path) is of less importance. It is also not necessary (or desirable) to maintain routes between every source destination pair at all times. The overhead expended to establish a route between a given source destination pair will be wasted if the source does not require the route prior to its invalidation due to topological changes.

TORA is designed to minimize reaction to topological changes. A key concept in its design is that it decouples the generation of potentially far-reaching control message propagation from the rate of topological changes. Control messaging is typically localized to a very small set of nodes near the change without having to resort to a dynamic, hierarchical routing solution with its attendant complexity. TORA includes a secondary mechanism, which allows far-reaching control message propagation as a means of infrequent route optimization and soft-state route verification. This propagation occurs periodically at a very low rate and is independent of the network topology dynamics.

TORA is distributed in that nodes need only to maintain information about adjacent nodes (i.e., one-hop knowledge). It guarantees all routes are loop free, and typically provides multiple routes for any source-destination pair that requires a route. TORA is "source initiated" and quickly creates a set of routes to a given destination only when desired. Because multiple routes are typically established and having a single route is sufficient, many topological changes require no reaction at all. Following topological changes that do require reaction, the protocol quickly reestablishes valid routes. This ability to initiate and react infrequently serves to minimize communication overhead. Finally, in the event of a network partition, the protocol detects the partition and erases all invalid routes.

TORA (Temporally Ordered Routing Algorithm) is one of a family of link reversal algorithms [1] for routing in ad hoc networks. For each possible destination in the ad hoc network, TORA maintains a destination-oriented directed acyclic graph (DAG)[1].

Figure 2.1 illustrates how link reversal is performed in TORA. An arrow connecting a pair of nodes in this figure implies that the two nodes can communicate with each other. That is, the physical link between the two nodes is bidirectional. However, the TORA algorithm imposes a logical direction on the links, as illustrated in Figure 2.1(a) this figure shows the destination-oriented DAG with node G being the destination. Observe that, starting from any node in the graph, the destination G can be reached by simply following the directed links. Now assume that the link between nodes D and F breaks (perhaps because node F moves away from node D). Then, in the destination-oriented DAG, node D does not have any outgoing logical link. In response, TORA reverses logical direction of the (D,B) and (D,C) links, as shown in Figure 2.1,

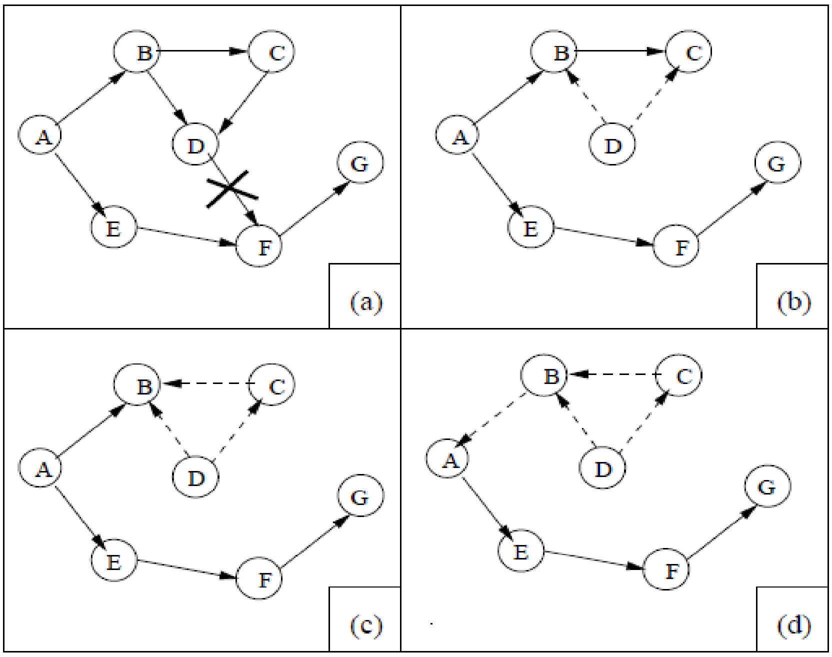

Figure 2.1: Route Maintenance in TORA: Route maintenance is required due to failure of the link between nodes D and F. In the figure, a link that has been reversed since the initial state is shown as a dashed line.

Now, node C does not have any outgoing logical link. In response, logical direction of link (B,C) is reversed, resulting in the graph in Figure 2.1(c). Now since node B does not have any outgoing logical link, the logical direction of link (A,B) is reversed, resulting in the destination-oriented DAG in Figure 2.1(d). In this state, each node (other than the destination G) has an outgoing logical link, and is able to reach the destination node G by following the directed links.

The main characteristic of TORA is the centralization of control messages in a very small set of near local nodes in which topological changes have been made. To achieve this property, nodes maintain routing information for the adjacent nodes for some interval. This protocol has three duties: route formation, route renovation and route cleaning. Route formation is performed with QRY and UPD. A route formation algorithm starts by determining a zero set for height of destination node and empty set for height of other nodes.

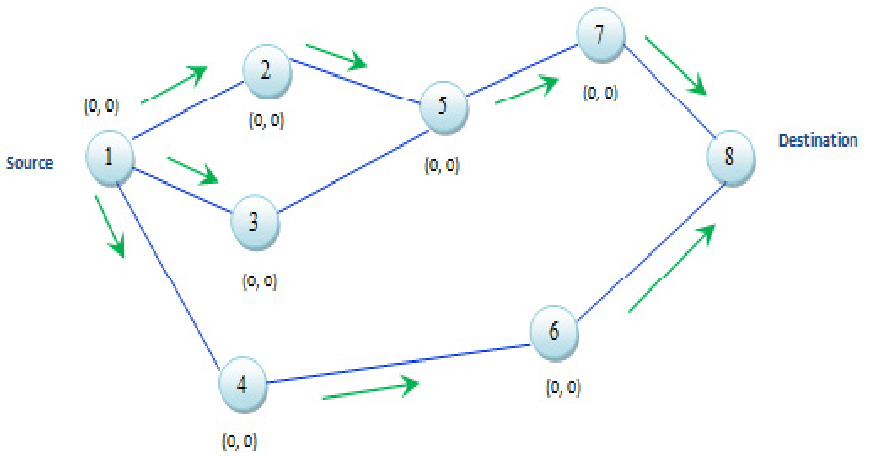

Figure 2.2: Broadcast QRY

The origin distributes a QRY packet in which destination node identifier is located. In this method, a non-circular graph is created from origin to destination. Figure 2.2 indicates a process of route formation in TORA. As shown in Figure 8 (a), node 5 receives the QRY packet from node 3 but it doesnt publish it because this packet has reached this node through node 2 previously. In Figure 2.3, the origin, i. e. , node 1 can receive the UPD packet from node 2 or node 3 but it doesn't receive it from node 4 of which the height is lower.

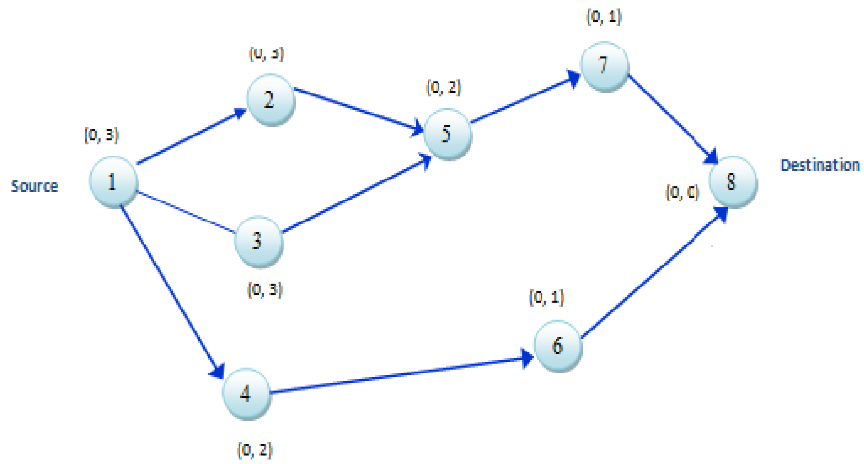

Figure 2.3: Distribute UPD packet

2.2 Problems

A wireless communications network needs to be implemented to support a fully interconnected collection of computers and/or computer networks when a wired network solution is not available or undesirable. The wireless devices in the network must support mobility with each other allowing for the dynamic discovery of wireless links to adjacent devices. The wireless devices must coordinate with each other to provide routing services over multiple wireless hops. The network routing process in the devices must search for and make use of any available path between the source of the IP traffic and the desired destination. To accomplish all of these requirements the routing process must possess the following attributes:

- Minimization of communication overhead
- Reactive (on-demand) and proactive (prior to demand) route establishment and maintenance
- Distributed execution
- Loop-free routing
- Multi-path routing

An advantage to TORA is that it supports multiple routes to any source-destination pair. Failure or removal of one node is quickly resolved without source intervention by switching to an alternate route. Unfortunately, there are drawbacks to TORA as well. The most glaring being that it relies on synchronized clocks among nodes in the network. While external time sources are present (GPS for example), it makes the hardware to support it more costly, and introduces a single point of failure if the time source became unavailable. TORA also relies on intermediate lower layers for certain functionality. It assumes, for example, that link status sensing, neighbor discovery, in-order packet delivery, and address resolution are all readily available.[1] [3]

2.2.1 Solution

Temporally Ordered Routing Algorithm by Park et al [1] is a distributed routing protocol for multi hop networks. The unique feature of this protocol is that it endeavours to localize the spread of routing control packets. The protocol is basically Lightweight Mobile Routing [13] protocols. It guarantees loop freedom,multiple routes and minimal communication overhead even in highly dynamic environments.The protocol targets at minimizing routing discovery overhead and in doing so prefers instant routes to optimal routes. The protocol supports source-initiated on-demand routing for networks with a high rate of mobility as well as destination oriented proactive routing for networks with lesser mobility. TORA maintains state on a per-destination basis and runs a logically separate instance of the algorithm for each destination. TORA assigns directional heights to links so as to direct the flow of traffic from a higher source node to a lower destination. The significance of these heights, which are assigned based on the direction of a link towards the destination, is that a node may only forward packets downstream but not upstream i.e. to another node that has a higher, undefined or unknown height. The height is represented by a quintuple (τ, oid, r, δ, i) where the first three values represent a reference level and the last two represent the delta with respect to the reference level. Each time a node loses its downstream link due to a link failure, a new reference level is computed using either a partial or full link reversal mechanism. The values in the height quintuple indicate the following:[3]

- τ Logical time of a link failure
- oid : Unique ID of the router that defined the ref. level

- r : Reflection indicator bit
- δ Propagation ordering parameter
- i : Unique ID of the router

TORA performs three basic functions: route creation, routemaintenance, and route erasure. Three control packets " query (QRY), update (UPD), and clear (CLR) " are used to accomplish these functions. Creating routes from various sources to the destination corresponds to establishing a sequence of directed links from each source to the destination. This is accomplished by maintaining a directed acyclic graph (DAG) rooted at the destination. A query/reply process with QRY and UPD packets is used for building the destination-oriented DAG. Figure 2.4 illustrates the process of route creation, with time increasing from Figure 2.4(a) to Figure 2.4(f). [1] [3]

A logically separate version of TORA is run for each destination to which routing is required. The following discussion focuses on a single version running for a given destination, j. TORA can be separated into three basic functions: creating routes, maintaining routes, and erasing routes. Creating a route from a given node to the destination requires establishment of a sequence of directed links leadingfrom the node to the destination. This function is only initiated when a node with no directed links requires a route to the destination. Thus, creating routes essentially corresponds to assigning directions to links in an undirected network or portionof the network. The method used to accomplish this is an adaptation of the query-reply process, which builds a Directed Acyclic Graph (DAG) rooted at the destination (i.e., the destination is the only node with no downstream links). Such a DAG will be referred to as a "destination-oriented DAG." "Maintaining route" refers to reacting to topological changes in the network in a manner such that routes to the destination are reestablished within a finite time-meaning that its directed portions return to a destination-oriented DAG within a finite time. TORA incorporates a new algorithm, in the same general class, that is more efficient in reacting to topological changes and capable of detecting a network partition. This leads to the third function, erasing routes. Upon detection of a network partition, all links (in the portion of the network that has become partitioned from the destination) must be marked as undirected to erase invalid routes.

