DYNAMIC VIRTUAL BACKBONE ROUTING PROTOCOL: A HYBRID ROUTING PROTOCOL FOR ADHOC NETWORKS

Melvin John

Supervised By

Dr. Cheryl Pope & Dr. Cruz Izu

Submitted to the School of Computer Science

The University of Adelaide

In partial fulfillment of the requirements for the

Masters Degree in Computer Science

November 2006

Abstract

Ad hoc networks are the collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such a scenario a mobile host can act as both a host and a router forwarding packets for other mobile nodes in the network.

The thesis focuses on the routing aspect of ad hoc networks; there are three kinds of routing protocols in ad hoc networks proactive, reactive and hybrid (mixture of both proactive and reactive components). According to ad hoc network literature hybrid routing should perform better than the other two types of protocols. The thesis aims at validating this particular argument by comparing the performance of a hybrid routing protocol with a popular ad hoc network routing protocol.

The thesis discusses some of the hybrid routing protocols discussed in the literature and then selects the Dynamic Virtual Backbone Routing (VBR/DVB) protocol. A brief introduction to the architecture of NS2 and the steps to implement a new routing protocol for ad hoc networks is also discussed. Dynamic Virtual Backbone Routing (VBR/DVB) protocol is newly implemented in NS2 and is compared with the Dynamic Source Routing (DSR) protocol which is already implemented in NS2

The comparison results show that DSR outperforms VBR in mobility scenarios; but the reasons for this was the optimizations which have been done to the DSR implementation in NS2; therefore the results obtained were biased towards DSR.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

1.1 Introduction

The introduction of wireless technology has given us mobility. Laptops, PDA's can all be connected to each other without wires but still they have one thing in common. All these devices have to be under some central management to talk to each other, through the introduction of ad hoc networks one tries to decentralize this management.

Ad hoc networks basically deal with formation and maintenance of temporary networks. In these networks the nodes (i.e. the computer) act as both the host and the router. In wireless ad hoc networks the nodes are more dynamic in nature i.e. the nodes can move around with any velocity. Thus the nodes can go out and come into the range of transmission of different nodes. Figure 1 shows a simple ad-hoc network of three nodes. Nodes A-B and B-C are in communication range of each other so they can send data directly to each other, if the data has to be send from A-C, then node B has to forward the data to C on behalf of node A.

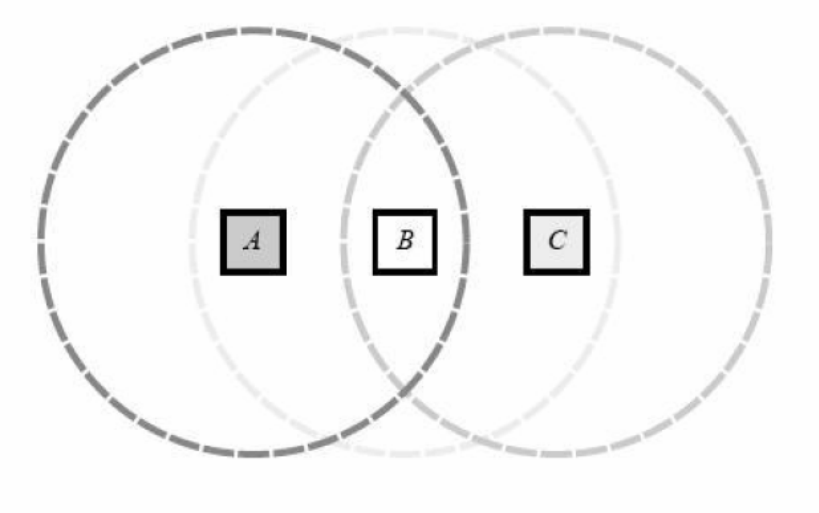

Figure 1: A simple MANET consisting of 3 nodes

Ad hoc networks were first mainly used for military applications. Since then, they have become increasingly more popular within the computing industry. Applications include emergency search-and-rescue operations, deployment of sensors,

conferences, exhibitions, virtual classrooms and operations in environments where construction of infrastructure is difficult or expensive.

Routing in ad hoc network has certain challenges firstly; the nodes in the ad hoc network can move around with any velocity, this implies that the nodes can move out of and into the range of different nodes. Therefore the routing protocols in ad hoc network must be able to adapt themselves to the dynamic changes in the topology of the wireless networks. The routing protocols can keep up to date information about the topology with periodic update messages. However, since the nodes have limited resources like CPU speed, power constraints, storage memory and bandwidth the exchange of control packets has to be minimized.

1.2 ROUTING IN MANETS

There are three different kinds of protocols, proactive, reactive and hybrid protocols which is a mixture of the other two kinds of protocols.

1.2.1 Proactive (Table driven) Routing

Proactive [2] routing tries to keep up to date information about the entire network, therefore when there is a routing request, the request is fulfilled without any delay e.g. is Destination Sequenced Distance Vector (DSDV), link state routing protocol.

In simple link state routing protocol each node will maintain a table containing routes to all the nodes in the network. This is done by transmitting the node's neighbor list to all the nodes in the network. Figure 2 shows the snapshot of a network using link state routing protocol. When a route is required; the routing table is looked up. Since the routing table has routes to all the nodes in the network the route is satisfied without any delay. For example, if node 0 has to find the route to destination 3, then node 0 does a lookup in its routing table for node 3. According to the routing table the data packet is forwarded to node 2. The same procedure is repeated at node 2 and finally the packet reached the destination node 3.

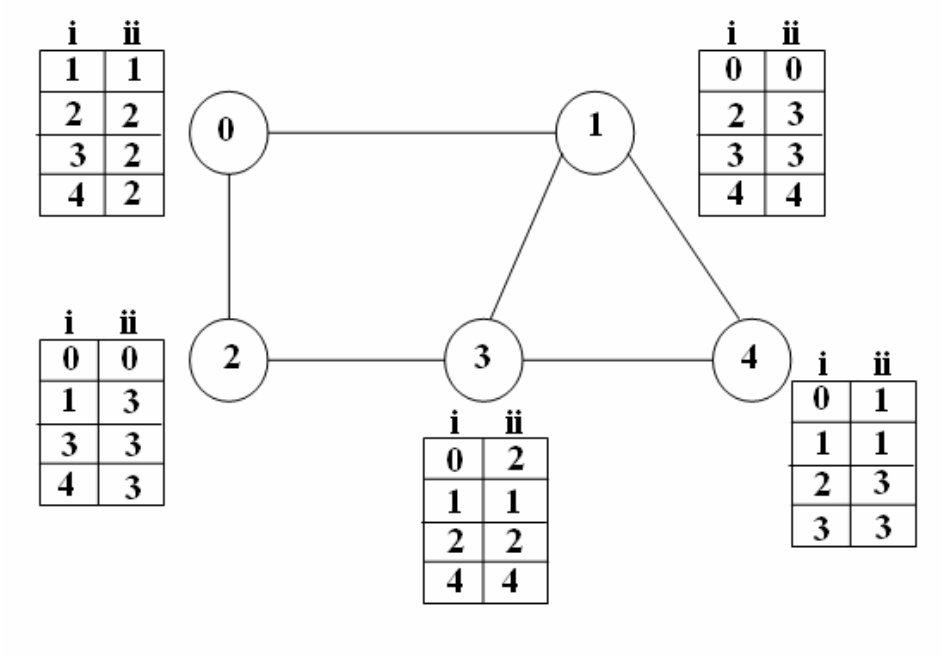

Figure 2: Example of a simple Link State Routing protocol. Col i – represents the destination id, Col. ii represents the next-hop id. Each node has route information for all other nodes in the network

Such a protocol will work very effectively and efficiently in a small network scenario but when the network is scaled it's difficult for each node to keep information about each and every other node. If the topology of the network changes very frequently then the number of control messages which are exchanged will increase. Thus, most of the bandwidth will be used up by these control messages rather than the actual messages which is not acceptable in a bandwidth limited scenario. To reduce the control message traffic and for achieving better convergence rate new algorithms combining the features of distance vector and link state protocols are being used e.g. is the wireless routing protocol (WRP) [1]. In WRP, routing nodes communicate the distance and second-to-last hop information for each destination in the wireless networks. It avoids the "count-to-infinity" problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. This ultimately eliminates looping situations and provides faster route convergence when a link failure event occurs [6].

1.2.2 Reactive (On-demand) Routing

Reactive [2] routing does not gather and keep information regarding the network topology. When a message has to be routed, a route discovery procedure is invoked in which a route discovery message is flooded through out the network. The original message is delayed till the source gets the reply for the route discovery message. The reply message for the route discovery message contains the path from the source to the destination. Examples of reactive routing protocols are Dynamic Source Routing (DSR) and Ad Hoc On-demand Distance Vector (AODV) protocol. Once a path is discovered the path is saved in the source nodes cache with a time stamp. If any of the nodes in the path disappear (i.e. the node may be switched off or have moved away) then the path until that node is retained and a new route discovery procedure is invoked to find the rest of the path to the destination.

The most popularly used ad hoc routing protocol is the DSR protocol. The source node starts the route discovery process by sending a route request to find the destination node. Figure 3 shows and example of this process to find a route form source node 1 to destination node 8. Whenever a route request packet visits a node, the packet adds that node to its visited list, as indicated by the link tags of figure 3.

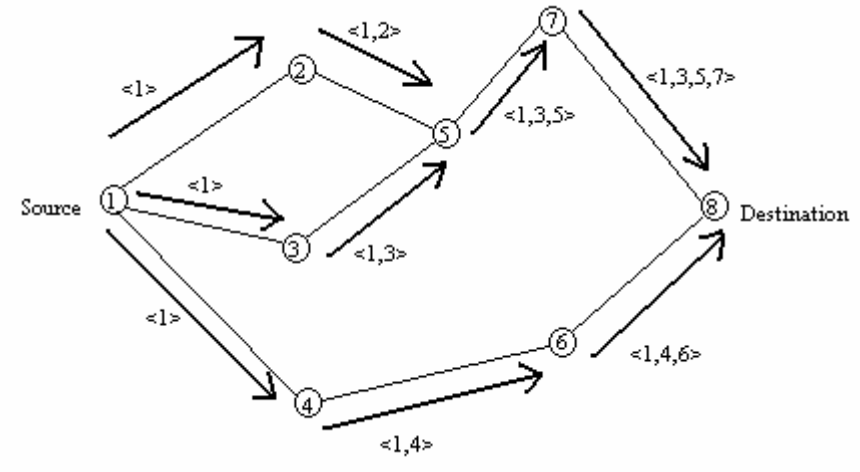

Figure 3: Building Record Route during Route Discovery

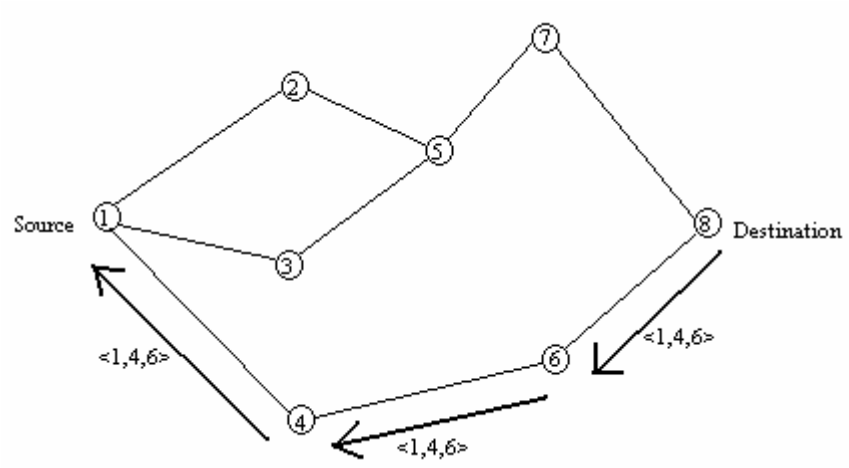

Figure 4: Propagation of Route Reply with the Route Record

The first route request packet which reaches the destination replies to the route request. Figure 4 shows how the route reply is sent back through the accumulated path (visited nodes) from node 8 back to the source.

In case of Ad Hoc On-demand Distance Vector (AODV) a similar procedure is followed, but instead of storing the routing path in the packet, each node on the path saves the next hop address for the destination.

The main advantage of reactive routing is that the control traffic in the network is minimized, but this is the cost of long setup delay therefore this scheme is not suitable for routing real-time traffic. Another drawback of this scheme is that the message size increases because the entire path information is in the message [1, 2, and 3].

1.2.3 Hybrid Routing

Ad hoc networks are used in diverse applications and it's not possible to create a single protocol which will work efficiently for all the applications. Proactive and reactive routing schemes operate efficiently in small regions of the ad hoc network design space.

As seen in figure 5 proactive routing is suited for areas where the ratio of node mobility to call rate is low i.e. the network topology does not change frequently, whereas reactive routing is suited for areas where the ratio of node mobility to call rate is high. The performance of both the protocols suffer when call rate and node mobility increase. Therefore new protocols have to be created which combine the strengths of existing protocols i.e. hybridization. A lot of hybrid routing protocols

have been suggested for ad hoc networks e.g. are Zone Routing Protocol (ZRP) [1], Independent Zone Routing Protocol [2], and Routing via dynamic group construction [4], AntHocNet [5] and many more.

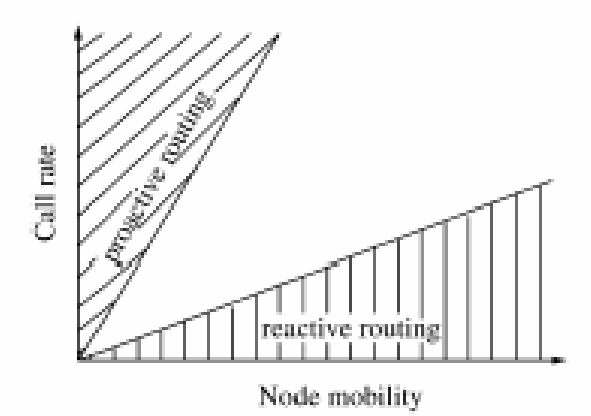

Figure 5: Ad hoc network design space

Most of the hybrid routing protocols have a common structure, they contain two sub protocols say A and B. Protocol A covers some local area and protocol B covers some global area. The basic idea of most of the protocols is to organize a group of nodes or a single node to maintain some minimal information about the topology of its neighbors; say protocol A governs the messages passing in this small group: now to pass messages between such groups a protocol B (global) is used. In most of the hybrid protocols; protocol A will be some kind of proactive (table based) protocol and the protocol B will be some kind of reactive (on-demand) protocol. The main factor which affects the performance of the hybrid protocols is the decision of when to switch between proactive and reactive protocols. One of the aims of this project is to find optimal solution for this switching problem by which the performance of the hybrid protocol can be improved.

1.3 PROPOSED WORK

Objective: *The goal of the project is to find out whether hybrid protocols can perform better than the currently implemented protocols. This goal is achieved by implementing a hybrid routing protocol in NS2 and compare its performance with DSR routing protocol which is implemented in NS2.*

The following are the main stages of the project:

1. Investigating different hybrid routing protocols for ad hoc networks.
    - Deciding if a particular protocol can be implemented or not. For example, does it rely on information that is difficult or costly to obtain, does it rely on an algorithm with a high order of complexity which could lead to scaling issues. This is the most important step because the further project revolves around the implementation of this protocol
    - Selecting the most promising protocol based on expected performance over a wide range of scenarios.
2. Implementing and testing the selected hybrid protocol in NS2.
    - Studying the working of NS2
    - Investigating the relevant sections of NS2 code which have to be changed
    - Actual implementation and testing.
3. Comparative study of routing performance and overhead, using NS2 comparing DSR with the implemented hybrid protocol.

The rest of the thesis is organized as follows: chapter 2 explains the different kind of hybrid routing protocol which were considered, some comparisons between the main routing protocols and the reasons for selecting the Dynamic Virtual Backbone routing protocol. Chapter 3 explains the Dynamic Virtual Backbone in detail; Chapter 4 give the overview of the NS2 architecture and the changes made to NS2, chapters 5 and 6 explain the testing scenarios and the results.

Chapter 2

Many hybrid routing protocols have been suggested, this section describes a list of hybrid routing protocols which were considered to select a good candidate for implementation.

