Resilience Routing in AdHoc Networks

A decision based routing tree mechanism that can establish routes in adhoc network, which may than be configured into logical dual ring. Also a system is proposed to embed the QoS mechanisms, resilience and reliability features from RPR

Tehmina Karamat Khan

Submitted for the degree

of Master of Philosophy

Department of Informatics

University of Bradford

2008

CONTENTS

Acknowledgement

I would like to thank Professor Mike Woodward, my advisor for the research and simulations, for his support and advisory work during the course of my research, and Mr. John Mellor for his aid as my Tutor and the help in writing and reviewing my simulations and thesis. Also, I would like to credit my parents, brothers, my daughter Khadija and son Ibrahim for there continuous support and help through out my research. I would like to thanks the department of computing, School of Informatics for providing me with resources to continue with my research work, for which I am very grateful.

ABSTRACT

As the number of people using wireless networks is increasing, the need to reduce the vulnerability of wireless networks from node or link failures that cause loss of data is becoming a priority. Also the present techniques and topologies used for wireless networking are not sufficient to handle the traffic load even if we solve the issues of reliability and resilience. Packet loss or delay is increasingly likely due to the increase in the number of packets as technology is evolving and more video and voice packets along with the data packets are being transmitted. Only the efficient and intelligent use of the shared medium can solve the problem and help in avoiding the collision or delay among the packets using a newly proposed intelligent topology.

Wireless technology offers the potential to replace wires from many applications, particularly for the rapid deployment of networks for permanent or temporary use. Fiber_ optic metropolitan area networks (WAN) provide security and resilience. A target of the research was to match this in the wireless environment. This research investigates the suitability if using wireless technology for the establishment of a MAN by adding features to enhance resilience. We proposed a mechanism that may be rapidly deployed and provide automatic configuration.

Research work and simulation design has been used to develop a new wireless network topology for an efficient and intelligent packet transmission by identifying reliable routes. This novel idea will help give wireless as well as mobile technology a clear edge over wired technology, not only in the case of mobility but also in the case of security of data and other services. A decision based routing tree mechanism has been developed, that can establish routes in an ad-hoc network which may than be configured into a logical dual ring. At the same time the proposed system proposes to embed the quality of service mechanisms, resilience, and reliability features from RPR.

The simulations were created using Microsoft Visual Studio.Net for the Decision based routing algorithm. The results were compared with an existing LAR algorithm. We have obtained 95% confidence intervals on all the performance analysis results to indicate accuracy.

Chapter 1

INTRODUCTION

1.1 Motivation

Wireless communication as the name implies is the transfer of data or information without the use of wires, which give the users more mobility freedom, it gives the designers more connectivity options and the capability to connect new devices to the network. But as the number of users in the network increases more the possibility of attacks and threats to the network increases and also the traffic load on the network increases. Even if the attack issues on the network are resolved the packet collision cannot be avoided as the number of packets increases because of more video and voice packets are being transferred along with the data packets.

This problem of packet collision may be solved if the medium can be used efficiently and intelligently in a way such that the packet collision can be avoided.

These problems of packet collision motivate an investigation to design a network where the packet transmission can be done efficiently.

1.2 Research Aims and Objectives

The ultimate aim of this thesis is to develop a new wireless (virtual) network topology and routing algorithm for efficient and intelligent packet transmission by identifying reliable routes.

This aim is to be achieved through following objectives

1) To identify through a literature review efficient wireless topologies and routing algorithms.
2) To design an efficient ring based resilient routing algorithm.
3) To develop a simulation environment that can be used to assess the performance of the algorithm.
4) Compare the performance of the resilient routing algorithm against an existing routing algorithm.

1.3 Contributions

The following contributions are made in the thesis

1) A novel resilient packet ring technology for Wi-Fi networks is proposed.
2) An efficient routing algorithm has been proposed that aims to optimally identify a ring of nodes.
3) A detailed simulation model has been developed.
4) The performance of the algorithm is evaluated and benchmarked.

1.4Structure of the Thesis

The thesis is organized as follows:

1) Chapter 2 describes different types of routing techniques and protocols and their advantages and disadvantages, their architecture design are discussed in detail.
2) In Chapter 3 different types of networks their features and usage and in particular Wi-Fi and WiMAX technologies and their advantages and disadvantages, their architecture design and flaws are discussed in detail Also in this chapter the Resilient Packet Ring is discussed in detail.
3) Chapter 4 explains the development of a decision tree based routing algorithm. The pseudo code for the algorithm is also provided in the chapter.
4) In Chapter 5 the performance measures which were mean delay and throughput was analyzed for our algorithm and evaluates its performance against LAR.
5) Chapter 6 concludes the thesis and suggests further future work and possible directions for the enhancement of the proposed decision based routing algorithm are pointed out.
6) The simulation code and its descriptions are contained on Appendix.

Chapter 2

ROUTING IN MOBILE AD-HOC NETWORKS

2.1 INTRODUCTION

Limited resources in MANETS have made designing an efficient and reliable routing strategy a challenging problem. An intelligent routing strategy is required to efficiently use bandwidth, which should be optimal, feasible, and should quickly and accurately adapt to a variety of network circumstances such as network bandwidth, router queue size, and network delay.

2.2 Wired Routing Protocols

Prior to the increased interests in wireless networking, two main algorithms were used in wired networks. These algorithms were commonly referred to:

- Link State Algorithm (LSA)
- Distance Vector Algorithm (DVA)

2.2.1 Link State Algorithms (LSA)

LSAs are known as shortest path first algorithms. These algorithms use flooding by periodically broadcasting link-state costs of its neighbouring nodes to all other nodes in inter-network. Each node, however sends only the portion of routing table that describes the state of its own link. When each node receives an update packet, they update their view of network and their link state information. But, LSAs do not scale well. . This is because periodic or frequent updates in large networks result in high storage overhead, high overhead in path computation requiring more CPU power [2].

2.2.2 Distance Vector Algorithms (DVAs)

DVAs believe in sending all or some portion of routing table, to its neighbors only. DVAs send larger updates, as they only concern their neighbors. These algorithms are not scalable as LSAs, as significant part of bandwidth is consumed by large number of updations. These increases channel contention, converges slowly and suffer from looping problem [2].

2.3 Classification of Routing Protocols in MANETS

To overcome problems associated with LSAs and DVAs, a number of routing protocols are proposed. These are classified generally as:

- Proactive Routing Protocols (PRPs)
- Reactive Routing Protocols (RRPs)
- Hybrid Routing Protocols (HRPs)

2.3.1. Proactive Routing Protocols (PRPs)

This class of routing protocols uses routing tables and continuously updates the routing tables on fixed intervals. Routing information is kept in a number of different tables. Difference between these protocols exists in ways routing information is updated, detected and type of information kept at each routing table.

Some of existing proactive routing protocols are described below:
Destination sequenced distance vector routing protocol (DSDV).
Wireless routing protocol (WRP)
Global state routing protocol (GSR)
Cluster-Head gateway Switch protocol (CGSR)

2.3.1.1 Destination sequence distance vector (DSDV)

DSDV routing protocol is a table driven algorithm based on classical Bellman Ford routing mechanism. Every node in network maintains a routing table in which all of the possible destinations and number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. In order to reduce the amount of overhead transmitted through network; two types of update packets are used. These are referred to “full dump” and “incremental” packets. Full dump packet carries all available routing information and can require multiple Network Protocol Data Units (NPDUs). Smaller incremental packet carries only information changed since last full dump. Incremental update messages are sent more frequently than full dump packets. However, DSDV still introduces large amount of overheads to network due to required periodic update messages [27]. Overhead involved grows according to O (N^2). This protocol does not scale well, as large portion of network bandwidth is used by updation procedure. Therefore the protocol will not scale in large) networks, since a large portion of the network bandwidth is used in the updating procedure[2].

2.3.1.2 Wireless Routing Protocol (WRP)

WRP is a distance vector routing protocol. This protocol guarantees loops freedom and avoids temporary routing loops by using predecessor information. However, WRP requires each node to maintain four routing tables, which are distance table, routing table, link cost table and message retransmission list table (MRL). MRL contains sequence number of update message, a retransmission counter and a list of updates sent in update message. These tables introduce a significant amount of memory overhead at each node as size of the network increases. Another disadvantage of WRP is that nodes learn of the existence of their neighbours from acknowledgments and other messages[2]. If a node is not sending messages it sends a hello messages within a specified time period to ensure connectivity. These hello messages are exchanged between neighbouring nodes whenever there is no recent packet transmission. This will consume a significant amount of bandwidth and power as each node is required to stay active all the time [27].

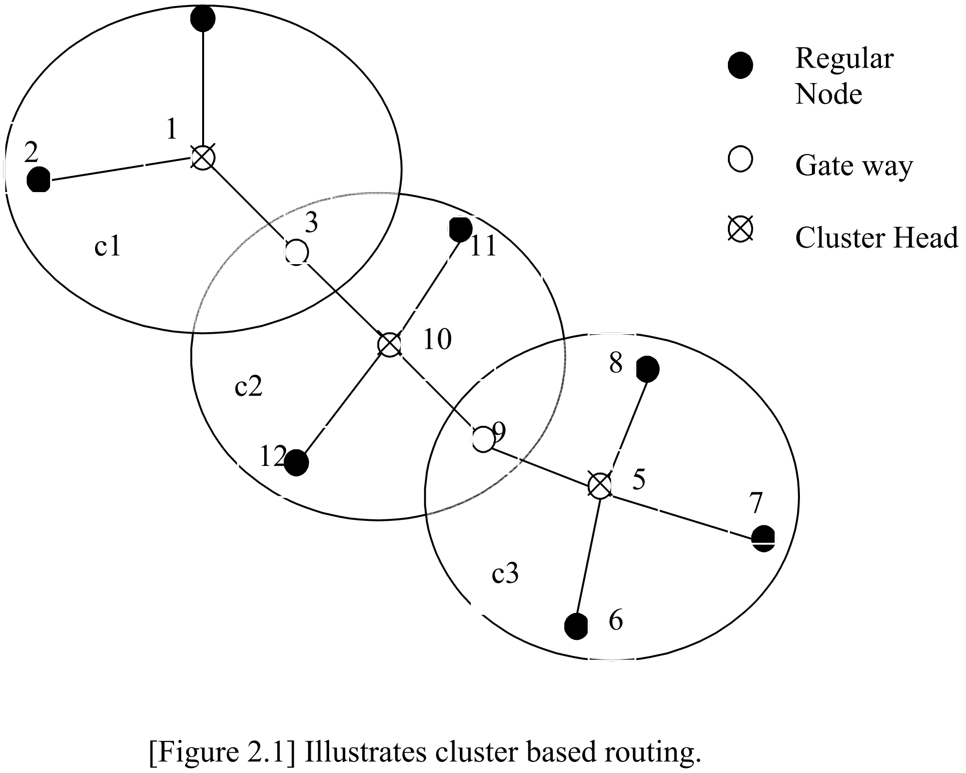
2.3.1.3 Global State Routing (GSR)

GSR is similar to DSDV. It takes the idea of link state routing but improves it by avoiding flooding of routing messages. In GSR, each node maintains a link state table based on up-to-date information received from neighbouring nodes, and periodically exchanges its neighbour node only (no global flooding). This means GSR keeps overhead of control message low. However, size of update messages is relatively large and as the size of the network grows they will get even larger. Therefore a considerable amount of bandwidth is consumed by these update messages [27].

2.3.1.3 Cluster-Head Gateway Switch Routing (CGSR)

CGSR is a clustered multi hop mobile wireless network with several heuristic routing schemes. In this scheme nodes are grouped into cluster and a cluster head is elected. All nodes that are in communication range of the cluster-head belong to its cluster. The head node controls transmission medium and all inter cluster communications occur through this node.

CGSR is a hierarchical cluster head-to-gateway routing approach to route traffic from source to destination. Gateway nodes are nodes that are with in communication range of two or more cluster heads. A packet sent by a node is first routed to its cluster head and then it is routed from cluster head to a gateway to another cluster head and so on until the cluster head of the destination node is reached. The packet is then transmitted to destination.



[Figure 2.1] Illustrates cluster based routing.

The advantage of this protocol is that each node only maintains routes to its cluster-head, which means that routing overheads are lower compared to flooding routing information through the network. Each node needs to periodically broadcast its clustered member table and update its table based on the received updates. This turns into significant overhead maintaining clusters [2].

2.3.2 Reactive Routing Protocols

In RRPs, routing tables do not exist and updation is once demanded either by user or service. This means that routes are determined and maintained for nodes that require sending data to a particular destination. Routes remains valid till the destination is reachable or until the route is no longer valid. Route discovery usually occurs by loading a route request packets through the network.

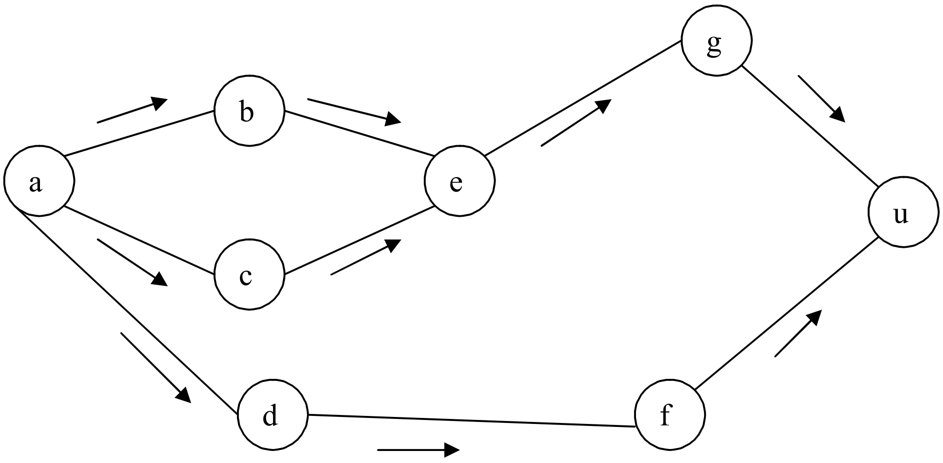
Reactive routing can be classified into two categories; source routing and hop-by-hop routing. A number of different reactive routing protocols have been proposed to increase performance of reactive routing. Some of these routing protocols are outlined and discussed below:

Ad-hoc on demand Distance Vector (AODV) Routing
Dynamic Source Routing (DSR)
Light-Weight Mobile Routing (LMR)
Location Aided Routing (LAR)

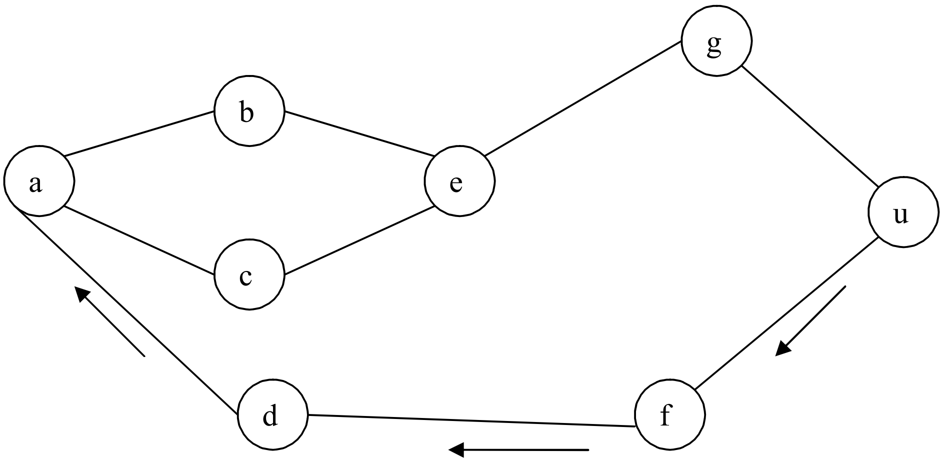
2.3.2.1 Ad-hoc on demand Distance Vector Routing (AODV)

AODV routing is improvement of DSDV and DSR algorithms. It uses periodic beaconing and sequence numbering procedure of DSDV and a similar route discovery procedure as DSR. However, AODV minimizes number of broadcasts by creating routes on demand as opposed to DSDV that maintains the list of all routes. This way AODV achieve less routing overhead. AODV route replies only carry the destination IP address and sequence number, whereas in DSR route replies carry address of every node along the route. AODV builds routes using route request/route reply query cycle. To find a path to destination, source broadcast a route request packet called RREQ to its neighbours. If one of these neighbours has a route to the destination then it replies back with a route reply RREP, other wise neighbours in turn rebroadcast request till it reaches destination.

When a node forwards a route request packet to its neighbours, it also records in its tables, the node from which the first copy of the request came. As long as route remains active, it will continue to be maintained. A route considered active as long as there are data packets periodically travelling from source to destination along that path. Once the source stops sending packets, the link will time out and eventually is deleted from intermediate node routing tables. If a link break occurs while route is active, the node upstream of the break propagates a route error message (RERR) to source node. After receiving the RERR message, if the source node still desires the route, it can re-initiate route discovery.



[Fig 2.2] a Propagation of Route Request



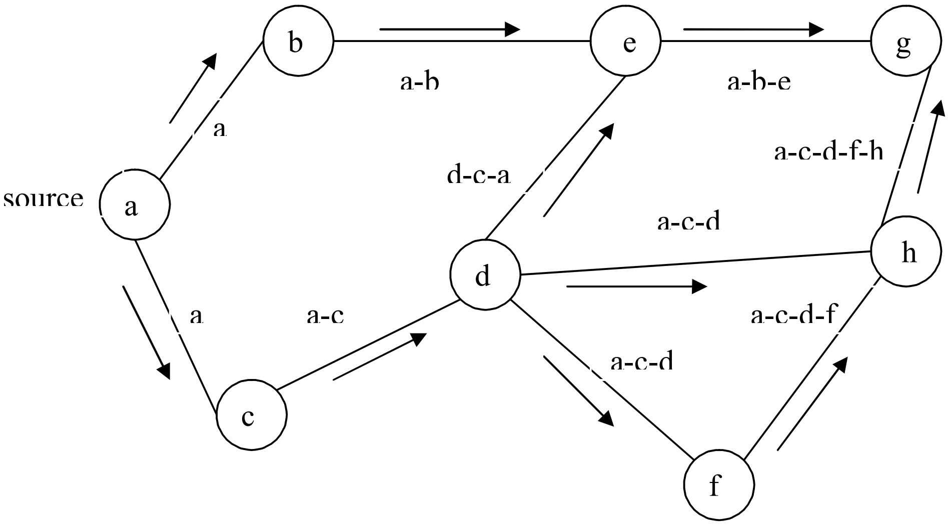
[Fig 2.2] b Path of the Route Reply (RREP)

The advantage of AODV is that it is adaptable to highly dynamic networks. However, node may experience large delays deriving route reconstructions. With increase in network size, more bandwidth is consumed [27].

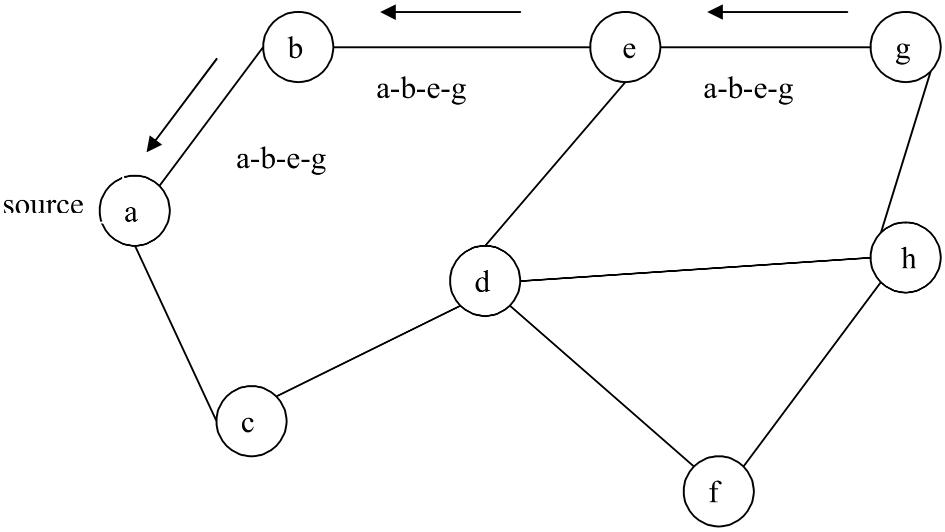
2.3.2.2 Dynamic Source Routing (DSR)

Dynamic source routing is an on demand routing protocol based on the concept of source routing. Nodes are required to maintain route caches that contain the source routes of which the node is aware of. Entries in the route cache are continually updated as new routes are learned.