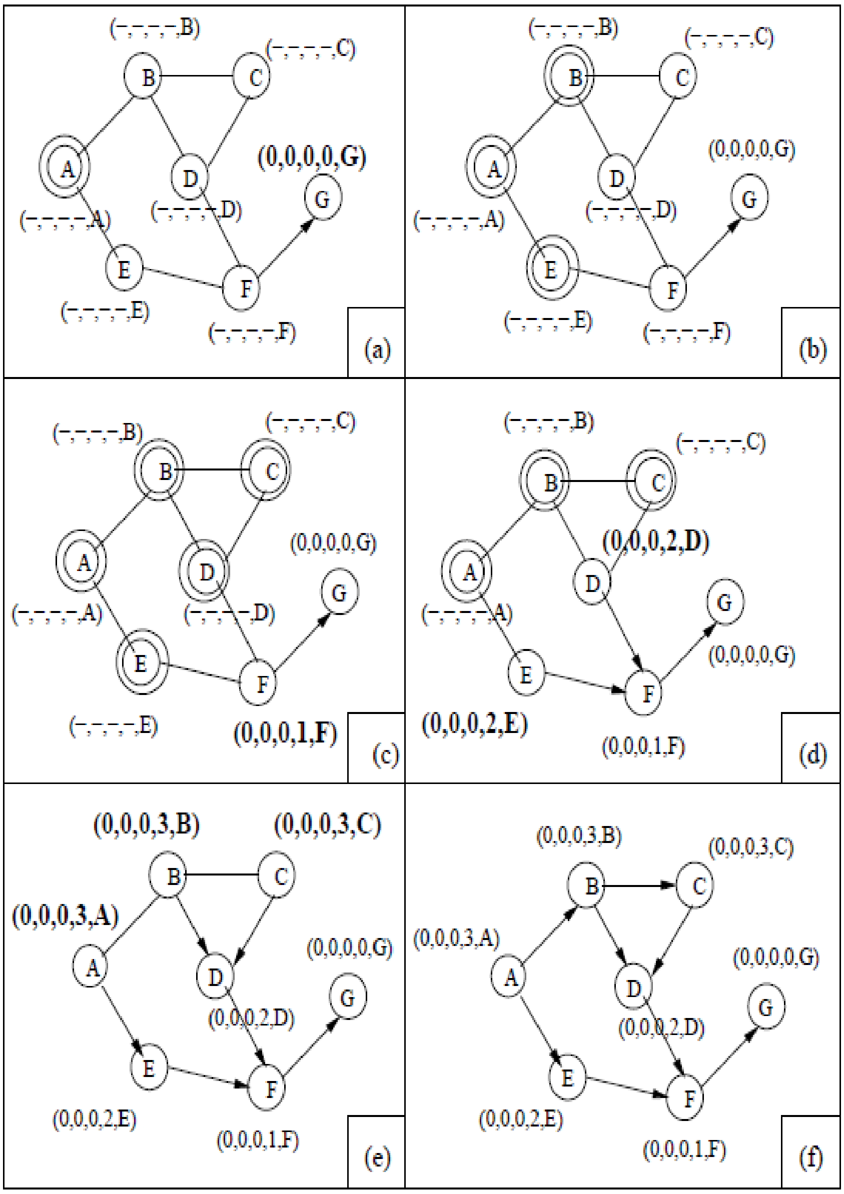

Figure 2.4: Route Creation Phase in TORA: In the figure, a circle around a node indicates that the routerequired flag is set. Arrows on each wireless link points from the higher height node to the lower height node. The height is depicted as a 5-tuple, as explained in the context.

TORA accomplishes these three functions through the use of three distinct control packets: query (QRY), update (UPD), and clear (CLR). QRY packets are used for creating routes, UPD packets are used for both creating and maintaining routes, and CLR packets are used for erasing routes.

2.3 Geocasting

Geocast[2] refers to the delivery of information to a group of destinations in a network identified by their geographical locations. It is a specialized form of multicast addressing used by some routing protocols for mobile ad hoc networks. see in figure-2.5

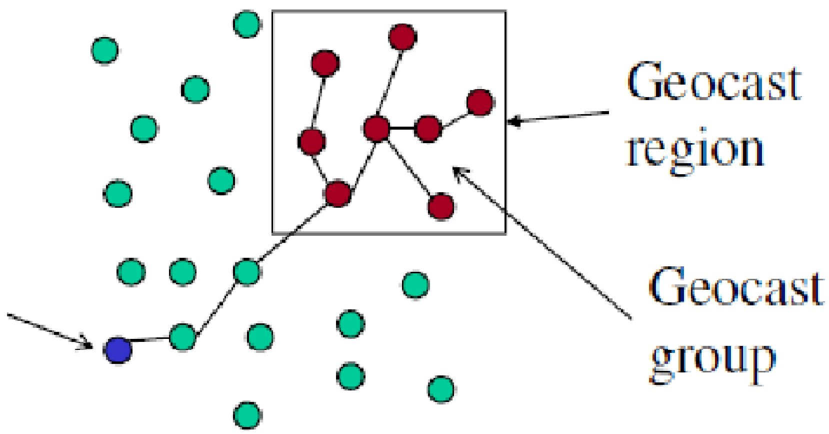

Figure 2.5: Geocasting

Geocasting has been proposed as a mechanism to deliver messages of interest to all hosts within a given geographical region [4]. In traditional multicasting, a host becomes a member of the multicast group by explicitly joining the multicast group. On the other hand, a host automatically becomes a member of a geocast group if its location belongs to the region specified for the geocast " this region is referred to as the geocast region [2]. Thus, the set of nodes in the geocast region is said to form the geocast group. For a node to be able to determine whether it belongs to a geocast group or not, the node should be able to know its own physical location. A node can determine its location, for instance, using the Global Positioning System (GPS). Imielinski and Navas [2] proposed approaches for geocasting in the internet. Ko and Vaidya [4] have proposed the use of geocasting in mobile ad hoc networks, and presented a protocol based on flooding.

A mobile ad hoc network consists of mobile hosts that communicate with each other over wireless links. In a mobile ad hoc network, typically, all mobile hosts behave as routers. A route between a pair of nodes in a mobile ad hoc network may go through several other mobile nodes. These routes can change when hosts change location. Therefore, there has been significant research on the development of (unicast) routing protocols for mobile ad hoc networks. On unicast routing in adhoc networks, there has been significant work on multicasting as well, and several approaches have been proposed [16] [17] [19]. The schemes for multicasting can be broadly divided into two types: flooding-based schemes and treebased schemes. Both approaches have their advantages and disadvantages. Flooding-based schemes do not need to maintain as much network state as the tree-based protocols. On the other hand, flooding-based schemes[18] can potentially deliver the multicast packets to a large number of nodes who do not wish to receive them (i.e., nodes which do not belong to the multicast group). Tree-based schemes tend to avoid this drawback of flooding-based schemes, at the cost of increased overhead in tree maintenance.

The concept of geocasting was introduced by Imielinski and Navas [2]. They also presented an architecture to implement geocasting in the internet. Ko and Vaidya [4] presented the so-called Location-BasedMulticast (LBM) algorithm that uses flooding to deliver a geocast packet. However, to reduce propagation of the flood, LBM limits the flood to a forwarding zone " the forwarding zone covers a subset of the network, and is determined based on the location of the sender and coordinates of the geocast region. Although the algorithm[4] is able to limit the flood of geocast packets to a relatively small region, still many nodes outside the geocast region tend to receive the geocast packet.

2.4 Routing challenges in wireless networks

The design of routing protocols in Wireless Network is influenced by many challenging factors[19]]. Which includes:

- **Ad hoc deployment:** Sensor nodes are deployed randomly. This requires that the system be able to cope with the resultant distribution and form connections between the nodes.

- **Energy consumption without losing accuracy:** Due to limited supply of energy performing computations and transmitting information in a wireless environment with reliability.

- **Computation capabilities:** Due to limited computing power of nodes and there- fore may not be able to run sophisticated network protocols.

- **Communication range:** Inter communication exhibits short transmission ranges. Therefore,it is most likely that a route will generally consist of multiple wireless hops.

- **Scalability:** The number of sensor nodes deployed in the sensing field may be in the order of hundreds or thousands or more.

- **Hardware constraints:** Consisting of many hardware components, a sensor node may be smaller than a cubic centimeter. These components consume extremely low power and operate in an unattended mode; nonetheless, they should adapt to the environment of the sensor network and function correctly.

- **Control overhead:** When the number of retransmissions in a wireless medium increases due to collisions, latency and energy consumption will also increase.

2.5 Clustering in Ad hoc Network

2.5.1 What is Clustering?

Clustering[7] is the method which divides the network into separate or overlapping zones. Clustering selects a set of nodes from the whole network such that from these nodes any of the node of the network is reachable and it does not require to maintain all the links between all the nodes in the network. The selected subset of the nodes lead to all the other nodes in the network. These leading nodes are called Cluster head. The cluster heads are either directly connected or connected via any other node. These intermediate nodes are called Gateways. Clustering is useful and provides following advantages.

- There is a back bone created considering only special nodes like cluster heads and gateways. So it requires less no of connections to be maintained.
- If cluster based routing is implemented then only cluster heads have to maintain route information.
- Mobility of node affects only when the movement of node is inter cluster.

2.5.2 Node Role

There are different node roles assigned to each node while running the clustering algorithm and according to the node role node may contribute in the management of the network.[6]

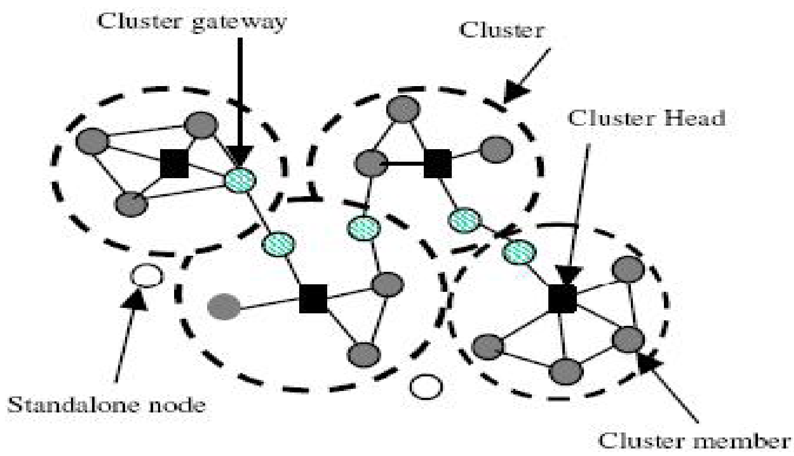

Figure 2.6: Cluster formation and node role

- **Cluster Head** They are the nodes selected by different clustering techniques to lead the network to create a back bone.This node serves as the head to the subset of ordinary nodes.
- **Gate Ways** These are some of the ordinary nodes which are connected to more than one clusters. Thus they connect two clusters and also contribute in creation of the back bone of network.
- **Ordinary node** The nodes which are connected directly or by k -hop to any of the cluster head in the network are called ordinary nodes.

Figure 2.6 illustrates node role and cluster formation in the network.

2.6 Survey of Clustering Techniques

2.6.1 Lowest - ID Technique

This is the simplest heuristic technique depended on the ID given to the node of the network. In this algorithm each node has the list of its direct neighbor. Then every node compares its own ID with ID of its neighbors. The node having the Lowest ID becomes the cluster head in the network. The method applied to find out the special nodes is simple but it has some drawbacks. If a node has lowest ID but the highest mobility in the network then it will not provide efficient balancing in the network.

Also it should be considered that the node having Lowest ID moves anywhere in the network It would always be selected as cluster head. So as the cluster head plays the special role in the network it may have more packet reception and delivery work to do. So in the Lowest ID, load balancing is not done due to bias ness of the node having lowest ID.

2.6.2 Max Degree Heuristic and K - CONID

The heuristic is the same as lowest ID but here it is considered that the higher connectivity provides efficient network connectivity. So while electing the cluster heads the parameter considered is the connectivity of each node i.e. degree of each node. Node knows the degree of itself and also the degree of its neighbors. Then the comparison is done between the node connectivity values gathered. The node having the highest connectivity becomes the cluster head and other work as ordinary nodes. Max Degree Provides less clusters compared to lowest ID but there may be ties between nodes who have the same degree, to become cluster heads.

K-CONID [10] has merged two concepts of lowest ID and maximum connectivity of node. In this a pair of ID and degree did=(d,id)is assigned to each node. First the degree value (d) of the pair is compared if any tie occurs then the node having lowest ID is selected as cluster head. This method is extended to k hop neighbor. The efficiency is tested by calculating the ratios of cluster head nodes to border nodes.

2.6.3 Max - Min D Hop clustering

The algorithm, Max-Min D -Cluster Formationin Wireless Ad Hoc Networks[9] elects the cluster head by considering the nodes which are at most d hop away from itself. Thus it may help to reduce the number of clusters in the network. Algorithm runs for 2d rounds and data structures used are simple i.e. two array WINNER(winning node ID of particular round) and SENDER(node which has sent the winning ID for particular round) at each node of size 2d.

Algorithm runs in four phases. First, the larger node ID is propagated through network in flood Max phase. Second, Lower Id is propagated in flood min. Third, Cluster head selection is done. Fourth back bone is created by linking the clusters.