2.1 Node-Centric Hybrid Routing

The motivation behind Node centric hybrid routing is that there are special node in an ad-hoc network which provide special services to all the other nodes in the network. This special service can be e.g. DNS service, web proxies and Internet access. These special nodes which provide the services are called as netmarks; they can be mobile or stationary depending on the problem scenario. All the non-netmark nodes in the network maintain route to these netmarks, the paths between the non-netmark nodes are set up on demand. The netmarks are distinguished from the normal nodes by explicit tagging or addressing. There are two approaches for node-centric hybrid routing. In one approach the netmarks force the other nodes to maintain the route to it for long periods of time, thus extending the caching of netmark routing information. In the other approach the netmarks send out proactive routing updates to push into the routing tables of the rest of the nodes in the ad-hoc network. This protocol can only be implemented in specific scenarios like a conference or educational institution because this method relies on special nodes which provide special services [10].

2.2 AntHocNet

AntHocNet is a hybrid multi-path algorithm, which uses the concept of ant based routing. They are based on the pheromone trial laying-following behavior of real ants and the related framework of ant colony optimization. Continuously sampling possible paths with ant-like agents and the quality of the path is indicated by the artificial pheromone variables. Ant based routing algorithms can work in a distributed, are highly adaptive, robust and provide automatic load balancing.

When a data session is started at node S with destination D, S checks whether it has up-to-date routing information for D. If not, it reactively sends out ant-like agents, called reactive forward ants, to look for paths to D. These ants gather

information about the quality of the path they followed, and at their arrival on D they become backward ants which trace back the path and update routing tables. On the backward path the pheromone tables in different nodes are updates thus indicating multiple paths between S and D, and data packets can be routed from node to node as datagrams. They are stochastically spread over the paths: in each node they select the next hop with a probability proportional to its pheromone value. Once paths are set up and the data session is running, S starts to send proactive forward ants to D. These ants follow the pheromone values similarly to data packets. In this way they can monitor the quality of the paths in use. Moreover, they have a small probability of being broadcasted, so that they can also explore new paths. In case of link failures, nodes either try to locally repair paths, or send a warning to their neighbors such that these can update their routing tables. AntHocNet uses a heuristic method like Ant Colony Optimization (ACO) as compared to the other hybrid routing protocols mentioned in the literature [5].

2.3 Routing in wireless/mobile ad hoc networks via dynamic group construction

This protocol aims at grouping nodes based on some metric, the proactive scope of a node is reduced to the group it belong to, thus reducing the message complexity and provides a more efficient way of infrastructure update. The degree of each node is calculated and compared with its neighbor, based on this the “dominating value (DV)” of the nodes are calculated. The DV’s can be positive, negative or zero. Depending on the DV the node are positive nodes, negative node and zero nodes. All neighboring positive nodes are group to form the P-cluster; all the neighboring non-positive nodes are grouped to form the N-clusters. A P-cluster and the adjacent N-clusters form the Routing Group. The two types of constructed clusters are interleaved with each other. The routing groups are connected to each other by bridge clusters, i.e. an N-cluster which is belonging to more than one routing group. The P-clusters stores the topology information for its routing group; the N-clusters have information about its own topology.

When a mobile node has to communicate it first perform a route search in the local topology (N-cluster), if it can’t find the route then it forwards route request to its adjacent P-cluster. If the route is still not found then it means that the destination is not in the current routing group so a route search is issued to the whole network. In

route maintenance the variation in the network are seen as link variation, each link variation is called a micro event and for each micro event there is a micro action. There are 6 micro events and its micro action. Figure 6 shows the initial constructed network topology. The N-cluster {M, N} is the bridge cluster which connects the two routing groups.

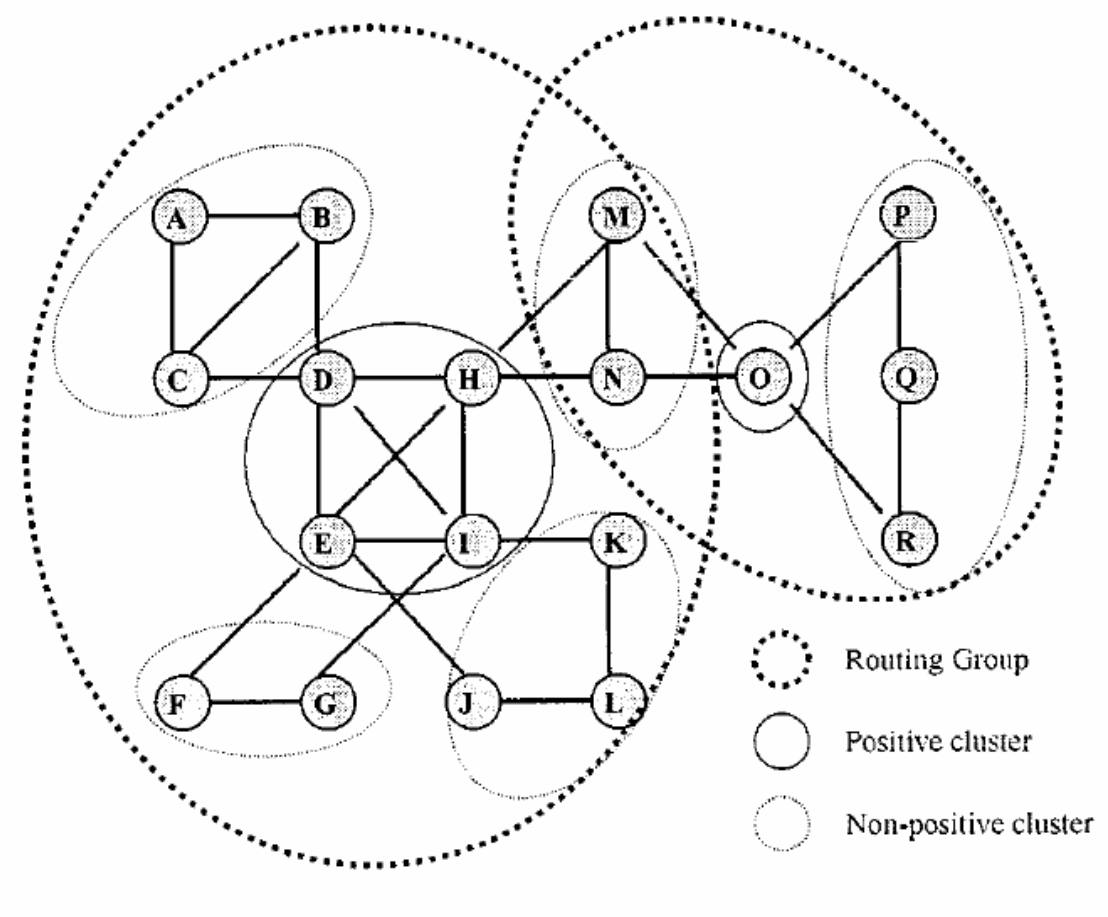

Figure 6: The initial constructed network topology

In their evaluation [4] three variables were considered; number of mobile hosts, changing mobility and call rates (route query frequency). The protocol performed better than Zone routing protocol (ZRP) and Minimum Connected Dominating Set (MCDS) protocol when any two variables are constant and the other variable is varying. High node density can generate more overhead because of frequency of topology updates [4].

2.4 Zone Based hierarchical link state (ZHLS) routing protocol

ZHLS is a routing protocol which incorporates the concept of location based routing in a hierarchical routing protocol. In this kind of routing the network is divided into non-overlapping zone and each zone is identified by a zone ID, each node in the zone is assigned a node ID. The zones have to be worked out during the design stage of the network. A node knows its physical location by using the GPS; this information is used to map the nodes to the respective zones. The zone can be divided based using simple geographic partitioning or radio propagation partitioning, the zone size depends on factors such as node mobility, network density, transmission power and propagation characteristics. The zones are connected to each other through virtual link i.e. if there is at least one physical link connecting any two zones.

Like any other hybrid routing protocol, ZHLS has a proactive component and reactive component. The proactive component is called as Intrazone Clustering and the reactive component is called as Interzone Clustering. Figure 7 shows the node and zone level topology in a ZHLS network

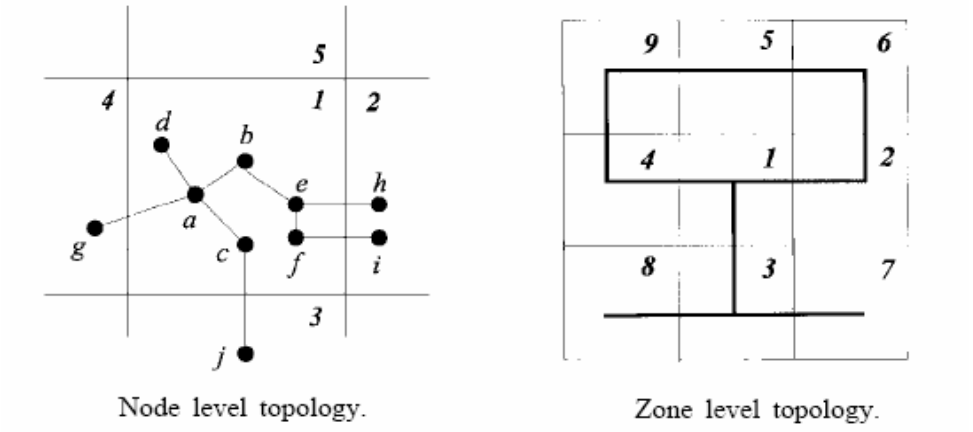

Figure 7: Node level and zone level topology in ZHLS

Each node exchanges neighboring information with all its neighbors and thus creates a local routing table in the Intrazone clustering component. All the nodes in a given zone have routing information about all the other nodes in that zone. In this scenario if the node density in one zone is higher than the other then the traffic load in that zone is higher than the other zones which can affect the overall routing. In the Interzone clustering component special gateway nodes are identified which connect one zone to another through a physical link. Gateway nodes are defined as nodes which receive responses from nodes of their neighboring zones. For example, in

figure 7 nodes e and f are gateway nodes as they connect zones 1 and 2. Using this information another table is created which stores the information for zone routing. Unlike other hierarchical protocols this protocols does not have cluster heads therefore avoiding single points of failure, bottle necks and simplifies mobility management

The biggest advantage of using this protocol is that the reactive component keeps some information about the network thus the latency is reduced, but this would be a problem in a high mobility case. The biggest disadvantage is that the zones have to be predefined, which is not a good idea in ad-hoc networks; another drawback is the dependency on a centralized system (GPS) [8]. Besides, since the nodes are assigned to zones based on GPS information, it is possible that nodes in the same zone can be out of transmission range (because there can be some obstacle between the two nodes preventing them from communicating), thus it's unable to deal with partitioning of the network. Another issue is the synchronization required among all the nodes in the same zone, this is not easy in a realistic environment.

2.5 Hybrid Routing Protocols with Routing Zone

The value of routing information decreases with respect to the distance from the information source, this relationship between value and distance is extremely valuable for routing protocol design. There is more value in providing nearby nodes with fresher and detailed information, at the expense of keeping more distant nodes less precisely informed, this idea is called as multi-scope routing. The below mentioned hybrid routing protocols, use this concept by making a reactive or proactive protocol to work on a limited scope. This limited scope is nothing but the routing zone which is measures as radius 'r' (i.e. number of hops). All the nodes which are within 'r' hops from the node are said to be in the node's routing zone.

2.5.1 Zone Routing Protocol

It is one of the first hybrid routing protocols. The most important variable in this protocol is the zone radius. A zone is defined around each node in the network, of radius equal to zone radius. The nodes of the zone are divided into peripheral nodes and interior nodes. Peripheral nodes are nodes whose minimum distance to the central

node is exactly equal to the zone radius. The nodes whose minimum distance is less than the zone radius are interior nodes.

The routing in the zone is done proactively (intra zone - IARP) and the routing between zones is done reactively (inter zone - IERP). The routing information which is stored by IARP is used by IERP to route globally; ZRP uses a concept called border casting to spread the route request throughout the network. When a node has to communicate, it searches for the destination locally in its routing zone using the proactive protocol (IARP), if there is no success then reactive routing is used. Reactive routing is divided into 2 parts route request phase and route reply phase. In the route request, the source sends a route request packet to its peripheral nodes using BRP (Border cast Resolution Protocol). If the receiver of a route request packet knows the destination, it responds by sending a route reply to the source. Otherwise, it continues the process by border casting the packet. In this way, the route request spreads throughout the network, zone by zone. To create the border cast tree 2 methods have been suggested Root-Directed Border casting (RDB) and Distributed Border casting (DB). In order to direct the quires away from the covered regions three query control mechanism; namely query detection, early termination and random query-processing delay have been suggested.

Figure 8, shows the routing zones for two nodes: S and T, assuming a zone radius of two hops. Consider node S wanting to send a packet to T, so it will send a route request to its peripheral nodes H, I, J, G. As node T is a peripheral node for node I, a path will be returned by node I.

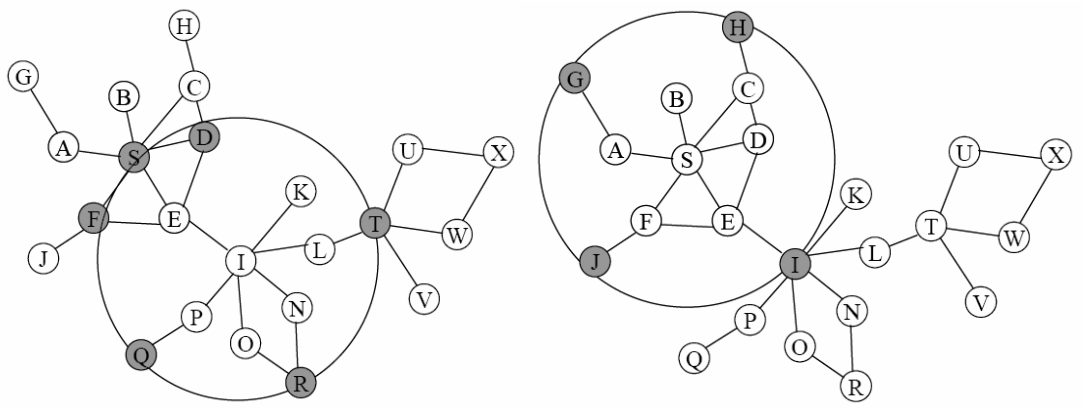

Figure 8: Routing zone and peripheral nodes for nodes S and I

The number of nodes is a zone radius must be optimal or else the proactive routing traffic will increase. The zone radius has to be determined before the network goes online which is one of the major drawbacks of ZRP.

2.5.2 Independent Zone routing protocol

The IZRP is a specialized version of the Zone routing protocol. In ZRP the routing zone radius is predefined; all the nodes need to know the default routing radius for the network. Another disadvantage, which is evident from the test results in [1], is that high mobility and/or low call rates favor smaller zone radius and vie versa, low mobility and/or high call rates favor larger zone radius. This kind of decision making regarding the zone radius cannot be predetermined, therefore the zone radius has to adapt according to the network conditions. It is not necessary that all the nodes in the network must have the same routing zone radius because different parts of the network may have different network conditions. The most important factor of IZRP is the dynamically adjusted zone radius, thus each node in the network has zone radius fine tuned according to the network conditions around the node.

In IZR there are two kinds of zones:

- **Routing zone** (Receive zone) is defined as the neighborhood around each node about which a node proactively maintains routing information about all the nodes whose minimum distance, in hops, from the node is not more than zone radius, 'r'. Routing information is maintained by receiving proactive updates from these nodes in the neighborhood, hence this zone is also called it’s receive zone. The receive zone has a regular shape i.e. it can be represented by a circle of radius proportional to the zone radius of the node.
- **Send zone** is defined as the set of nodes that require proactive updates from the node in question in order to maintain their Intrazone routing information. A node is expected to broadcast proactive updates to the members of its send zone; the send zones may not have such a regular shape and also may not even be a connected (contiguous) area. In case of ZRP the send zone and the receive zones have equal radius. Peripheral nodes are nodes which are exactly 'r' hops from the current node.

Figure 9 shows the routing zone/receive zone and send zone for node S. Since the send zone need not be of regular shape the Intrazone routing protocol (IARP) introduced in ZRP has to be modified to in order to distribute the proactive updates in

such a send zone. A controlled broadcasting scheme is used to send the proactive updates to all nodes in the send zone. The bordercasting resolution protocol (BRP) is also affected because with independent sized routing zone, it is possible that some node in the bordercast tree of the source node have a routing zone which is small, so that it lies completely within the source node's routing zone.