The protocol is composed of the two main mechanisms. Route discovery and route maintenance which work together to allow nodes to discover and maintain routes to obituary destinations in the network. When source node wants to send a packet to the destination, it first look up its route cache to determine whether it already has a route to the destination, if it has an unexpired route, it will use this route to send the packet to the destination, but if the node does not have such a route, it initiates route discovery by broadcasting a route request packet. This route request contains the address of both the source and destination along with a unique identification number. Each node receiving the packet checks whether it knows of a route to the destination, if it does it adds its oven address to the route record of the packet and then forwards the packet to its neighbours. To limit the number of route requests propagated on the outgoing links of a node, a node only forwards the route request packet if the request packet has not yet been seen by the node, and its address is not present in the route record of the packet.



[Fig2.3a] Building of the Route during Route Discovery.



[Fig 2.3 b] Propagation of the Route Reply with the Route Record

A route reply is generated when either the route request reaches the destination or when it reaches an intermediate node which contains in its route caches on unexpired route to the destination.[27] A route request packet reaching such a node already contains, its route record, the sequence of hops taken from the source to this node. As the route request packet propagates through the network, the route record is formed as shown in fig 2.3a. If the route reply is generated by the destination then it places the route record from route request packet into the route reply packet. On the other hand if the node generating the route replies is an intermediate node than it appends its cached route to destination, to the route record of route request packets and puts that in the route reply packet. Fig 2.3 b shows that route reply packet being sent by destination node itself. To send a route reply packet, the responding nodes must have a route to the source. If it has a route to the initiator in its route cache, it may use that route other wise if symmetric links are supported the node may reverse the route in the route records. If symmetric links are not supported, the node may initiate it's over route discovery and piggy back the route reply on the new route request.

Route maintenance is accomplished through the use of route error packets and acknowledgments. When node encounter a fatal transmission problem at its DLL it generates a route error packet, it removes the loop in error from its route cache. All routes that contain the hop are truncated at that point. Acknowledgment packets are used to verify the correct operation of the route links.

The advantage of DSR is that nodes can store multiple routes to their route cache, which means that the source node can check its route cache for a valid route before initiating route discovery and if a valid route is found, then there will be no need for route discovery. Another advantage is it does not require periodic beaconing, therefore nodes can enter sleep node to conserve their power which also saves network bandwidth. But in DSR each packet to carry the full address from source to destination which make it not very effective in large networks, as the amount of overhead carried in the packet will continue to increase as the network increases [2].

2.3.2.3 Light Weight Routing (LMR)

The light weight routing algorithm belongs to the class of link reversal strategy of the fafti-Bertsekas algorithm [2]. The LMR protocol is another on demand routing protocol, which uses a flooding technique to determine its routes.

The light weight mobile routing algorithm was maintaining paths to a destination from all other nodes. This algorithm consists of two parts:

- Route Establishment
- Route Maintenance

Route Establishment: If a route is needed for a destination, a node issues a Query packet for route discovery. A query packet consist of

Source node ID (which is sending the query)

Destination node ID

Sequence counter, a sequence counter in a query produces increasing sequence number.

Transmitting node ID, ID of the node which is forwarding the ID

The query starts the initialization phase and floods the network. Each node that receives the query broadcasts it to all of its neighbours only once. Queries travel over the unassigned link. A node with a route to the destination D initiates a RPY reply packet. If a node receives a RPY copy over unassigned link, it marks the link to the neighbour (from when the RPY came) as downstream. If there is no RPY for a certain period of time, the initiator of the QRY may start another QRY.

Route Maintenance: LMR is on demand routing protocol. Route maintenance is only triggered when a route to a particular destination is still needed i.e. for a given destination D, instead of maintaining routes from all nodes to D, the algorithm guarantees route maintenance only for those sources that actually need the routes. This property ensures that the control overhead of the algorithm is quite low. This can be determined by monitoring traffic at the router nodes. A route is considered inactive if there is no traffic after a certain time-out period. A node which has lost its link to a still active destination issues a Failure Query Packet (FQ), which has an effect that other nodes will not issue a FQ anymore that previously routed through this node.

For route maintenance to work correctly nodes shall retransmit FQ packets regardless if they need themselves a route to the destination. A node which has no upstream links for a destination will not retransmit a FQ but instead will issue a normal query message [27]. The advantages of LMR are that the nodes maintain multiple routes to destination. Which increase the reliability of the protocol by allowing nodes with out initiating a route discovery technique; select the next available route to a particular destination. Also storage overheads and extra delays are avoided as each node only maintains its neighbour routing information however LMR may produce temporary invalid routes, which results in extra delays in determining a correct link [2]

2.3.2.4 Location Aided Routing (LAR)

LAR is an on-demand protocol; which is based on dynamic source routing (DSR). The Location –Added Routing protocol uses location information to reduce routing overhead and relies on the global positioning system (GPS) for location information. With the availability of GPS the mobile hosts know their physical location.

2.3.2.4.1 Expected zone and request zone

2.3.2.4.1.1 Expected Zone

When a source node S wants to send a packet to some destination node D, suppose node S knows that at time t0 D's position was L and that the current time is T1. On the basis of this information S is able to determine the expected zone of D from the viewpoint of node S by time t1 (1). For instance if D travelled with an average speed v, the source node S expects D to be in a circular region around the old position L with a radius v (t1-t0). The expected zone is only estimated by S to determine the possible locations of D. If D traveled with a higher speed than actual speed, the destination node may be outside the expected zone at time t1.If node S has no information about the position of node D, then the entire region of the ad-hoc network is assumed to be expected zone. In general, we can say that having more information about destination node D can result in a smaller expected zone.

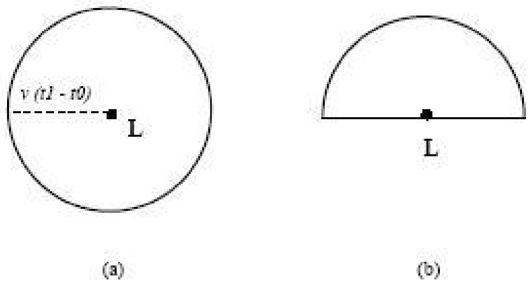


Fig 2.4[23]

As from fig2.4 when S knows that destination D is moving north, then the circular expected zone can be reduced to semi-circle.

2.3.2.4.1.2 Request Zone

The request zone is somewhat different from the expected zone, node S defines a request zone for the route request. A node forwards a route request only if it belongs to the request zone. To increase the probability that the route request will reach D, the request zone should include the expected zone.

The request zone includes regions outside the expected zone due to two reasons which are as follows:

1) When host node S is not in the expected zone, then additional regions must be included in the request zone.
2) The request zone as in fig 2.5(a) includes the request zone from fig2.4 (a) but in fig2.5 (b) all paths between S & D are outside of the request zone, which means that to find the path between S and D cannot be guaranteed in the request zone. LAR protocol allows S to expand the request zone so the path to destination can be more easily found, but when the expected zone is increased as in fig2.5(c), the overhead for route discovery also increases.

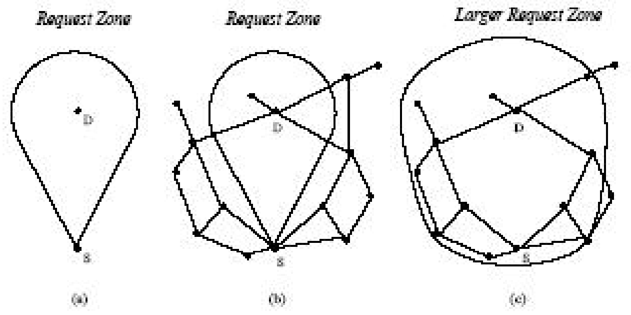


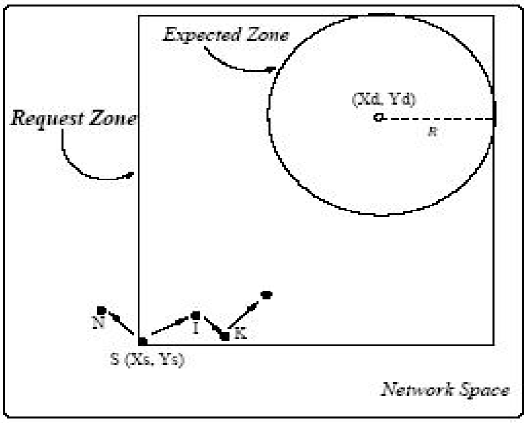
Fig2.5 Request Zone [23]

2.3.2.4.2 Request Zone Membership

LAR defines two different types of request zone based two protocols or algorithms which are LAR scheme 1 and 2.

2.3.2.4.2.1 LAR Scheme 1

This Scheme uses a rectangular shape request zone. In the route request message the node S includes the coordinates of the four corners of the request zone. When a node receives a route request message, if the node is not with in the specified rectangle it discards the request. This reduced the flooding in ad-hoc network.



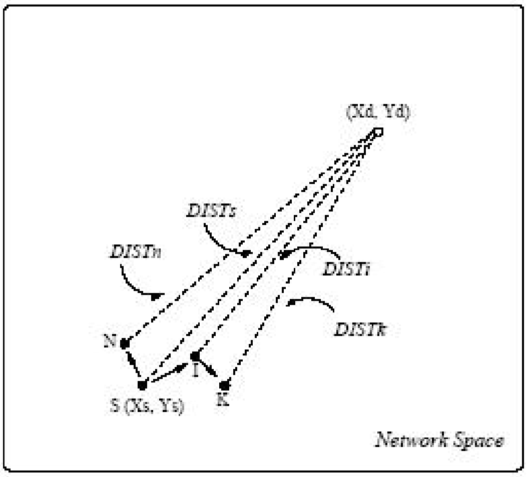
(a) LAR scheme 1

Fig2.6 [23]

When destination Node D receives the route request send by Node S, it replies with a route reply message and includes it current location and current time in the route reply message., when node S receives this route reply message it records the location of node D and this information can be used in future for route discovery.

2.3.2.4.2 LAR Scheme2

In the LAR scheme 2 the source node S includes two pieces of information with its route request message, the destination coordinates plus the distance to the destination. When a node receives the route request message it only forward the request message to the destination D. if it is closer to the node D otherwise it discards the route request.



(b) LAR scheme 2

Fig 2.7 [23]

The above figure shows the LAR scheme 2

LAR reduces routing overheads present in traditional flooding algorithms by using location information. Both LAR schemes limit the control overhead transmitted through the network and hence conserve bandwidth. They will also determine the shortest path to the destination since the route request packets travel away from the source towards the destination [2]

2.3.3 Hybrid Routing Protocols

Hybrid routing protocols are the combination of both proactive and reactive routing. In this class of routing protocols it partially formulates the routing tables and it also does the partially updating on demand. These protocols are designed to increase the scalability by allowing nodes with close proximity to work together to form some sort of a back bone, to reduce the route discovery overheads. Mostly this is achieved by maintaining near by nodes routes pro-actively and using route discovery strategy for determining the routes to far away nodes. Following are some hybrid routing protocols:

Zone Routing Protocol (ZRP)

Zone-based Hierarchical Link State Routing (ZHLS)

2.3.3.1 Zone Routing Protocol (ZRP)

ZRP is the first hybrid routing protocol with both proactive and reactive routing component. Zone routing protocol is based on the concept of zone. Intra-zone routing protocol (IARP) is used in the zone which is based on proactive routing scheme whereas inter-zone routing protocol (IERP) is used for communication between the zones which is reactive routing scheme. IERP and IARP are not specific routing protocols. Instead, IARP is a family of limited-depth, proactive link-state routing protocols. IARP maintains routing information for nodes that are within the routing zone of the node. IERP enhance route discovery and route maintenance services based on local connectivity monitored by IARP [2].

When a source wants to send packets, it first checks if the destination is within the zone. If yes the routes are immediately available. But if the destination node lie outside the routing zone, the ZRP uses a concept called border casting utilizes the topology information provided by IARP to direct query request to the border of the zone. The border nodes check there local zone for destination, if the destination is the member of the local zone of the node, it sends a route reply on the reverse path back to the source. The source node uses the path saved in the route reply packet to send data packets to the destination. But if the destination node is not member of this local zone the node adds its own address to the route request packet and forwards the packet to its border nodes. The advantage of this protocol is that it has significantly reduced the amount of communication overhead as compared to pure proactive protocols. It also has reduced the delays associated with pure reactive protocols such as DSR, by allowing routes to be discovered faster. lapping zone and each zone is identified by a zone id, and each node in the zone has a node id which is calculated using a GPS. Each node knows only the node connectivity with in its zone and the zone connectivity of the whole network.

ZHLS has a proactive component known as intra zone clustering i.e. when the destination node is in the same zone as the source node and reactive component called intra zone clustering when the destination node isn’t with in the source node. In inter zone clustering component special gateway nodes are identified which connects one zone to another through a physical link. Gateway nodes are nodes which receive responses from node of there neighboring zones node. If a node wants to send data it broadcast a link request. Nodes with in its communication range reply with the link response containing their node id and zone ID. After receiving the replies the node generate link state packet (LSP) which contains the node ID of its neighbors in the same zone and zone ID of its neighbor in different zones. The LSP is forwarded to all nodes in the same zone. The node LSPs from other zones will not be stored, because they are only propagated in there own zone. Now the node level topology will be known to the node of its zone. The intra zone routing table is built by using shortest path algorithm. Each LSPs of the nodes contain zone ID on the bases of which node will know the zones which are connected to its zone. After receiving all nodes LSPs, same zone LSP is generated by each node of the same zone. The gateway nodes send the zone LSP to every node in the network. This procedure is performed by every zone and every node stores a list of zone LSPs. So every node will know the zone level topology of the network. To find the shortest path, shortest path algorithm is used and inter zone routing table is built. Given the zone ID and node ID of the destination, the packet is routed based on zone ID till it reaches the destination zone. Then on the bases of node ID the packet is routed.

The advantage of using this protocol is that the reactive component keep some information about the network thus the latency is reduced [2] but disadvantage of ZHLS is that all nodes must have a pre program static zone map in order to function, which may not be feasible in dynamic geographic boundary [2]Because if the node has to determine a route outside the routing zone, it only has to travel to a node which lies on the boundaries of the required destination. But protocol behaves like a pure proactive protocol for large values of routing zone while it behaves like a reactive protocol for smaller values. [2]

2.3.3.2 Zone-based hierarchical link state (ZHLS)

Zone-based hierarchical link state protocol is another hybrid protocol and employs hierarchical structure. In this kind of routing the network is divided in to non-over

Summary

In this chapter, we discussed the different routing protocols and techniques in wired and mobile ad-hoc networks. We discussed the different approaches of routing techniques, and briefly illustrated the differences, advantages and disadvantages of these protocols.

Chapter 3
OVERVIEW OF TECHNOLOGIES FOR RESILIENT WIRLESS NETWORKS

3.1 Introduction:

Over the past few years the Information Technology (IT), Mobile and Telecom industries have rapidly changed and revolutionized, with management of links and nodes connectivity of network infrastructure, performance of emerging services and medium control.

Although wireless networking concepts and technology have been around for decades, there are still many corporate hurdles and technological obstacles to be overcome. During last couple of years, the industry has begun cooperating to modify and integrate the concepts and equipment necessary to make the 802.11x standards for wireless Internet connection as a requirement and necessity of time even for domestic users. The IEEE standard of 802.11b has had widespread acceptance within the various sectors of the networked world. From corporate and government networks to law enforcement, military, academic, retail and home users, everyone is discovering benefits, utilities, services and new ways of conducting business through the use of low-cost WLAN technology**[1].** The low cost of WLAN along with its increased flexibility, mobility, ease of installation and reduced logistical overheads, makes it a natural choice for networking where access is difficult and to reach places where little or no infrastructure exists [1].

3.2TYPES OF NETWORKS

3.2.1 Wireless Networks

Wireless data networks are frequently divided into a number of categories according to how a user views the network. There are mainly two different types of networks, fixed and mobile. Characteristics that are used to define the networks are. Fixed or mobile, point-to-point (PTP) or point-to-multipoint (PTM), licensed or unlicensed, and standard-based or proprietary. Fixed networks for the purpose of definition include all those networks that connect two or more stationary locations as well as systems like 802.11

based networks designed to support “roaming” users (can also be called as nomadic users). A nomadic user is technically a fixed user controlled by the boundaries and limitations of coverage available on the network. In a functional mobile system, the service will be available world over or in wide area, and it supports its use while the user is in a state of motion. Based on the original GSM technology by adding EDGE, GPRS, 1xRTT, and 1xEVDO overlays on to the voice networks, cellular and PCS carriers have taken the first step which may be was a bit tenuous but was in a right direction of providing true mobile data [17].

The speeds at which most of the current networks generally function could not be classed as a broadband in mobile systems. But in recent years the mobile technology took a boost and a later version which succeeded GSM was introduced, which is called Universal Mobile Transmission System (UMTS) mainly implemented in the European region. On this new technology the High Speed Downlink Packet Access (HSDPA) overlay has been added and this advancement enhanced the cellular technology to the level of mobile broadband. In parallel to this, wireless technology has also been improved and enhanced to the level of wireless broadband, by introducing IEEE 802.16 WiMAX technology which theoretically provides connectivity at the range of 30 miles.

3.2.2 Fixed Networks

The simplest network of all is the fixed network which can also be called a point-to-point network. As it is obvious from its name these types of networks are facilities that connect two or more fixed locations such as buildings.

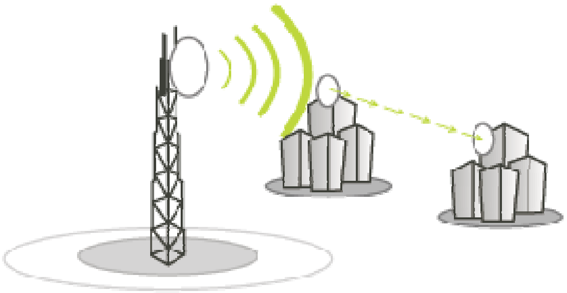


Fig 3.1 A fixed Network with directional Antennas [11]

They are designed in a manner to extend data communications to those locations which are physically separate from the rest of the network. A fixed network can also be considered as a less vulnerable system because of its dedicated nature and fixed addresses are used to communicate within the terminals. A fixed network solution is mostly used to connect buildings together, to provide a network connection to a home user, or to connect multiple elements of one or more networks together. These links may be familiar as the traditional microwave link. In order to achieve range and control over interferences they use highly directional antennas.

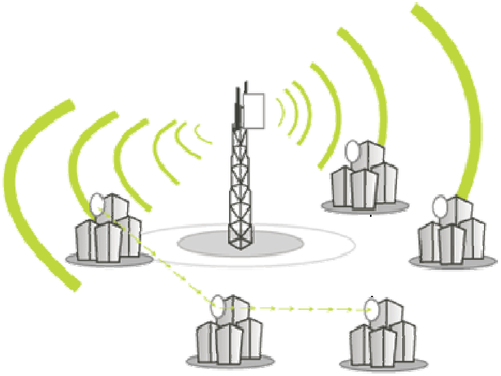


Fig 3.2 Multiple directional Antennas [11]

The range, efficiency and data transfer rate depends on the technology selected for the purpose of communication and data transfer and also effected due to the frequency of operation These links can be designed in such a way that it may cover distances as short as several hundred feet or as long as 20 or more miles, with different capacities which may be under 1 Mbps to nearly 1 Gbps**.** [3]

An alternative type of point-to-point network is a point-to-multipoint network. In these networks a central server station or master does not use an individual antenna for every station separately but instead, it uses a wide aperture antenna, which is commonly known as an omni antenna that is capable of serving many stations in its line of sight. By using this type of antenna and the network connection topology it is easy to provide access of the system to the number of users to share its capacity and they can transfer the data securely and efficiently. To accomplish Point-to-point and point-to-multipoint networks a service provider can use licensed or unlicensed bands which exist in frequency ranges starting from 1 GHz to over 90 GHz [3].

3.2.3 Nomadic Networks

Another option which has been to provide the users to connect to the systems in an environment of point-to-multipoint networks is the network that directly supports a user's connection to the system and is commonly called a nomadic network. In this type of data network instead of connecting buildings together, it connects roaming users or individual computer users to the network existing at a fixed location or at a central place. In the case of a laptop computer or PDA, these computing devices can be considered as partially mobile, because the network is designed in such a way that it offers a low level of mobility to the users in the network. The devices which are currently in use for establishing this type of network are 802.11b, which is a common standard for this type of network, but the 802.11g and 802.11a can also support and facilitate connections with the network in an almost similar environment to the 802.11b.

Nomadic networks are becoming quite common among current network users. As an example, if a user is in a place like a coffee shop or in a super market, the requirement is for giving some entertainment to the customer and the coffee shop or super market owner can get few terminals connected using an 802.11b network; this will not only offer internet access but also improves the look of the shop, and will provide an edge compared to other businesses.