Cluster head(CH) is decided if node gets its own ID back after 2d rounds of flooding. If this does not happen then node checks for the Node Pair(A node ID which at least occurs as WINNER in both 1st and 2nd round of flooding.). If any node pair does not exists then the node having maximum ID in the 1st d round of flooding is selected as cluster head. For the selection of gateways node checks that itself and its neighbors' cluster head are same or different? If they are same then node is not a gateway but if they are different then node becomes gateway.

Thus the algorithm provides less clusters and good cluster formation but it requires much flooding of messages in the network which may lead to congestion. The special nodes are selected as the local parameter node ID. if the mobility is also considered then the algorithm may give more efficient results. They have considered the highest connectivity, Node mobility and node ID to select the cluster heads.Thus this heuristic provides small overhead , fast convergence speed and good scalability for large scale networks.

2.7 Observation

As the survey of different algorithms[8] is done, all clustering algorithm mainly try to divide the network in such a way that maximum connectivity can be achieved in the presence of mobility of the node by handling the battery and energy constraints of the nodes of the network. The algorithms mainly have two major parts.

Method	**Details**
Hop count	1-hop.(Can save energy but creates more clusters),Multi hop. (Energy aware)
Node role	Cluster head based. Fully distributed. (but requires generally more rounds for clustering.)
Mobility Control	Have some Equations to find the mobility of nodes and based on that Clusters are formed.Generally node with lowest speed of mobility is selected as CH in the nbrs.
Passive clustering	Does not need any extra control messages for the cluster formation and maintanenence.

Table 2.1: Observation done from clustering scheme

- **Cluster formation**: Generally using graph theoretic approach of dominating sets and spanning trees to finding out the leading nodes in the network.
- **Cluster Maintenance**: This process is used to either periodically or on demand reform and select the cluster heads in the network.

The other main aspect to observe performance of any algorithm is the parameters used to form the clusters. Resulting clusters may be overlapping or non overlapping as per the methods used to form the cluster.

Chapter 3

GeoTORA Protocol

3.1 For GeoTORA Protocol

Now further elaborate on GeoTORA. Since GeoTORA is quite similar to TORA, Primarily highlight the differences between TORA and GeoTORA in this section. First, the route creation andmaintenance in GeoTORA is discussed, followed by the procedure for delivery of geocast messages using GeoTORA. Recall that, TORA Protocol, for each geocast group, GeoTORA maintains a single directed acyclic graph (DAG). This is similar to the DAG maintained by TORA, with the difference being that all nodes that belong to the geocast region have a ZERO height link between a pair of nodes is not assigned a direction if both nodes have ZERO height. This is unlike TORA, where only a single node (the destination) has ZERO height. [15]

3.1.1 Route Creation in GeoTORA

In order to deliver packets to the geocast group, a source should have a route to the given geocast region. To establish routes initially, GeoTORA uses a route creation process that is essentially identical to that for TORA, but with the difference noted above (i.e., all geocast members have ZERO height). Figure 3.1 provides an illustration for the process of geocast route creation in GeoTORA.

In Figure 3.1, the dotted circle represents the geocast region ” nodes G, H and I are within the geocast region (in this example, the set of nodes in the geocast region does not change). Figure 3.1(a) represents the initial state of the system. Since nodes G, H and I are within the geocast region, they set their height to ZERO. Any other

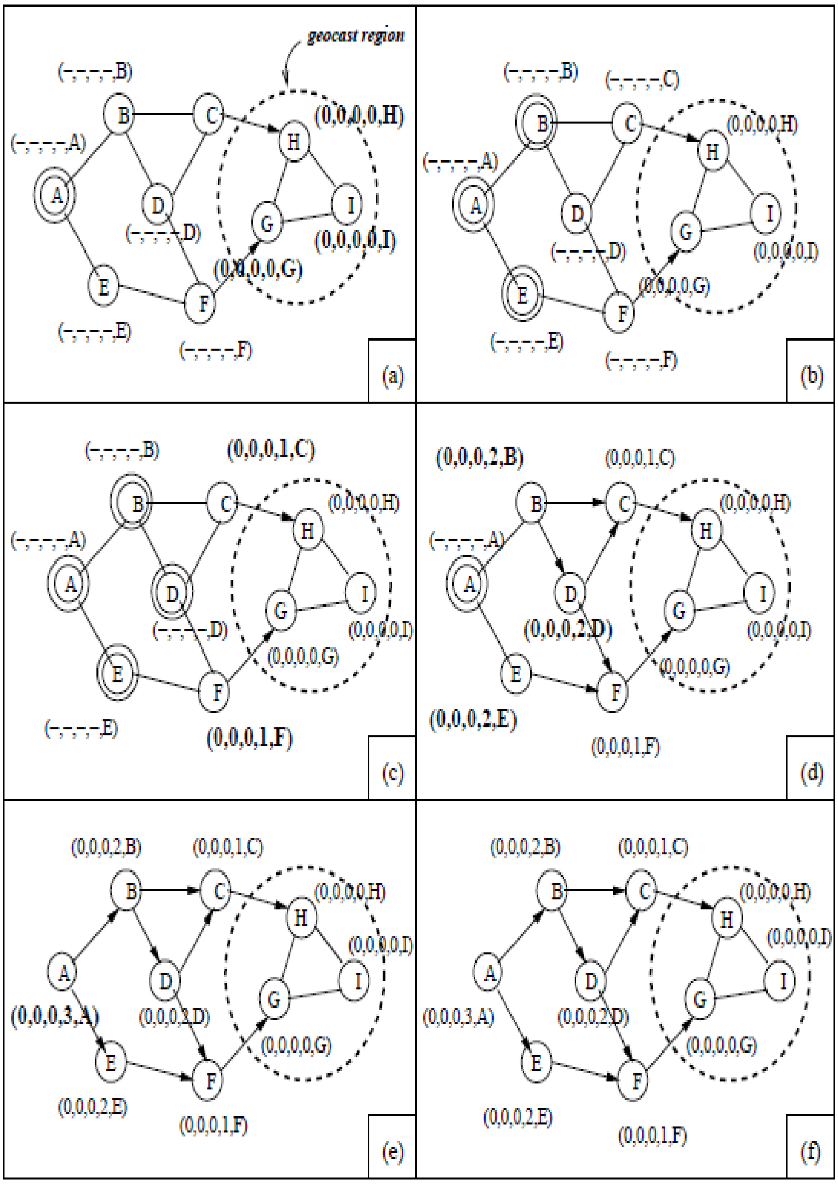

Figure 3.1: Geocast Route Creation in GeoTORA.

node, say i, sets its height to NULL " specifically, node isets its height to be (-,-,-,-,i). Note that links between two nodes with ZERO height are not assigned any direction; similarly links between two nodes with NULL height are also not assigned any direction. Nodes C and F (whose height is NULL) have links, respectively, to nodes H and G (whose height is ZERO). Therefore, the links (C,H) and (F,G) are assigned a direction " recall that NULL height is considered to be greater than any non-NULL height. Rules for route creation process in GeoTORA are identical to those described for TORA.

Assume that node A wishes to perform a geocast to the geocast group. Since node A does not have any outgoing link, it transmits a QRY packet to its neighbors, and sets its route-required flag. Note that in Figure3.1, a double circle around a node indicates that the route-required flag at that node is set. The QRY packet transmitted by node A reaches nodes B and E, and they, in turn, forward the packets to their neighbors, and also set the local route-required flag (refer Figure 3.1(b)). Nodes C and D receive the QRY message from node B, and node F receives from node E. In Figure 3.1(c), observe that nodes C and F have outgoing links to geocast group members, but node D does not. Therefore, only node D forwards the packets to its neighbors, and sets its route-required flag. On the other hand, on receiving a QRY, nodes C and F change their height from NULL to (0,0,0,1,C)and (0,0,0,,1,F), respectively, and send UPD message to their neighbors informing the new height. Response of a node on receiving an UPD message is identical to that in TORA. Figures 3.1(d) through (f) show evolution of the algorithm beyond the stage shown in Figure 3.1(c). At the end, as seen in Figure 3.1(f), a DAG is established wherein each geocast group member is a sink.[11]

3.1.2 Route Maintenance in GeoTORA

Now illustrate route maintenance in GeoTORA. In GeoTORA, the DAG may need to be modified when: (a) a link failure occurs, or (b) when a node enters or leaves the geocast region.[14] The GeoTORA routemaintenance procedure in response to link failures is similar to TORA. Figure 3.2 illustrates how the DAG is modified in GeoTORA in response to link failures, considering several link failure scenarios. Figure

3.2(a) shows the case where no maintenance reaction is taken by node D, as a result of breakage of link (D,F), since node D still has an outgoing link (D,C). Next, as shown in Figure 6(b), the link from node C to node H breaks. Now, node C is left without any outgoing links ” let us assume that the link failure occurred at time 1. Node C then updates its height using a new reference level representing the fact that node C has lost all downstream links at time 1. The new height of node C is (1,C,0,0,C), as shown in Figure 3.2(c). Node C also generates an UPD containing its new height and broadcasts the UPD to neighbor nodes ” the procedure for handling the UPD messages in GeoTORA is identical to TORA. Since node C increases its height, now node D also has no outgoing links in response, node D chooses height (1,C,0,-1,D)and sends an UPD to its neighbors. The new height chosen by nodes C and D results in the loss of the only outgoing link from node B. Therefore, node B lowers its height to (1,C,0,-2,B), to be lower than the current height of node D, and transmits an UPD containing its new height. The new height chosen by node B again causes reversal of the link between nodes A and B to, now, point to node A. However, node A still has another outgoing link (A,E), so no further action is needed. The final state of the DAG, after the failure of the link between nodes C and H, is shown in Figure 3.2(c).

Sometimes, a link failure causes a network partition, such that some nodes may not have any path remaining to any node in the geocast group. For instance, Figure 3.2(d) depicts the case where link between nodes F and G is broken. Now, assume that the time when failure occurred is 2. The reaction to this link failure is similar to the reaction following failure of link (C,H) in Figure 3.2(b). As a result of the failure of link (F,G), nodes F and E choose new height. The resulting state is shown in Figure 3.2(e). Observe that, before all the link failures, node A only had outgoing links (i.e., no incoming links in Figure 3.2(a)). Now, however, due to the new height chosen by node E in Figure 3.2(e), node A has no outgoing links remaining. Node A realizes that all its outgoing links are broken when it receives an UPD message from node E containing node E's new height. Subsequently, following the ”reflection” procedure as defined in TORA, the fact that A is partitioned from the geocast group is detected. Therefore, route erasure phase is initiated. Details of the route erasure phase are not illustrated here for brevity however, note that the procedure is identical to TORA.

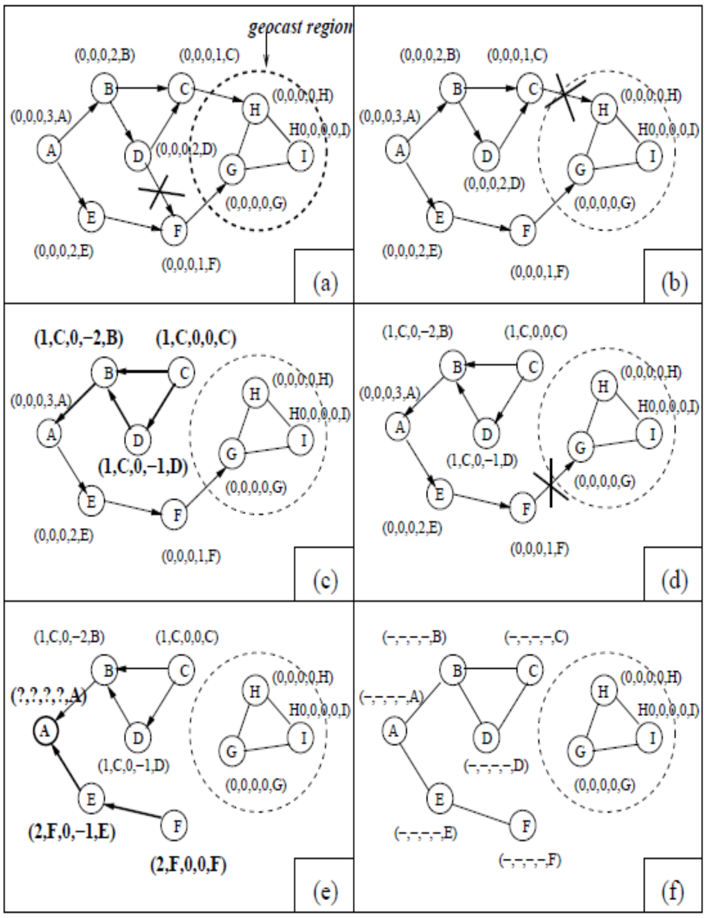

Figure 3.2: Geocast RouteMaintenance in GeoTORA: Three different scenarios of link failure

Figure 3.2(f) shows the network state after route erasure process has been completed. Until a new route to the geocast region is detected, a source that is partitioned from the geocast group is not able to send geocast data packets.