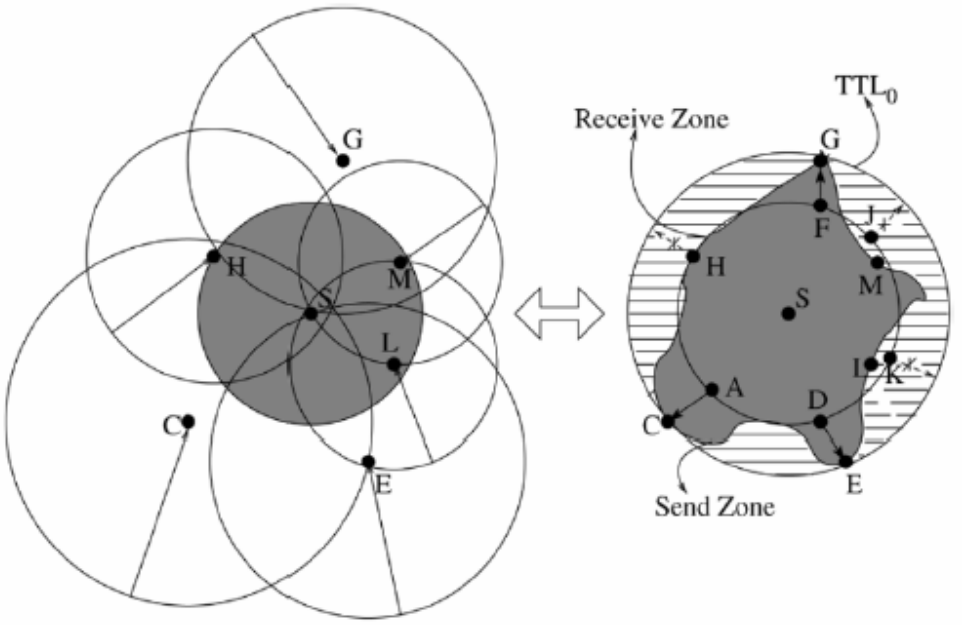

Figure 9: The receive zone is regular in shape but the send zone need not be.

Therefore some special nodes such as rebordercasting and forwarding nodes are defined. Rebordercasting node is defined as the node closest to the source node on the bordercast path from the source node to the peripheral node, such that its routing zone extends beyond the source node's routing zone. Nodes lying on the bordercast path between the source node and a rebordercasting node belong to the set of forwarding nodes corresponding to that rebordercasting node. The query control is done by the rebordercasting and forwarding nodes. Figure 10 shows bordercasting tree for node S.

The zone radius determination algorithm is the most important component of the entire routing protocol; it tunes the radius depending on the network characteristics and operating conditions. The algorithm determines optimal zone radius and is quick to adapt to changes in the network topology, in the process creating less overhead and without assigning any special roles to any node.

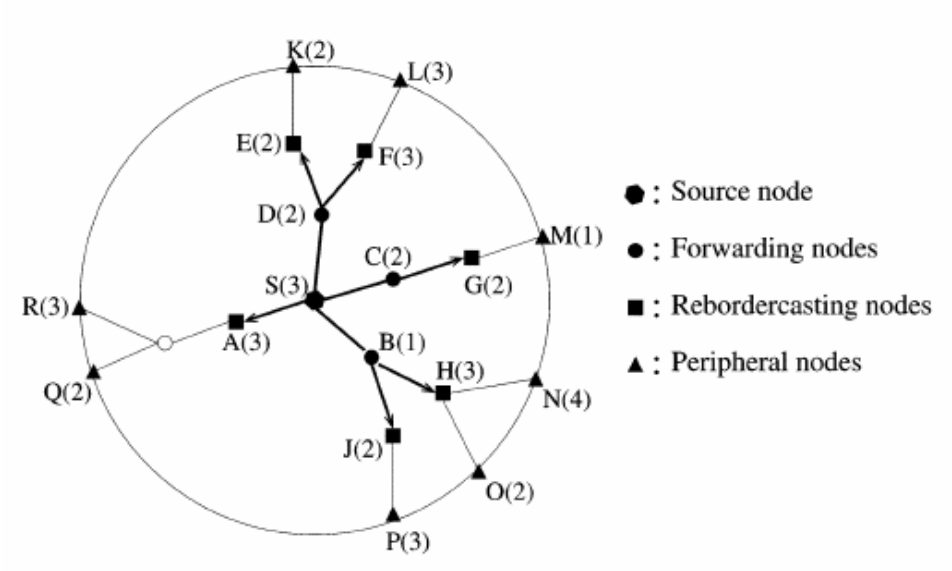

Figure 10: The bordercast tree of the source node S. The zone radii are indicated in parentheses next to the node labels.

A combination of Min Searching [2] and Adaptive Traffic Estimation [2] schemes is used for zone radius determination. The algorithm starts with Min Searching and a zone radius of one; it searches for a zone radius which gives the least routing overhead. Once the optimal value of radius is found, a threshold is set which is equal to the ratio of the reactive component to the reactive component at the optimal radius. Then the Adaptive Traffic Estimation takes over Min Searching and tries to maintain the zone radius adaptively using the threshold which was calculated by Min Searching.

In a hybrid routing protocol the most important factor is determining the scope of the proactive component of the protocol. All the hybrid routing protocols discussed so far the zone radius is pre-determined except in IZRP.

In short, the key features of IZRP are: protocol hybridization, multi-scope operation and dynamic reconfiguration. Although IZRP is adaptable and flexible its main disadvantage is that its very complex protocol from the point of view of implementation.

2.5.3 Hybrid Routing in Ad hoc networks with a Dynamic Virtual Backbone

Virtual Backbone routing (VBR) scheme is a hierarchical routing protocol that is hybrid in nature. The proactive component is similar to ZRP, in terms of the each node having its own local zone. Reactive routing relies on a hierarchical network or backbone, which is formed by a subset of the nodes that covers the entire network. These special nodes are called virtual backbone (VB) nodes as they form the backbone of the network.

VB nodes are not special nodes inserted into the network but ordinary nodes which become VB nodes based on the construction algorithm of the protocol. Ordinary nodes have a routing zone of radius ‘r’, in which routing information is maintained proactively. The VB nodes are selected in such a way that the entire network is covered. Therefore all the nodes in the network must be connected to some VB or must be a VB. The VB nodes are connected to each other by links which span multiple non-VB nodes, these links are called as virtual links. The VB nodes are also called as database because they contain the routing information for all the nodes in its routing zone and have paths to other databases. . This part is helpful (in terms of saving time) when doing the reactive part because we have to direct the queries to only the VB’ s where as in ZRP we have to query the entire network.

When a node wants to communicate with a destination, it first checks its local routing zone. If it cannot find the destination, the route request is given to the database it’s connected to; if the destination is not found in the routing zone of the database then the route request is forwarded to its adjacent databases through the virtual links. After receiving the route query, a database whose routing zone contains the destination, sends a route reply, thorough the reversed VB path, back to the query originating database. Each database along the way computes the best route segments that it can see within its zone; and appends them in the route reply packet.

Figure 11 shows an example of the VBR protocol assisted routing. The source node is 17 and destination is node 8. The VB node in node 17’s routing zone is node 41 and the VB in node 8’s routing zone is 12.

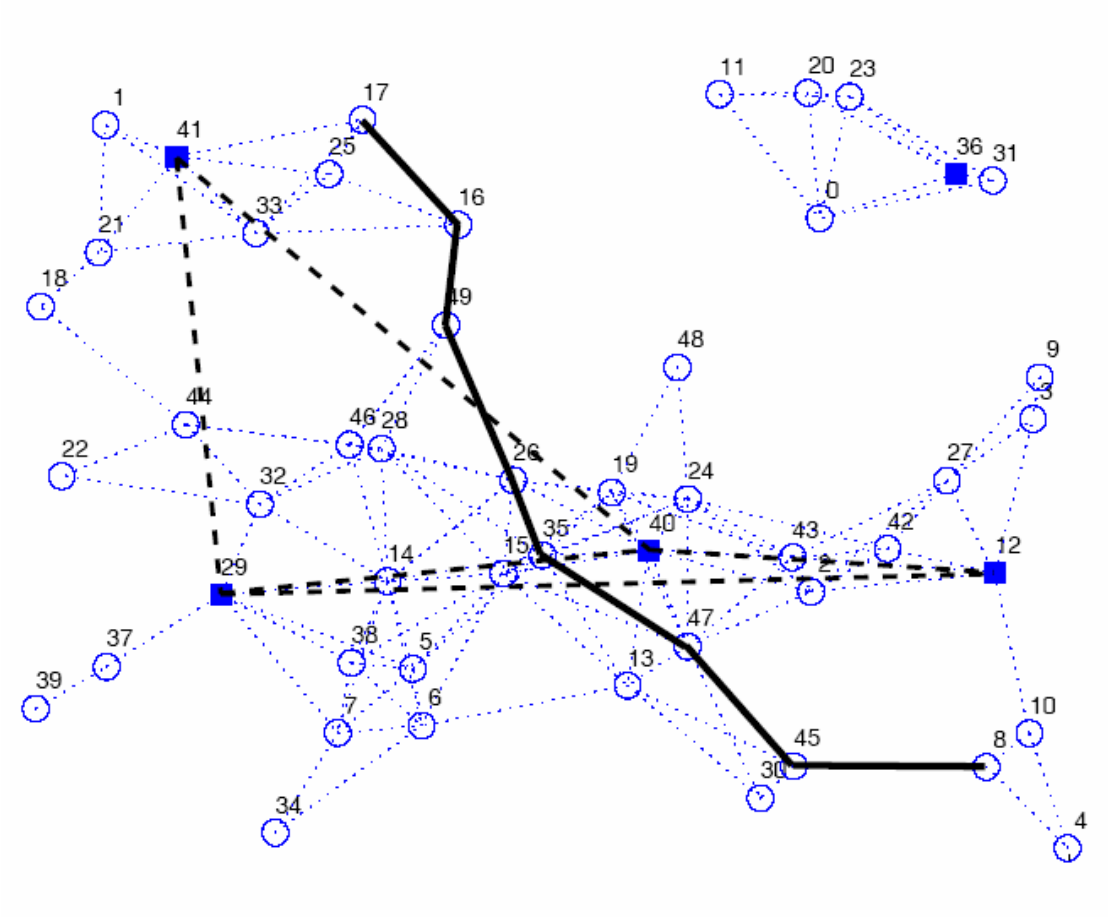

Figure 11: Snapshot of the VBR simulation with 50 nodes and r = 2. Circles: regular node; solid squares: VB nodes/databases; thin dotted lines: radio links; dashed lines: multi-hop virtual links; solid line: the actual data route between two nodes.

The advantage of this scheme is that the reactive route queries are always directed to certain location in the network, instead of flooding them through out the network. Hence the overall control traffic in the network is reduced. Even though it's a hierarchical protocol, the potential occurrence of congestion (hot-spots) is less as the routing paths need not always pass through the VB nodes. Node mobility might be a problem for this because the process of re-generation of the VB and stabilization of the network (means each node associating itself with at least one database) are time consuming operations and as mobility increases, more of these operation have to be performed very often. This approach provides multiple paths to the destination which can lead to proper load balancing. The reactive part involves a lot of control traffic transmissions.

2.5.4 SHARP: A Hybrid Adaptive Routing protocol for Mobile ad hoc networks

SHARP automatically finds the balance point between proactive and reactive routing by adjusting the degree to which routing information is propagated proactively versus the degree to which it needs to be discovered reactively. Initially all the nodes are routing reactively by using some reactive routing protocol, but as time passes and depending on the data traffic and network characteristics proactive routing zones are created about destination nodes which are popular. The radius of these routing zones varies along with time, depending on network and traffic characteristics.

The specialty of these proactive zones is that the nodes in the zone have route to the central (destination) node around which the zone is defined. The proactive protocol used is called SPR (SHARP proactive routing) protocol which creates DAG rooted at the destination. The DAG is built using a construction algorithm which is initiated every construction interval. To maintain the links in between the construction cycles update protocol are used. Ad Hoc On-demand Distance Vector (AODV) is used as the reactive protocol.

The proactive component of the protocol has got a fixed over head for a given zone radius, but as the zone radius increases the overhead also increases as for the reactive component it's very difficult to determine the overhead. The decision to vary the zone radius is taken independently by each destination. The destination can vary the zone radius to achieve some application specific goals like say reducing packet overhead or delay jitter.

The simulation shows that for destination centered traffic the algorithm perform well but for non-destination centered like traffic the overheads are high but the loss rate is kept low. There might be cases when different destination nodes try to achieve different application specific goals and this might lead to an over all deterioration in the network efficiency.

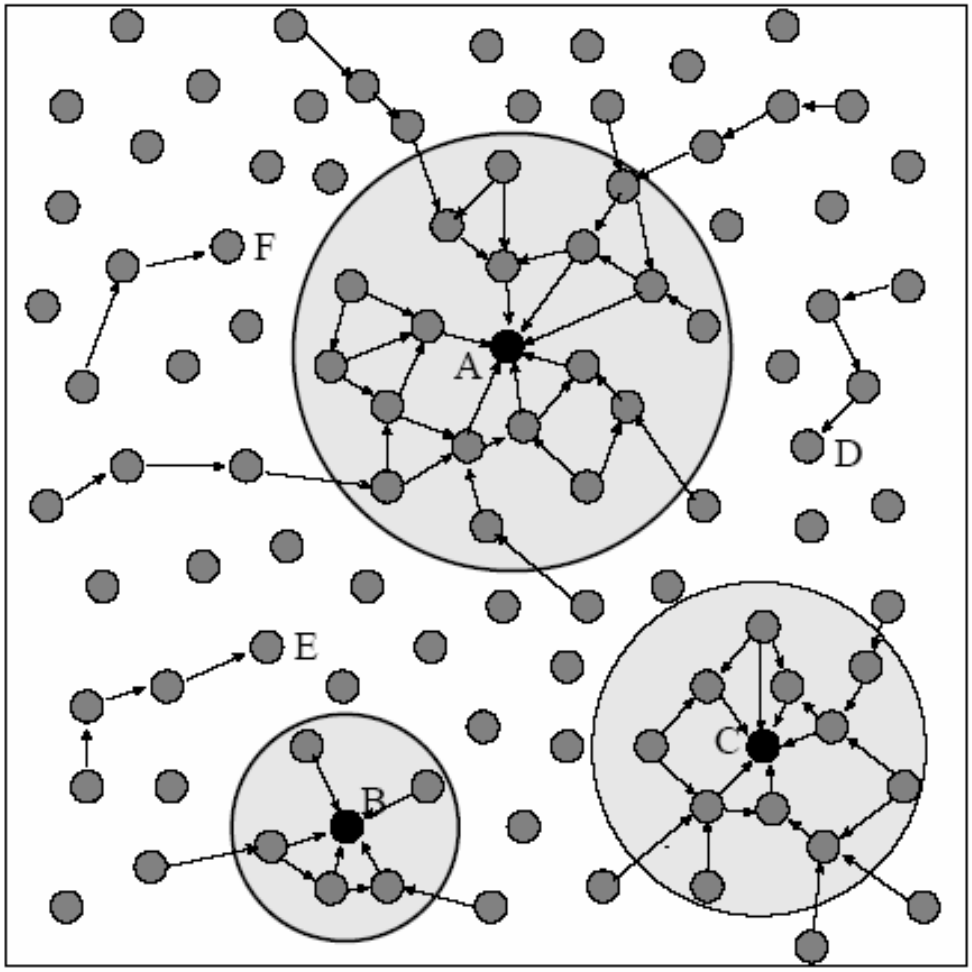

Figure 12: A snapshot of a network which uses SHARP for routing. The three circles are the proactive routing zones defined around three popular destinations. The size of the proactive zone is adjusted independently by each destination. The nodes which lie outside the proactive zone use a reactive protocol to find the route.

2.6 Comparison

The comparisons among the four main hybrid protocols ZRP, Dynamic group construction, VBR and SHARP is based on the following factors

2.6.1 Node Density

It is defined as the number of nodes per square unit. In an ideal network the nodes should be evenly distributed throughout the network. But there can be situation when nodes are concentrated in some regions of the network. This can lead to partition of the network and uneven usage of the network bandwidth therefore it is assumed in all the four protocols that the node density is evenly distributed. Partition

of the network can lead to situation in which nodes cannot communicate with each other because there are no intermediate nodes. Node density also affects bandwidth because the uneven usage of the bandwidth can affect the throughput of the entire network. Higher node density can lead to more overhead for the proactive part of the hybrid routing protocol.