Truly these networks cannot be graded as mobile networks except in very few scenarios. But it can be said that with every thing there are some trade-offs. This is a localized low mobility solution which is fairly easy and inexpensive to implement. But the better aspect of this type of network is that, that there is an unlicensed spectrum available for use. There is a large installed base of customer equipment built to operate on and with the 802.11b Wi-Fi standard that already exists. These were the factors which led the Wi-Fi industry towards the rapid development of all sorts of nomadic networks. Nomadic networks can be as small as a home and others can be as large as a community or may be of a size of a comparatively small city [4].

3.2.4 Mobile Networks

The networks that are designed for true mobility are the most complex in nature, For example a voice-based cellular or a Narrowband Personal Communications Services (PCS) network, the high-speed mobile data network must provide an all time multidirectional coverage, and it is necessary that it must support high velocity mobility. Requirements which are to be fulfilled by a mobile network are neither easily achievable nor inexpensive. Mobile systems require a licensed spectrum that is many tens of megahertz, and also require a technology that can deal with an environment congested due to heavy transmission of different RF which will be the truly an environment for an applied mobile system. The 802.16e, 802.20 and CDMA2000 standards are several of the standards that may eventually bring true broadband mobile data solutions to large areas of the earth. Because of their cost, complexity, and need for interference managed dedicated spectrum, large solutions, will be the most likely owner of these networks.

The focus for the early stage of this research project has been a study of security protocols and mechanisms for wireless networks. Several issues have been identified, and the future research will be to develop a new mechanism to enhance the security and reliability of wireless networks. [5].

3.3 TECHNOLOGIES IN WIRELESS NETWORKS

In this section major types of wireless network technologies 802.11 (Wi-Fi) and 802.16 (WiMAX) and 802.17(RPR) protocols are discussed in detail. The aim of this review is to identify the best wireless technology for a Resilient Packet (Wireless) Ring.

3.3 Wi-Fi Technology (802.11)

Modern Wireless Technology is a viable alternative to Wired Technology. Wireless technology is commonly used for connecting devices in wireless mode.

Wi-Fi (Wireless Fidelity) is a terminology which is used for referring to the IEEE 802.11 communications standard for Wireless Local Area Networks (WLANs) [10].

Wi-Fi Networks can be used to connect computers to each other, to the internet and to the wired network.

Wi-Fi Networks use Radio Technologies to transmit and receive data at relatively high speed. The technology includes [10]:

a. IEEE 802.11a
b. IEEE 802.11b
c. IEEE 802.11g
d. IEEE 802.11i

3.3.1 802.11 Physical Layer

There are three sub-layers in physical layer of 802.11 they are as follows:

a. Direct Sequence Spread Spectrum (DSSS)
b. Frequency Hoping Spread Spectrum (FHSS)
c. Diffused Infrared (DFIR) - Wide angle

3.3.1.1 Direct Sequence Spread Spectrum

The Direct sequence signalling technique divides the 2.4 GHz band into 11 22-MHz channels. Adjacent channels overlap one another partially, with three of the 11 being completely non-overlapping. Data is sent across one of these 22 MHz channels without hopping to other channels.

3.3.2 Data Link Layer

The data link layer possesses two sub-layers those are:

a. Logical Link Control (LLC)
b. Media Access Control (MAC).

3.3.2.1 Logical Link Control (LLC)

The 802.11 uses similar LLC to 802.2 and 48-bit addressing as other 802 LANs, allowing for very simple bridging from wireless to IEEE wired networks, but the MAC Layer is unique to WLANs

3.3.2.2 Media Access Control (MAC)

3.3.2.2.1 Carrier Sense Medium Access with Collision Avoidance protocol (CSMA/CA)

The main features of this are:

a. Listen before talking
b. Avoid collision by explicit Acknowledgement (ACK)
c. Problem: additional overhead of ACK packets, so slow performance

3.3.2.2.3 Request to Send/Clear to Send (RTS/CTS) protocol

a. Solution for “hidden node” problem
b. Problem: Adds additional overhead by temporarily reserving the medium, so used for large size packets only since retransmission would be expensive

3.3.3 Power Management in 802.11

The main features of this are:

a. MAC supports power preservation to extend the battery life of portable devices
b. Power utilization modes
   (i) Continuous Aware Mode
      a) Radio is always on and drawing power
   (ii) Power Save Polling Mode
      a) Radio is “dozing” with access point queuing any data for it
      b) The client radio will be activated periodically in time to receive regular **beacon** signals from the access point.
      c) The beacon includes information regarding stations having traffic, waiting to be transmitted.
      d) The client activates on beacon notification and receives the data
   (iii)Fragmentation
   (iv)CRC checksum
      a) Each packet has a CRC checksum calculated and attached to ensure that the data was not corrupted during its transmission
   (v) Association & Roaming

3.3.4 Present Wi-Fi Network Topologies

The main features of this are:

a. AP-based topology (Infrastructure Mode)

b. Peer-to-peer topology (Ad-hoc Mode)

c. Point-to-multipoint bridge topology

3.3.4.1 AP-based topology

The features of this are:

a. The client communicates through Access Point.

b. Business Software Alliance - Radio Frequency (BSA-RF) coverage provided by an Access Point (AP).

c. Enterprise System Architecture (ESA) -It consists of 2 or more BSA.

d. Enterprise System Architecture (ESA) cell includes 10 to 15% overlap to allow roaming.

3.3.4.2 Peer-to-peer topology

The features are:

a. AP is not required in this particular case.

b. Client devices within a cell can communicate directly with each other.

c. It is useful for setting up of a wireless network quickly and easily.

3.3.4.3 Point-to-multipoint bridge topology

This Topology is used to fabricate an organizational network using this a reliable connection can be maintained from a LAN in one building to a LANs in other buildings even if the buildings are miles apart. The only condition which applies to such a topology that it receives a clear signal if the line of sight between buildings is clear. The range and reception of such a network varies due to variation in thee type of antennas used, types of wireless bridges used as well as the environmental conditions.

3.3.5 802.11x (Wi-Fi) Applications

a. Home user for file transfer or internet

b. Small Businesses or Small Office/Home Office (SOHO)

c. Large Corporations and University Campuses

d. Health Care centres or Hospitals

e. Wireless ISP (WISP)

f. Travellers Guidance

3.4 Wireless Interoperability for Worldwide Microwave Access (WiMAX)

3.4.1 The Family of WiMAX Standards

WiMAX is considered to be a homogenous technology by the people who are not well conversant with its features; on the contrary it is an IEEE wireless standard group's trade name. As a matter of fact we can compare WiMAX with Wi-Fi in a manner that Wi-Fi is not an IEEE standard name rather it is also a trade name which is applicable to a series of IEEE 802.11 standards, that includes: 802.11b, 802.11a, and 802.11g and it may also apply to the 802.11n once the problems in its development have been ratified. The WiMAX technology includes:

a. 802.16 V 2004 (802.16d)

b. 802.16e.

802.16e utilizes Orthogonal Frequency Division Multiple Access (OFDMA) and can serve multiple users simultaneously by allocating sets of "tones" to each user. **[7] [8] [9]**

3.4.1.1 IEEE 802.16d

The IEEE 802.16d is a fixed wireless access technology. It is designed in such a manner that it can become a wireless DSL replacement technology [9]. It generally provides the basic voice and broadband access in such areas where there are no other means to connect the users with internet or broadband such as using copper cable or fiber optics. To serve multiple users in a time division it utilizes Orthogonal Frequency Division Multiplexing (OFDM). It can also serve the purpose as a wireless backhauling solution for Wi-Fi APs and may be so used for cellular networks in future. A round-robin technique is used to manage the traffic in a very efficient and fast manner so that users do not notice any difference and feel that they are simultaneously transmitting and receiving. In certain configurations, like fixed WiMAX, it can replace T-1 because it can be used to provide much higher data rates. It is very useful for corporate subscribers which are of high-value. The fixed WiMAX technology may introduce a degree of nomadic capability and may allow the subscriber to travel with the "Customer Premises Equipment" (CPE) and may allow its use at many other fixed locations. Self-installable CPEs should make it more economical for the users. It operates in 2-11 GHz (Unlicensed frequencies). This version was specifically designed for non line of sight operation and ssupports single carrier that is OFDM 256-FFT. In this case the Fast Fourier Transform (FFT) size is 256 bit which comprises of 192 data carriers, 8 pilots, 56 Nulls [9].

3.4.1.2 IEEE 802.16e

This standard is developed to provide the features which 802.16d lacks those are:

a. Portability
b. Full-scale mobility.

It is based on the same OFDM technology adopted in previous standards and is designed to deliver services across many more sub-channels than the OFDM 256-Fast Fourier Transform (FFT). It also supports single carrier, OFDM 256-FFT and at least OFDMA 1K-FFT. The 802.16e standard also supports OFDMA 2K-FFT, 512-FFT and 128-FFT capability. And operates in licensed frequencies of approximately 11 GHz to 66 GHz [7], [8], [9].

3.4.2WIMAX & WI-Fi Infrastructure

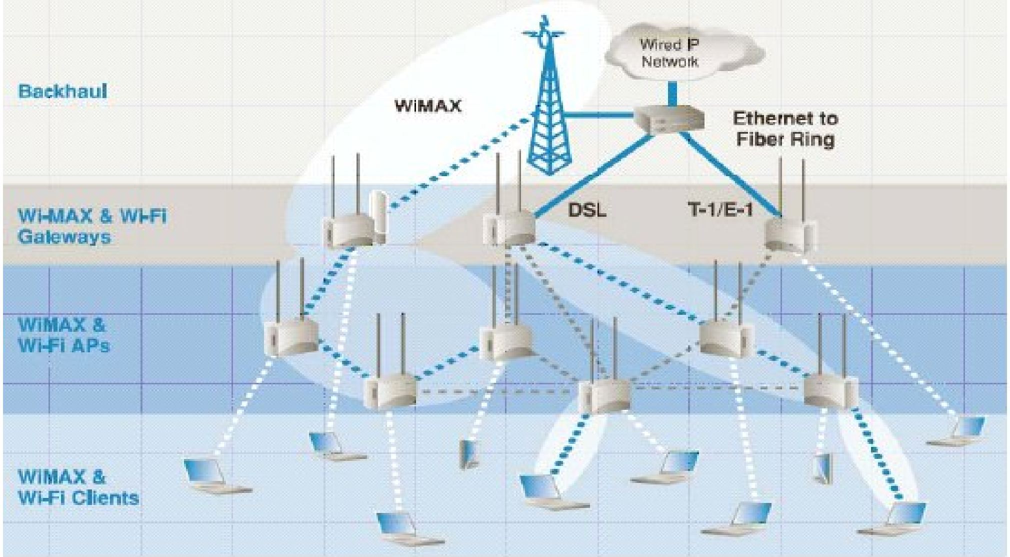


Figure3.3 WiMAX as Client Connection Option [32]

In fig3.3- above, the infrastructure required for the deployment and communication establishment of WiMAX is given. It is in four layers and they are as given below:

a. Backhauling Infrastructure
   (i) Wired IP Network
   (ii) Terminals
   (iii)Fiber Ring
b. WiMAX Gateway
c. WiMAX Access Point
d. WiMAX Client

The basic Infrastructure required for WiMAX and also the equipment for installation is very expensive. It is not very cost effective if we compare it to Wi-Fi, even though the bandwidth provided by WiMAX is much more than Wi-Fi but till today it is a compromise on mobility. In the case of WiMAX, mobility is restricted and is almost equivalent to fixed networks, the only advantage in case of fixed WiMAX is that it can provide a broadband facility at a remote location where fiber networks do not exist [9] [16].

3.4.4 The Protocol Layer of WiMAX

Fig 3.4 represents the architecture of 802.16 standards. It can be seen that the 802.16 standard defines only two layers, the physical (PHY) layer and the MAC Layer which is the main part of the Data Link Layer with the Link Layer Control very often applying the IEEE 802.2 standard[16][29]. The MAC layer is further made up of three sub layers, the Convergence sublayer which describes how wireline technologies such as Ethernet, ATM and IP are encapsulated on the air interface and how the data is classified, the Common Part sub layer which is responsible for bandwidth allocation, connection establishment and maintenance of the connection and the privacy sub layer describes how secure communications are delivered.

The physical layer establishes the physical connection between two sides. It defines the type of signal used, transmission power, and modulation and demodulation kind.

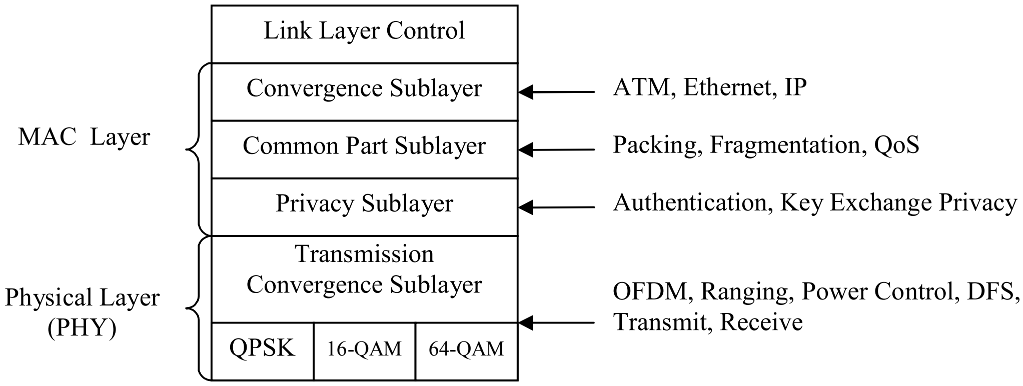

Link Layer Control
Convergence Sublayer
Common Part Sublayer
Privacy Sublayer
Transmission Convergence Sublayer
QPSK
16-QAM
64-QAM
MAC Layer
Physical Layer (PHY)
ATM, Ethernet, IP
Packing, Fragmentation, QoS
Authentication, Key Exchange Privacy
OFDM, Ranging, Power Control, DFS, Transmit, Receive

Figure-3.4IEEE 802.16 MAC and PHY layers **[16]**

3.4.5Security of WIMAX

WiMAX is a highly secure network because it has its own improved security measures. WiMAX has security built into the standard (unlike WiFi).The PHY layer defines how the data will be transmitted physically on the medium.

The privacy sublayer is responsible for implementing security in 802.16 and two main protocols work in this security sublayer, one is an encapsulation protocol which is responsible for encrypting packet data across the fixed Broadband Wireless Access (BWA), and another one for distributing data from Base Station (BS) to Subscriber Station (SS) securely, called Privacy and Key Management Protocol (PKM). RSA public-key algorithm, X.509 digital certificates, and strong encryption algorithm are being used by the PKM protocol to perform key exchanges between SS and BS securely. This Privacy protocol enhanced to fit seamlessly into the 802.16 MAC and accommodate stronger cryptographic methods such as AES otherwise it was based on the PKM protocol of the DOCSIS BPI+ specification.[16] Hence the entire security of the IEEE 802.16 communication or data transmission relies on the Privacy Sub layer (PS) **[15][16]**.

3.4.5.1 Security Risks

a. Longer range makes for a larger “listening area” for attackers.

b. Management frames in the WiMAX are not encrypted which permits, without being in knowledge of the user, allowing an attacker to collect not only the information about clients in the network but also other sensitive network information too[16].

c. The original specifications of 802.16 did not specify an authentication method for base stations (BS). This made the initial version of 802.16 standard vulnerable to man-in-the-middle attacks. To resolve the problem of man – in –the –middle attacks the 802.16e adds EAP authentication for base stations[16].

3.4.5.2 Strengths

d. Ensures interoperability between different equipment of different vendors.

e. Unlike WiFi WiMAX has built in security.

f. Ensures Quality of service (QoS) by providing built in Service level agreements (SLA's) which is the standard to provide different levels of service for a tiered pricing model[16].

g. WiMAX deployment is quicker and cheaper as compared to shorter range technologies.

3.5 IEEE 802.17 Resilient Packet Ring (RPR)

In this section I explain the features, node architecture, general operations, functioning and implementation of IEEE 802.17 which was named by the Work group of 802.17 as Resilient Packet Ring (RPR), because of its implementation of dual ring topology. It is also called Dynamic Packet Transport (DPT). It can have up to 128 nodes in a ring.

3.5.1 Features of RPR

The topology it uses is a "Dual Ring" topology in which outer ring has clockwise data transmission and inner ring has anti clockwise data transmission pattern.

It has an inbuilt routing or transmission technique, that a packet must take the shortest path to the destination [18]

The entire ring of 802.17 is considered to be a single subnet.

3.5.2 Node Architecture

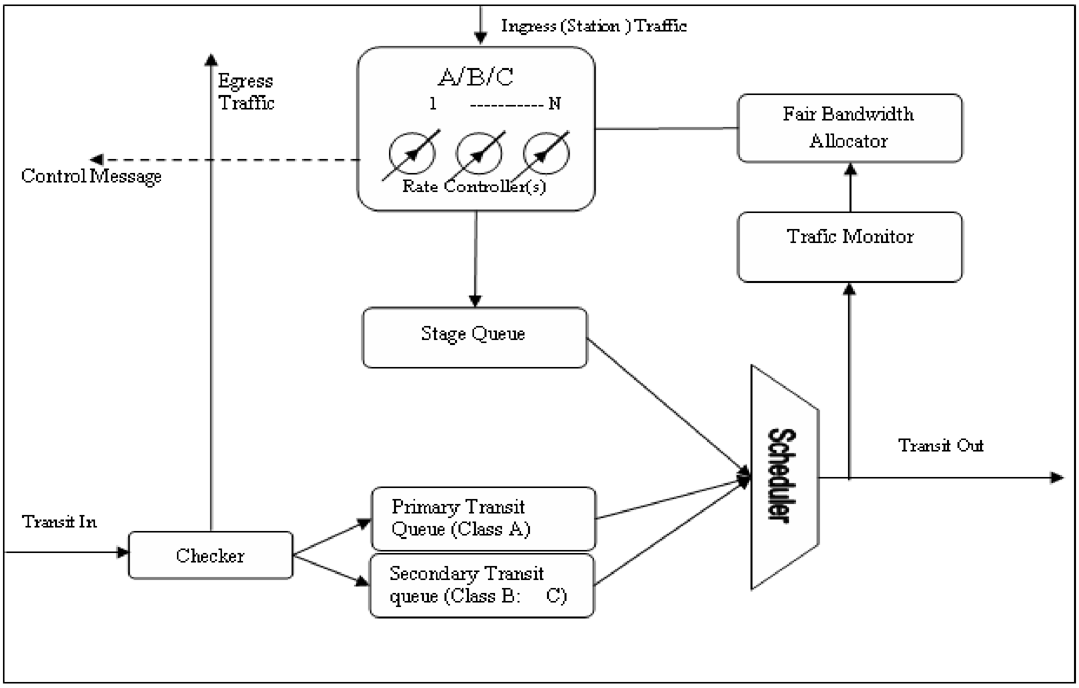


Fig-3.5 Node Architecture of 802.17 (RPR) [14]

3.5.3 RPR Ring Architecture

3.5.3.1 Ring Operation

An RPR network employs two counter-rotating rings for carrying working traffic in opposite directions. Adjacent stations are connected to each other through unidirectional links. Each ring is composed of multiple such links with data flowing in only one direction.

RPR frames carry data from one station to another by traveling around on one of the two rings. All stations in an RPR network are identified by an IEEE 802 48-bit MAC address [18]. A frame starts off on ringlet from its source station and travels around the ring trying to find its destination. Every station that receives this frame checks for the destination address in the frame's header. If the station's address matches the destination address, the frame is removed in moved from the ring and passed on to the client. Therefore, a frame transits stations till it reaches a station which matches its destination address.

The two rings not only carry data but also transfer control information between stations. Control frames convey control information related to topology, protection, and bandwidth control in an RPR network. A control frame is usually sent in the opposite direction to a data frame across the same span.

Since every station usually transits frames sourced by its upstream stations, it makes the upstream stations depend entirely on them to provide adequate bandwidth to transit frames. RPR uses an effective algorithm to ensure fair distribution of bandwidth between all stations. RPR also provides a mechanism by which a lower class of service can reclaim unused bandwidth from upper class of service. This ensures efficient usage of available bandwidth.

3.5.4 RPR Layer Model

The RPR MAC layer is divided into two sublayers which are

- Control sublayer
- Datapath sublayer

The MAC provides a service interface used by MAC clients to transfer data with one or more peer clients on the ring, or to exchange local control information between the MAC and the MAC client.