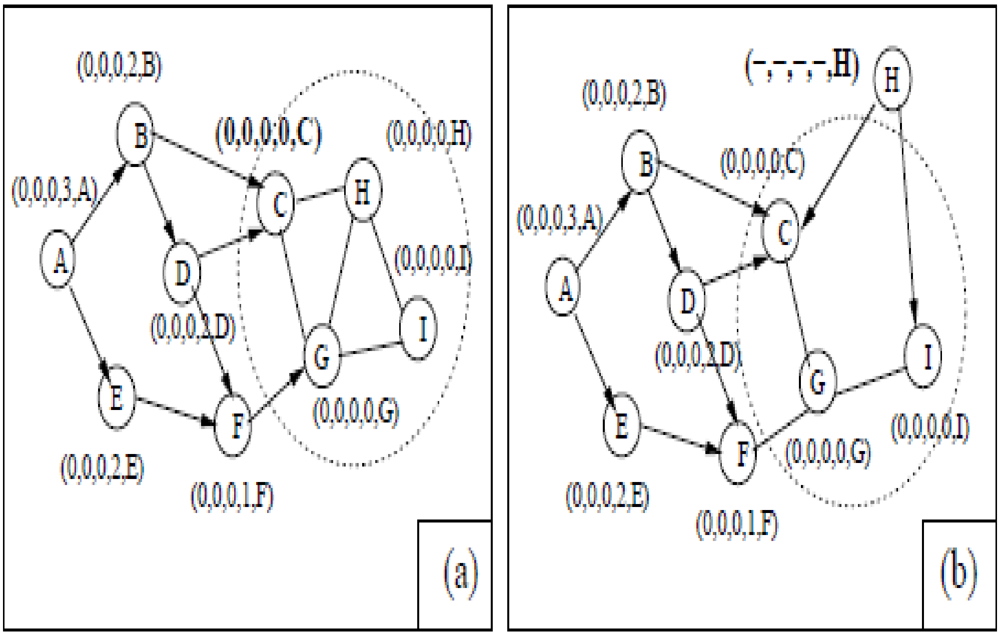

Figure 3.3: Route Maintenance in GeoTORA for handling dynamic change of geocast group

Figure 3.3 illustrates how GeoTORA handles geocast group membership changes. Consider Figure 3.1(f) as an example network. In Figure 3.1(f), when node C moves into the geocast region and becomes a sink, it simply updates its current height to be ZERO, and then a UPD is transmitted by node C to inform its new height to its neighbors. The resulting state is shown in Figure 3.3(b). Now, let us assume that node H leaves the geocast group by moving out of the geocast region. In this case, the height of node H is set to NULL. Note that a NULL height is considered greater than any non-NULL height. Therefore, undirected links (H,C) and (H,I) in Figure 3.3(b) now have logical directions from node H to C and from H to I, respectively, as shown in Figure 3.3(c).

3.1.3 Geocast Packets Delivery

Geocast delivery using GeoTORA consists of phases: anycasting phase and local packet sending phase, as discussed below.

Anycasting Phase: When a node wishes to send a packet to the geocast group, it forwards the packet on any of its outgoing links. Each node that receives the packet forwards the packet on an outgoing link. Provided the source node is not partitioned from the geocast group, the packet eventually reaches one member of the geocast group.

Local packet sending Phase: Once a hello packet is delivered to one node in the geocast group (by the anycasting phase above), that node initiates local broadcast of the hello packet. The purpose of local broadcast, described below, is to deliver the hello packet to the remaining geocast group members. The node, say X, that initiates the flood, tags the specification of the geocast region to the hello packet, and broadcasts it to its neighbors. Any node, say Y, that receives the hello packet verifies whether it is within the region whose specification is tagged to the hello packet. If node Y is outside the region, then it simply discards the hello packet. On the other hand, if node Y is within the tagged region, then node Y broadcasts the hello packet to its neighbors. Caution is taken to ensure that a given node would not broadcast the same hello packet more than once.

3.2 Challenges

The overhead of TORA consists of data packets as well as control packets (QRY, UPD, and CLR) used to create and maintain routes.

Also, Few more overhead packets consider when creating a cluster(geo-region) for routing in small area. The overhead increases with increasing node mobility (i.e., decreasing pause time) for all schemes. However, note that the main reason for increasing overhead in GeoTORA is the control packets, not the data packets. With low mobility rate in GeoTORA, routes for forwarding packets are likely to be fixed and, therefore, the number of control packets to maintain the routes is relatively small. As mobility rate goes up, the cost for a route maintenance process, i.e., number of QRY and UPD packets, also becomes higher.

Chapter 4

Proposed Solution

4.1 Proposed Solution

We suggest the following approaches to reduce overhead of control packets. Firstly we assume that nodes are stationary node, for that we can modify setdest utility available in network simulator.

Secondly, when the controls packets communicate between source node to destination node through intermediate nodes, there is a problem of packets being dropped due to collisions since packets are routed on same channel at same time. This can be solved by adding timer functionality, where in these packets are timely broadcast.

To further reduce the overhead of control packets, small regions of topology and clustering scheme is employed within these small regions. The intra-region control packets overhead is further reduced by employing routing between cluster head to cluster head of different regions.

Chapter 5

Tools and Technique

5.1 Tools:

5.1.1 Network Simulator

Network simulator is tool used to stimulate different network scenarios. We can build ns either from the the various packages (Tcl/Tk, otcl, etc.), or We can download an 'all- in-one' package. I start with the all-in-one package, especially if we're not entirely sure which packages are installed on your system, and where exactly they are installed. The disadvantage of the all-in-one distribution is the size, since it contains some components that we don't need anymore after we compiled ns and nam. It's still good for first tests, and we can always switch to the single-package distribution later.[21]

Network simulator is tool used to stimulate different network scenarios. Using this we need to achieve following goals.

- Optimization on Energy, Lifetime of network and packet delivery.
- Analysis of overhead delivery for different scenario
- Find out the gain achieved by traffic patterns.

NS is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS-2 is written in C++ and an Object

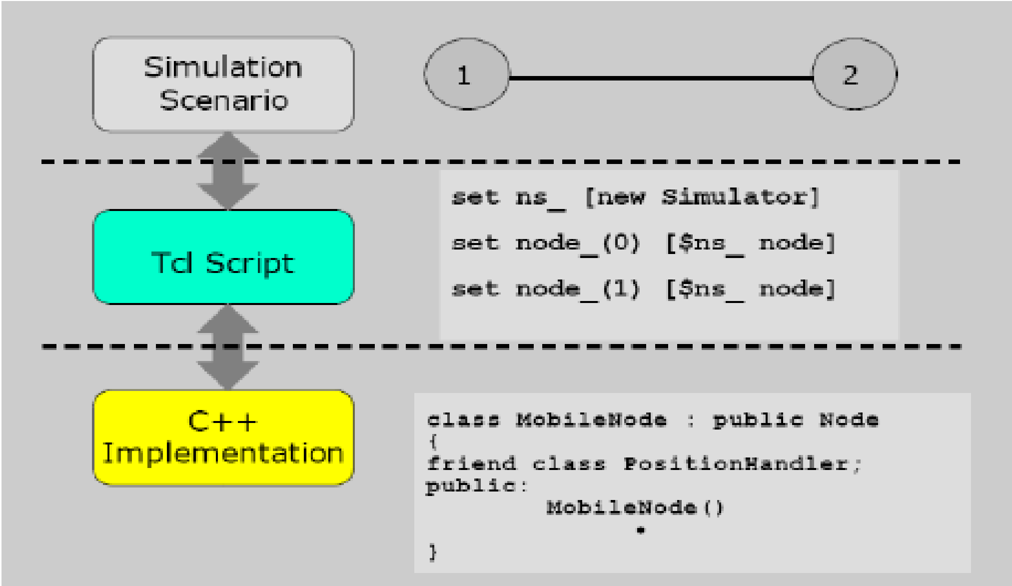

Figure 5.1: NS-2 Overview

oriented version of Tcl called OTcl. NS is not a polished and finished product, but the result of an on-going effort of research and development. In particular, bugs in the software are still being discovered and corrected.

5.1.2 Resource requirements for NS-2

To build NS we need a computer and a C++ compiler. ns-2 is designed to run from on most UNIX based operating systems. It is possible to run ns-2 on Windows machines using Cygwin. If you don't have a UNIX install, you can also use a virtual linux machine and run that under Windows. Simple scenarios should run on any reasonable machine, but very large scenarios benefit from large amounts of memory.
NS is fairly large. The allinone package requires about 320MB of disk space to build. Building ns from pieces can save some disk space.

5.1.3 NS-2 Components

Ns-allinone is a package which contains required components and some optional components used in running ns. The package contains an "install" script to automatically configure, compile and install these components. After downloading, run the install script.
Currently the package contains:

- Tcl release 8.5.10 (required component)
- Tk release 8.5.10 (required component)
- Otcl release 1.14 (required component)
- TclCL release 1.20 (required component)
- Ns release 2.35 (required component)
- Nam release 1.15 (optional component)
- Xgraph version 12 (optional component)
- CWeb version 3.4g (optional component)
- SGB version 1.0 (optional component, builds sgblib for all UNIX type platforms)
- Gt-itm gt-itm and sgb2ns 1.1 (optional component)
- Zlib version 1.2.3 (optional, but required should Nam be used).

5.1.4 NS-2 installation

NS-2 version 2.34 is available on the following link[21]:
http://nsnam.isi.edu/nsnam/index.php/Downloading_and_installing_ns-2

5.2 Installation Procedure

5.2.1 Installation of FEDORA 10

The various platforms which supports Network Simulator(NS) are Linux,Solaris, Windows XP using Cygwin. We have used Fedora core 10 and 12 (Linux environment) because Fedora is open source. Following are the steps to install Fedora 10 OS.

- Remove data from one of the drives where fedora is to be installed.
- Load the CD in drive.
- Skip for disc checking.
- Then select the English US language.

- Select Custom layout for partitioning hard disk and click next.
- Where we want to install, make that disk space free first (at least 5GB). The drives are named as dev/sda1 (which is C drive in the system) and so on for the other drives.
- Then select New for that free space.
- Give a mount point as '/'(root) and select ext3, give a space equal to total free space minus 2*RAM. This space is given in MB.
- Then make it ok.
- Then on the remaining free space select New and then select file format as Swap Give space to swap 2*RAM size. This space is to be given in MB.
- Select grub boot loader and make windows or other OS as by default.
- Select all the software if needed.
- Give the computer name.
- Select region as Asia/Calcutta.
- Enter Root password.
- Then installation will start (1103 packet should be installed for proper functioning of fedora).

5.2.2 Installation of Network Simulator

Network Simulator 2.27

Step 1: Prepare necessary files for installation:

1. NS-2.27 package: ns-allinone-2.27.tar.gz
2. Patch for compiling NS-2.27 with GCC 4.1.x: ns-2.27-gcc410.patch
3. MIT's LEACH extension: mit.tar.gz

4. LEACH's Makefile patch: leach makefille-2.27.patch Download this file from: http://www.tekno.chalmers.se/yusheng/reports/ns-2.27-gcc410.patch

Step 2: Download NS-2.27, apply ns-2.27-gcc410.patch, and install it
Download NS-2.27 from

1. http://www.isi.edu/nsnam/dist/ns-allinone-2.27.tar.gz or
2. http://www.internetworkflow.com/downloads/ns2leach/ns-allinone-2.27.tar.gz
3. tar zxvf ns-allinone-2.27.tar.gz
4. patch -p0 < ns-2.27-gcc410.patch
5. cd ns-allinone-2.27/
6. ./install

Step 3: Set the environment variables to make NS-2.27 works:

1. cd∼
2. gedit /.bashrc
   #LD LIBRARY PATH
   OTCL LIB= /ns-allinone-2.27/otcl-1.8
   NS2 LIB= /ns-allinone-2.27/lib
   X11 LIB=/usr/X11R6/lib
   USR LOCAL LIB=/usr/local/lib
   export
   LD LIBRARY PATH=$LD LIBRARY PATH:$OTCL LIB:$NS2 LIB:
   $X11 LIB:$USR LOCAL LIB
   #TCL LIBRARY
   TCL LIB= /ns-allinone-2.27/tcl8.4.5/library
   USR LIB=/usr/lib
   export TCL LIBRARY=$TCL LIB:$USR LIB
   #PATH

PATH=$PATH: /ns-allinone-2.27/bin: /ns-allinone-2.27/tcl8.4.5/unix: /ns-allinone-2.27/tk8.4.5/unix

3. source /.bashrc

Step 4: Download, copy, and extract MIT's LEACH extension

1. Download files from: http://www.internetworkflow.com/downloads/ns2leach/mit.tar.gz
2. cp mit.tar.gz /ns-allinone-2.27/ns-2.27/
3. cd /ns-allinone-2.27/ns-2.27/
4. tar xzvf mit.tar.gz
5. rm mit.tar.gz

Step 5: Modify NS-2 source code

1. gedit /ns-allinone-2.27/ns-2.27/mac/wireless-phy.cc
   Goto line 59, that is after line 58:
   #define max(a,b) (((a)<(b))?(b):(a))
   Insert:
   #define min(a,b) (((a)>(b))?(b):(a))

Step 6: Add environment variables for LEACH extension

1. gedit /.bashrc
   Goto line 59, that is after line 58:
   export RCA LIBRARY=NS=mit=rca
   exportuAMPS LIBRARY =NS/mit/uAMPS
2. source /.bashrc

Step 7: Download and apply patch for Makefile.vc, edit Makefile and recompile NS-2.27 with LEACH extension

1. patch -p0 < leach makefile-2.27.patch

2. gedit Makefile
   Add -DMIT uAMPS to the DEFINE list Add -I./mit/rca -I./mit/uAMPS to the INCLUDE list Add the following just prior to the line gaf/gaf.o mit/rca/energy.o mit/rca/rcagent.o mit/rca/rca-ll.o mit/rca/resource.o mac/mac-sensor-timers.o mac/mac-sensor.o mit/uAMPS/bsagent.o make clean make

Step 8: Run default LEACH Go to the root¿ns folder using command prompt And then run ./leach-test

5.2.3 GNUPlot

Gnuplot is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three-dimensional plots in many different formats. It is designed primarily for the visual display of scientific data. gnuplot is copy- righted, but freely distributable; you don't have to pay for it. GNUPlot provides following functionalities.

a. Plotting two-dimensional functions and data points in many different styles.

b. Plotting three-dimensional data points and surfaces in many different styles.

c. Algebraic computation in integer , oat and complex arithmetic.

d. Support for a large number of operating systems, graphics file formats and output devices.

e. TEX-like text formatting for labels, titles, axes, data point.

f. Extensive on-line help.