2.6.2 Scalability (Number of nodes)

As the number of nodes increases the routing protocol should be able to handle the traffic efficiently. This is called as scalability. The entire four hybrid routing protocols are scalable but there are limitations to their scalability. The problem with scalability is that as the number of nodes increases the routing protocol has to search more nodes to reach the destination, thus increasing the convergence time. Out of the four routing protocol three of them namely dynamic group routing, ZRP and SHARP protocols have limited scalability. As the network size increases the convergence time also increases, but in case of VB protocol the entire network is represented by a set of nodes called as VB nodes. A single VB node covers a large set of nodes therefore if the number nodes increase the count of VB nodes might increase but this increase is very small. It is due to these VB nodes that the VB protocol has a faster convergence rate.

2.6.3 Node Mobility

It defines the rate of movement of the nodes. In a network where the nodes move around frequently the node mobility is high. SHARP is the only protocol out of the four protocols which is not actually affected by high node mobility. SHARP increases the size of proactive zone as the mobility increases. In case of ZRP as the mobility increases there is more link formation and breakage and this make the stored routing information invalid. Therefore more time and bandwidth is spent in control traffic than in data traffic. In dynamic group routing protocols high node mobility is a problem because link breakages lead to a lot of computation and restructuring of the network. As there is a lot of routing information which is stored in the routing group, frequent mobility makes this information invalid. In VB protocol mobility can be a problem as the local routing information which is stored will become invalid, but the good aspect of VB protocol is that to reconstruct the VB local information is used which is obtained quickly.

2.6.4 Call rate frequency

It defines the number of routing queries generated per unit time. A high call rate can be easily handled if the nodes keep some local routing information. If no routing information is saved then the time required to answer a route request query will increase thus degrading the performance of the network. VB protocol, ZRP and dynamic group routing protocol save local routing information which helps in answering the route request query, but in the case of SHARP; a reactive protocol is used therefore every time a new route request will result, in degraded performance of the network.

2.6.5 Routing overhead (proactive + reactive)

Hybrid protocol consists of proactive and reactive components, so the routing overhead is generated by both the components. This factor is a major concern for all hybrid protocol. In an ideal routing protocol the routing overhead should be to the minimum. The proactive component will always produce more overhead than the reactive component. In ZRP and VB protocols the proactive component has similar overheads. In case of dynamic group routing protocol the proactive component overhead is more than the ZRP and VB protocols because local routing information has to be maintained for P and N clusters. In case of SHARP the routing zone is only formed around the popular destinations, unlike the other protocols in which routing zone is around all nodes therefore the proactive routing overhead is less compared to the other protocols. The reactive component in ZRP and dynamic group routing protocols have similar overheads as both perform some kind of multicasting to spread the route request query. In VB protocol the reactive component has less overhead than the other three protocols because the route request queries are directed to specific parts of the network through the VB nodes. In case of SHARP the reactive component overhead is the most because it uses flooding technique to spread the route request query to all parts of the network.

2.6.6 Effect of traffic load

In the communication between source and destination multiple links are used, these links are also used for communication between different sets of sources and destinations. So there is a possibility that as the usage of the link increases, this leads to a decrease in the link bandwidth. Hence in spite of having an optimal path between

source and destination, the data traffic might take more time to reach the destination. In case of ZRP, dynamic group routing and SHARP only one optimal path is found therefore there is no way of balancing the traffic load, but in case of VB protocol traffic load can be handled efficiently. Multiple paths are provided between the source and the destination; the most optimal path is selected, but as the traffic load increases the other routes can also be used thus balancing the traffic load over multiple links.

2.6.7 Implement able network scenarios

The table below shows the scenarios for which the four protocols are best suited

Protocols	Scenarios
Dynamic group routing	Low-High call rate, Low-Moderate node mobility
ZRP	Low-High call rate, Low node mobility
SHARP*	Low-Moderate call rate, Low-High node mobility
VB	Low-High call rate, Low-Moderate node mobility

*the performance for SHARP also depends on the data traffic characteristics

2.6.8 Path Optimality

Optimum path is the best path between the source and destination. In all the routing protocols the main aim is to provide this shortest path, but some of them just provide the first available path rather than the optimal path between the source and destinations. If a proactive protocol is used it can be guaranteed that the path which is obtained is the best path, as the routing information stored by the proactive protocols will be used to determine the best path. In case of a reactive protocol there is no routing information which is stored therefore there is no guarantee that the path obtained is the best path.

In ZRP, SHARP and dynamic group routing protocols the reactive part uses a multicast mechanism to spread the route queries therefore there is not necessary that the path which is obtained is the best.. In case of VB protocol the reactive part is

directed to specific VB. Each VB knows the nodes in its routing zone and the best path to these nodes. When determining the path between the source and the destination, the segment which gives the overall best path is selected in each segment.

2.7 Selection of hybrid protocol: DVB

The reasons for selecting the Dynamic Virtual Backbone protocol to implement in NS2

- The convergence time for the reactive component of the protocol is the least compared to the other protocols considered?
- The path obtained is the shortest
- The protocol can be extended to take care of traffic load balancing
- Local information is used to reconfigure/reconstruct the VB therefore it's a faster process
- As there is no broadcasting method used in this protocol, it is more reliable
- Simpler representation of the whole network because a few VB nodes can represent the whole network
- Advantageous for the network with high call rates

These observations were determined from the study of 4 different wireless networks [11, 12, 13, 14].

- Looking at some real world network traffic it can be observed that most people do not make maximum use of the mobility. The percentage of the population which is mobile is very small compared to the others. The network traffic distribution is evenly spread among different nodes and applications.
- Another aspect which was observed was that the movement of people was in neighboring areas i.e. if a node was mobile the movement would be in near by area, very seldom did a node move huge distances.
- The call rate for the nodes was high compared to their mobility. Hence to handle real world ad hoc networks the protocols must be able to handle higher call rates.

Chapter 3

Dynamic Virtual Backbone Routing

Dynamic virtual backbone [7] routing method is a hierarchical routing protocol which is hybrid in nature. Virtual backbone routing (VBR/DVB) is influenced by the Zone routing protocol (ZRP) [1] and hierarchical routing architecture. Hierarchical routing is a prefect example of selective representation. In this routing method nodes are grouped into super nodes based on some predefined metric, these super nodes are further grouped into super-super nodes and so on. The only way the nodes at lower level have information about faraway nodes is through the higher level nodes.

Selective representation of the network topology is advantageous as it reduces the control overhead of the routing protocol; because each node need not maintain the routing information about all the other nodes in the network, only nodes which are in its routing zone. Another advantage of selective representation is that the route query messages can be directed to different region of the network, instead of flooding throughout the network. Since VBR is a hybrid routing protocol it has two routing components, proactive component and reactive component.

The proactive component is similar to the zone routing protocol, in which a routing zone of radius 'r' is defined around each node and each node has routing information about each other nodes in its routing zone, along with this functionality the proactive component in VBR is also responsible for the construction, maintenance and connectivity of the VBs.

The reactive component of VBR is applied if the source does not find the destination in its local routing zone, in this case the route query is send to the nearest VB, which tries to resolve the query by looking in its routing zone, if the destination is still not found then the VB broadcasts the route query to all the VB through the multi-hop paths connecting the Vbs.

Some interesting facts about the VB nodes are that any node in the network can serve as a VB during its time as part of the network. The VB nodes are connected to each other through virtual links which span multiple regular nodes. VB nodes are also called as “database”, because during the reactive routing phase the source nodes query the VB nearest to them. The VB are selected in such a way that all the nodes in

the network are connected to at least one VB i.e. in other words the VB represents the entire network.

3. 1 Proactive Component

The proactive component in VBR uses a concept of zones, similar to the ones used in ZRP. A zone of radius' r' is defined around each node in the network this zone is called as routing zone of a node. The zones of neighboring nodes overlap each other. The radius 'r' is measured in terms of the number of hops. The proactive component can be implemented as simple link state routing protocol or any of the complex proactive protocols like DSDV; IARP (which is implemented in ZRP) can be implemented. In this case a simple link state routing protocol is used.

Each node sends out a connectivity packet periodically to the entire node in its routing zone. Figure 13 shoes the format of the connectivity packet:

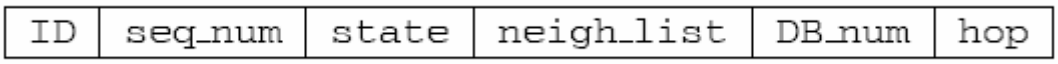

Figure 13: Connectivity Packet

- *ID* is the identity number of the node sending the packet
- *seq_num* is the unique sequence number, which increases every time a new connectivity packet is send
- *state* is the current state of the node which is sending the packet
- *neigh_list* is the list of neighbors of the node sending the packet
- *DB_num* is the number of VBs (databases) in the sending nodes routing zone
- *hop* is initialized to the 'r' which is the routing zone radius and is decremented every time the packet is forwarded; when the hop count is zero the packet is dropped, the packet is also dropped if the packet is seen earlier.

3.2 Virtual Backbone Construction

The network topology can be considered stable when all the nodes have at least one VB node in its routing zone or are a VB itself. In other words a set of VBs with some minimum cardinality must cover the entire network; this problem is similar to the Minimum Set Covering (MSC) problem. Figure 14 shows the simple representation of the VB selection problem

$VB = \phi$
while VB does not cover all nodes,
    1. find node v with the max dependency number
    2. add v to VB
    3. re-compute dependency number
end

Figure 14: The VB selection problem

There is no centralized controlling in ad-hoc network, therefore implementing the MSC algorithm is not possible, a distributed version of the algorithm called as Distributed Database Coverage Heuristic (DDCH) is implemented. The DDCH is implemented asynchronously i.e. the DDCH algorithm is executed asynchronously on each node.

3.2.1 Distributed Database Coverage Heuristic (DDCH)

The nodes in the VBR routing must be in any one of three states; panic, samaritan or normal. In a fully functional network all the nodes which are connected to a VB node are in normal state (normal node). If a node is not connected to a VB then it is in panic state (panic node). If a node is connected to a VB node, and if there are panic nodes in its routing zone then such a node is in samaritan state (samaritan node). A node in samaritan state can potentially become a VB to cover the panic nodes in its routing zone under some conditions.

When the network start-up initially, there are no VB nodes so all the nodes are in the panic state. All nodes in panic or samaritan state send node state packet periodically to all nodes within '2r' hops. Figure 15 shows the node state packet format:

ID	seq_num	state	dep_num	hop

Figure 15: Node State Packets

where

- *ID* is the identity of the node sending the packets
- *seq_num* is the sequence number which always increases each time a new node state packet is send

- *state* is the current state of the node sending the packets
- *dep_num* is the dependency number of the node, dependency number is defined as the number of panic nodes in the nodes routing zone including the current node
- *hop* is initialized to '2r' and is decremented every time the packet is forwarded, when the hop count is zero then the packet is dropped, the packet is also dropped if the packet was seen earlier.

Nodes in panic or samaritan state collect these node state packets and extract the dependency numbers from all the panic and samaritan within '2r' hops. If the dependency number of the current panic node is higher than the other nodes then, the current node becomes the VB; else there are three possibilities for the current panic node. If no new VB appears in the routing zone within a threshold time, then the current node remains in panic , re computes the dependency number and send a new node state packet; if a VB is found in the current node's routing zone and there are no panic nodes in the current node's routing zone then the current node changes its state to normal; if a VB is found in the current node's routing zone and there are panic nodes in the current node's routing zone then the current node changes its state to samaritan.

If the current node is a samaritan node, it computes its dependency number and check with the other nodes in its routing zone, if its dependency number is higher then the current node changes its state to VB; else there are two possibilities for the current samaritan node. If there are no panic nodes in the current node's routing zone then the current node changes its state to normal else if there are panic nodes in the current node's routing zone, then it remains in samaritan state, re computes its dependency number and send out new node state packets.

The dependency number is calculated from the previously collected node state packets at each panic and samaritan node before sending new node state packets.

The above algorithm is run on all panic and samaritan nodes until no panic nodes are remaining and the entire network is covered by VBs. There are proofs in [2] which prove that the number of VBs generated by DDCH is same to the number of VBs generates by MSC algorithm.

3.2.2 VB Structural Maintenance

There are two possible ways of updating the VB in the network, firstly the DDCH algorithm can be executed periodically, thus generating the VB nodes after fixed interval of time, and secondly the VB can be generated only when there are changes in the network. The second method is much better because changes in one part of the network will not affect the VB in other parts of the networks. There are two possibilities when a VB node moves, the VB node can move to a new location and still be connected to the network or it can detach itself from the network.

In either case the neighboring nodes must find a new VB in their routing zone or if no new VB is found must set themselves to panic state. Now the panic nodes and the samaritan nodes induced by them must apply DDCH to generate a new VB to cover all the nodes. If a node in normal state starts moving, it does not affect the structure of VB nodes in the network.

The DDCH algorithm is applied asynchronously to all panic and samaritan nodes; therefore there is a possibility of more number of VB nodes being generated. A redundancy check is performed on all VB nodes periodically to keep optimal number of VB nodes. This is where the DB_num field mentioned in the connectivity packet comes into picture. DB_num field gives the number of VB in a nodes routing zone, using this information the redundancy number for a VB is calculated. Redundancy number of a VB is defined as the minimum DB_num among all the nodes within the VBs routing zone. A VB is redundant if its redundancy number is greater than or equal to 2. A procedure similar to DDCH is used in which the redundancy number is exchanged between near by VB nodes to eliminate the redundant VB nodes is a distributed greedy manner.

Each VB is made aware of all the VBs (database) within '2r+1' hops from it through the gateway nodes (explained below). A VB resigns if its redundancy number is the highest among all the VBs '2r+1' hops away. Resigning of a VB has the same effect as detachment of a VB on the nodes covered by the VB.

3.2.3 VB Connectivity Maintenance

Adjacent databases are connected to each other through multi-hop links. A database is said to be adjacent if it’s within '2r+1' hops from the current database. These databases are connected to each other through gateway nodes. A node is a

gateway from database DB1 to database DB2 if it's within 'r' hops from DB1 (upstream DB) and it is exactly 'r+1' hops from DB2 (downstream DB)

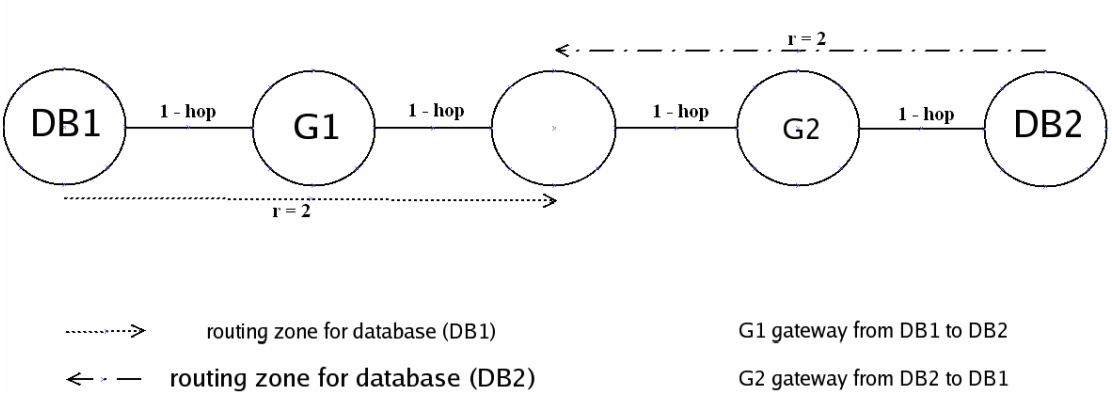

Figure 16: Gateway nodes between DB1 and DB2; r = 2

When ever a node 'r' hops away from database (DB/VB) receives a connectivity packet from the DB, it sends a gateway notification packet to all its neighbors. The set of neighbors that are not 'r+1' hops away from the DB ignore the packet, but those that are 'r+1' hops away are gateways which update their record of the DB. Thus gateway nodes monitors the connectivity to its downstream databases and reports any changes of downstream database membership to all upstream databases within 'r' hops. Therefore through the local routing table and the gateway nodes the VB nodes keep up-to-date information about other VB nodes '2r+1' hops away. There can be more than one gateway node connecting two databases.