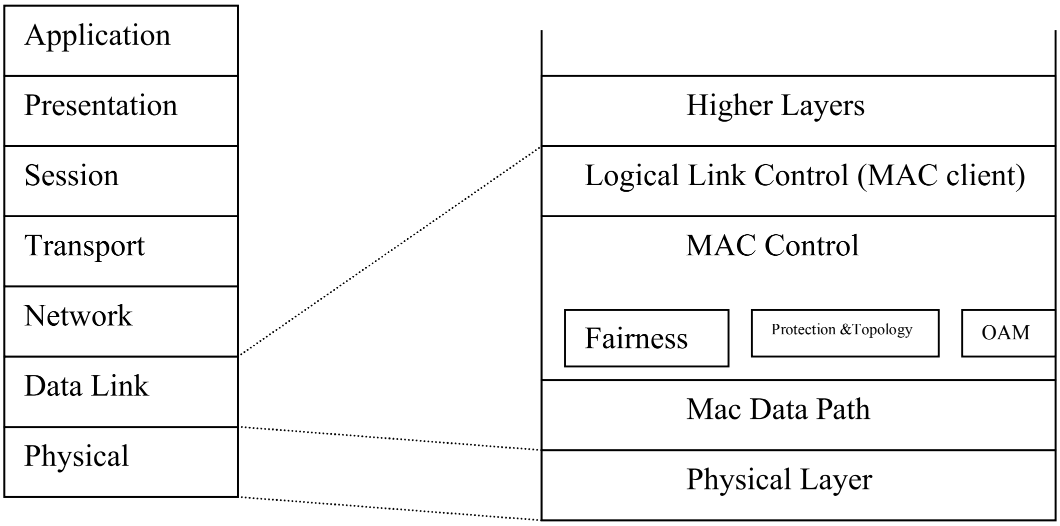


Fig 3.6 MAC Sublayer for RPR

The above fig 3.6 shows where RPR fits into the OSI model. Since an RPR station uses ring selection for sending a frame, it gives an initial impression that it is a layer 3 protocol, however, functionalities like destination address matching, fairness, and bandwidth reclamation, which are related to medium access control layer, make it a layer 2 MAC protocol.

3.5.4.1 RPR MAC_ Station Architecture

3.5.4.1.1 Station Structure

An RPR station consists of two PHYs (one for each span), a MAC and its client. PHYs transmit and receive frames over a span at the physical layer. An RPR frame is sent or received through these PHYs using the primitives defined by its service interface. A frame from the local station can originate from the MAC Client or the MAC's control entity and is passed to one of the datapath entities by ringlet selection. The respective data path then transmits the frame using PHY's service interface.

A transit frame is passed from one PHY to the other after the required processing in the associated MAC datapath entity. A frame whose destination address matches the address of local station is passed to MAC client or MAC control entity.

3.5.4.2 MAC Service Interface

An RPR MAC offers three main classes of service to its client for sending data frames. The services are distinguished based on their commitment for bandwidths, end-to-end delay, jitter.

The classA service provides transfer of traffic with a guaranteed bandwidth and low delay-jitter. Although bandwidth for classA traffic is allocated, some of that allocated bandwidth may be kept unreserved for lower classes to reclaim when needed. MAC rejects any classA traffic that goes beyond the allocated rate. It does however, provide classA related policing indications to its client, which can be used to shape or police its classA traffic before giving it to MAC.

The classB service provides transfer of traffic at or below the committed information rate (CIR) with bounded delay and jitter commitments. All of the classB bandwidth which is allocated can be reclaimed by lower classes if it is not being used. Unlike for classA, MAC does not reject any excess traffic for classB. But instead it delivers it as best-effort traffic with no bandwidth or jitter commitment.

The classC service provides best-effort service with no allocated bandwidth or bounds on jitter or delay. This class of traffic is opportunistic and utilizes any available bandwidth for transmission.

3.5.4.3 MAC Datapath sub layer

This sublayer describes the handling of data traffic in the map. The RPR MAC on a station needs to process two types of data packets: ingress traffic is sent by the MAC client for transmission on the ring and transit traffic coming from the upstream neighbor on the ring meant for some node downstream on the ring. There are two types of transit queuing architectures supported in RPR- single and dual. In single queue architecture, all transit frames are placed in a small queue, called Primary Transit Queue (PTQ). The intent is to temporarily hold transit frames before they are transmitted. To support this intent, transmit logic gives priority to all frames (irrespective of their class) in PTQ over local client's frame.

In dual queue architecture, only classA transit frames are placed in PTQ, while classB and classC transit frames are placed in a larger queue called Secondary Transit Queue (STQ). The transit logic therefore gives priority only to classA transit frames (i.e. PTQ frames) over local client's frames. The frames in STQ are transmitted only when there are no client frames or classA transit frames available for transmission. However, it STQ becomes nearly full the transit logic temporarily gives priority to STQ frames to avoid any loss of transit data.

3.5.4.4 MAC Control Sublayer

MAC control sublayer maintains information related to fairness, network topology, protection, and management, which it uses to control MAC datapath sublayer. The information is kept up-to-date by communication with MAC control sublayers of other MAC's present in the ring via control frames. MAC control sublayer also controls the transfer of data between the MAC and its client.

3.5.4.4.1 Fairness algorithm and protocol

The MAC fairness section defines the access control protocol that ensures fair access to ring resources. The RPR MAC includes mechanisms for detecting the level of congestion on any link in the ring. Each MAC monitors the utilization of the links it is attached to. Locally, the flow control algorithm provides policing indications that control its add traffic. The algorithm also calculates a fair rate at which it expects the upstream stations to transmit. The fair rate information is communicated to the corresponding fairness algorithm on the upstream stations using a protocol. The upstream fairness algorithms use this information along with their local rate statistics to control their ingress traffic.

3.5.4.4.2 Protection database and Topology database

MAC control sublayer maintains information about the failure of links, spans, or PHYs associated with the local station that effects normal operation of a MAC.

It also maintains information about the latest ring topology obtained from stations active in the ring. The topology database is updated whenever the ring topology changes. Topology changes whenever the station is added or removed on purpose or when it fails.

3.5.4.4.3 OAM functionality

Operation, administration, and maintenance (OAM) is an important part for an easily manageable network. OAM gives a network operator ability to add, remove, configure, or manage a station in the ring.

3.6 Features of RPR

In this section a general overview of some of the RPR features are provided

3.6.1 Reliability

RPR network has a ring topology with two counter-rotating rings, both of which carry working traffic. The two counter-rotating rings in RPR are capable of providing a protection path for each other within 50ms of a failure. This feature makes RPR a reliable network in case of node failure.

3.6.2 Quality of Service

A Metro network should be capable of providing quality of service demanded by every class or type of traffic that it transports. An RPR network provides three classes of service to its MAC client. Each RPR station is capable of differentiating between these classes of service at both ingress and transit points in the network.

The QoS guarantees for a class are not only dependent on how a station treats its ingress traffic but also on how that traffic is treated by stations it transits. A station in RPR uses dual queue architecture for doing so.

3.6.2 Dynamic Bandwidth

Another feature of RPR is bandwidth reclamation. If a lower class of service is in need of bandwidth, it is given the freedom of reclaiming unused bandwidth from higher classes. This makes RPR bandwidth efficient.

3.6.3 Fairness

When a network resource is shared amongst a group of stations, fairness becomes important feature to have that network since it directly affects ring utilization.

Since each node or station in RPR also transits frames from upstream stations, RPR has to ensure that upstream are not starved for bandwidth, because if the stations are given the freedom to transmit then its not difficult to imagine that the common ring bandwidth would be unfairly distributed between them. RPR uses a feedback flow control mechanism that achieves fairness amongst its stations. The fairness mechanism is dynamic in nature, in order to maximize the ring utilization.

3.7 Proposal for Metro-Fiber

Layer 2 in OSI reference model is Data link layer. Conceptually Data link Layer (DLL) is further divided into two sub layers those are logical link control (LLC) and media access control (MAC). LLC refers to the functions required for the establishment and control of logical links between local devices on a network, Provides services to the network layer and hides the rest of the details of the data link layer to allow different technologies to interact impeccably with the higher layers. Usually LAN technologies use Logical Link Protocol (LLC). To control the access to the network medium Media Access Control (MAC) uses predefined procedures. Many networks use a shared medium so it is necessary to have rules so that traffic could be managed and there should be no conflict and collision among the packets.

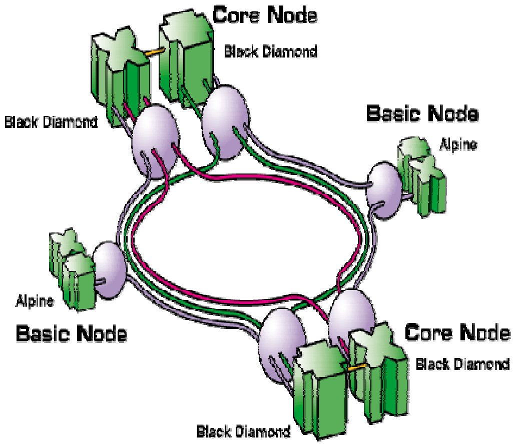


Fig 3.7 Adopted Dual bus ring topology in Metro Fiber Network [19]

Token ring is different from dual bus ring topology, Token Ring uses token passing but “IEEE 802.17 standard is an alternative Layer 2 technology that is optimized to deal with the multi-service transport requirements over metro ring topologies. RPR functionality is built into routers, switches and add/drop multiplexers (ADMs)” **[19]**. In my last research paper I anticipate that once the RPR will be adapted and integrated then it may be possible to transmit Voice Video and Data in a single frame instead of separating them from*bit* each other and transmitting them in different chunks using additional bandwidth and additional protocols using the dual-ring architecture.

3.8 Proposed MAC Replacement for Wi-Fi

It was a fact that 802.17 were designed to replace the MAC layer in metropolitan fiber networks. And also to provide the services and functions with flexibility while maintaining Quality of Service. It was considered a worthwhile experiment to consider the possibility of adapting dual bus ring topology in Wi-Fi layer2, the design which is in my consideration may provide encapsulated voice video and data in one packet and its transmission over a wireless network using this topology may be easier and efficient.

It may be very early to say that whether this will be a functional proposal or not but I am pretty sure that by the end of this document it will be proved that the effort is not a waste and I am sure it will be fruitful to help me further establish my work for the PhD Module.

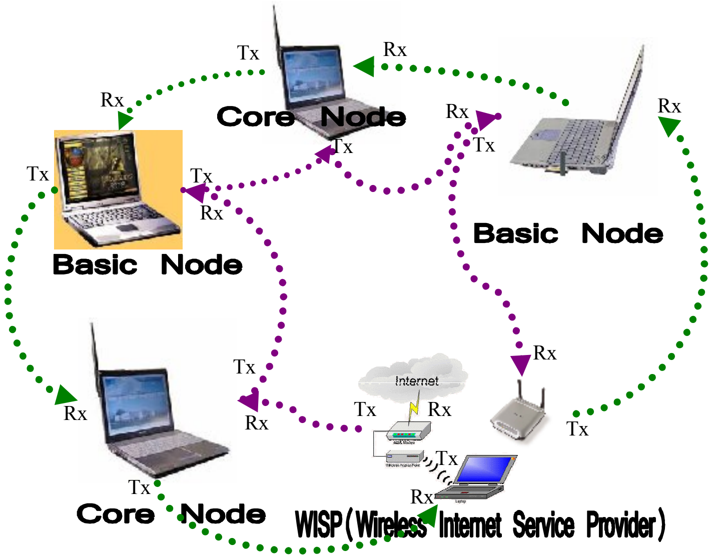


Fig-3.8 Proposed Topology for Wi-Fi Resilient Packet Ring [19]

I hypothetically proposed a design for the topology to be used and implemented for the experiment, given in Fig-3.8. In this two core nodes and two basic nodes are shown although in case of wireless nodes there is no such restriction as any node in the network can be master or server node and any node can be basic or slave node but it is thought that for forming a basic network it is required that there must be one master node and one backup for the master node should be present just to give the idea of dual ring as if there is a failure their will be a backup, so keeping that in mind two core nodes will act as master and secondary master respectively. Master node can be initiator of the ring and may perform the checks for the admission of the nodes in the group **[19]**. Users in the group are called basic nodes there task is just to send and receive the data and they do not have to perform the actions which a master node is suppose to perform for example transmission control over the medium and network management etc, but to maintain the ring topology it will retransmit the packets over the medium. For the fail safe transmission and maintaining the flow control it may be the case that there are more than one core nodes so that if one core node stops working there will be another core node to take over to prevent network failure or denial of services.

Summary

In this chapter different types of networks their features and usage have been discussed. Also WiMAX and Wi-Fi and Resilient packet ring have been discussed in detail. The purpose of this review was to find the best approach designing a faster, more efficient and reliable network topology, and for this purpose most of the features, advantages, disadvantages, architecture design and flaws in present topologies are discussed in detail for WiMAX, Wi-Fi and RPR. Wi-Fi and WiMAX and RPR are likely to be the future of wireless networking and communication. They provide mobility and access of the network and communication facilities in such areas where wired communication can not be provided. In addition to the technical features of wireless networks, economic factors have also been considered to determine which type of wireless network are more economical. But, my design considerations are mostly concentrating towards RPR network topologies used.

Chapter 4

DEVELOPMENT OF THE DBR ALGORITHM

4.1 Introduction

There are many ways of doing routing and one of the purposes in the parent context efficiently link randomly placed wireless nodes into a logical ring structure. The Decision based routing algorithm identifies next hops on the basis of physical distance and routing signal strength, to give a ring topology. In the simplest of networks this would be done on the basis of signal strength, for sophisticated networks GPS or real location could be used.

4.2 Decision Based Routing Tree Algorithm

The decision based routing algorithm is a technique to allow the use of the Packet Ring Architecture in a wireless environment.

There are two portions of the algorithm i.e.

- Decision Tree Formation
- Routing based on the Decision Tree.

4.2.1 Decision Tree Formation

The tree formation is based on the distance of nodes from each other in the network. Initially a tree is formed and the connection graph identifies loops and a further mechanism is used for including nodes which are not part of the ring after the first iteration of the mechanism. Every parent node has a left descendent and a right descendent. The left descendent node is the nearest node to the parent and the right descendent node is the second nearest. In the first step we start from the first node, calculate the distance of this node to all other nodes in the network and on the basis of the calculated distance, the algorithm decides on the left descendent or the right descendent.

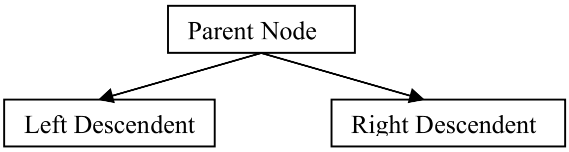


Fig 4.1 Tree Formation

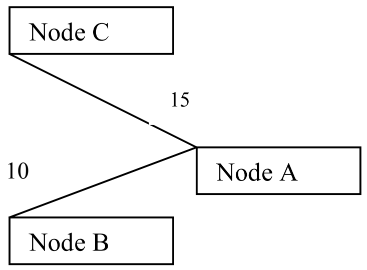


Fig 4.2 an example

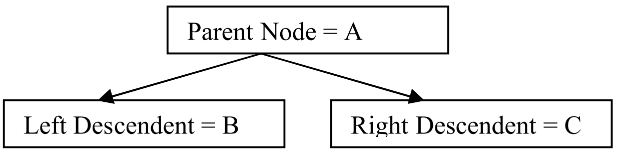


Fig 4.2 b Formation of the tree

If we have three nodes, as in the network of fig 4.1, the decision tree formation will be based on the distance of each of the descendent nodes to the parent node. For example, we take the node A and calculate its distance to Node B and to Node C, get Node B is 10 units away from node A and Node C is 15 units away from Node A so the left descendent will be B and the right descendent will be C, [Fig 4.2].

4.2.1.1 Addition of Missed Nodes (Completion of Tree)

After the first run of the tree formation, i.e. when the algorithm calculates all the descendents, to make sure that the ring is established and all the nodes are reachable, another check is run i.e. all the nodes in the network should be a descendent of at least one node. If any of the nodes are not the descendent of any node in the network, then the distance of these node to all other nodes are obtained and the nodes become the right descendent of their closest node.

In this way the nodes will be added in the network. Also the algorithm marks the parent node, so that other nodes not added to the network do not replace existing parent nodes.

If there is more than one node which were initially excluded then these nodes will be added in the same way, but if the shortest node is already marked, i.e. another node is added as right descendent, and then this node will be added to the second closest node. This algorithm is a left descendent driven algorithm, meaning that tree formation and packet delivery are mostly based on left node, e.g. when a tree is made the nearest node is added as the left descendent .During packet delivery if none of the descendents is the destination node then any packets will be forwarded to the left descendent. So, adding a node in the network as a right descendent will not effect the current routing, and this node will be reachable.

4.2.1.2 Prevention of Multiple loop Formation

As this algorithm is based on the distances and it is much possible that a node can have same parent and descendent node which can cause the formation of small loops. To avoid these loops, we make sure that the descendent of a node is not the same as its parent node and in case we have such a situation then the second nearest node is assigned as the left or right descendent.

4.2.2 Routing Based On the Decision Tree

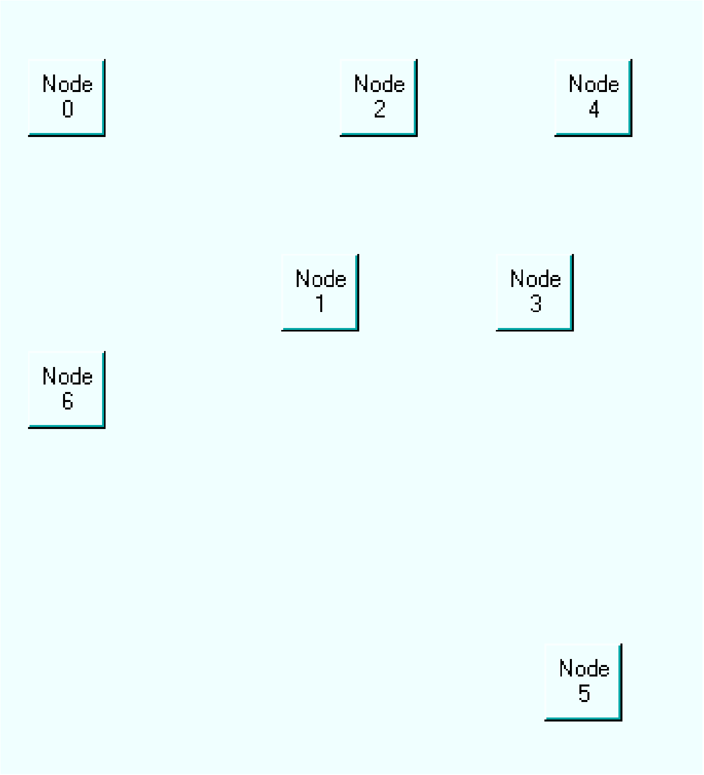
Each packet has the originating and destination addresses, plus the information of nodes visited by this packet. A packet originated/received by a node is transferred to one of its descendents based on some checks, since every node knows its left descendent and right descendent.

When a node receives a packet it is put in a queue in the node and when this packet reaches the top of the queue, the receiving node checks the destination. If the node is the destination node then it processes the packet accordingly, otherwise it checks its left descendent and right descendent and if one of them is the destination of the packet the node transfers the packet to that node. If this is not the case then the receiving node checks the nodes visited by the packet. If any of the descendent nodes have not been visited by this packet before arriving at the current node then the packet is transferred to a node that has not been visited by the packet. If both of the descendents have not been visited by the packet then the receiving node will transfer the packet to the left descendent by default.

For loop prevention the hop count can be used. When a node sends a packet, it will contain a hop counter which will be incremented by every visited node. Every node will also contain a threshold value, i.e. when a packet reaches a node with hop count more then the threshold the node automatically discards the message a “Message Discarded” packet will be sent to the originating node.

4.2.2.1 Example Routing Tree Formation:

In fig 4.3 a complete example of the tree formation for 7 Nodes.



[Fig 4.3] Node Positions

4.2.2.2 Distance Calculation:

Distance of each node with all other nodes is calculated, and based on these calculated distances the tree formation takes place. Following is the example of the calculated distances for 7 Nodes.