Open the terminal and Type the following commands

cd /Desktop/gnuplot-4.2.6

. /configure

make

make install

Installation of gnuplot is complete.

To plot the graphs write the .txt file which containts all the information about axis,label etc.

5.2.4 AWK

Awk has two faces: it is a utility for performing simple text-processing tasks, and it is a programming language for performing complex text-processing tasks.The two faces are really the same, however. Awk uses the same mechanisms for handling any text-processing task, but these mechanisms are exible enough to allow useful Awk programs to be entered on the command line, or to implement complicated programs containing dozens of lines of Awk statements. The Awk text-processing language is useful for such tasks as:

- Tallying information from text files and creating reports from the results.
- Adding additional functions to text editors like "vi".
- Translating files from one format to another.
- Creating small databases.
- Performing mathematical operations on files of numeric data.

Awk statements comprise a programming language. In fact, Awk is useful for simple, quick-and-dirty computational programming. Anybody who can write a BASIC program can use Awk, although Awk's syntax is difierent from that of BASIC. Anybody who can write a C program can use Awk with little dificulty, and those who would like to learn C may find Awk a useful stepping stone.

Awk is not really well suited for extremely large, complicated tasks. It is also an "interpreted" language that is, an Awk program cannot run on its own, it must be executed by the Awk utility itself. That means that it is relatively slow, though it is eficient as interpretive languages go, and that the program can only be used on systems that have Awk. There are translators available that can convert Awk programs into C code for compilation as stand-alone programs, but such translators have to be purchased separately.

5.3 Simulation Overview

A typical simulation with ns and the mobility extension is shown Figure 6.3. Basically it consists of generating the following input files to NS: A scenario file that describes the movement pattern of the nodes, communication file that describes the traffic in the network[21]. These files can be generating completely randomized movement and

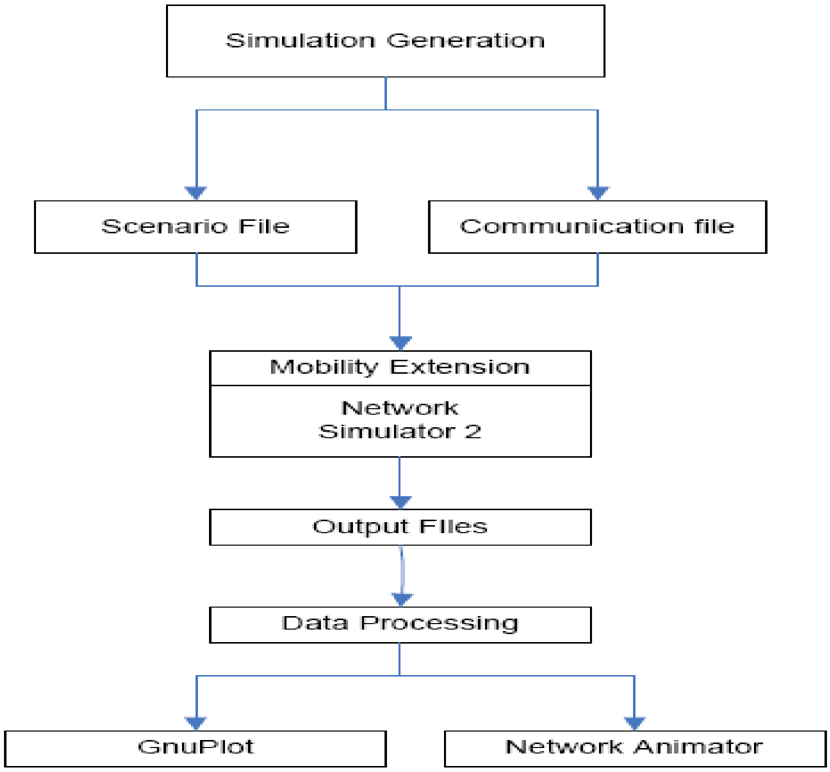

Figure 5.2: Simulation Overview

communication patterns with a script. These files are then used for the simulation and as a result from this, a trace file is generated as output. Prior to the simulation, the parameters that are going to be traced during the simulation must be selected. The trace file can then be scanned and analyzed for the various parameters that we want to measure. This can be used as data for plots with for instance Gnuplot.

5.4 Needed Changes

There are some files that are to be modified when one adds its new protocol in Network Simulator.

Packet.h Since we add new protocol packet so it must be added in the common/paket.h file

cmu-trace.h/cc A trace object is used to write wanted information of a packet every time it is received, sent or dropped. For that we have to write function in the cmu-trace.h file and its implementation is in cmu-trce.cc file.

ns-packet.h In this file we have to add the name of the new protocol in this case it is TORA.

ns-lib.tcl We need a add procedure for creating a node. We have to create wireless node with TORA as routing protocol. The procedure node from tcl file calls the createwireless ndoe procedure. So we have to write that procedure in ns.lib.tcl file.

Makefile The last step after adding all the required function into the files, it should be compiled. The compilation of file is done by the make command. Before making the make, the object files are to be added into the object list of the Makefile. After adding and running the make command, if no errors comes than the protocol is successful and can be used for the simulation purpose.

Chapter 6

Simulation and Result Analysis

6.1 Evaluation Metrics:

This chapter describes performance of topology based routing protocols along their effectiveness and underlying limitations within certain density levels of city and highway scenarios. Ad hoc routing protocols: AODV, DSR, DSDV and TORA are separately incorporated by simulation models with the precise parametric values of defined scenes and simulations are carried out with NS-2.34. The evaluative metrics for the examination of these protocols are measured by Packet Delivery Ratio (PDR), Average End-to-End delay, Dropped packets, Overhead respectively. Mainly concentrate on TORA Protocol.

The following metrics are chosen for evaluating the protocols:

Packet Delivery Ratio (PDR): This metric gives the ratio of the total data packets successfully received at the destination and total number of data packets generated at source.

Average End-to-End Delay (E2E Delay): It is the calculation of typical time taken by packet (in average packets) to cover its journey from the source end to the destination end. In other words, it covers all of the potential delays such as route discovery, buffering processes, various in-between queuing stays, etc, during the entire trip of transmission of the packet. The classical unit of this metric is millisecond (ms). For this metric, lower the time taken, more privileged is the routing protocol.

Dropped TCP packets: It defines a total number of TCP packets dropped during transmission of packet from source end to destination end. For this metric lower the dropped packet, more privileged the routing protocol is considered.

Overhead: This is calculated as a ratio between the average number of packets received by the node and the average number of packets generated by the node. Higher the overhead the higher is the power usage since power is unnecessarily spend in forwarding the redundant packets received by the node.

Scalability: Scalability of the protocol is calculated on the basis of the above three metrics. The three metrics are carefully examined as the number of nodes in the network increases. The number of nodes deployed in the sensing area are increased. Any routing scheme must be able to work with this huge number of nodes.

Before running simulation, Has to use the tools to generate the traffic and scenario patterns.

- `Traffic pattern (change the path to ~ns\indep-utils\cmu-scen-gen)`

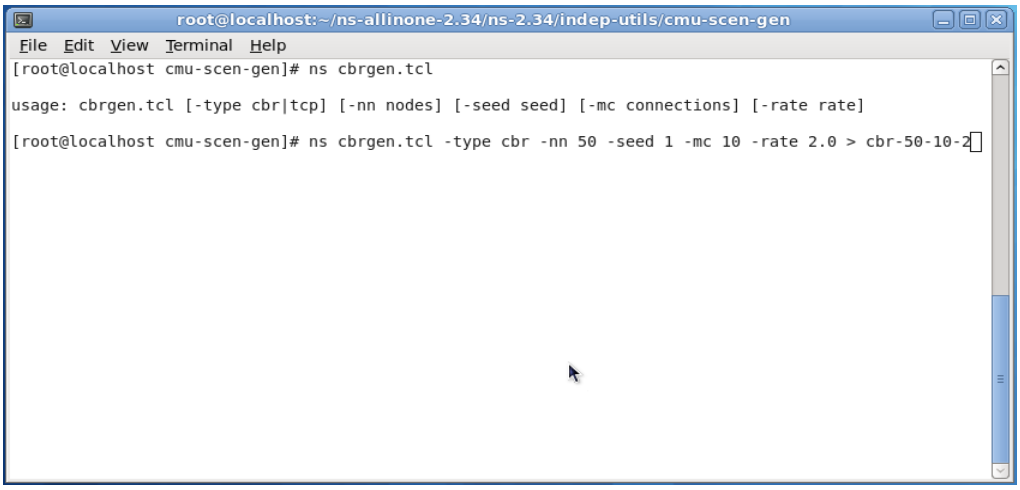

Figure 6.1: Generate Traffic and Scenario

In figure 6.1, traffic models were generated for 50 nodes with CBR traffic sources, with maximum connections of 10 at a rate of 8kbps.

(-rate 2.0: in one second, 2 packets are generated. The packet size is 512 byte. Therefore the rate is 2*512*8=8kbps)

- `Scenario pattern (change the path to ~ns\indep-utils\cmu-scen-gen)`

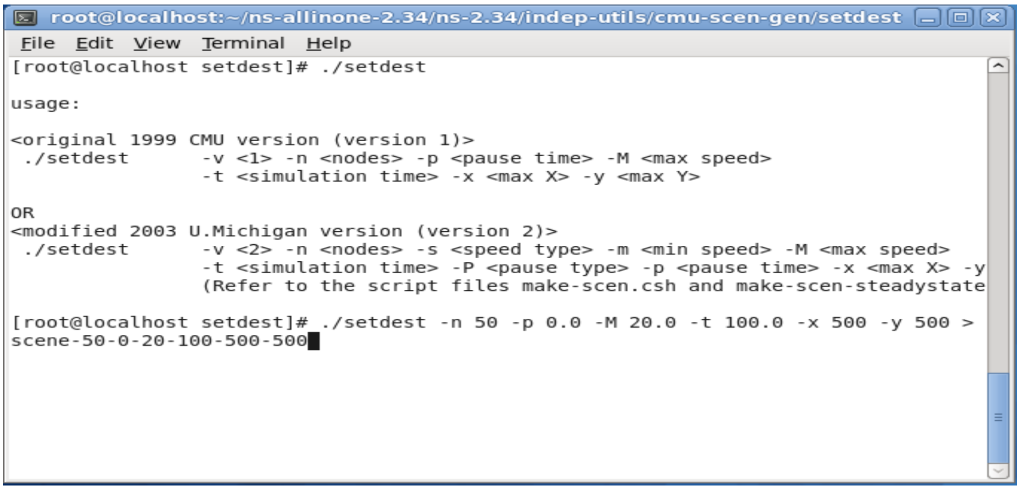

Figure 6.2: Scenario Pattern

In figure 6.2, mobility models are created for the simulations using 50 nodes, with time of 0 seconds, maximum speed of 20m/s, topology boundary of 500x500 and simulation time of 100 secs.

After creating the traffic and scenario pattern files, copy them to the place where you put the adhoc.tcl.