3.3 Route Determination (Reactive Component)

The VB routing process is carried through three control packets route query, route feedback and route designation. When the destination is not in the source nodes routing zone a route query packet is send to the nearest VB node. Figure 17 shows the format of the route query packet:

S	D	seq_num	DB_path

Figure 17: Route query packet

where

- S, D are the source and destination
- *seq_num* is the sequence number which is used to limit the redundant transmissions

- *DB_path* is accumulated path containing the identity of all databases that have forwarded the packet.

Each VB will check to see if its local zone contains D, otherwise it will forward packet to next VB and append its name to the DB_path. In figure 18, DB3 can see the destination in its routing zone, when the route query packet arrives on DB3 the field DB_path contains DB1, DB2 and now DB3 starts the route feedback i.e. when D is local the VB starts the route feedback.

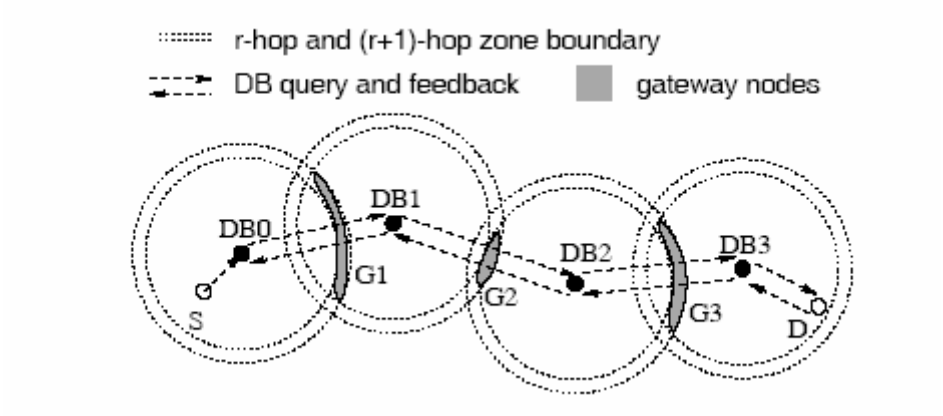

Figure 18: Route query and feedback process

In the route feedback phase each database calculates the best path between the gateway nodes and the destination and adds the path to the packet. In the figure 18, DB3 determines the shortest route from every node in its gateway node set (G3) to the destination D, using only links it can see, it then sends the length of these routes, one for each node in its gateway node set (G3), together with the identity of the respective starting node in its gateway node set to DB2. At DB2 all possible routes from its gateway node set (G2) to (G3) are calculated using links it can see. The path from G2 to G3 is added to all the paths passed from DB3, thus getting the paths from G2 to D. This procedure continues until DB0 receives the packet from DB1 containing the constrained shortest route from every node in G1 to D. Using the routes it knows about, between S and G1, DB0 finds a node V1, in G1, through which the S-to-D route is the shortest. It then sends the selected S-to-V1 route to S, and passes along the identity of V1 to DB1. This starts the route designation process.

In the route designation step, the databases pass down segments of the selected constrained best route along DB_path. DB1 receives the identity of the selected gateway node V1 from DB0, this process happens till DB3.

Thus a multi-hop route has been established between S and D, with V1, V2, and V3 as the intermediate nodes. In the above example hop count is used as the measure of optimality of the path, some other metric like quality of service can be used.

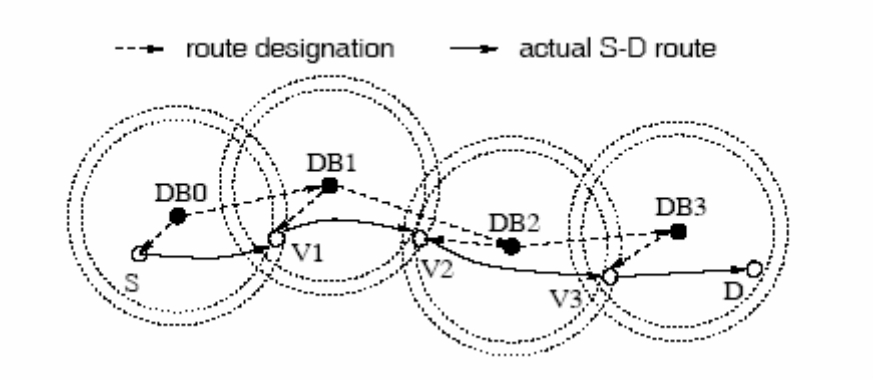

Figure 19: Route designation process

3.4 Caching Enhancement to VBR

VBR supports two types of route caching direct route caching and VB path caching. In direct route caching a source node caches routes so that a route is available when an application, running within the same node demands it. As an extension to the above, an intermediate node can cache the route and reply to the source with the cached route. In VB path caching, whenever a VB assisted route query is successful, all VB nodes on the VB path that receive the route feedback stores the destination node's ID, its associated VB node, and the VB path that leads to that VB node for future use.

Chapter 4

NS2 Simulator Overview

NS2 [17] is an object-oriented simulator developed as part of the VINT project at the University of California in Berkeley. This simulator is the most widely used tool in the networking research community. It provides support for simulating TCP, routing, multicasting protocols over wired and wireless (local and satellite) network. The simulator is event-driven and runs in a non-real-time fashion. It consists of C++ core methods and uses Tcl (Tool Command Language) and Object Tcl shell as interface allowing the input file (simulation script) to describe the model to simulate. Users can define arbitrary network topologies composed of nodes, routers, links and shared media. A rich set of protocol objects called agents can then be attached to nodes. The simulator suite also includes a graphical visualizer called network animator (nam) to assist the users get more insights about their simulation by visualizing packet trace data.

4.1 Basic Wireless Model in NS2

The wireless model essentially consists of the MobileNode at the core, with additional supporting features that allows simulations of multi-hop ad-hoc networks, wireless LANs etc.

The MobileNode object is a split object. The class MobileNode is derived from the class Node. A MobileNode is the basic Node object with added functionalities of a wireless and mobile node like ability to move within a given topology, ability to receive and transmit signals to and from a wireless channel. A major difference between them, though, is that a MobileNode is not connected by means of Links to other nodes or mobile nodes. The mobility features including node movement, periodic position updates, maintaining topology boundary etc are implemented in C++ while plumbing of network components within MobileNode itself (like classifiers, dmux , LL, Mac, Channel etc) have been implemented in Otcl [17]. The figure 20 shows the basic architecture of a wireless node in NS2.

The function of a node when it receives a packet is to examine the packet's fields, usually its destination address, and on occasion, its source address. Depending on the value of field the packet is either forwarded or processed at the current node.

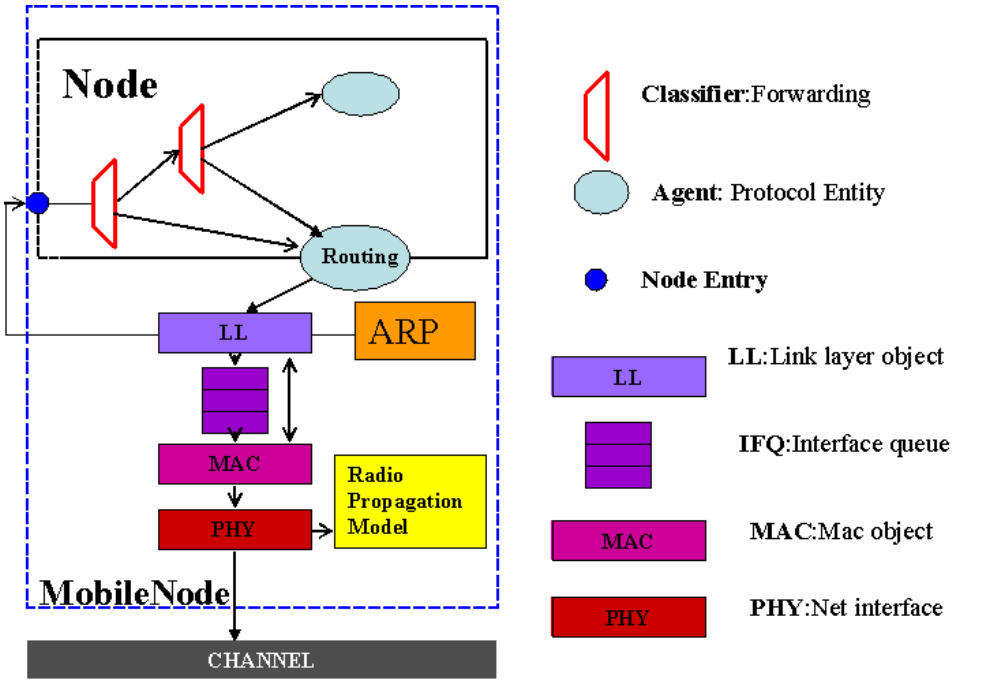

Figure 20: Architecture of wireless node in NS2

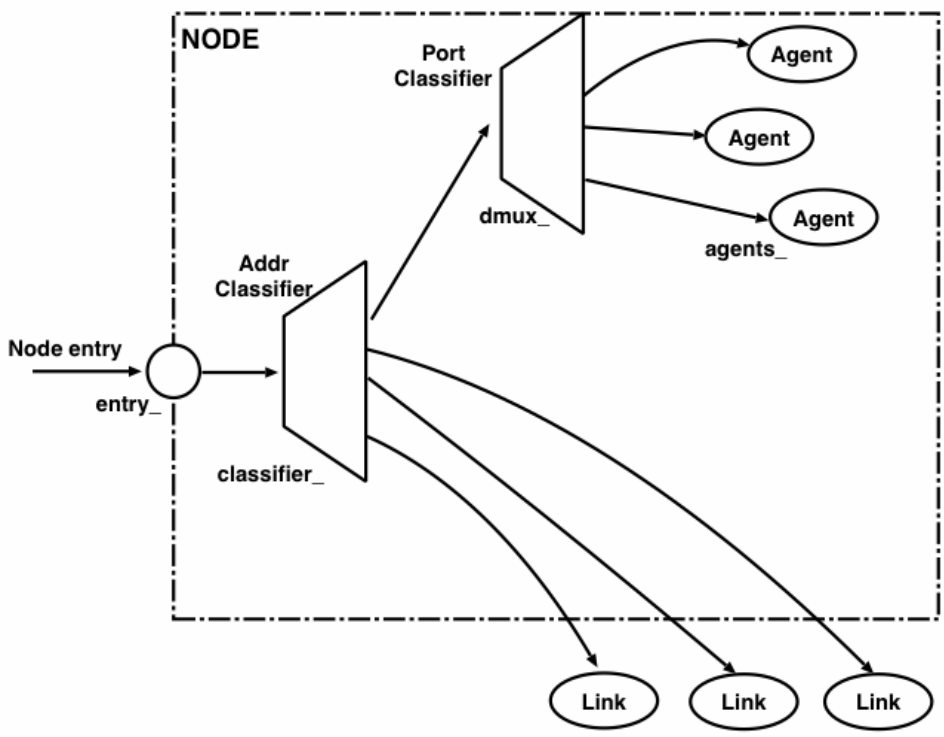

Figure 21: A simple of Unicasting Node

In NS, this task is performed by a simple classifier object. A classifier provides a way to match a packet against some logical criteria and retrieve a reference to another simulation object based on the match results. There are multiple classifier

objects, each looking at specific portion of the packet. Each classifier contains a table of simulation objects indexed by slot number. The job of a classifier is to determine the slot number associated with a received packet and forward that packet to the object referenced by that particular slot.

An NS node is essentially a collection of classifiers. The simplest node (unicast) contains only one address classifier and one port classifier. The routing agent is the actual implementation of the routing protocol in NS2. Once the packet enters the node, depending on the address, ports and packet type fields the packet is forwarded to the respective routing agent. The figure 21 shows the structure of a unicasting.

4.2 Network Stack

The network stack for a MobileNode consists of a link layer (LL), an ARP module connected to LL, an interface priority queue (IFq), a MAC layer (MAC), a network interface (netIF), all connected to the channel. These network components are created and plumbed together in OTcl.

4.2.1 Link Layer

The link layer simulates the data link layer of the network stack. This layer implements protocols such as packet fragmentation and reassembly and reliable link protocol. The link layer also has to set the MAC destination address in the MAC header of the packet. In the MobileNode this functionality of address resolution from IP to MAC is provided by the ARP module. Normally for all outgoing (into the channel) packets, the packets are handed down to the LL by the Routing Agent. The LL hands down packets to the interface queue. For all incoming packets (out of the channel), the MAC layer hands up packets to the LL which is then handed off at the node_entry_ point.

4.2.2 Address Resolution Protocol (ARP)

This module receives packets from the LL layer. If the ARP has the MAC address for the IP then it writes it into the MAC header of the packet. Otherwise it broadcasts an ARP query, and caches the packet temporarily. There is a buffer for a single packet for each unknown destination hardware address. This buffered packet is

dropped if additional packets are sending to the ARP for the same destination. Once the hardware address is resolved, the address is written into the packet header and inserted into the Interface Queue.

4.2.3 Interface Queue

It is priority queue which gives higher priority for routing packets (over data packets) and inserts the routing packets at the head of the queue. It also supports running filters over all packets in the queue and removes those with a specified destination address. The interface queue is implemented by the class PriQueue in NS2. In this project the VBR routing packets are set to higher priority like the other routing protocol packets.

4.2.4 Mac Layer

This layer simulates the medium access protocols which are used in a shared medium environment such as the wireless and local area networks. The MAC object is duplex, when sending a packet the MAC layer adds the MAC header to the packet and send it to the physical layer. When receiving the packet, the packets are received asynchronously from the physical layer and after processing it the packets are passed to the link layer. In NS2 two MAC layer protocols are implemented for mobile networks, which are 802.11 and TDMA.

4.2.5 Network Interfaces

The Network Interface layer serves as a hardware interface which is used by MobileNode to access the channel. The wireless shared media interface is implemented as class Phy/WirelessPhy. This interface subject to collisions and the radio propagation model receives packets transmitted by other node interfaces to the channel.

4.3 Changes in NS2

In general, every routing implementation in ns consists of three function blocks: Routing agent, Route logic (uses the information gathered by routing agents or the global topology database in the case of static routing to perform the actual route computation) and Classifiers sit inside a node (they use the computed routing table to

perform packet forwarding). When implementing a new routing protocol, one does not necessarily implement all of these three blocks. For instance, when one implements a link state routing protocol, one simply implement a routing agent that exchanges information in the link state manner, and a route logic that does Dijkstra on the resulting topology database. It can then use the same classifiers as other unicast routing protocols.