Node 0

Distance with Node 0: 0 units

Distance with Node 1: 10130 units

Distance with Node 2: 160 units

Distance with Node 3: 10240 units

Distance with Node 4: 270 units

Distance with Node 5: 90265 units

Distance with Node 6: 22500 units

Node 1

Distance with Node 0: 10130 units

Distance with Node 1: 0 units

Distance with Node 2: 10030 units

Distance with Node 3: 110 units

Distance with Node 4: 10140 units

Distance with Node 5: 40135 units

Distance with Node 6: 2630 units

Node 2

Distance with Node 0: 160 units

Distance with Node 1: 10030 units

Distance with Node 2: 0 units

Distance with Node 3: 10080 units

Distance with Node 4: 110 units

Distance with Node 5: 90105 units

Distance with Node 6: 22660 units

Node 3

Distance with Node 0: 10240 units

Distance with Node 1: 110 units

Distance with Node 2: 10080 units

Distance with Node 3: 0 units

Distance with Node 4: 10030 units

Distance with Node 5: 40025 units

Distance with Node 6: 2740 units

Node 4

Distance with Node 0: 270 units

Distance with Node 1: 10140 units

Distance with Node 2: 110 units

Distance with Node 3: 10030 units

Distance with Node 4: 0 units

Distance with Node 5: 40005 units

Distance with Node 6: 22770 units

Node 5

Distance with Node 0: 90265 units

Distance with Node 1: 40135 units

Distance with Node 2: 90105 units

Distance with Node 3: 40025 units

Distance with Node 4: 40005 units

Distance with Node 5: 0 units

Distance with Node 6: 22765 units

Node 6

Distance with Node 0: 22500 units

Distance with Node 1: 2630 units

Distance with Node 2: 22660 units

Distance with Node 3: 2740 units

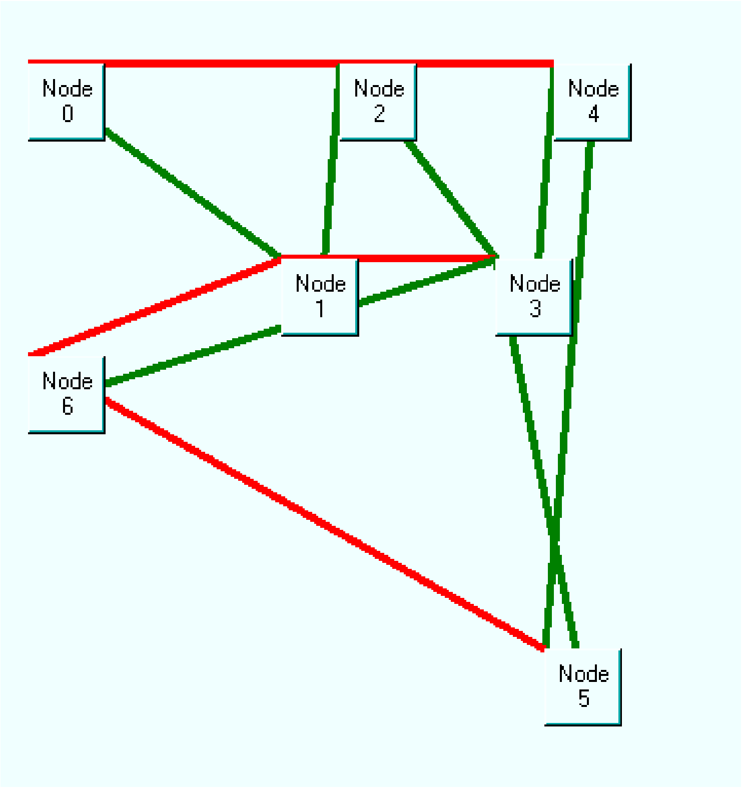
Distance with Node 4: 22770 units

Distance with Node 5: 22765 units

Distance with Node 6: 0 units

4.2.2.3 Decide Left Descendents and Right Descendents

Based on all the node distances, the left descendent and the right descendents are obtained as in fig 4.4.



[Fig 4.4] Left and right descendents based on node distances

According to the algorithm all the nodes define their left and right descendents. Fig. 4.5 shows the tree formation with endless loops and unreachable nodes.

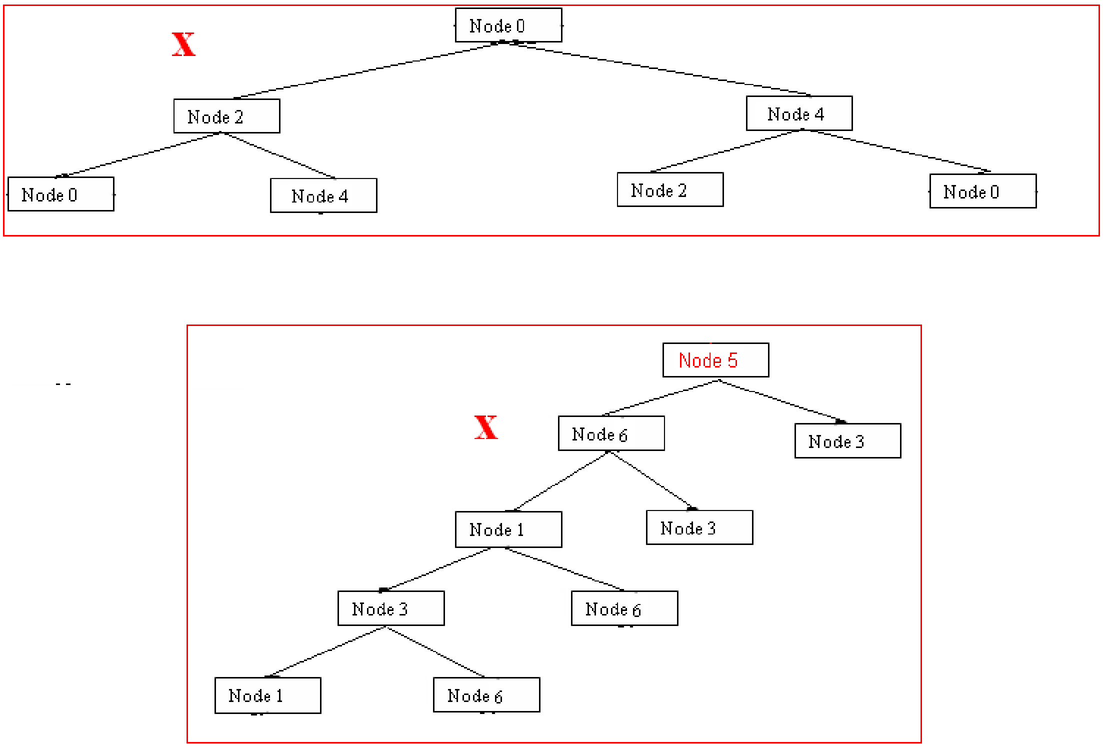
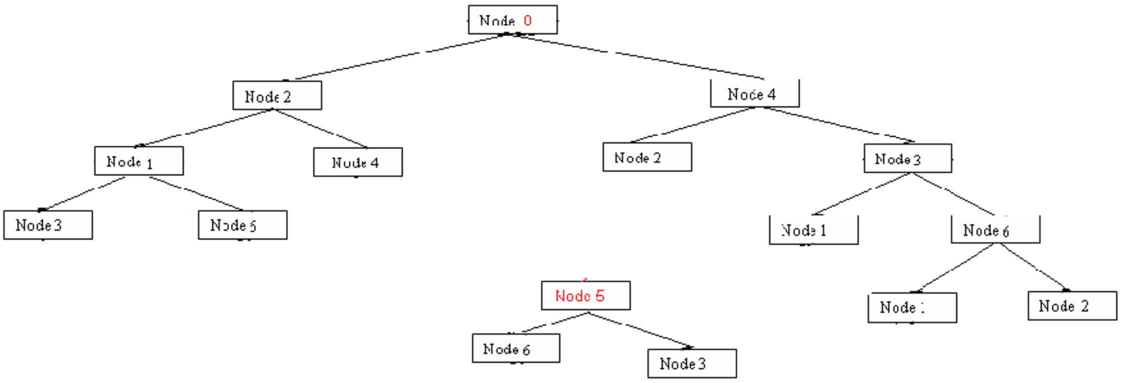


Fig 4.5 Tree with endless loops and unreachable Nodes

In fig 4.5, the nodes 0, 2, and 4 have formed an endless loop which means that if a packet is sent from node 0 to node 1 it will move in a loop between nodes 1, 2 and 4. Secondly Node 5 is unreachable because it is not defined as the descendant of any node. As discussed above in the algorithm for loop prevention the descendants of all nodes whose descendents are the same as the parent are calculated again, excluding the already assigned two descendents from the distance list. Fig 4.6 shows the tree formed after applying this check.



[Fig. 4.6] Tree without the endless loops

To cater for the unreachable node, as explained above, we add this node as the right descendent of its nearest node. Fig 4.7 shows the complete tree formed.

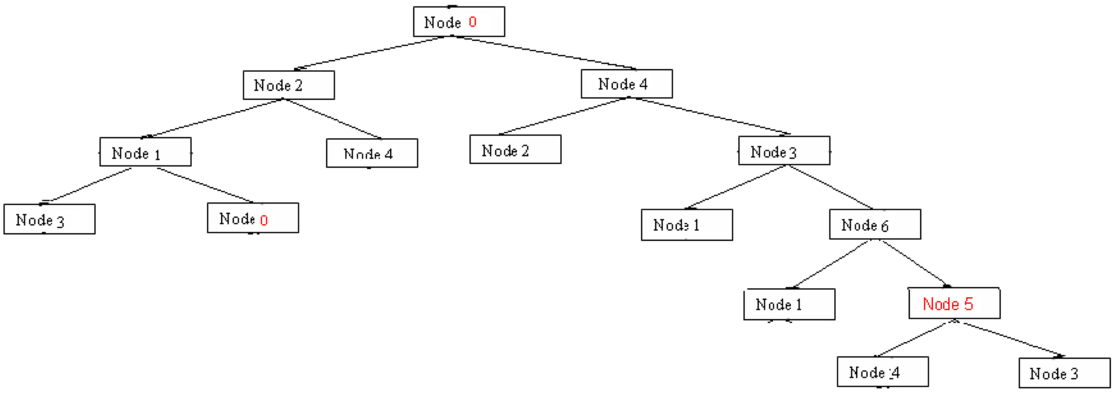


Fig. 4.7 Final Tree

Hence;

For node 0 the left descendent will be node 2 and the right descendent will be node 4.

For node 1 the left descendent will be node 3 and the right descendent will be node 0.

For node 2 the left descendent will be node 1 and the right descendent will be node 4.

For node 3 the left descendent will be node 1 and the right descendent will be node 6.

For node 4 the left descendent will be node 2 and the right descendent will be node 3.

For node 5 the left descendent will be node 4 and the right descendent will be node 3.

For node 6 the left descendent will be node 1 and the right descendent will be node 5.

4.2.2.4 Routing a Packet:

Source node = 0, destination node=6 no of nodes = 7 (As Fig 4.5).

1- Start from the originating node i.e. “node 0”.
2- Check left as well as right descendent node for the destination point i.e. “node 6”. (None of the them is destination so move to the left descendent node i.e. node 2
3- Add node 0 in the visited node Array.

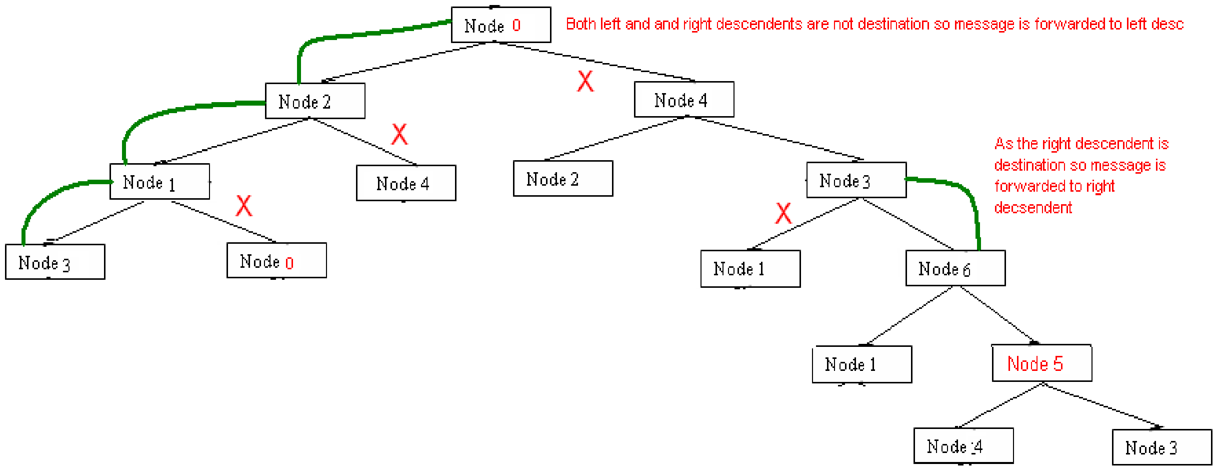

Node 0
Both left and and right descendents are not destination so message is forwarded to left desc
Node 2
X
Node 4
Node 1
X
Node 4
Node 2
Node 3
As the right descendent is destination so message is forwarded to right decsendent
Node 3
X
Node 0
X
Node 1
Node 6
Node 1
Node 5
Node 4
Node 3

Fig 4.8 Routing Decision 1

4- Check the left as well as right node of node2.

5- None of them is a destination.

6- Move to the left descendent i.e. node 1 which is not in the visited node list .

7- Check the left and right descendents of node1, none of them is a destination so move to the left node i.e. node 3 and add node 3 in visited node list.

8- For node 3 check the right and left descendents, since the right descendent is the destination node move the message to this node.

4.3 Implementation of the Algorithm

The algorithm was implemented in a simulation using a C/C++ dot Net platform and was run for different network scenarios.

The following are algorithms, the scenarios and the results

4.3.1Algorithms Implementation:

```
Tree Formation()
Start
  Loop j ←1 to No Of  Nodes   ;; first of all calculate the distance with all nodes
     Loop i  ← 1 to No of Nodes
```

```
        Node[j].Distance[i] = CalculateDistance(node[j],node[i]) ;; Calculates distance between Node[j] and Node[i]
     End Loop
  End Loop
;; Calculate the shortest of all the distances and make the corresponding node the left node.
Integer smallest
  Loop j  ←1 to No Of  Nodes
      Loop i  ← 1 to No of Nodes
         if(node[j].Distance[i]< smallest)
           smallest=i ;
      End Loop
      node[i].LeftNode=node[shortest]
  End Loop

  Loop j  ←1 to No Of  Nodes
      Loop i ← 1 to No of Nodes ;;comp the distance with all nodes to find closest node
         if(node[j].Distance[i]< smallest AND smallest > node[j].LeftNode)
           smallest=i ;
      End Loop
      node[i].RightNode=node[shortest]
  End Loop
 Loop j ← 1 to No of Nodes
   Loop i←1 to No of Nodes  ;; Search for the nodes outside loop
      If(node[j]== node[i].leftNode OR node[j]==node[i].rightNode)
          Flag==0;
         Continue
      Else
           Flag==1;
   End Loop
   Loop k←1 to No Of Nodes   ;; Add the nodes outside the loop with
```

```
      If( node[node[j].NearestNode].rightDescendentNodeCheck==FALSE)
         node[node[j].NearestNode].rightDescendent==node[j]
        node[node[j].NearestNode].rightDescendent==TRUE
     else

find2ndClosestNode(node[node[j].NearestNode].rightDescendentNodeCheck==FALSE)
           node[node[j].NearestNode].rightDescendent==node[j]
        node[node[j].NearestNode].rightDescendent==TRUE
     END IF
;; Preventing small loops formation
If (Node[i].parent == Node[i].leftdescendent or Node[i].rightDescendent )
      Node[i].descendent== CalculateSecondNearestNode(Node[i])
  END LOOP
END

RoutingPacket()
Start
  Integer OriginatingNode
  Integer DestinationNode
  Integer CurrentNode ← OriginatingNode
  Integer  NodesVisited[]
  Integer NodeIndex=0
  Integer HopCountThreshold
  Integer HopCount=0
  While( CurrentNode NotEqual DestinationNode AND HopCount LessThen HopCountThreshold )
    NodesVisited[NodesIndex]=CurrentNode
     NodesIndex++
    If(CurrentNode.LeftNode==DestinationNode)
      CurrentNode=CurrentNode.LeftNode
    Else if (CurrentNode.RightNode==DestinationNode)
      CurrentNode=CurrentNode.LeftNode
```

```
   Else If( NotExistsVisitedNodes(CurrentNode.LeftNode))
     CurrentNode=CurrentNode.LeftNode
   Else
     CurrentNode=CurrentNode.RightNode
   End If
   HopCount++
 End Loop

END
```

Summary

In this chapter the development of the algorithm has been explained. There are two steps. Decision Tree Formation and Routing based on Decision Tree. The routing based on the decision tree has been explained in a step by step manner and the pseudo code for the algorithm has also been provided.

Chapter 5

PERFORMANCE EVALUTAION

The first step in the performance evaluation of any new protocol is to benchmark this against an existing protocol. For this purpose we intend to use the Location Aided Routing (LAR) algorithm. The justification for using LAR for benchmarking purposes is as follow

a) Although LAR is mostly used for mobile nodes, LAR is used in the simulation with the velocity parameter set to zero (static nodes). This implies that any overhead incurred by LAR due to node movement will also be close to zero.
b) Also, since both LAR and the DBR are reactive algorithms and both rely on distance to destination, then it is deemed that LAR should prove adequate for use as a benchmarking tool for DBR.

5.1 Simulation Implementation

5.1.1 Simulation Environment

The simulation was developed in C# dot net platform. This environment was preferred since it lends itself to rapid development, when the only possible way to get any issues fixed is by redeveloping the algorithm code and doing the testing.

For LAR an already developed C# simulator was used.

5.1.2 Simulation Limitations

Although the simulation is efficient for routing and network establishment testing, the performance modelling is a bit tricky to analyze, because both the simulators have the limitation that only one packet can be sent at a time so the routing and the network formation is analyzed efficiently, but for the measurements like congestion control, security etc the simulator needs some more development.

5.1.3 Simulation Features

As already discussed the main focus of the simulator is to analyze the routing and network generation so all the features related to these topics are added in the simulation, these are:

- Node` creation so that the required no of nodes can be incorporated in the simulation.
- Ring Generation, i.e. the network is created and the routing for all the nodes is defined.
- Nodes Mobility, i.e. the nodes can be moved and the effect of the new position on performance can be analyzed.
- The simulator calculates the time taken for each task and saves all the inputs for analysis.
- Packet Delivery; for a given number of packets the simulator picks a random node as the source node and a random node as the destination node and routes the packet to the destination.
- After running the program the simulator saves the results for analysis.

5.2 Simulation

The Simulations are created using Microsoft Visual Studio.Net. Fig 5.1 explains how the simulation works. The simulation is created according to the proposed algorithm. The simulation runs according to the following steps:

- After running the simulation nodes are created as specified.
- When nodes are created, the simulation calculates the distance between the nodes.
- According to the distances, the shortest distances are calculated and according to the algorithm the left descendents and right descendents are assigned.
- When all the left descendents and right descendents are defined according to the algorithm the logical ring is established.

- To run the simulation, random packets are generated from a random node to a random destination.
- After a specified no of packets are generated, a document is generated listing the number of packets sent, received, failed and time taken to deliver a packet.

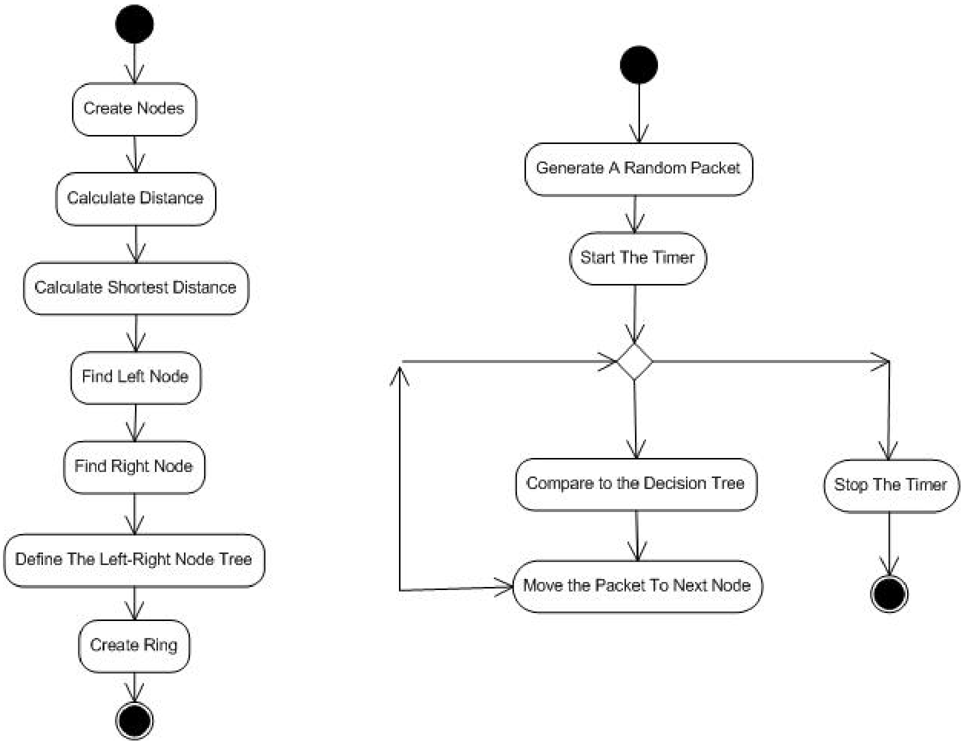


Fig 5.1 Activity Flow

5.3 Simulation Results

Different test scenarios were run and the mean delay and throughput of the network were analyzed. Also, the effect of disabling a node and not allowing the traffic transfer to pass through it was analyzed.