Start Running Simulation shows in figure 6.3

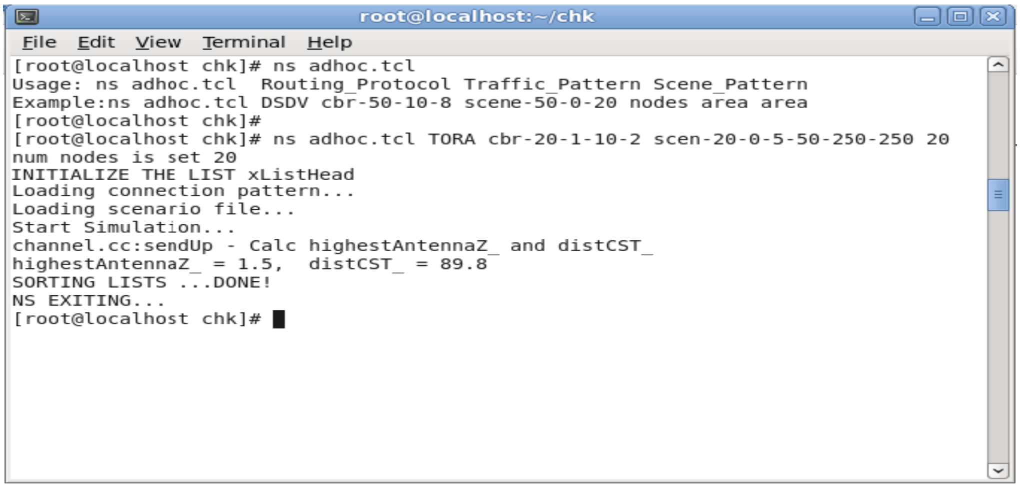

Figure 6.3: Simulation

6.2 Result Analysis

Here comparison between the routing protocols on the basis of packet delivery fraction, throughput using different number of traffic sources. Throughput describes the loss rate as seen by the transport layer. It reflects the completeness and accuracy of the routing protocol. From these graphs it is clear that throughput decrease with increase in mobility. As the packet drop at such a high load traffic is much high.

Table 6.1: Simulation Parameter

Parameter	value
Simulator	NS-2
Protocols studied	AODV,DSDV,DSR,and TORA
Simulation time	100 sec
Simulation area	100x100, 200x200
Node movement	Random way point
Bandwidth	2Mbit
Traffic type	CBR
Data payload	Bytes/packet
Nodes	10,20,30,40,50

TORA performs better at high mobility but in other cases it shows to have a lower throughput figure 6.4 and figure 6.5. AODV in our simulation experiment shows to have the best overall performance. On-demand protocols (DSR and AODV) drop a considerable number of packets during the route discovery phase, as route acquisition takes time proportional to the distance between the source and destination.

The situation is similar with TORA. Packet drops are fewer with proactive protocols as alternate routing table entries can always be assigned in response to link failures.

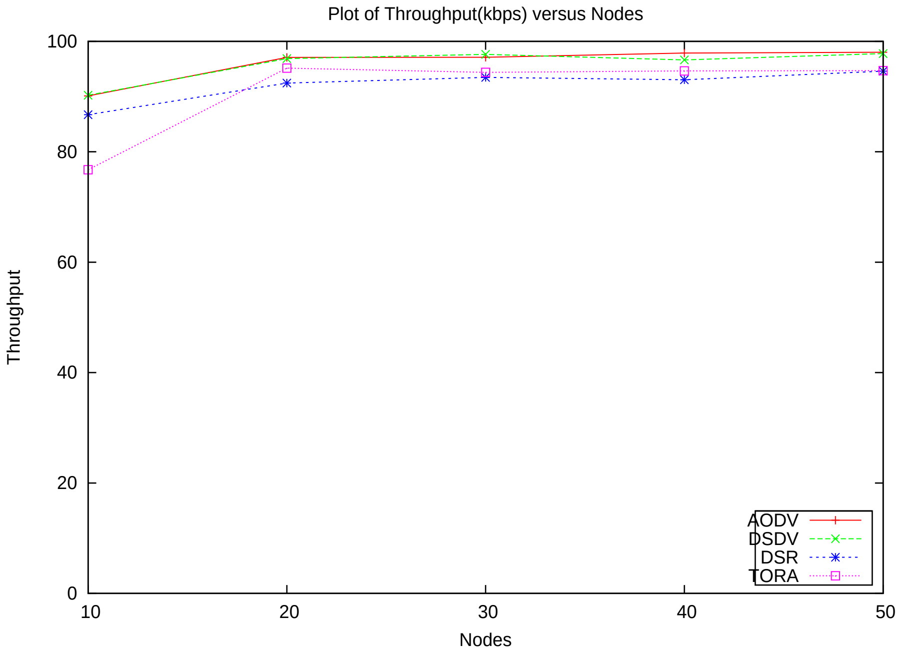

Figure 6.4: Graph of Throughput Vs Node

AODV performs a little better delay-wise and can possibly do even better with some fine-tuning of this timeout period by making it a function of node mobility. TORA too has the worst delay characteristics because of the loss of distance information with progress. Also in TORA route construction may not occur quickly. This leads to potential lengthy delays while waiting for new routes to be determined.

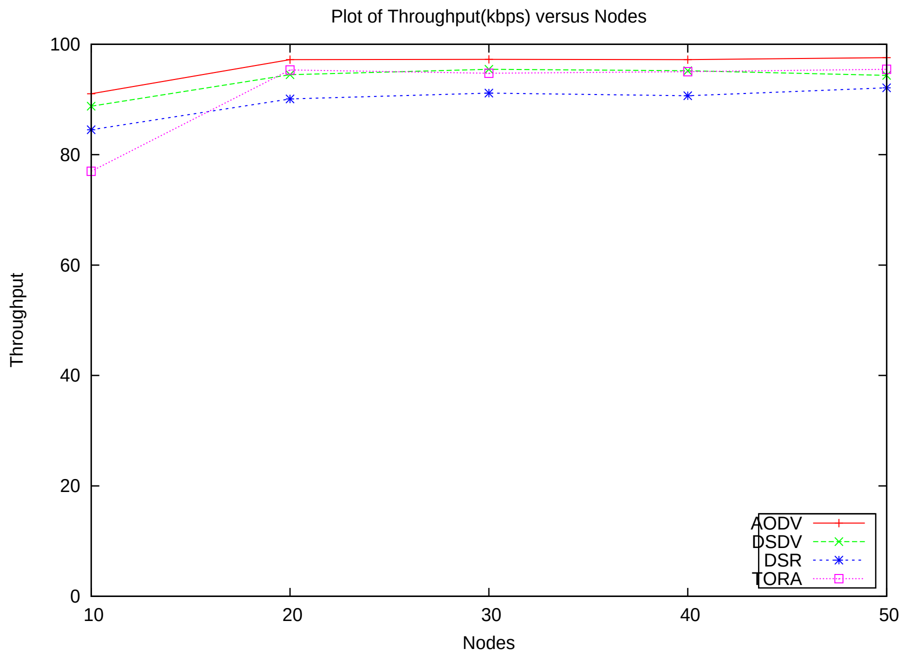

Figure 6.5: Graph of Throughput Vs Nodes

Performance of TORA is not very competitive with the distance vector and on-demand protocols. We conjecture that it is due to the fact network partitions cause TORA to do substantial work to erase routes even when those routes are not in use. However, TORA shows a better performance for large networks with low mobility rate.

TORA can be quite sensitive to the loss of routing packets compared to the other protocols. Buffering of data packets while route discovery in progress, has a great potential of improving DSR, AODV and TORA performances shows in figure 6.6 and figure 6.7.

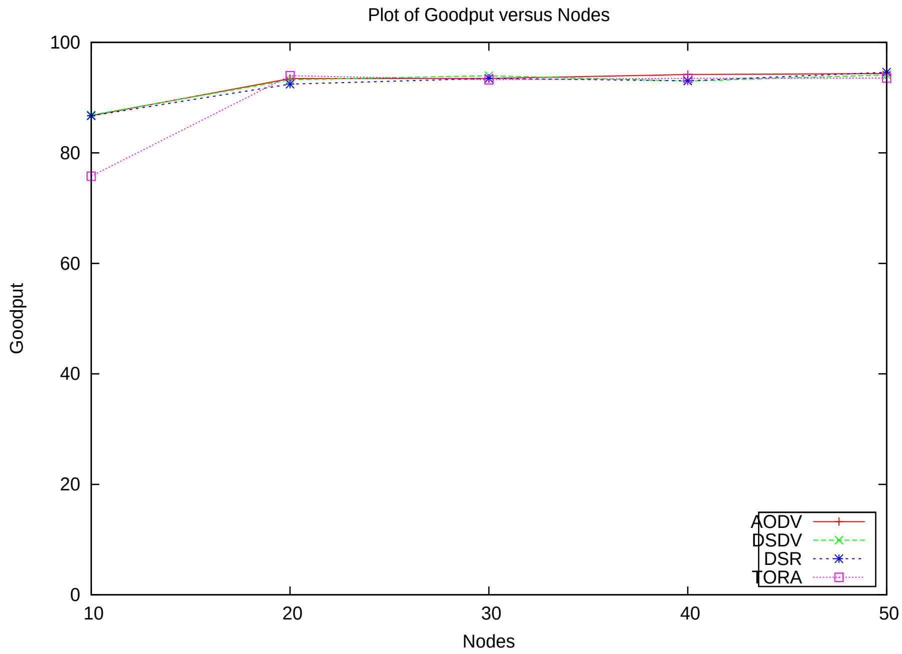

Figure 6.6: Goodput Vs Nodes

In DSR Route Discovery is fast, therefore shows a better delay performance than the other reactive protocols at low pause time (high mobility). But in case of congestion (high traffic) DSR control messages get loss thus eliminating its advantage of fast establishing new route.

In short, AODV has the best all round performance. DSR is suitable for networks with moderate mobility rate. It has low overhead that makes it suitable for low bandwidth and low power network. Whereas TORA is suitable for operation in large mobile networks having dense population of nodes. The major benefit is its excellent support for multiple routes and multicasting for large region.

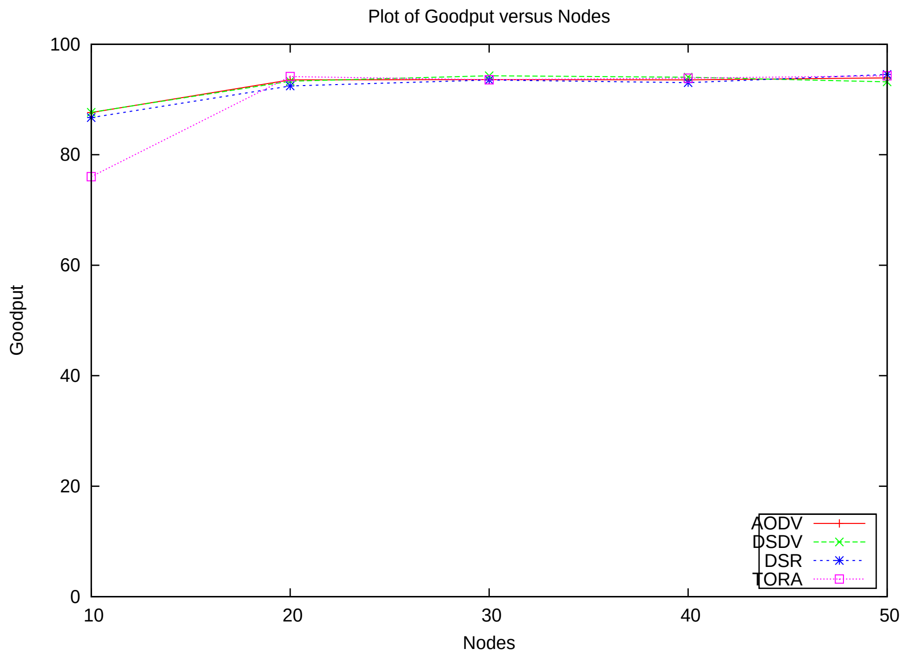

Figure 6.7: Goodput Vs Nodes

6.3 Result of Proposed Scheme

The Simulation is done to evaluate the performance index Packet Delivery Ratio. It is mathematically calculated by the given formula:

$$P = \frac{1}{c} \sum_{f=1}^{e} \frac{Rf}{Nf} \tag{6.1}$$

Where P is the fraction of successfully delivered packets, C is the total number of flow or connections, f is the unique flow id serving as index, Rf is the count of packets received from flow f and Nf is the count of packets transmitted to f.

For Average end-to-end delay includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times. It can be defined as:

$$D = \frac{1}{N} \sum_{i=1}^{s} (ri - si) \tag{6.2}$$

Where N is the number of successfully received packets, i is unique packet identifier, ri is time at which a packet with unique id i is received, si is time at which a packet with unique id i is sent and D is measured in ms. It should be less for high performance.