The following are the basic steps to implement a new MANET routing protocol (routing agent) in NS2; detailed explanation can be found in [18]:

- Creating a new packet type and declaring the contents of this new packet, then this new packet is bound to a Tcl Interface.
- Creating a new routing agent for the new routing protocol; this is done by creating a new class and then it is bound to a Tcl interface.
- Creating a new routing table for the new routing protocol

The following changes have to be made in the existing NS2 files to integrate the new protocol with NS2:
common/packet.h - the new packet type must be specified
trace/cmu-trace.h & trace/cmu-trace.cc -- to provide trace support for the newly implemented protocol
tcl/lib/ns-packet.tcl -- adding the new routing protocol to list of existing routing protocols
tcl/lib/ns-default.tcl -- specifying the default value for the parameter of the routing protocol
tcl/lib/ns-lib.tcl -- add procedure for creating wireless node with the new routing protocol as the routing protocol
queue/priqueue.cc -- setting the priority for routing packets of the new routing protocol
Makefile -- adding the object files, so that the new protocol is compile when NS2 is built

4.4 Protocol Implementation Details

The Dynamic virtual routing protocol is implemented in NS 2.29.3. The code

for the VBR can be found at ns2.29/dvb containing six files.

debug.h - declares some macro for debugging purpose

dvb_pkt.h - declares the structure of the new packet (PT_DVB) for VBR

dvb_table.h - declares the different data structure used in VBR

dvb.h - declares the function and defines the miscellaneous parameters used in the VBR protocol

dvb.cc - implements the VBR algorithm

dvb_table.cc - implements the routing table and the other data structures for storage

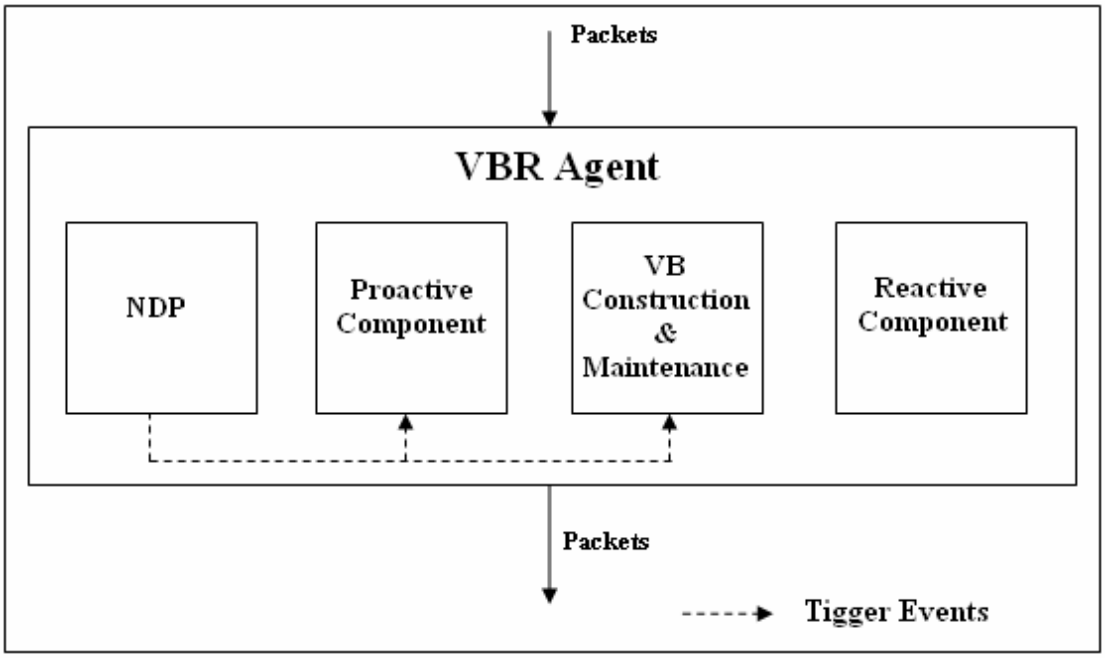

Figure 22: The component in the VBR implementation

Initially when the network is new, it takes some time for the network to stabilize (means time to form the virtual backbone and get routing information), the probability of finding a routing during this interval is very low thus it is recommended the packet transmission must start after the stabilization time.

There are four major modules in my implementation of the VB routing protocol, these are the Neighbor Discovery protocol (NDP), IARP (Intrazone routing protocol), VB construction & maintenance and the Reactive component. The NDP, proactive component and VB construction and maintenance modules have timers associated with them because some operations have to be performed periodically. The values (start time of timer varies between 1-3 seconds) for these timers are assigned randomly at each node but the order of the start time for these timers are fixed.

4.4.1 Neighbor Discovery Protocol (NDP)

In the Neighbor Discovery protocol (NDP) nodes advertise their presence to each other by periodically transmitting "HELLO" packets. On receiving a "HELLO" packet a node first checks if it has received a packet from that node, if this is the first time then it reports a new neighbor found and adds the node to the neighbor list else it will just update the expiry time for the old entry. The neighbor list table is scanned periodically to remove expired entries from the list, thus reporting a neighbor lost. A neighbor found or lost event is reported to the IAPR module. The packet type field of a VB routing packet will have the value "HELLO" [16]. The timers associated with the NDP module is the “HelloTransmitTimer“ which schedules “HELLO” packets transmission are regular intervals and the “NeighborhoodTableUpdateTimer” which schedules scans of the neighborhood list of a node at regular interval to removes expired entries (lost neighbors).

4.4.2 Intrazone Routing Protocol (IARP) – Proactive Component

In this version of IARP nodes compute Intrazone routes based on the link state of each routing zone node. A node may receive link state updates either from an IARP link state packet (IARP_UPDATE) or from an interrupt generated by the Neighbor Discovery Protocol. Link states are maintained in a Link State table. When a link state packet is received, if the update is already seen then the expiry time of the entries is updated and if the time to live field (TTL) is still valid then the packet is forwarded else packet is dropped but the routing table is not recomputed. If the update packet reports a change, then the change is made and the routing table is recomputed. The link state table is scanned frequently to remove expired links. When a new neighbor is reported then the entire nodes in its routing zone transmit their respective link state tables to that node [16].The timer associated with the proactive component is the “IarpTablesUpdateTimer” which schedules the sending of the current node’s link state information to nodes ‘r’ hops away from the current node and schedules updates to the current nodes link state and routing tables.

4.4.3 VB Construction & Maintenance

When the network is formed all the nodes are in PANIC state, therefore they

broadcast connectivity packets (CONN_PKT) periodically. Before sending a connectivity packet a node calculates its dependency number based on the information gathered in its node state table (explained later). When sending a connectivity packet the following information is also send with it: state of the node, DB_num of the node (number of VB in the node's routing zone). When a node receives a connectivity packet, it checks if the packet is received from a new node, if so a new entry is made in the connectivity table (stores all the information from the connectivity packet), else the entry in the connectivity table is updated. Immediately after this the connectivity table is purged, to remove invalid entries. In this implementation the connectivity packets are transmitted even if the node is in VB or normal state.

The redundancy packet (RED_PKT) is send if the connectivity packet was received on a VB node. Before sending the redundancy packet the redundancy number (minimum DB_num among all nodes within the VBs routing zone) of the node is calculated based on the information in the Connectivity table. When a redundancy packet is received it check if the redundancy packet was received from new VB node, if so it makes a new entry in the redundant VB list storing all the information from the redundancy packet, else it just updates the expiry value of the entry.

If the connectivity packet was received from a VB node, the "ttl" value in the packet was valid and the current node is not a VB, then it sends a gateway notification packet (GATEWAY_PKT) to all its neighbors. On receiving a packet, a node first checks if its 'r+1' hops from the VB which is being advertised by the gateway notification packet. If so then that node makes an entry in its gateway list for the VB (downstream) and also the neighbor from which it received the gateway notification packet. After this the gateway list is purged to remove invalid gateway entries.

When the node is in a PANIC state node state packets (NODE_STATE_PKT) are also broadcasted along with the connectivity packet. Before sending the node state packets, the current node calculates its dependency number (number of panic node in the node's routing zone) and stores it the node state packet. On receiving a node state packet, a node first checks if it’s from a new node, if so then makes a new entry in the node state list, else it just updates the expiry interval of that entry.

The node state packets are broadcasted periodically. Before broadcasting the node state packet the VB construction algorithm (DDCH) discussed in chapter 3 is

applied. There were situation when a single node in the entire network was in panic state and all the nodes in its routing zone where in samaritan state; in the normal DDCH algorithm the dependency of the samaritan node is checked and the node with higher dependency number is made the VB, but in this case the dependency number of all the nodes remains same, thus resulting in an unstable network. An additional constraint has been added to the algorithm; when a node is in the samaritan state it calculates the number of samaritan (s), panic (p) and VB (v) nodes in the node's routing zone, if the s >= v >= p then the node is made a VB node. When this condition is applied, the first node which executes the algorithm will become the VB thus resolving the situation. Thus by applying the above procedure a set of VB nodes are selected with cover the entire network. In this implementation node state packets are transmitted even if the nodes are in VB or normal state.

There three timers associated with VB construction and maintenance module are “ConnectivityPktTransmitTimer” which schedules the transmission of the connectivity packets at regular interval, ”NodeStateTransmitTimer” which schedules the transmission of the node state packets at regular interval depending on the state of the node and the ”RedundancyCheckTimer” which schedules the transmission of the redundancy packets at regular intervals.

4.4.4 Reactive Component

The reactive part of the implementation comes into picture when a route is not found using the local routing information. If the source of the route is a normal node then, a route request (ROUTE_REQUEST) is send to the nearest VB node else if the source is a VB node then directly it starts with the route query (ROUTE_QUERY).

In the route query phase first the VB is added to the VB list and the visited node list in the route query packet, then the gateway list is checked, if the VB node is a gateway node then it forwards the route query to the downstream VBs using the information stored in the gateway list and then the VB just broadcast the packet to all the nodes in its routing zone (in the hope that there can be other nodes which are gateway to some other VBs). The route query can be processed by three kinds of node VB nodes, normal state and samaritan nodes.

If the route request is received on a VB node then, it check its routing zone for the destination, if found a route feedback is send else the VB is added to the VB list and the visited node list and the above procedure is followed.

If the route request is received on a normal state or samaritan node, first the node is added to the visited list in the packet and then if the node is a gateway node then it sends the packet to the respective VBs using information in its gateway list else the packet is forwarded to the next hop address.

Once the packet reaches a VB which can see the destination, the route feedback (ROUTE_FEEDBACK) is started. In this implementation the route feedback reverses the path in the visited node list present in the packet and thus reaches the VB which had started the route query.

Once feedback reaches the last VB the route reply (ROUTE_REPLY) (containing the route) is send to the source node which requested the path. Once the route reply (source is a normal node) or route feedback (source is a VB node) return the entire path, the path is cached on the source node. After the first data packet reaches the destination the reverse path to the source is cached at the destination, this path is used by the acknowledgment packet to reach the source node.

When the data packet is being routed through the network, the next hop address is checked at each node, if the next hop address is not a neighbor of the current node then a route error packet (RERR_PKT) is send to the source node by reversing the path in the packet.

Chapter 5

Performance Analysis

5.1 Testing Methodology

VBR is a new protocol therefore two sets of test are performed; the first test is called as tuning tests and the other is the comparison tests. Tuning tests are aimed at showing the effects of different parameter settings on the performance of the VBR protocol. Comparison testing is aimed at comparing the performance of two protocols DSR and VBR using NS2. The same traffic models and mobility pattern for each protocol are used and our simulations repeated couple of times for each scenario to obtain an average and unbiased result.

We always use 20 nodes in each scenario with an approximate radio range of 40 m. The simulation area is a square flat ground of 500 by 500 meters. The simulations are run for 250 seconds, with no data being sent for the first 40 seconds. This allows the VBR to configure the virtual backbones.

From the observation of the four different wireless networks mentioned earlier in section 2.7 it is clear that the actual numbers of nodes which are mobile are very few and also the movement of nodes is usually between neighborhood areas. Thus only medium mobility pattern is considered in the testing and comparison of the VBR protocol. In the medium mobility scenario 20% of the nodes in the network move to new destinations with speed of 3-5m/s.

Constant bit rate (CBR) is used to generate the traffic. We generate four types of traffic which we generalize as representing different combinations of local and global traffic. A random number of packets to be sent are selected uniformly with each data packet payload being of a fixed size of 512 bytes.

5.2 Tuning VBR

5.2.1 Different Data Rate

All the other parameters are kept constant and the data rates are changed, this test is performed to check if the performance of VBR is affected by different data rates. The source nodes are sending packets (data rates) at 600 packets per min, 120

packets per min, and 66 packets per min. The routing zone radius is set to 2 hops and the default timer values are used.

It can be seen from figure 23 and 24, that when the data rates is kept very high, there are time intervals when the throughput reduces to zero for e.g. in figure 23 at time 210 seconds and in figure 24 at time 150 seconds, at these points the majority of the network bandwidth is used for control traffic. Such a condition arises because there are small time segments in simulation when all the timers schedule their operations at almost the same times thus using the entire bandwidth for control traffic. Such scenarios will also occur when the data rates are low, but since the data rates are low the balance of control and data traffic is maintained, therefore the throughput never become zero.

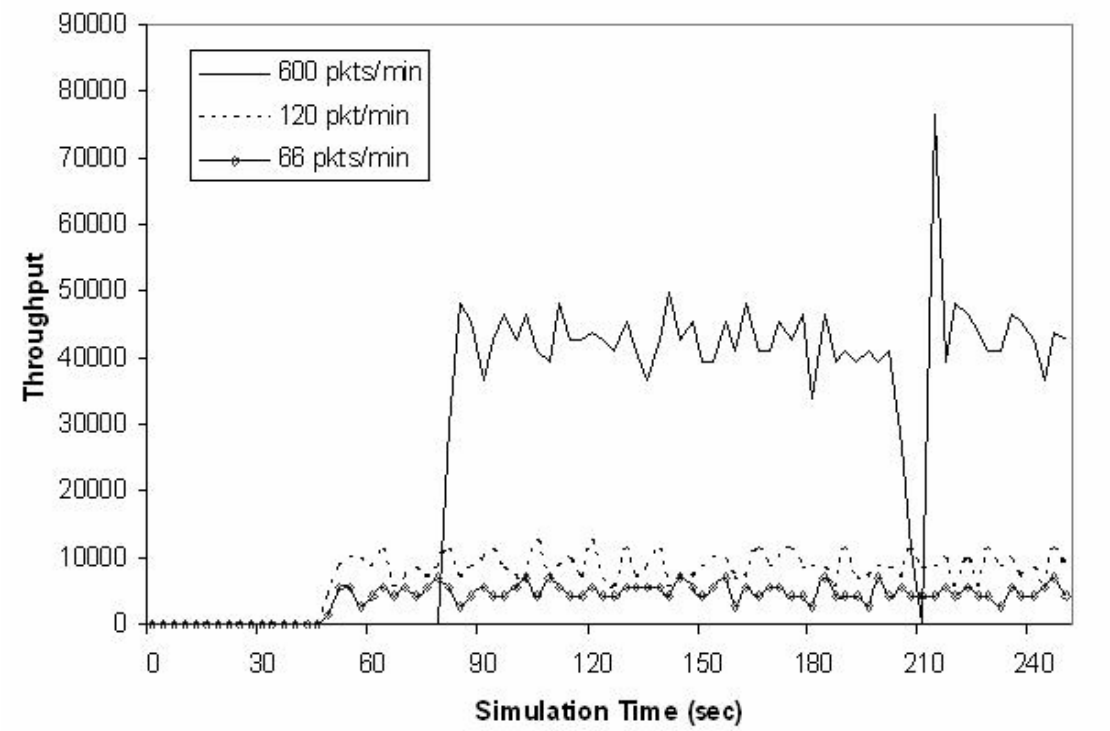

Figure 23: VBR network throughput for no mobility and different data rates in 20-node network

Another point to be observed from figure 23 and 24 is that the latency to find a route for higher data rates in both the figures is different. In the no mobility scenario the time taken to find the route is higher for higher data rates compared to the other data rates for e.g. for a data rate of 600 packets/min the destination starts receiving data at around 80 seconds. This is because when a data packet has to be sent, if the route is not present then the VB routing process is started for each request to send

data. Since in the higher data rates scenario there are more requests to send data, more route request packets are introduced into the network, thus increasing the control traffic and the latency time.

In case of medium mobility, the nodes start moving close to each other therefore the source and destination came into each other routing zone and thus the traffic become local, therefore reducing the route finding latency.

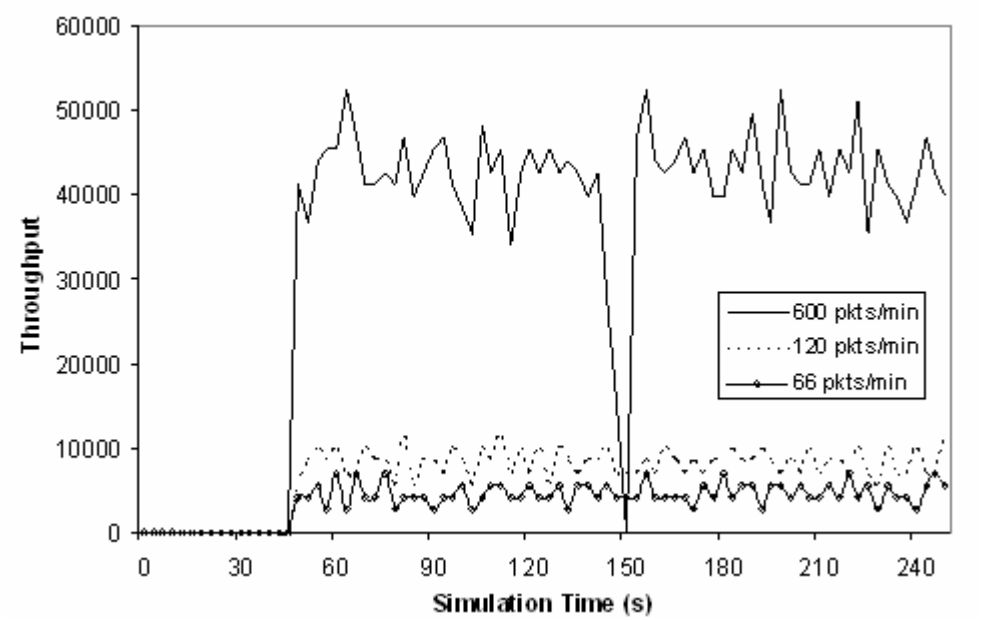

Figure 24: VBR network throughput for medium mobility and different data rates in 20 –node network

5.2.2 Routing Zone Radius

Routing zone radius is an important factor that affects the performance of the hybrid routing protocol. Routing zone is defined as the neighborhood around each node about which a node proactively maintains routing information about all the nodes whose minimum distance, in hops, from the node is not more than zone radius, 'r'. The routing zone radiuses are varied from 1 to 5, the data rate is kept constant at 66 packets per minute and the default timer values are used.