The results of the simulation are compiled in the following table. Different scenarios were run on the simulation keeping the nodes constant and varying the no of packets, and vice versa keeping the number of packets constant and varying the number of nodes.

Table 5.2 shows results with different number of nodes and the number of packets constant at 500.

Sent Packets	No Of Nodes	Lost Packets	Received Packets	Mean Delay DBR	Mean Delay LAR	Throughput DBR	Throughput LAR	Over Head LAR	Over Head DBR
500	5	0	500	**20**	**26**	**2.484**	**6.784**	**750**	**161**
500	10	0	500	**32**	**40**	**6.8708**	**15.422**	**1200**	**695**
500	15	0	500	**35**	**48**	**11.5008**	**22.814**	**2000**	**1553**
500	20	0	500	**39**	**54**	**15.5896**	**25.745**	**4000**	**2880**
500	25	0	500	**41**	**62**	**21.1922**	**29.732**	**7000**	**4372**
500	50	0	500	**46**	**70**	**44.7182**	**51.2442**	**57000**	**17540**

Table 5.2

To calculate the throughput and mean delay the following formula were used.

Throughput = Amount of data transferred/time taken

$$\textbf{Mean Delay} = \left[\sum_{i=1}^{N} \textbf{(Time Taken by Node (i) to Deliver a Packet)}\right] / N$$

Figure 5.3 shows the throughput calculated for the Decision Based Routing algorithm.

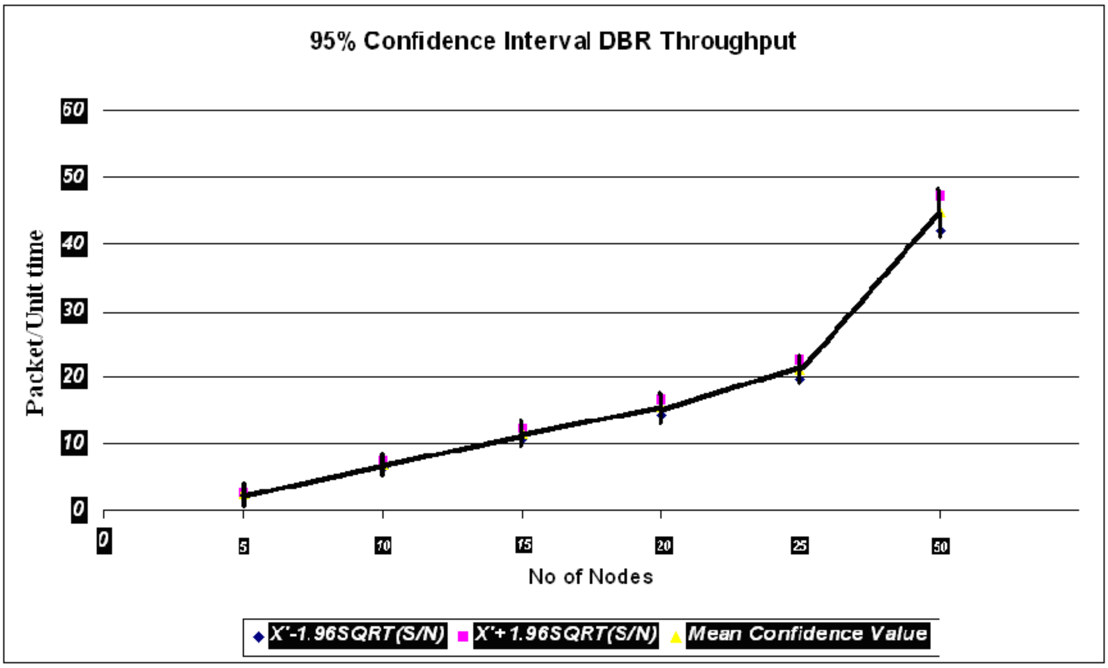


Fig 5.3 Throughput DBR

Fig 5.4 shows the Mean Delay calculated for the Decision based routing algorithm

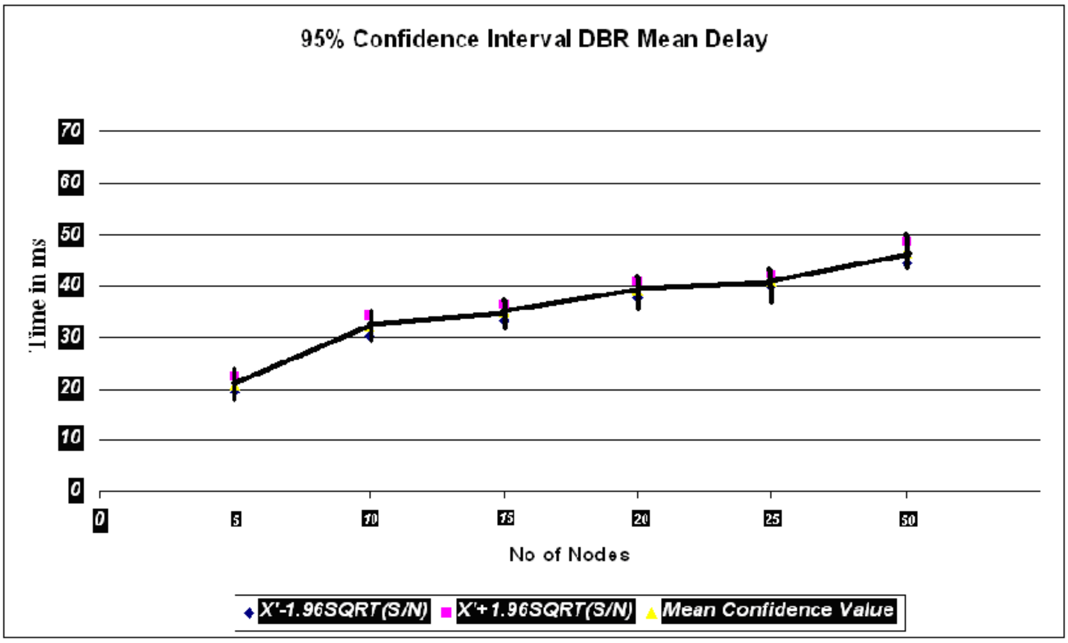


Fig 5.4 Mean Delay DBR

The following figure [fig 5.5] shows the throughput calculated for LAR.

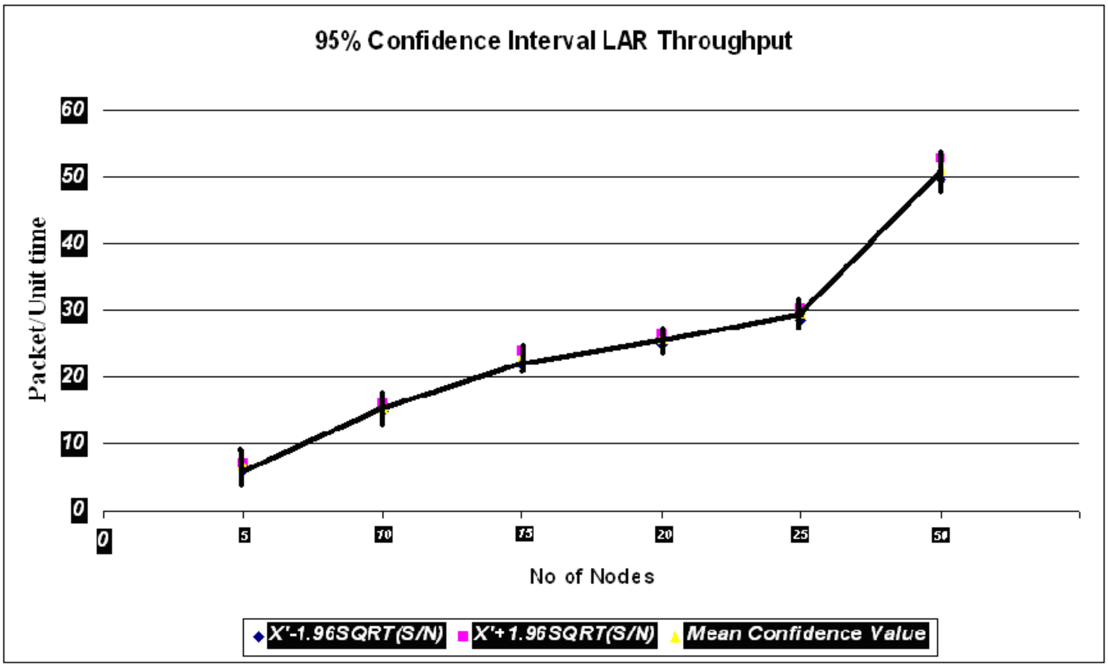


Fig 5.5 Throughput LAR

The following figure 5.6 shows the Mean Delay calculated for LAR.

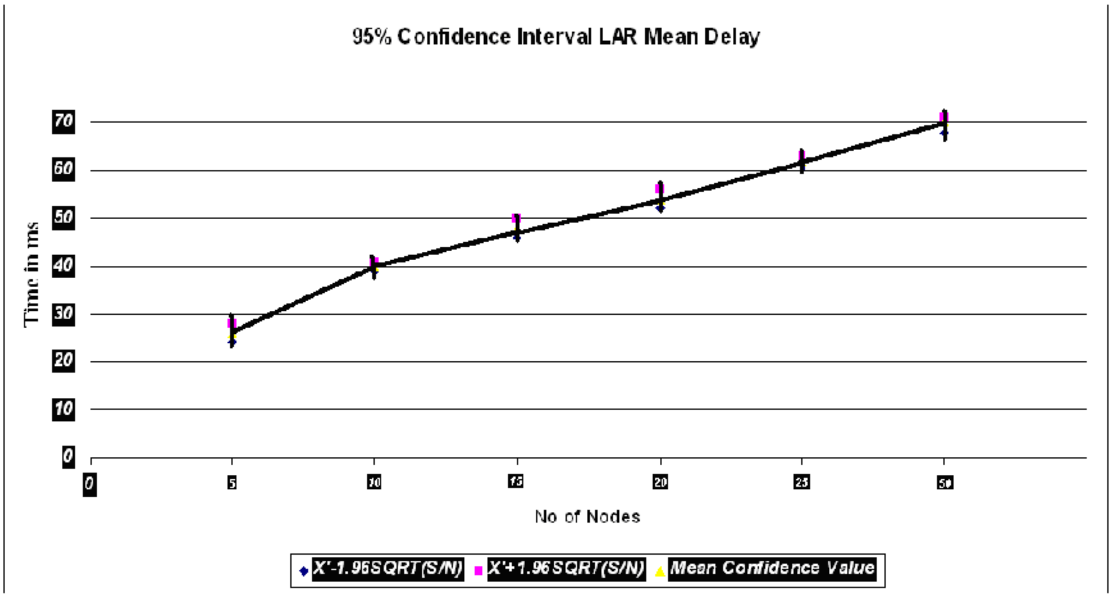


Fig 5.6 Mean Delay LAR

The fig5.7 compares the through put of both algorithms.

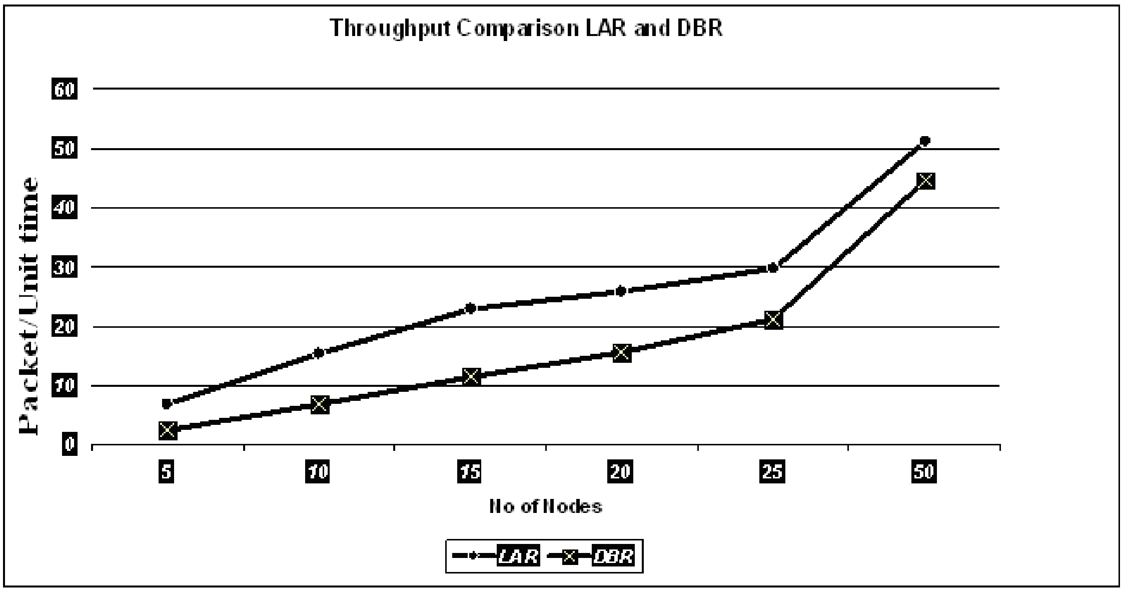


Fig 5.7 Throughput Comparison LAR and DBR

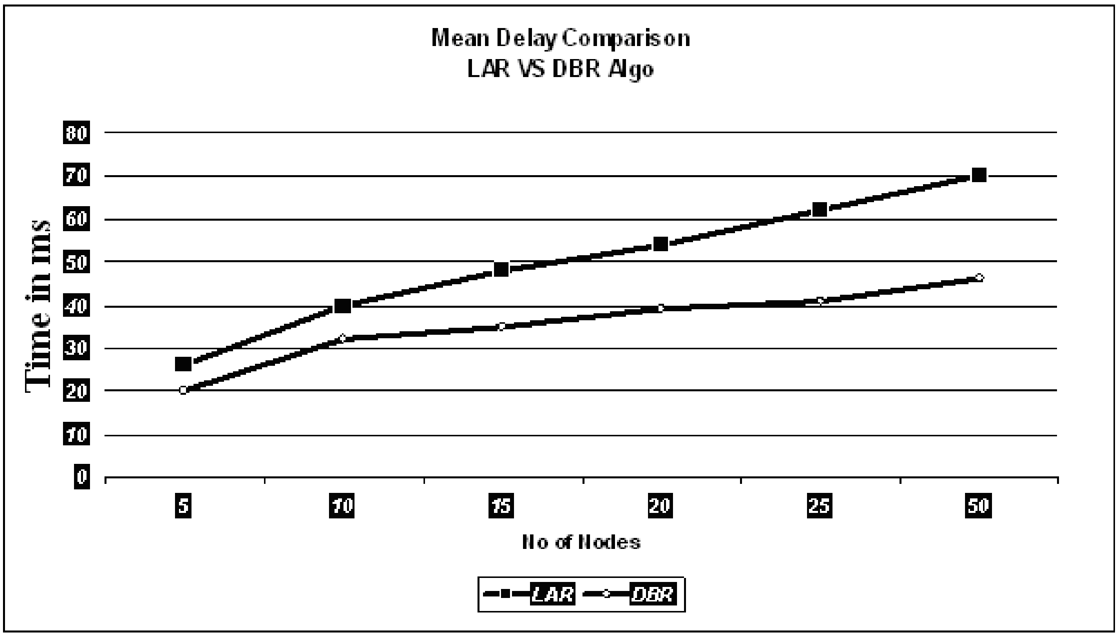


Fig 5.8 Mean Delay Comparison LAR and DBR

Overhead Comparison

The overhead of both the algorithms is calculated in terms of the removal of a node from the network or a change in a node's location.

For the DBR algorithm when a node leaves the network or the descendents change their position the algorithm calculates the new descendents/parents for the effected nodes that are affected by changes.

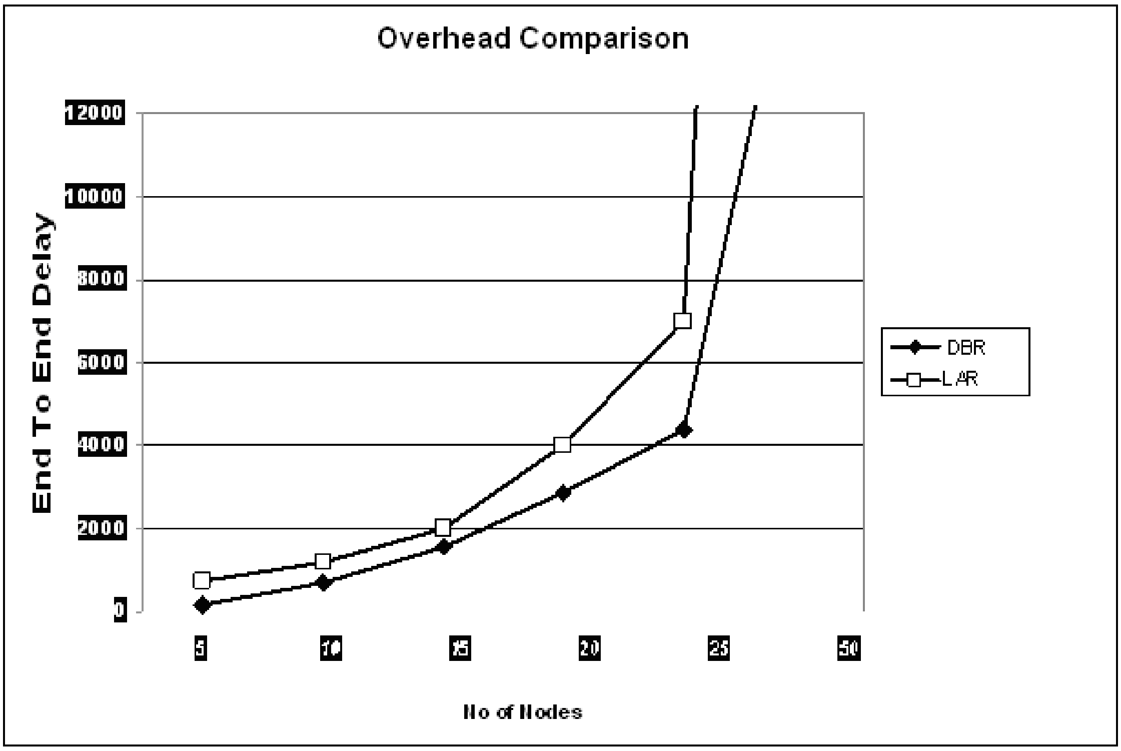


Fig 5.9 Overhead Comparison LAR and

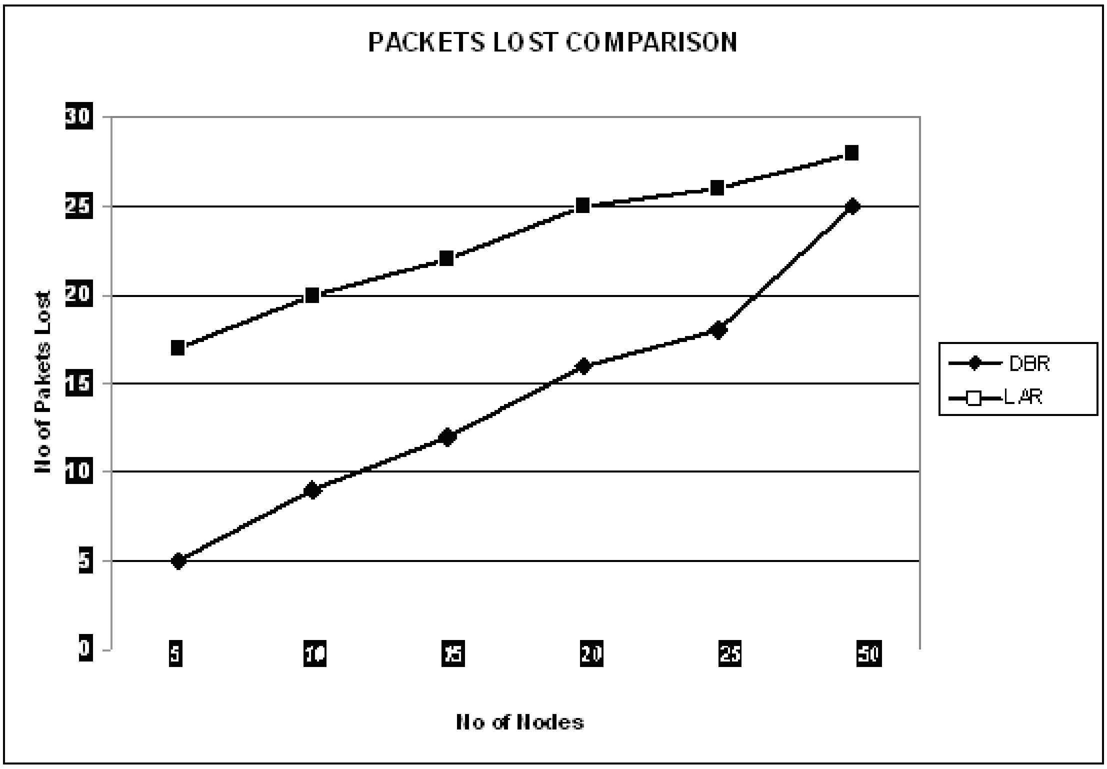


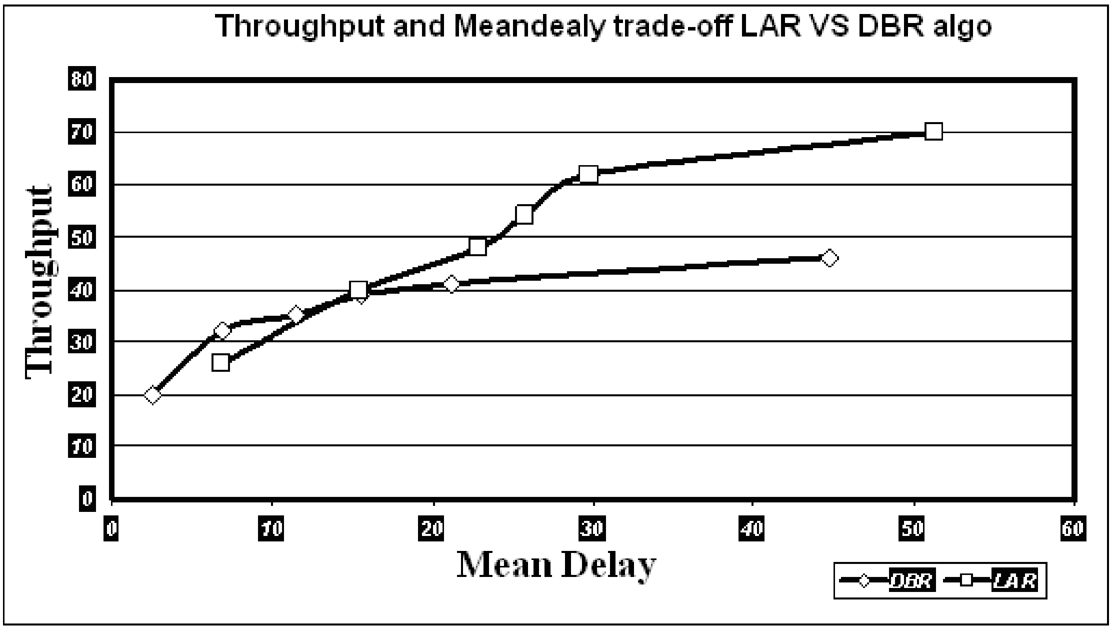
Fig 5.10 Packets Lost Comparison LAR VS DBR