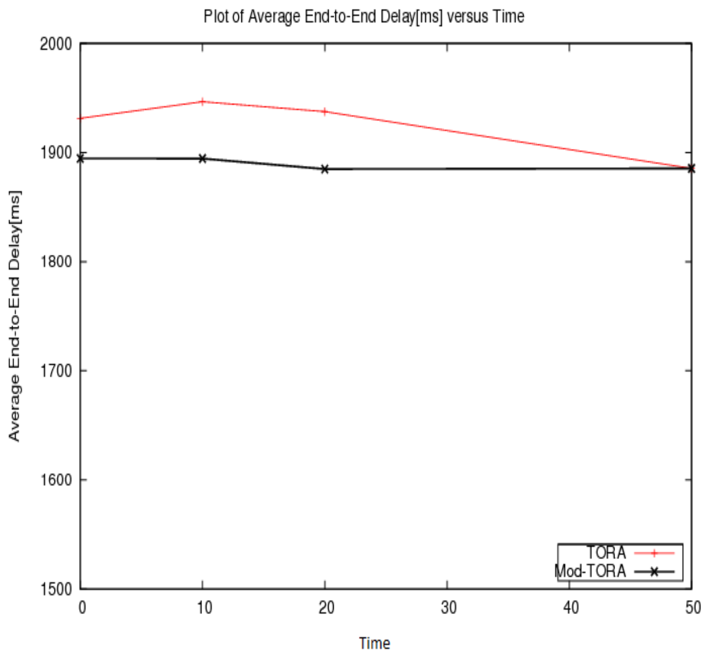

Figure 6.8: End to End Delay Vs Time

The results of the simulation shows in figure 6.8 that as we increase the number of nodes and time the end to end delay for protocols increases. But the effect of increasing the number of nodes can be seen in TORA the most. Figure shows the graphs of the End to end Delay with variable parameter time 0, 10, 20, 50 with number of nodes 20 and 50 respectively for area of 250X250 meters.

When mobility increase it will decrease throughput and same situation occurs when we are using different number of nodes increase like 50, 70 and 90 respectively. Shown in figure 6.9.

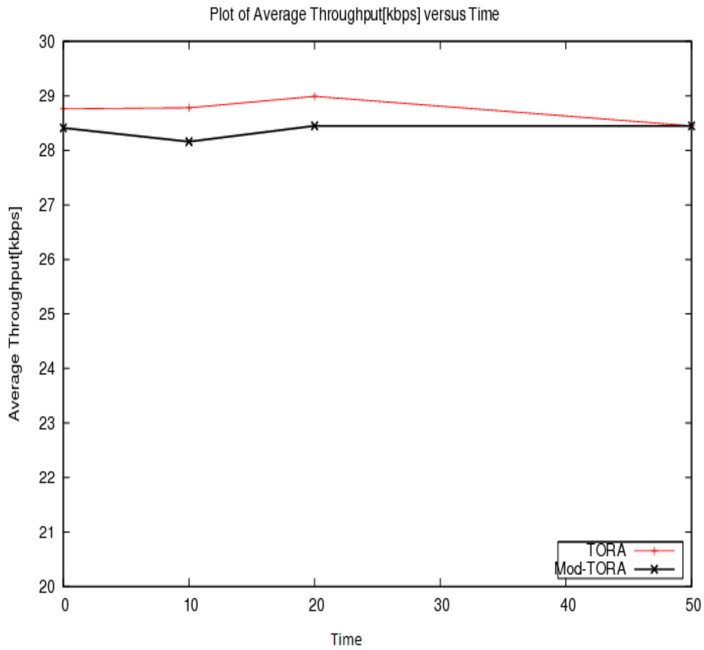

Figure 6.9: Throughput Vs Time

Table 6.2: Simulation Parameter

Parameter	value
Simulator	NS-2
Protocols used	TORA, Modified TORA
Simulation time	100 sec
Simulation area	250X250, 5000X5000
Node movement	Random way point
Bandwidth	2Mbit
Traffic type	CBR
Data payload	Bytes/packet
Nodes	20,50,70,90
Time	0, 10, 20, 50

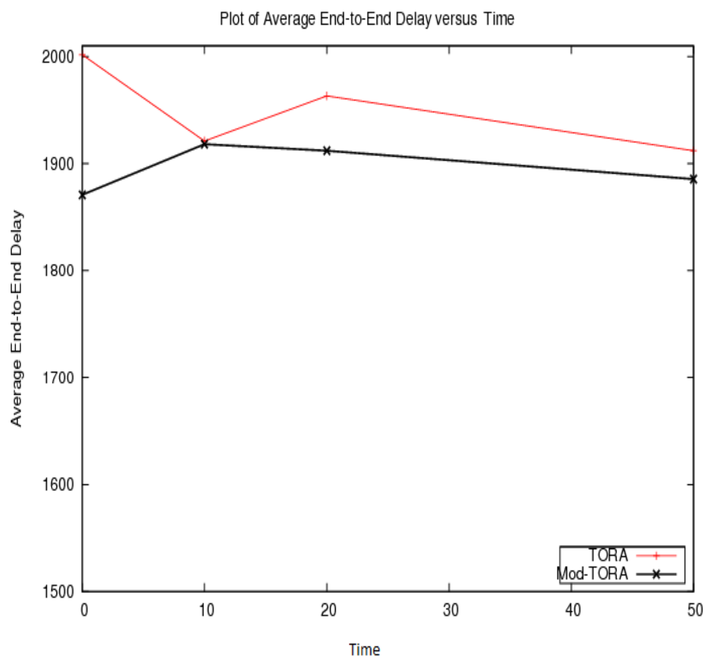

Figure 6.10: End to End Delay Vs Time

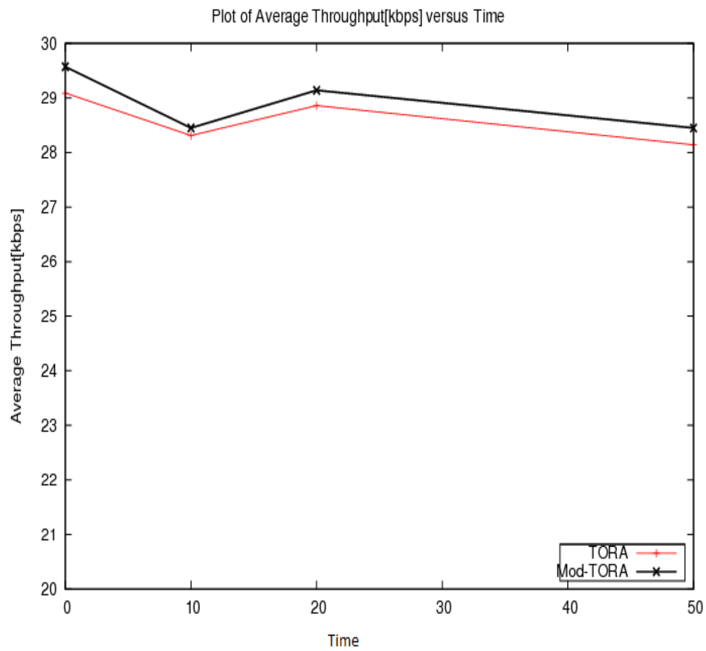

Figure 6.11: Throughput Vs Time

Figure 6.10-6.13 shows throughput versus time statistics with traditional TORA and Modified TORA Protocol. Results of Simulation shows that when increasing a mobility of nodes then delay of delivery increase and due to simultaneously generate packets QRY, UPD and CLR packets from route link creation/failure and collision occurs.

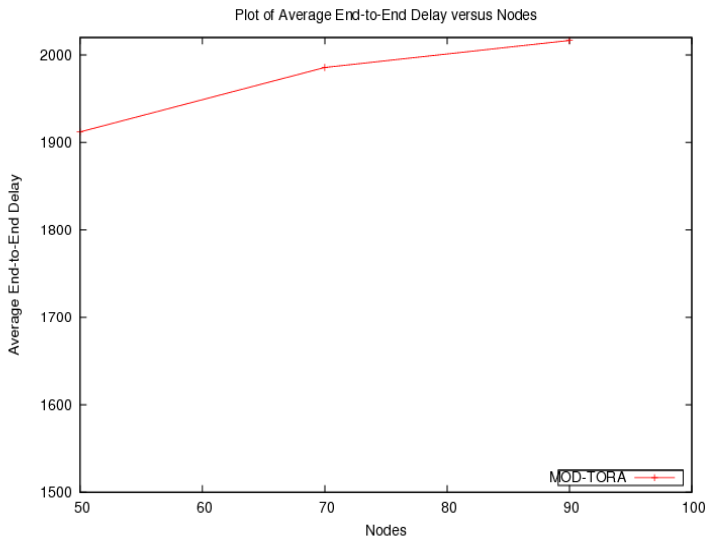

Figure 6.12: End to End Delay Vs Nodes

When we have increase quantity of nodes then control packets are increase and due to that delay is too much higher than previous. Solution of this situation is to send packet timely, so using this we have reduced few of overhead of control packets and in resultant we got improved throughput than previous throughput which is shown in figure 6.12 in modified TORA Protocol.

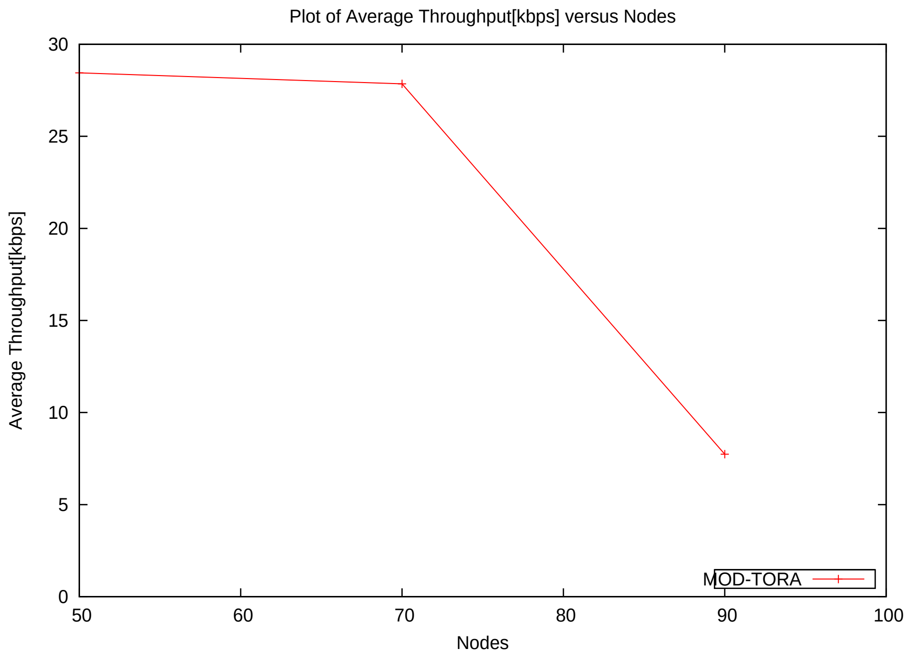

Figure 6.13: Throughput Vs Nodes

Problem arises when we are increasing nodes then end to end delay increases and throughput decreases. In figure 6.12-6.13 shows that when we are increasing number of nodes at that situation control packets are generates too much high and due to this phenomenon delay increases drastically and throughput decreases drastically. To solve this we proposed another approach which is clustering technique with TORA Protocol routing shows figure 6.14-6.16.

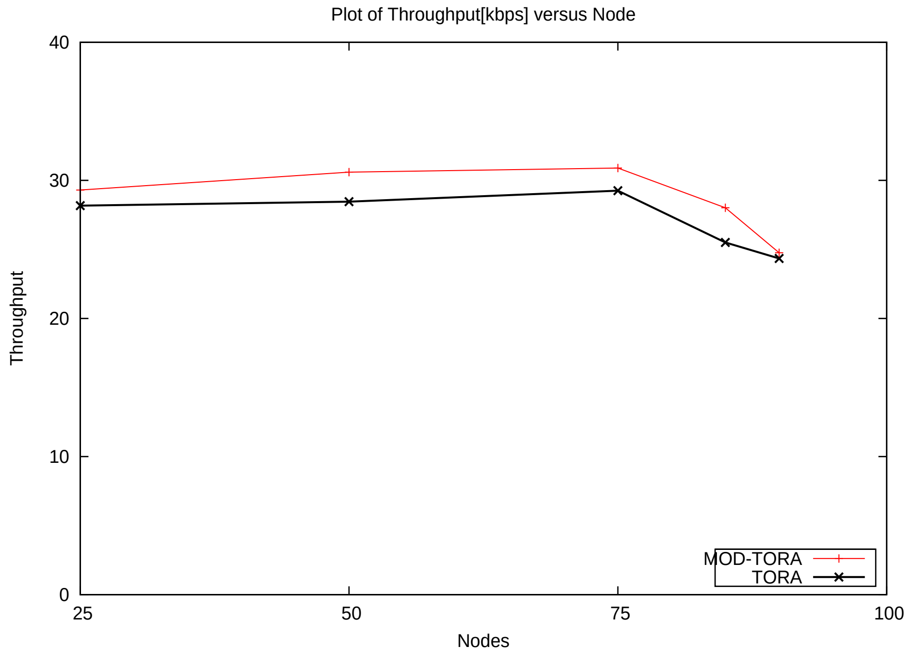

Figure 6.14: Throughput Vs Nodes

Figure 6.14 shows that modified protocol gives higher throughput than traditional protocol because of in modified protocol modified with clustering scheme to reduce overhead control packets.