Figure 25 shows the amount of routing overhead as the routing zone radius increases. The node state packets (VB construction and maintenance) and the IARP update packets (proactive component) are the major components of the generated routing overhead.

Figure 26 shows the effect of routing zone radius on the packet delivery fraction. When the routing zone is 1 the route is not found to the destination therefore

the packet delivery fraction (PDF) for r=1 is zero, but as the routing zone radius increases the overhead also increases therefore more of the network bandwidth is used by the control packets. In this thesis the default value for the zone radius is 2, as it can be seen from figure 4 the PDF values are almost 100% for routing zone of radius 2 and 3.

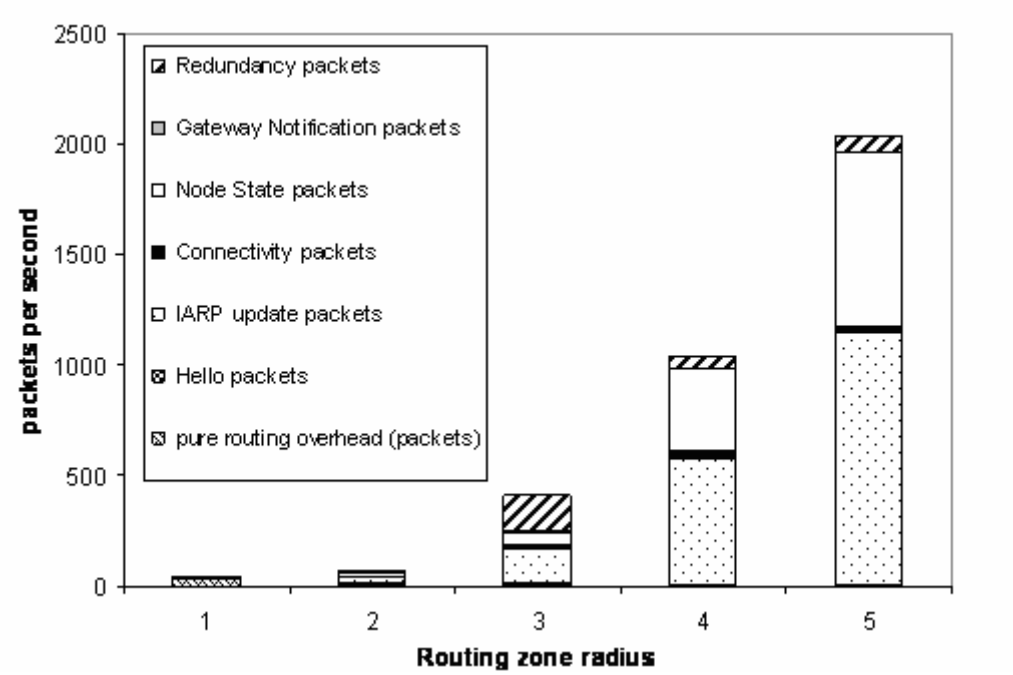

Figure 25: Number of different VBR control packets in a 20-node network versus different routing zone radius for a 250 second simulation

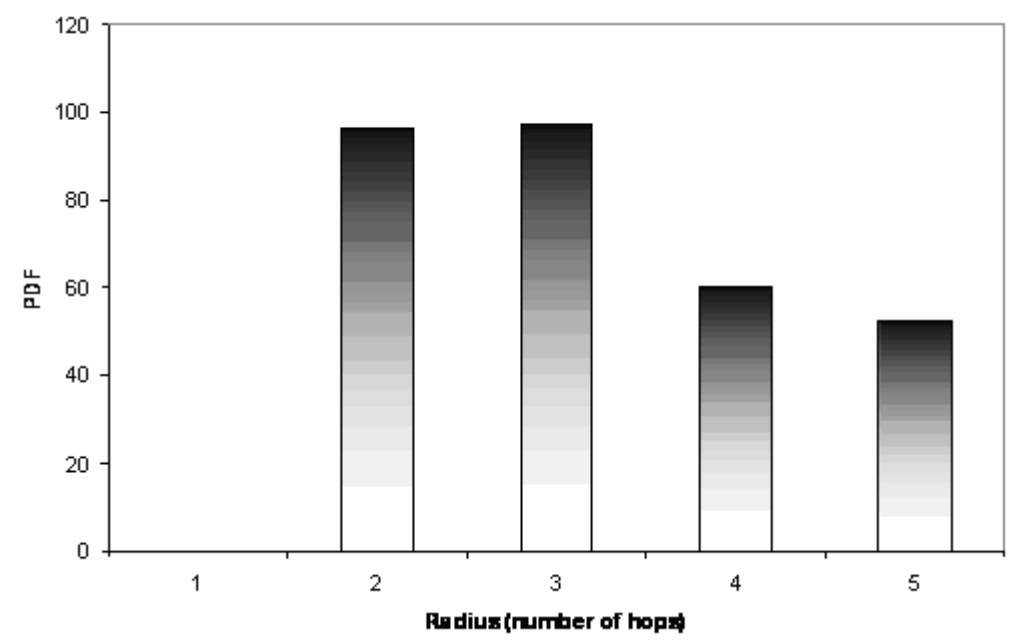

Figure 26: VBR packet delivery fraction in 20-node network for different routing zone radius

5.2.3 Different Timer values

Timer values decide how often a particular function needs to be scheduled. If that function is scheduled to often then it can increase the control traffic or if it's scheduled seldom then changes in network condition will not be detected quickly, so the values selected for the timers must be such that the overhead must not be increase and the network changes must be detected as early as possible. This test is performed to see the effect of the timer values on the throughputs of the network and overheads generated. The timer value is represented as a tuple:

```
<HelloTransmitTimer, IarpTablesUpdateTimer,
ConnectivityPktTransmitTimer, NodeStateTransmitTimer,
RedundancyCheckTimer >
```

The default values are <2, 10, 15, 10, 12>. The values are varied appropriately to show the effect of timer values on the performance of the VBR protocol. The other parameter like the zone radius is set to 2 hops and a data rate of 66 packets per min is used. Figure 27 and 28 show the effects of different timer values on the throughput and the routing overheads. There are no major changes in the throughput of the network for the varied timer values as observed from figure 5.

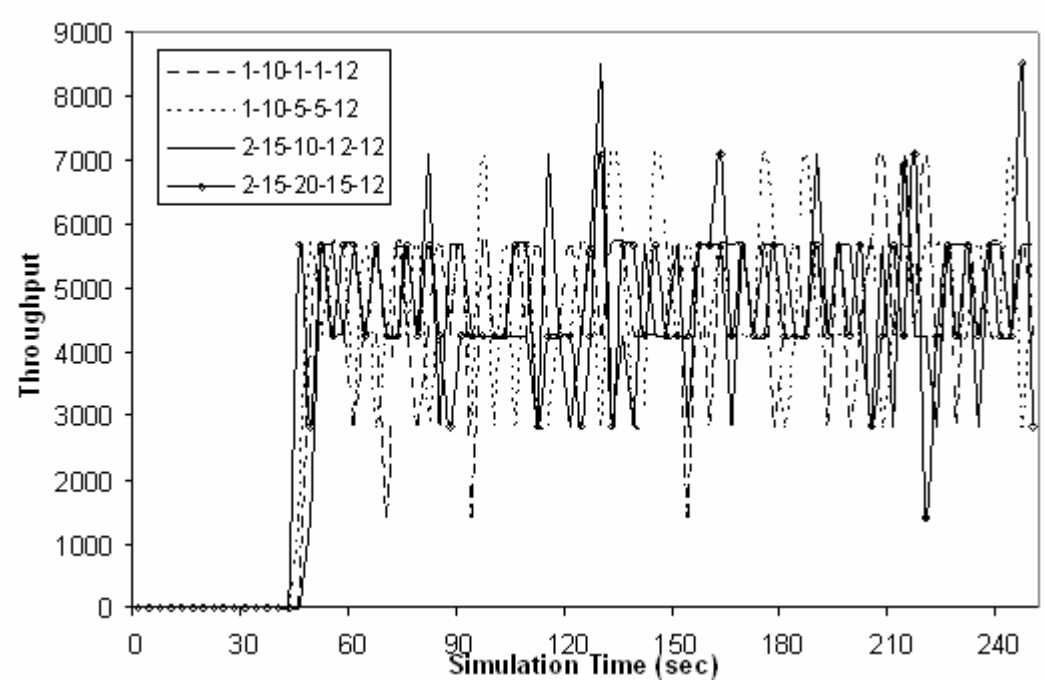

Figure 27: Throughput of VBR for different timer values in a 20-node network at 66 packets per second and a routing zone of radius 2 hops

In the fist case <1, 10, 1, 1, 12> the interval between consecutive "HELLO" packets (HelloTransmitTimer), connectivity packets (ConnectivityPktTransmitTimer)

and node state packets (NodeStateTransmitTimer) was reduced. As can be observed from figure 6 the resultant overhead was very high. In the other cases the values for HelloTransmitTimer, ConnectivityPktTransmitTimer, NodeStateTransmitTimer have been increased, therefore the volume of corresponding control packets have reduced, decreasing the overall routing overheads. The other timers IarpTablesUpdateTimer and RedundancyCheckTimer have not been changed because; changes in their value will not effect the routing overhead as these timers are associated with table scans and not packet transmission. As can be seen from figure 6 the default values gives less overhead compared to the others.

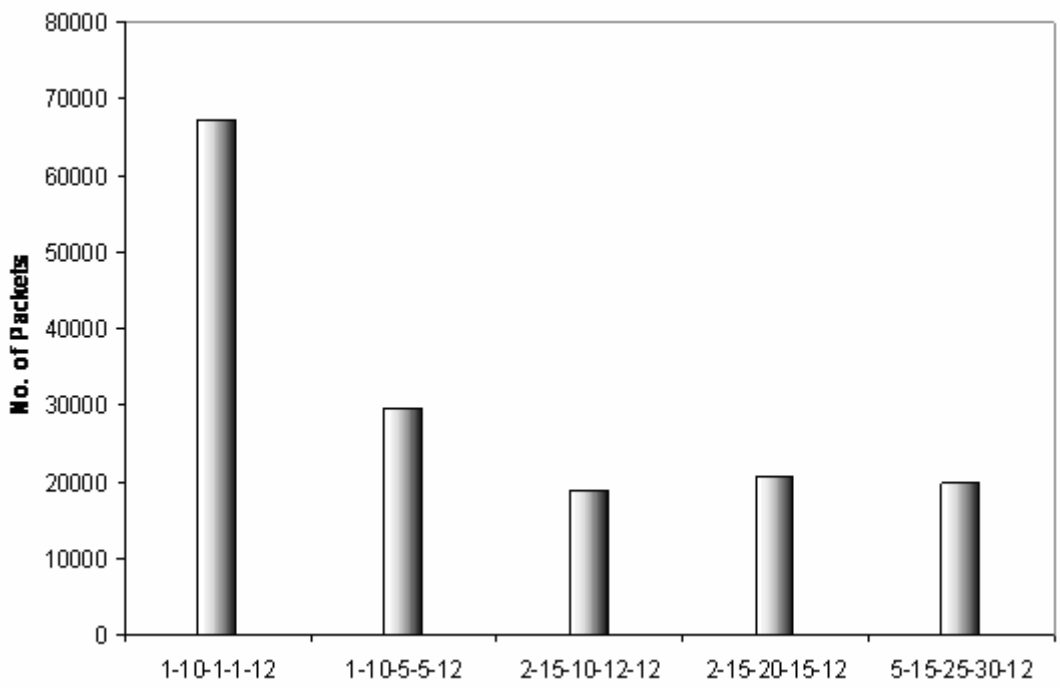

Figure 28: Routing Overheads of VBR for different timer values in a 20-node network for a 250 second simulation

5.3 Virtual Backbone Routing (VBR) versus Dynamic Source Routing (DSR)

This section compares the DSR and VBR protocol on the basis of throughputs, routing overheads, packet delivery fractions and average route discovery latencies. The routing zone is kept constant at 2 hops for all scenarios and the default timer values are used. As mentioned earlier in section 5.1.2 four types of traffic loads are used.

Type1 – 100% of the data traffic in the network is local i.e. the destination node is in the source nodes routing zone.

Type2 – 80% of the data traffic is local and the remaining 20% is global traffic.

Type3 – 20% of the data traffic is local and the remaining 80% is global traffic.

Type4 – 100% of the data traffic is global i.e. the destination is not in the source nodes routing zone therefore the VB routing process has to be used to determine the route.

5.3.1 No Mobility Scenarios

Figure 29 shows that in no mobility scenarios and 100% local traffic the throughput for VBR is almost as good as DSR. This is because in case of VBR the routing information is available, but in case of DSR a route discovery procedure has to be started for each new route. The performance is similar when there is 80% local and 20% global traffic. The performance of VBR reduces at some time intervals due to the proactive and VB maintenance overheads.

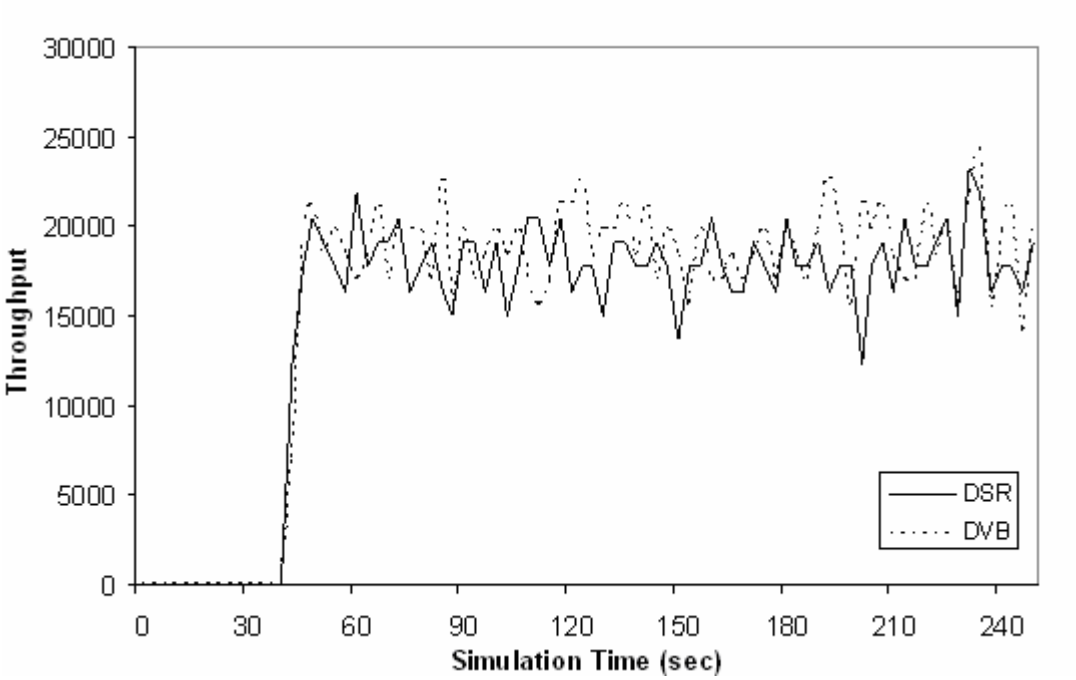

Figure 29: VBR network throughput for no mobility and 100% local traffic for a 20-node network

The effects of non-local route can be seen in figure 30, when there is 20% local and 80% global traffic the throughput of the network for VBR is slightly better

than DSR but at some time intervals the throughput reduces in comparison to DSR. This is because of the additional overhead of finding the global routes through the VB routing process. When the destination is not in the source nodes routing zone then the VB routing process is started, this process introduced more control packets into the network thus reducing the throughput of the network. In figure 29 the dotted line and the solid line are almost superimposed because the routing information is already available, but in figure 30 the two lines can be distinguished, this is the latency to find the route in VBR which is more than DSR. The two lines start at the same point because of the small percent of local traffic still present in the traffic. When the global route is discovered the throughput of the network jumps, this can be seen from the initial spike in the VBR graph in figure 30.