5.3.1 Results:

Table 5.2 show results of DBR compared to the LAR algorithm. One can clearly see that the packet loss is zero because of the efficiency of the algorithm. The simulation is run under ideal conditions, but the results suggest that DBR gives a lower packet loss. Fig 5.4 and Fig 5.6 show the mean delay of both algorithms separately, and Fig 5.8 shows the comparison of both. It clearly shows that the DBR algorithm takes less time to transfer a packet from one node to another because the route is predefined and the node has to only decide between its two descendents i.e. to which node the recipient node has to transfer the packet whereas in LAR we need to calculate the expected and objective zones which may include many nodes. For the throughput there is a linear relationship between the number of nodes and the number of packets transferred.

5.4 Performance Comparison of LAR and Decision Based Tree Algorithm

If we compare the mean delay and throughput trade-off of both algorithms, [Fig 5.11], shows that DBR gives a better performance than LAR over most of the range shown.



[Fig 5.11] Throughput and Mean Delay Comparison LAR and DBR

The following are the main points which indicate that DBR gives a better performance then LAR for the scenarios used.

5.4.1 Data Transfer:

It can be seen through the results that the average time taken by the Decision Based Routing algorithm to transfer a packet to its destination is lesser than the LAR algorithm. The reason is that the LAR algorithm has to transfer the packet to many nodes assuming that the destination node will be one of these nodes, whereas in the Decision based Routing Algorithm the destination is known and the path is also defined for the destination. This is the reason due to which the average time taken to transfer a single packet in LAR is higher then the Decision Based Routing Algorithm.

5.4.2 Data Loss:

It is likely that there will be more lost packets in LAR then the Decision Based Routing Algorithm because in DBR paths are defined such that if a path fails there is always and second path available to ensure the delivery of packet whereas in the LAR algorithm the packet is transferred to the Request Zone assuming that the destination node will be in that Zone, there is only one path and even that is not fully defined so more packet loss can occur in this algorithm.

5.4.3 Efficient Transfer

In the LAR algorithm the sender transfers a packet to all the nodes in the Request Zone which means that the sender must keep the information of all the nodes in the network, which is a burden. In DBR the sender must only know the left and right descendent and if these are known to the sender the packet can be transferred easily.

Summary

In this chapter simulation design was explained. Performance measures of mean delay and throughput were obtained for both LAR and DBR and a performance comparison with LAR was made. The results demonstrated a better performance by DBR in almost all cases examined.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

The DBR algorithm is a very efficient algorithm for small to medium size wireless networks because the routing strategy is very efficient due to the factors listed in the following sections. In general, the algorithm best suits networks without a server, since every parent node acts as a virtual server for its descendents or it can be said that every node in the network is effectively a server.

6.1.1 Minimum Packet Loss

The algorithm has multiple paths and this helps the packets to transfer efficiently across the network with low delay and without consuming their Time To Live (TTL). Also, due to availability of multiple paths for the packet transfer this reduces the chances for packet loss. The techniques used in the design are that every packet can choose different routes and every sender node checks the buffer size of its descendent before sending the packet. If the buffer of that particular descendent node which was expected to receive the packet is full then the parent node will send the packet to another descendent using the shortest path. Yet another reason of low packet loss is that the algorithm used for establishing the route selects the shortest path for packet transfer as the signal strength during the packet transfer from one node to another weakens if the path selected has more distance between the sending and receiving node. Thus the proposed technique saves on power consumption by choosing shortest path.

6.1.2 Best Possible Shortest Unique Route

With the help of the decision tree the packet is routed through the shortest path. The default path is always the left descendent which is the nearest node. Hence the packet is transferred without sending the packet to long distances, which is very helpful in wireless networks because in wireless networks the packet is transferred in the form of a modulated signal, if the distance is increased the signal strength reduces and the quality of the information is compromised, when the packet is transferred to the nearest node there is less chance that the signal power is reduced. The only problem for shortest path is possibly more hops, and this problem is solved by when the node checks the “nodes visited” array of the packet.

6.1.3 Fast Transfer of Packets

The algorithm ensures that the packets are transferred at a faster rate than most other connection topologies used in wireless networks. That is by using the shortest path and also the dual ring technique. Also, fewer acknowledgements are sent backwards to ensure fewer packet collisions which results in efficient packet transfer rate.

6.1.4 Link Termination

In case a node exits the network, one of the connections breaks or an established link is terminated due to unavoidable circumstances, such as node expiration in a wireless sensor network, in this case all the nodes that had this particular node as one of their descendents will redefine their descendents automatically. So the whole network doesn’t need to be re-initiated but only those nodes that have a link with that node re-initiate and establish the link with other nodes also they loop-back to maintain the link.

6.2 Future Work

Some possible future work directions are as follows:

1) The idea of decision based routing can be used for fuzzy logic implementation for the route / path selection. As in fuzzy logic the decision is based on the possibility of the successful result. E.g. in our case even though the decision is made by the node regarding the next node but the node is not sure about its decision i.e. how much the decision is correct say 40%, 50%, 100%. So this idea may improve the successful delivery of the packet to the destination.
2) Study the impact of congestion and loss on real time data transmission over wireless media in a similar way to that in fixed wired network. Generally deterministic protocols lend themselves much more readily to

real time data transmissions e.g. token ring compared to contention based protocols e.g. Ethernet. Wireless media is generally contention based and will pose research challenges in wireless media.

3) One would expect a ring based structure in a wireless medium, e.g. RPR, should lend itself much more readily to real time transmissions than the conventional IEEE 802.11 protocols. This thesis has studied the routing problem. Future work could thus regard to explore delay and loss and how they would impact on different services.
4) Reliability is another feature of the RPR that would lend itself to future work. The routing protocol studied provides resilience by reconfiguration. Future work is required to identify the full impact of node mobility and link failure.
5) Security was one goal for RPR and further work is required to identify how much of the existing RPR security mechanism is appropriate for decision based routing algorithms.
6) Future work might all nodes hear each other o consider widely dispersed nodes, where each node can only see a small subset of the complete system.
7) The use of multiple frequencies could be explored. This would open up frequency assignment problem, rather like those encountered in wavelength division multiplex networks.

References

[1] John W Mark, Weihua Zhuang, “Wireless communications and Networking,” Prentice Hall 2003.ISBN 81-203-2746-2

[2] Mehran Abolhasan,Tadeusz Wysocki,Eryk Dutkiewicz,” Review of Routing Protocols for Mobile ad-hoc networks” AdHoc Networks Volume 2, Issue1, Jan 2004, Pages 1-22.

[3] William Stalling, “Computer Networking With Internet Protocols and Technology,” Prentice Hall 2004, ISBN 0-13-911554.

[4] Birte Christensen-Dalsgaard, William Donnelly, Michael Griffith, “Flexible Working, New Network Technologies,” IOS Press Ohmsha 1999.ISBN 1-58-6030280.

. [5] Theodore S. Rappaport, “Wireless Communication Principles and Practice,” Second Edition Prentice Hall 2006, Chp2, 9, 10.

[6] Behrouz A. Forouzan, Sophia Chung “Data Communication and Networking,” 2nd edition Update McGraw-Hill year 2007. ISBN0-07-3250325.

[7] Introduction to 802.16 WiMAX: wireless broadband technology, market, operation and services. By Harte. Lawrence. Althos, 2006.Inc.ISBN: 978-1-932813-74-6.

[8] Vikki Lipset “802.16 vs. 802.20” 4 Sep 2003,Insights Wi-Fi Planet.Available: http://www.wi-fiplanet.com/columns/article.php/3072471 .

[9] Jeffrey G. Andrews, Arunabha Ghosh, Rias Muhamed. , “Fundamentals of WiMAX: understanding broadband wireless networking,” Prentice Hall, 2007.

[10] Matthew S. Gast, “802.11 Wireless Networks: The Definitive Guide,” O’REILLY 2002. ISBN0-596-00183-5.

[11]A fixed Network with directional Antennas. 2005 Rapid Computers Limited.
Available: http://www.rapid.co.uk/pointtopoint.html .

[12] SYN flood diagram assessed 20 May 2008 Wikipedia, the Free Encyclopedia. Available: http://en.wikipedia.org/wiki/SYN_flood .

[13] 802.11 security specification finalized 25 june 2004 Wi-Fi Planet News. Assessed 4 May 2008 Available: http://www.wi-fiplanet.com/news/article.php/3373441 .

[14]Metropolitan Area Networks (MAN): The Resilient Packet Ring and how it tops other technologies, SC441 independent Study 12 Aug 2004 By Kronfol Chao Productions.

[15] Jamshed Hasan, "Security Issues of IEEE 802.16 (WiMAX)", Proceedings of 4th Australian Information Security Management conference 5 Dec 2006.School of Computing and information science, Edith Cowan University, Australlia. Pages 107-129.

[16] WiMAX Technology for broadband Wireless Access by Loutfi Nuaymi John Wiley & Sons 2007. ISBN 9780470028087.

[17] William C.Y. Lee, "Mobile Cellular Telecommunications, Analog and Digital Systems," Second Edition, McGraw-HILL 2006:pp119.

[18] Wong Yew Fai, "Introduction to RPR," National University of Singapore Jan 2003.Available as: http://www.singaren.net.sg/activity/kranPT_6jan03.pdf

[19] Tehmina Karamat, Mr. John Mellor, "Proposal for a Wi-Fi Resilient Packet Ring Architecture," University of Bradford,PGNet(UK) 2005.pp 235-238

[20] Kevin Downes, "Internetworking Technologies Handbook," Fourth Edition, Cisco Systems, Inc, 2003. ISBN 1-58705-001-3.

[21] Prof Christian Schindelhauer, "Mobile Ad Hoc Networks," University of Freiburg, Computer Networks and Telematics 2007.Available as http://cone.informatik.uni-freiburg.de/teaching/lecture/manet-s07/slides/MANET-13.pdf

[22] Location-Aided Routing (LAR) in mobile ad hoc networks _Young-Bae Ko and Nitin H. Vaidya Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA. Wireless Networks 6(2000) pages 307-321.

[23] Location Aided Routing Protocol. 2004-2006 University of Luxembourg, SECAN-Lab Available:http://wiki.uni.lu/secan-lab/Location-Aided+Routing+Protocol.html .

[24] Implementing 802.11, 802.16 AND 802.20 wireless networks By Ron Olexa, Elsevier Inc 2005: pp 3-7.

[25] OFDM or OFDMA? By Mark E. Hazen, EWT Editor 25 Oct 2005 RF Design assessed 4 May 2008 Available As: http://rfdesign.com/ar/ofdm-or-ofdma/

[26] Performance Evaluation of Distance Routing Algorithm, By Nauhwar The Code Project 27 Dec 2005. Available: http://www.codeproject.com/useritems/MANET.asp .

[27] Routing Protocols for Ad-hoc Mobile Wireless Networks by Padmini Misra, Ohio

State University 1999. Available as http://www.cs.wustl.edu/~jain/cis788-99/index.html

[28] Holger Karl "A Short survey of wireless sensor networks", Telecommunication Network Group, Technische Universit¨ at Berlin August 18, 2003.TKN Technical Report TKN-03-018.

[29] IEEE 802.16, Wikipedia the Free encyclopedia, 3 May 2008 .Available: http://en.wikipedia.org/wiki/IEEE_802.16.

[30] Wireless Network Security by Y.Xiao,s.Shen and D.Du(Eds) Springer 2007. ISBN: 0-38-7289405.

[31] Tehmina Khan, Mr John Mellor," Security Issues in Wireless Communications and There Preventive Measures" PGNet (UK) 2004 pp 123-127.

[32] Understanding Wi-Fi and WiMAX as Metro-Access solutions By Intel 2004.Available as http://www.rclient.com/PDFs/IntelPaper.pdf

Appendix

USE CASE ANALYSIS

For Analysis and Design purposes the Unified Modeling Language (UML) is used. In - UML, a system is represented using five different “views” that describe the system from - distinctly different perspective. Each view is defined by a set of diagrams. The following views are present in UML.

User Model View

The use-case is the modeling approach of choice for the user model view.

Structural model view

Data and functionality are viewed from inside the system. That is, static structure (classes, objects and relationship) is modeled.

Behavioral Model

This model represents the dynamic or behavioral aspects of the system. It also depicts the interactions or collaborations between various structural elements described in the user model and structural models view.

A.1 User Model View

A.1.1 Identification of Actor

Actor of this system is (1) Users who are using the application.

A.1.2 Use Case Identification

Use cases are represented graphically in a use case diagram to allow the analyst to visualize each use case in the context of other use cases in the system or subsystem to show its relationship with actors and other use cases. Use case names are text strings that contain letters, numbers and most punctuation marks except for colon, which is used to separate use case names from the name of packages, and it is good idea to keep them short. Use case names are normally made up of an active verb and a noun or noun phrase that concisely describe the behavior of the system that you are modeling. There is only one actor of the system, which is the user of Simulator.

A.1.3 Use Case Diagram

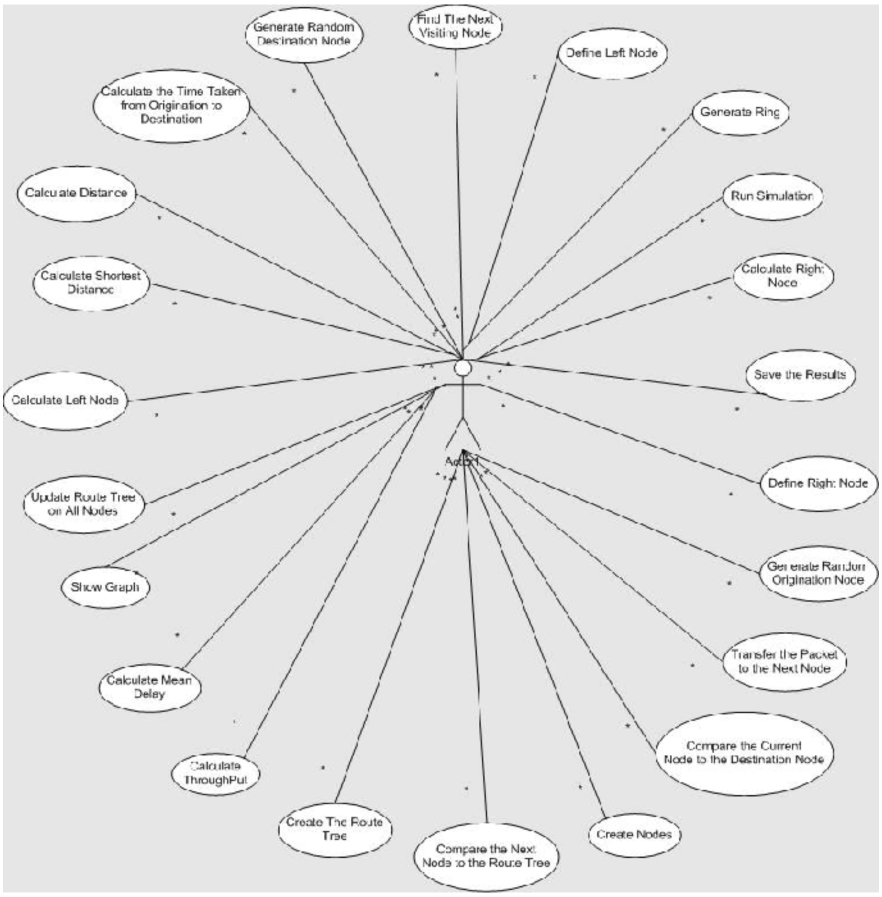
In use case diagram use case are drawn as an ellipse, the name of the use case usually written inside the ellipse, but can be placed beneath it. Do not mix these two styles in the same model.

A.1.4 Use Cases

Following are the possible identified use cases of the system.

- Create Nodes
- Calculate Distance
- Calculate Shortest Distance
- Find Left Node
- Find Right Node
- Define the Left-Right Node Tree
- Create Ring
- Generate A Random Packet
- Compare to the Decision Tree

A.1.5 Use Case Diagram



Generate Random Destination Node
Find The Next Visiting Node
Define Left Node
Calculate the Time Taken from Origination to Destination
Generate Ring
Calculate Distance
Run Simulation
Calculate Shortest Distance
Calculate Right Node
Save the Results
Calculate Left Node
Actor1
Define Right Node
Update Route Tree on All Nodes
Generate Random Origination Node
Show Graph
Transfer the Packet to the Next Node
Calculate Mean Delay
Compare the Current Node to the Destination Node
Calculate ThroughPut
Create The Route Tree
Create Nodes
Compare the Next Node to the Route Tree

A.1.6 Use Case Description (Use-Case Template)

Here is the description of the use cases identified above.