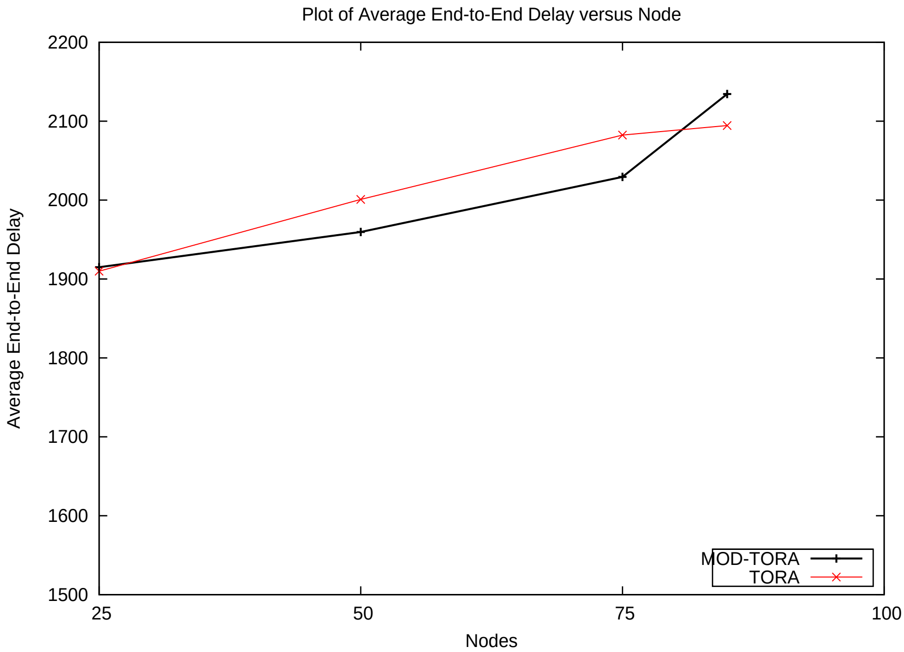

Figure 6.15: End to End Delay Vs Nodes

Figure 6.15 shows that end to end delay decrease compare to traditional routing protocol due to reduce overhead of packets. But when gradually increasing quantity of nodes then delay is too higher and throughput is also become down.

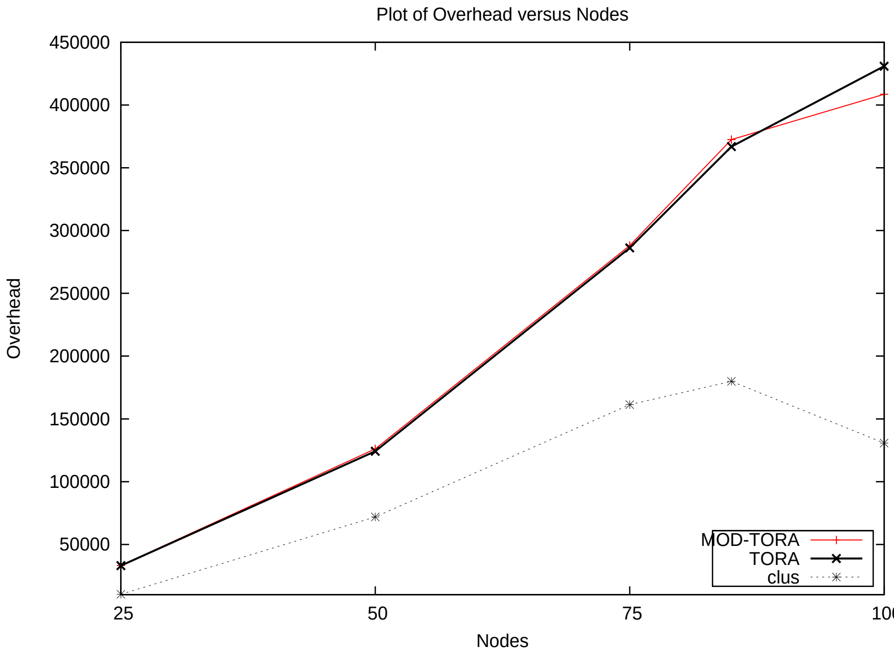

Figure 6.16: Overhead Vs Nodes

So, figure 6.16 shows that when we used modified TORA protocol which is modified by geocasting and clustering scheme. We have implemented optimize result of overhead in terms of control packets which is helpful to reduce delay, increase throughput and makes network more scalable.

Chapter 7

Conclusion and Future Work

The thesis includes, overview of wireless network and problems of resource constrains like overhead, limited available bandwidth etc. The work mainly concentrate on simulation of existing TORA protocol along with Geocasting and Clustering technique. The performance of TORA Protocol is compared with modified TORA Routing protocol. The modified TORA routing protocol is improved in terms of throughput, overhead and scalability of network.

We have implemented routing protocol simulation with different topologies. We have simulated the same with large area and dense networks. Also Geocasting technique has been implemented. The cluster are created wherein a node of the cluster is selected as a cluster head. After this step, the data from the cluster nodes are transferred to cluster head to reduce the number of overhead packets. Finally clustering hierarchy is created to overcome the overhead during routing process for all nodes.

At present the cluster nodes are considered to be stationary but in real deployment the nodes may not be stationary. Also the cluster head is not switched which if done can be helpful. When the size of network is increased the number of cluster heads may become an issue. To solve this problem multilevel clustering can be employed.

Appendix A

List of Publication

Paper titled "A COMPARATIVE ANALYSIS ON REACTIVE ROUTING PROTOCOL FOR MANET" published in International Journal of Advanced Engineering Research and Studies(IJAERS Journal), IJAERS/Vol. I/ Issue II/January-March, 2012.

Appendix B

Trace File Format

B.1 General trace formats

Usually the format looks like this:

```
ACTION: [s|r|D|f]: s -- sent, r -- received, D -- dropped,f -- forwarded
WHEN: the time when the action happened
WHERE: the node where the action happened
LAYER: AGT -- application,
 RTR -- routing,
 LL  -- link layer (ARP is done here)
 IFQ -- outgoing packet queue (between link and mac layer)
 MAC -- mac,
 PHY -- physical
flags:
SEQNO: the sequence number of the packet
TYPE: the packet type
  cbr -- CBR data stream packet
  DSR -- DSR routing packet (control packet generated by routing)
  RTS -- RTS packet generated by MAC 802.11
  ARP -- link layer ARP packet
SIZE: the size of packet at current layer, when packet goes down,
  size increases, goes up size decreases
```

```
[a b c d]: a -- the packet duration in mac layer header
  b -- the mac address of destination
  c -- the mac address of source
  d -- the mac type of the packet body
flags:
[......]: [
  source node ip : port_number
  destination node ip (-1 means broadcast) : port_number
  ip header ttl
  ip of next hop (0 means node 0 or broadcast)
  ]
```

So if you have a line like this

```
s 76.000000000 _98_ AGT  -  - 1812 cbr 32 [0 0 0 0] - [98:0 0:0 32 0]
```

in trace file.

It should interpret it as

Application 0 (port number) on node 98 sent a CBR packet whose ID is 1812 and size is 32 bytes, at time 76.0 second, to application 0 on node 0 with TTL is 32 hops. The next hop is not decided yet.

Similarly, It should be able to interpret a line such as this:

```
r 0.010176954 _9_ RTR  -  - 1 gpsr 29 [0 ffffffff 8 800] - [8:255 -1:255 32 0]
```

as

The routing agent on node 9 received a GPSR broadcast (mac address 0xff, and ip address is -1, either of them means broadcast) routing packet whose ID is 1 and size is 29 bytes, at time 0.010176954 second, from node 8 (both mac and ip addresses are 8), port 255 (routing agent).

```
r 40.639943289 _1_ AGT - 1569 tcp 1032 [a2 1 2 800] -  [0:0 1:0 32 1] [35 0] 2 0
```

The first field is a letter that can have the values r,s,f,D for received,sent,forwarded and dropped, respectively.It can also be ¡ for giving a location or a movement indication. The second field is the time. The third field is the node number. The fourth field is MAC to indicate if the packet concerns a MAC layer, it is AGT to indicate the transport layer(e.g. tcp) packet, or RTR if it concerns the routed packet. It can also be IFQ to indicate events related to the interference priority queue(like drop of packets). After the dashes come the global sequence number of the packet(this is not the tcp sequence number). At the next field comes more information on the packet type(eg. tcp,ack or udp). Then comes the packet size in bytes. The 4 numbers in the first square brackets concern the mac layer information. The first hexadecimal number,a2(which equals 162 in decimal) specifies the expected time in seconds to send this data packet over the wireless channel. The second number,1, stands for the MAC-id of the sending node, and the third,2, is that of the receiving node. The fourth number,800, specifies that the MAC type is ethertype ip. The next numbers in the second square brackets concern the IP source and destination addresses, then the ttl(Time To Live) of the packet(in our case 32), The third bracket concern the tcp information: its sequence number and the acknowledgment number.

Bibliography

[1] V. Park and S. Corson, Temporally Ordered Routing Algorithm (TORA) Version 1, Functional specification IETF Internet draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-01.txt, 1998.

[2] J.C. Navas and T. Imielinski. Geocast - geographic addressing and routing. In Proc. of ACM/IEEE MOBICOM, Budapest, Hungary, Sept. 1997.

[3] Asad Amir Pirzada and Amitava Datta and Chris McDonald. Trustworthy Routing with the TORA Protocol . Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on Digital Object Identifier. Pages 121-130, 2004.

[4] Y.-B. Ko and N.H. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In Proc. of IEEE WMCSA, New Orleans, LA, Feb. 1999.

[5] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In Proc. of IEEE INFOCOM, Kobe, Japan, Apr. 1997.

[6] Suchismita Chinara, Santanu Kumar Rath,"A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks", Journal of Network and Systems Management, Volume(17), Springer Science+Business Media, pp. 183207, 2009.

[7] S. Mehta, P. Sharma and K. Kotech. A Survey on Various Cluster Head Election Algorithms for MANET. In Proc. of 2nd International NUiCONE Nirma University, Ahmedabad, India, Dec. 2012.

[8] Ratish Agarwal, Dr. Mahesh Motwani, ”Survey of clustering algorithms for MANET”,International Journal on Computer Science and Engineering, Vol. 1 Issue 2, 2009, pp. 98-104

[9] A.D. Amis, R. Prakash, T.H.P Vuong, D.T. Huynh, ”Max-Min DCluster Formation in Wireless Ad Hoc Networks”, In proceedings of IEEE Conference on Computer Communications (INFOCOM) Vol. 1. pp. 32-41, 2000

[10] G. Chen, F. Nocetti, J. Gonzalez, and I. Stojmenovic, ”Connectivity based k-hop clustering in wireless networks,” In proceedings of the 35th Annual Hawaii International Conference on System Sciences. Vol. 7, pp. 188.3, 2002

[11] Tomoyuki Ohta, Munehiko Fujimoto, Shinji Inoue, Yoshiaki Kakuda, ”Hi-TORA: A Hierarchical Routing Protocol in Ad Hoc Networks,” hase, pp.143, 7th IEEE International Symposium on High Assurance Systems Engineering (HASE’02), 2002

[12] [1] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proc. of ACM/IEEE MOBICOM, Dallas, TX, Oct. 1998.

[13] Y.-B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In Proc. of ACM/IEEEMOBICOM, Dallas, TX, Oct. 1998.

[14] E. Gafni and D. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. IEEE Trans. on Comm., C-29(1):1118, 1981.

[15] Young-Bae Ko and Nitin H. Vaidya. Anycasting-based protocol for geocast service in mobile ad hoc networks.Computer Networks, Volume 41, Issue 6, Pages 743-760,22 April 2003.

[16] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In Proc. of IEEE WMCSA, Feb. 1999.

[17] J.J. Garcia-Luna-Aceves and E.L. Madruga. A multicast routing protocols for ad-hoc networks. In Proc. of IEEE INFOCOM, New York, Mar. 1999.

[18] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In Proc. of DIAL-M, Seattle, WA, Aug. 1999.

[19] K. Obraczka and G. Tsudik. Multicast routing issues in ad hoc networks. In Proc. of IEEE ICUPC, Oct. 1998.

[20] Wendi Beth Heinzelman, Application-Specific Protocol Architectures for Wireless Networks, [$PhdThesis$], June, 2000.

[21] "Network Simulator",[Online] available at: https://www.isi.edu/nsnam/ns/.