Use Case ID:	
Use Case Name:	**Create Nodes**
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	When User Clicks on the Create Nodes

	The program Initiate the Create Nodes function First it clears all the current Nodes Create the Nodes Compares the No of Nodes value and creates them on a specific distance
Post Condition:	

Use Case ID:	
Use Case Name:	**Calculate Distance**
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	Calculates the Distance of Each Node with all other nodes Find the Coordinates of Each Node Calculates the Distance using the Distance formula Stores Each Distance in an Array
Post Condition:	

Use Case ID:	
Use Case Name:	**Calculate Shortest Distance**
Actors:	Application User

Description:	
Preconditions:	
Main Flow:	Finds out the shortest distance of each node with any other node Takes a specific node compares the distance with the other nodes Finds the Shortest Distance node
Post Condition:	

Use Case ID:	
Use Case Name:	**Find Left Node**
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	Defines the Left Node The most nearer node is set the Left Node
Post Condition:	

Use Case ID:	
Use Case Name:	**Find Right Node**
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	Find the Right Node Searches the Distance Array

	Finds the second shortest distance and sets as the right node.
Post Condition:	

Use Case ID:	**Define the Left-Right Node Tree**
Use Case Name:	
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	Now that all the Left and Right Nodes are defined Start with the first Node set this node as the Parent node. Now set the left node as Left descendent Set the right node as the right descendent Go to left node set its left and right descendents Do the same for all levels till the end of nodes.
Post Condition:	

Use Case ID:	**Create Ring**
Use Case Name:	
Actors:	Application User
Description:	

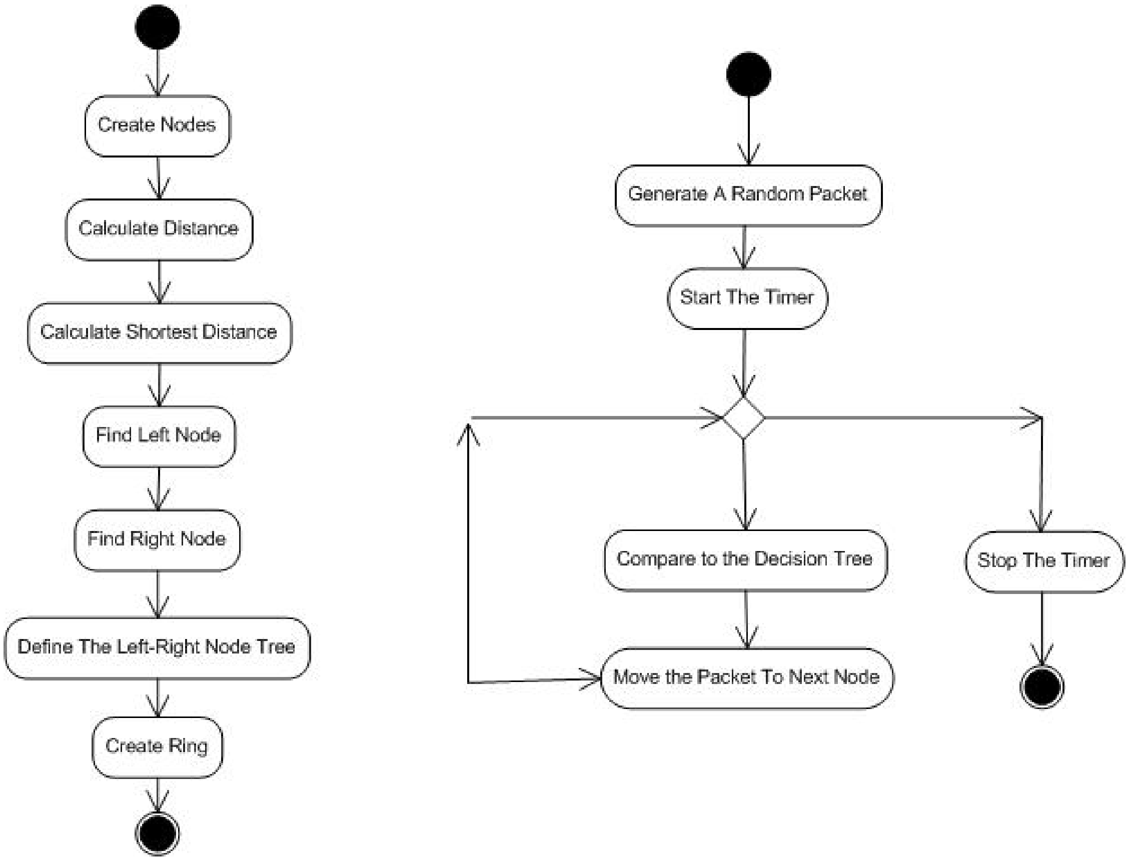
Preconditions:	
Main Flow:	Create the Ring Connect all the nodes to its left node as well as the right node
Post Condition:	

Use Case ID:	
Use Case Name:	**Generate A Random Packet**
Actors:	Application User
Description:	
Preconditions:	
Main Flow:	Create a Random Packet Create its random originating point Create a random destination point Add the origination node to the visited node
Post Condition:	

Use Case ID:	
Use Case Name:	**Compare to the Decision Tree**
Actors:	Application User
Description:	
Preconditions:	

Main Flow:	Compare the Left node to the visited array If the node is present in the visited node array Go to right node.
Post Condition:	

Activity Diagram:



IMPLEMENTED CODE

```
Using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace TokenRing
{

  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }

    public int noOfNodes;
    private void btCrSimulation_Click(object sender, EventArgs e)
    {

      if (btCrtRing.Enabled == true)
      {
        for (int i = 0; i < noOfNodes; i++)
        {
          nodesBtn[i].Hide();
          this.panSim.Refresh();
        }
        this.lb1.Hide();
        this.lb2.Hide();

      }
      noOfNodes = Convert.ToInt32(tbNoOfNodes.Text);
      if (noOfNodes > 0)
        createSimulation();
    this.btCrtRing.Enabled = true;

}
private Button[] nodesBtn;
private void createSimulation()
{
    nodesBtn = new Button[noOfNodes];
    Random rt = new Random(2);
    int y = 20;
    int x = 10;
    int j = -1;
    for (int i = 0; i < noOfNodes; i++)
    {
        nodesBtn[i] = new Button();
        if (j == 1)
        {
            x = 10;
            y += 50;
            j = -1;
        }
        if (x > panSim.Size.Width - 80)
        {
            x = 100;
            y += 50;
            j = 1;
        }

        Point pt = new System.Drawing.Point(x, y);
        x += 80;
        nodesBtn[i].Location = pt;
        nodesBtn[i].Size = new System.Drawing.Size(40, 40);
        nodesBtn[i].Text = "Node" + i;
        this.panSim.Controls.Add(nodesBtn[i]);
    }
}

private void btnUp_Click(object sender, EventArgs e)
```

```
        {
            int selectedNode = Convert.ToInt32(tbNodeMov.Text);
            nodesBtn[selectedNode].Location = new
System.Drawing.Point(nodesBtn[selectedNode].Location.X, nodesBtn[selectedNode].Location.Y - 25);
        }

        private void btnRight_Click(object sender, EventArgs e)
        {
            int selectedNode = Convert.ToInt32(tbNodeMov.Text);
            nodesBtn[selectedNode].Location = new
System.Drawing.Point(nodesBtn[selectedNode].Location.X + 25, nodesBtn[selectedNode].Location.Y);
        }

        private void btnLeft_Click(object sender, EventArgs e)
        {
            int selectedNode = Convert.ToInt32(tbNodeMov.Text);
            nodesBtn[selectedNode].Location = new
System.Drawing.Point(nodesBtn[selectedNode].Location.X - 25, nodesBtn[selectedNode].Location.Y);
        }

        private void btnDown_Click(object sender, EventArgs e)
        {
            int selectedNode = Convert.ToInt32(tbNodeMov.Text);
            nodesBtn[selectedNode].Location = new
System.Drawing.Point(nodesBtn[selectedNode].Location.X, nodesBtn[selectedNode].Location.Y + 25);
        }

        private NodeInfo[] nodes;
        private void btCrtRing_Click(object sender, EventArgs e)
        {
            this.panSim.Refresh();
            nodes = new NodeInfo[noOfNodes];

            // Initializes the nODES.
            for (int i = 0; i < noOfNodes; i++)
            {
                nodes[i] = new NodeInfo(noOfNodes);
```

```
          nodes[i].ownIndex = i;
          nodes[i].ownCor.x = nodesBtn[i].Location.X;
          nodes[i].ownCor.y = nodesBtn[i].Location.Y;
        }
        // end Initialize the Nodes



        calculateDistance();  // this method calculates the Distance Between the Nodes;

        /// Start Identify the Shortest Distance

        for (int i = 0; i < noOfNodes; i++)
        {
          int index = 0;
          index = compare(nodes[i]);
          nodes[i].LeftNodeIndex = index;
          nodes[i].leftNode.x = nodes[index].ownCor.x;
          nodes[i].leftNode.y = nodes[index].ownCor.y;
        }

        //end One Shortest distannce node calculated

        System.IO.StreamWriter sr1 = new System.IO.StreamWriter(@"C:/ringLeftNode.doc");

        for (int i = 0; i < noOfNodes; i++)
        {
          for (int j = 0; j < noOfNodes; j++)
            sr1.Write("Node " + i + " X = " + nodes[i].ownCor.x + "Y= " + nodes[i].ownCor.y + "
Distance Node " + j + " " + nodes[i].distance[j] + "\n");
          // sr1.Write("Node " + i + " Left Node " + nodes[i].LeftNodeIndex + " Right Node  " +
nodes[i].rightNodeIndex + "\n");

        }
        sr1.Close();
        ///end test code
```

```
//Confirm the Loop and If any Link is missing generate that link
for (int i = 0; i < noOfNodes; i++)
{
   if (nodes[nodes[i].LeftNodeIndex].ownIndex == i)
   {
      nodes[nodes[i].LeftNodeIndex].rightNodeIndex = i;
      nodes[nodes[i].LeftNodeIndex].rightNode.x = nodes[i].ownCor.x;
      nodes[nodes[i].LeftNodeIndex].rightNode.y = nodes[i].ownCor.y;
   }
   else
   {
      int index = 0;
      index = compare(nodes[i], nodes[i].LeftNodeIndex);
      nodes[i].rightNodeIndex = index;
      nodes[i].rightNode.x = nodes[index].ownCor.x;
      nodes[i].rightNode.y = nodes[index].ownCor.y;

   }

}

// if left node/right node is the node it self or left node== right node remove it
for (int i = 0; i < noOfNodes; i++)
{
   if (nodes[i].LeftNodeIndex == i)
   {
      int index = 0;
      index = compare(nodes[i], nodes[i].LeftNodeIndex);
      nodes[i].LeftNodeIndex = index;
      nodes[i].leftNode.x = nodes[index].ownCor.x;
      nodes[i].leftNode.y = nodes[index].ownCor.y;
   }
   if (nodes[i].rightNodeIndex == i)
   {
      int index = 0;
      index = compare(nodes[i], nodes[i].rightNodeIndex);
```

```
                nodes[i].rightNodeIndex = index;
                nodes[i].rightNode.x = nodes[index].ownCor.x;
                nodes[i].rightNode.y = nodes[index].ownCor.y;
            }
            if (nodes[i].rightNodeIndex == nodes[i].LeftNodeIndex)
            {
                int index = 0;
                index = compare(nodes[i], nodes[i].rightNodeIndex);
                nodes[i].rightNodeIndex = index;
                nodes[i].rightNode.x = nodes[index].ownCor.x;
                nodes[i].rightNode.y = nodes[index].ownCor.y;
            }

        }

        // end Create the loop (identify right nodes)

        //Start Drawing the loop
        Graphics graph = this.panSim.CreateGraphics();
        Pen penCurrent = new Pen(Color.Red, 4);
        Pen penCurrent1 = new Pen(Color.Green, 4);
        for (int i = 0; i < noOfNodes; i++)
        {
            graph.DrawLine(penCurrent, (float)nodes[i].ownCor.x, (float)nodes[i].ownCor.y,
(float)nodes[i].leftNode.x, (float)nodes[i].leftNode.y);
            graph.DrawLine(penCurrent1, (float)nodes[i].ownCor.x + 20, (float)nodes[i].ownCor.y + 20,
(float)nodes[i].rightNode.x, (float)nodes[i].rightNode.y);
            //MessageBox.Show(" "+nodes[i].rightNodeIndex+"  "+nodes[i].LeftNodeIndex);
        }
        //end Drawing the Loop

        System.IO.StreamWriter sr = new System.IO.StreamWriter(@"C:/ring.doc");

        for (int i = 0; i < noOfNodes; i++)
        {
            sr.Write("Node " + i + " Left Node " + nodes[i].LeftNodeIndex + " Right Node  " +
nodes[i].rightNodeIndex + "\n");
```

```
  }
  sr.Close();

}


public void calculateDistance()
{
  Coordinates tempDis;
  tempDis = new Coordinates();
  tempDis.x = 0;
  tempDis.y = 0;
  for (int i = 0; i < noOfNodes; i++)
  {

    // START cALCULATE DISTANCE WITH ALL THE OTHER NODES
    for (int j = 0; j < noOfNodes; j++)
    {
      tempDis.x = nodes[i].ownCor.x - nodes[j].ownCor.x;
      tempDis.y = nodes[i].ownCor.y - nodes[j].ownCor.y;
      nodes[i].distance[j] = new double();
      if (i != j)
      {
        nodes[i].distance[j] = Math.Sqrt((tempDis.x) * (tempDis.x)) + ((tempDis.y) * (tempDis.y));
      }
      else
      {
        nodes[i].distance[j] = 9999999;
      }
      //MessageBox.Show(" " + nodes[i].distance[j]);
    }
    // eND cALCULATE DISTANCE WITH ALL THE OTHER NODES
  }

}
```

```
//This Funtion will search the node's closest node.
private int compare(NodeInfo m_node)
{
   int shortest = 0;
   [          for (int i = 0; i < noOfNodes; i++)
   {
      if (m_node.distance[shortest] > m_node.distance[i])
         shortest = i;
   }
   return shortest;
}
// This function will search for the right node
private int compare(NodeInfo m_node, int lNode)
{
   int shortest = 0;
   for (int i = 0; i < noOfNodes; i++)
   {
      if (i != lNode)
      {
         if (m_node.distance[shortest] >= m_node.distance[i])
            shortest = i;
      }


   }
   return shortest;
}
private int[] time;
int packetsLost = 0;
private void btRunSim_Click(object sender, EventArgs e)
{


   System.IO.StreamWriter sr = new System.IO.StreamWriter(@"C:/transfer.doc");
   packetsLost = 0;
   for (int x = 0; x < noOfNodes; x++)
   {
      nodes[x].currentMessageBufferCount = 0;
   }
```

```
int[] ranPackets = new int[100];
int count = 0;
int k = 0;
time = new int[1000];


Packet tranPacket = new Packet();
int packetNode;
Random temp = new Random();
int toBeSentNode;
Random rm = new Random();
Random rm1 = new Random();

for (int i = 0; i < 500; i++)
{
   time[i] = Environment.TickCount;
   packetNode = rm.Next(0, noOfNodes);
   toBeSentNode = rm1.Next(0, noOfNodes);
   while (packetNode == toBeSentNode)
      toBeSentNode = rm1.Next(0, noOfNodes);

   for (int jK = 0; jK < nodes.Length; jK++)
   {
      nodes[jK].currentMessageBufferCount = 0;
   }
   int[] nodesVisited = new int[noOfNodes * 2];

   tranPacket.orginatingNodeIndex = packetNode;
   tranPacket.finalDestination = toBeSentNode;

   nodesVisited[k] = tranPacket.orginatingNodeIndex;
   if (nodes[nodes[packetNode].LeftNodeIndex].currentMessageBufferCount < 10)
      tranPacket.nextNode = nodes[packetNode].LeftNodeIndex;
   else
      tranPacket.nextNode = nodes[packetNode].rightNodeIndex;
   count = 1;
```

```
        sr.Write("\n Originating Index " + tranPacket.orginatingNodeIndex + "  Next Node" +
tranPacket.nextNode + "  Final Dest" + tranPacket.finalDestination);
        tranPacket.nextNode = nodes[tranPacket.nextNode].LeftNodeIndex;

        while ((tranPacket.nextNode != tranPacket.finalDestination) && (count <= noOfNodes * 2))
        {
          if (nodes[nodes[tranPacket.nextNode].LeftNodeIndex].currentMessageBufferCount > 100)
          {
            packetsLost += 1;

            break;
          }
          if (exists(nodesVisited, tranPacket.nextNode))
          {
            tranPacket.nextNode = nodes[tranPacket.nextNode].rightNodeIndex;
            nodes[nodes[tranPacket.nextNode].rightNodeIndex].currentMessageBufferCount += 1;
          }

          else
          {
            tranPacket.nextNode = nodes[tranPacket.nextNode].LeftNodeIndex;
            nodes[nodes[tranPacket.nextNode].LeftNodeIndex].currentMessageBufferCount += 1;
          }

          if (nodes[tranPacket.nextNode].LeftNodeIndex == tranPacket.finalDestination)//
(nodes[nodes[tranPacket.nextNode].LeftNodeIndex].currentMessageBufferCount < 99)
          {
            tranPacket.nextNode = nodes[tranPacket.nextNode].LeftNodeIndex;
            nodes[nodes[tranPacket.nextNode].LeftNodeIndex].currentMessageBufferCount += 1;
          }
          count += 1;

        }
        time[i] = Environment.TickCount - time[i];
        sr.Write("\nTotal Time Taken to delver the packet=" + time[i]);

      }
```

```
    sr.Close();
}
int i = 0;

Label lb1 = new Label();
Label lb2 = new Label();
private bool exists(int[] visited, int x)
{
    for (int i = 0; i < visited.Length; i++)
    {
        if (x == visited[i])
            return true;
    }
    return false;
}

int[] ranPackets;
    int count;
    int k;

    Packet tranPacket;
    int packetNode;
    Random temp;
    int toBeSentNode;
    Random rm;
    Random rm1;
System.IO.StreamWriter sr;
private void tmSimulation_Tick(object sender, EventArgs e)
{

    packetsLost = 0;
    time[i] = new int();
    time[i] = Environment.TickCount;
    System.Timers.Timer s = new System.Timers.Timer(1);
    packetNode = rm.Next(0, noOfNodes);
    toBeSentNode = rm1.Next(0, noOfNodes);
    while (packetNode == toBeSentNode)
```

```
        toBeSentNode = rm1.Next(0, noOfNodes);
      int[] nodesVisited = new int[noOfNodes * 2];

      tranPacket.orginatingNodeIndex = packetNode;
      tranPacket.finalDestination = toBeSentNode;

      nodesVisited[k] = tranPacket.orginatingNodeIndex;
      if (nodes[nodes[packetNode].LeftNodeIndex].currentMessageBufferCount < 99)
        tranPacket.nextNode = nodes[packetNode].LeftNodeIndex;
      else
        tranPacket.nextNode = nodes[packetNode].rightNodeIndex;
      count = 1;
      sr.Write("\n Originating Index " + tranPacket.orginatingNodeIndex + "   Next Node" +
tranPacket.nextNode + "   Final Dest" + tranPacket.finalDestination);
      tranPacket.nextNode = nodes[tranPacket.nextNode].LeftNodeIndex;
      this.nodesBtn[tranPacket.orginatingNodeIndex].ForeColor = System.Drawing.Color.Green;
      this.nodesBtn[tranPacket.finalDestination].ForeColor = System.Drawing.Color.Red;

      lb1.Location = new Point(nodesBtn[tranPacket.orginatingNodeIndex].Location.X - 22,
nodesBtn[tranPacket.orginatingNodeIndex].Location.Y - 22);
      lb1.Text="RTS";
      lb1.Show();
      lb2.Location=new Point(nodesBtn[tranPacket.finalDestination].Location.X-
22,nodesBtn[tranPacket.finalDestination].Location.Y-22);
      lb2.Text="CTS";
      lb2.Show();
      this.panSim.Controls.Add(lb1);
      this.panSim.Controls.Add(lb2);
      this.panSim.Refresh();
      while ((tranPacket.nextNode != tranPacket.finalDestination) && (count <= noOfNodes * 2))
      {
        if (exists(nodesVisited, tranPacket.nextNode))
        {
          tranPacket.nextNode = nodes[tranPacket.nextNode].rightNodeIndex;
          nodes[tranPacket.nextNode].currentMessageBufferCount += 1;
          this.nodesBtn[tranPacket.nextNode].Text =
nodes[tranPacket.nextNode].currentMessageBufferCount.ToString();
```

```
            }
            else
            {
               tranPacket.nextNode = nodes[tranPacket.nextNode].LeftNodeIndex;
               nodes[tranPacket.nextNode].currentMessageBufferCount += 1;
               this.nodesBtn[tranPacket.nextNode].Text                                    =
nodes[tranPacket.nextNode].currentMessageBufferCount.ToString();
            }
                        time[i] = time[i] - temp.Next(5, 50);
            count += 1;

            time[i] = Environment.TickCount - time[i];
            sr.Write("\nTotal Time Taken to delver the packet=" + time[i]);

         }

         if (timerCount >= 1)
         {
            tmSimulation.Stop();
            sr.Close();
         }
         else
            timerCount++;
       }
      int timerCount;
      private void button1_Click(object sender, EventArgs e)
      {
         graphForms bs = new graphForms(noOfNodes, time, nodes, packetsLost);
         bs.Show(this);
      }

      private void btSimSlow_Click(object sender, EventArgs e)
      {
         sr = new System.IO.StreamWriter(@"C:/transfer.doc");
         tmSimulation.Interval = 1000;
         tmSimulation.Start();
         timerCount = 1;
```

```
          for (int x = 0; x < noOfNodes; x++)
          {
             nodes[x].currentMessageBufferCount = 0;
          }
            packetsLost = 0;
            int[] ranPackets = new int[100];
             k = 0;
             time = new int[1000];
             tranPacket = new Packet();
             temp = new Random();

             rm = new Random();
             rm1 = new Random();
       }
    }
 }
```