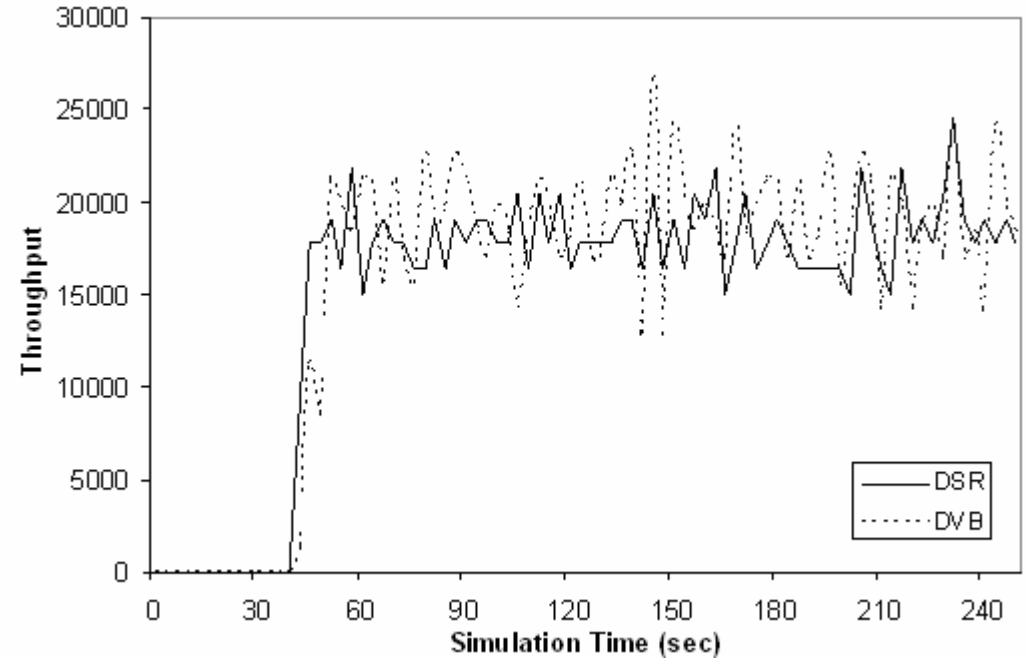

Figure 30: VBR network throughput for no mobility and 80% local 20% global traffic for a node 20-network

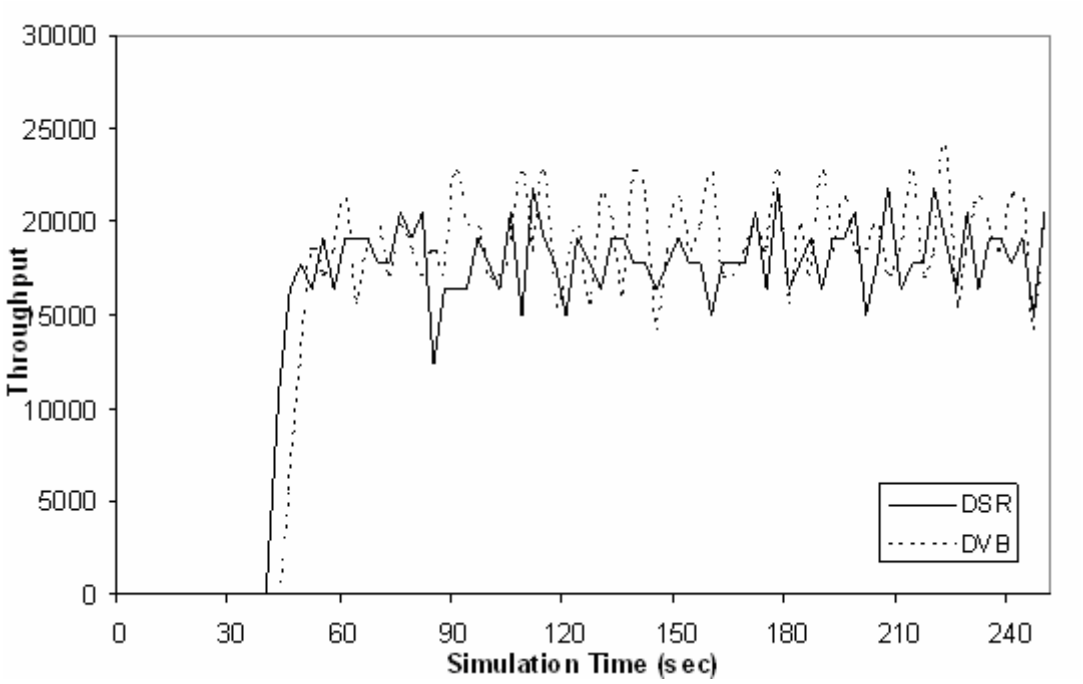

Figure 31: VBR network throughput for no mobility and 100% global traffic in 20-node network

When all the traffic in the network is global, the throughput of VBR is almost similar to DSR as shown in figure 31. The only difference is the latency time increases in case of VBR, which can be seen in figure 31, the dotted and the solid lines start at different times. The latency time for VBR will always be more than DSR because the network has additional control packet (proactive & VB maintenance packets) along with the route discovery packets, but in case of DSR the only control packets are the route discovery packets, but this difference in latency time can be reduced by applying the same optimization done to DSR to VBR.

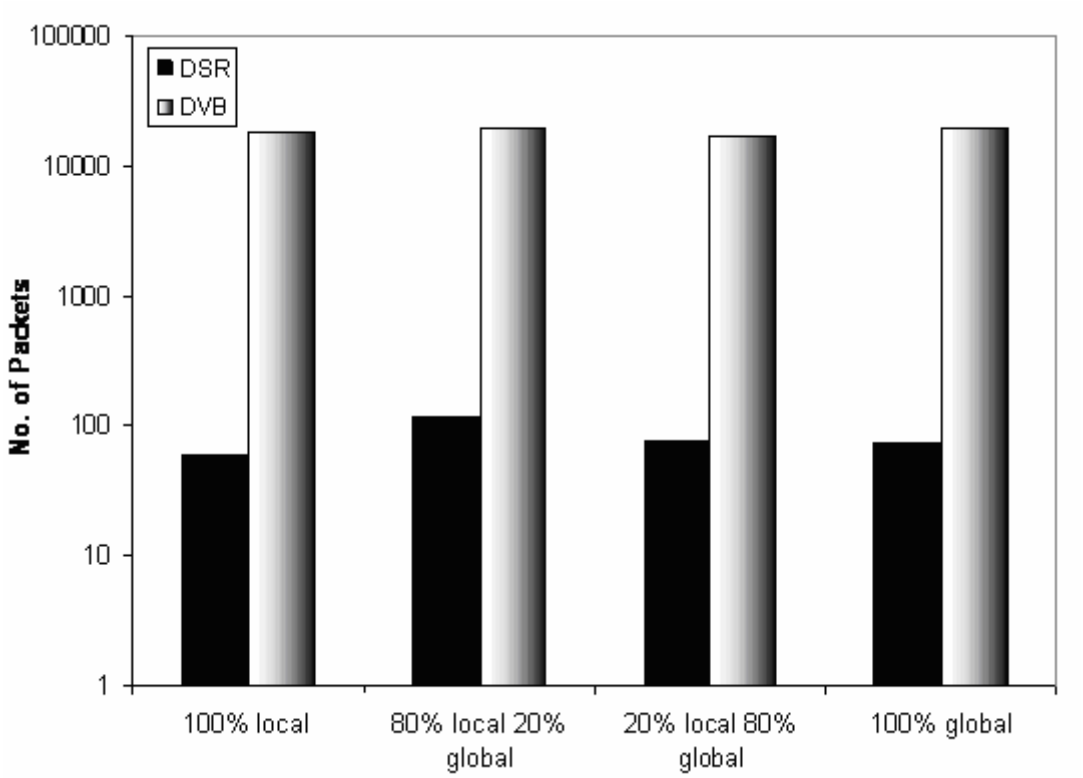

Figure 32: VBR routing overhead for no mobility in a 20-node network

The volume of control packets generated by each protocol can be seen in figure 32. As can be seen from figure 32 the overhead for VBR is very high compared to the DSR (note the y-axis of this figure uses a logarithmic scale); this overhead includes the proactive component overhead, VB construction & maintenance overhead and the routing overhead to find global route. The overhead for VBR is almost constant for all four traffic scenarios, but the overhead of DSR increases for 80% local and 20% global routes.

The number of data packets received at the destination for VBR and DSR protocol is shown in figure 33. In all the traffic scenarios the number of packets sends and received for DSR is almost same. In case of VBR the ratio of send to receive reduces as the amount of traffic become more global, this is because when the route information is not available the data packets are dropped. As the latency time increases to find the route, more data packets are dropped.

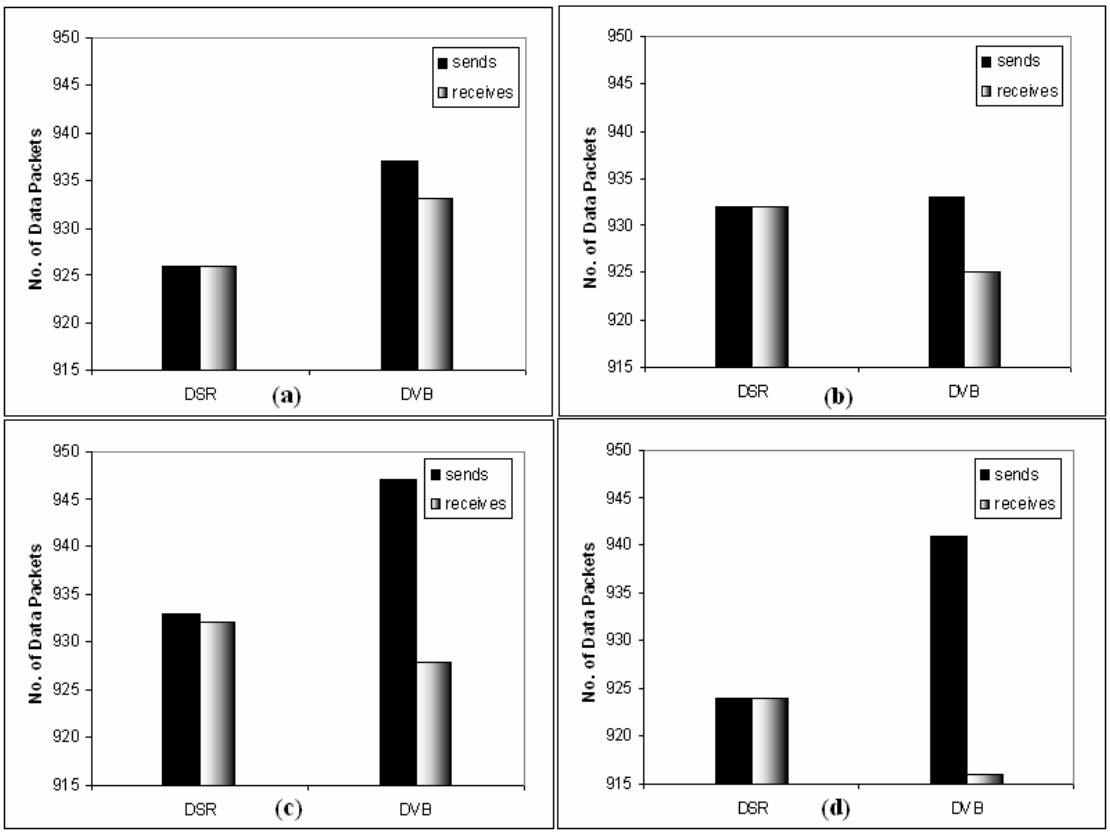

Figure 33: Packets send and received for DSR and DVB in no mobility and (a) 100%local, (b) 80%local 20% global, (c) 20%local 80% global (d) 100% global traffic in 20-node network

Figure 34 shows the latency time to find one route at variable distance away from the source node in DSR and VBR; it can be observed from the figure 34 that the latency time for VBR is more than DSR. When the distance from the source node is 1 hop or 2 hops, there is no latency in case of VBR because the scope of the proactive component is set to 2 hops radius and the route information is available. When the distance from source node increases route request packets have to pass through more nodes, thus facing more control traffic therefore increasing the latency time to find the route. In case of DSR control traffic is only generated when there is a route request, and figure 34 shows the latency time to find one route, therefore there is no other traffic which will affect the DSR route discovery packets, thus reducing the latency time.

Another important reason for this difference is the optimizations which have been applied to DSR code implemented in NS2. It can be observed from the figure 34

that DSR takes less time for shorter path (i.e. <= 2 hops) and as the number of hops increases (> 2 hops) the time to find the route remains almost constant.

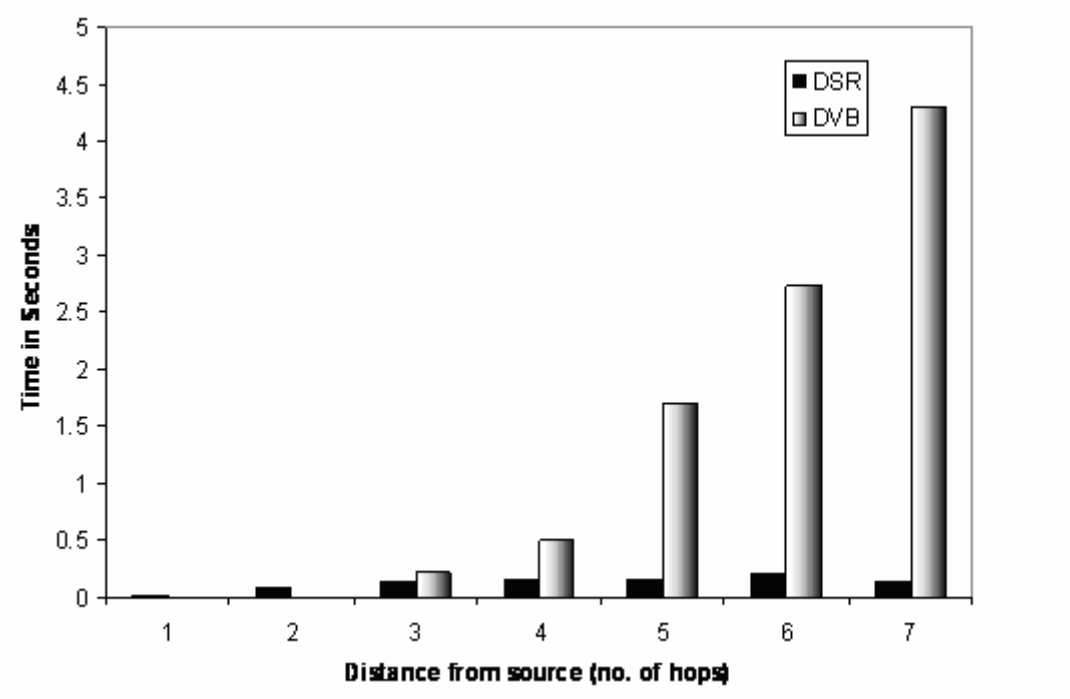

Figure 34: Latency time for finding one route 1-7 hops away from the source node in VBR and DSR

These are the optimizations which have been implemented in DSR for NS2 [19]

- Route Caching – each node caches a new route it learns by any means. “Any means” suggests that the route information can be obtained through route request, reply packets or data packets passing through that node. “use-Tap” is another optimization which allows a node to enter promiscuous mode, to overhear packets not intended for its MAC address and thus update its routes from the overheard packets.
- Data Packet Caching – If a route is not present for a data packet, the data packet is buffered, till the route is obtained. Once the route is obtained, the path is appended to all the buffered packets and transmitted. This explains the high delivery fraction of DSR compared with VBR
- Route Reply storm - When intermediate nodes reply Route Request by using local route cache, the route reply is transmitted after some delay, during this

delay period, the route reply is cancelled if overhearing a packet contains a route from the same initiator to the same target with a shorter length path.

- Salvage-with-cache – In case of a transmit failure, before sending the route error to the source node, the path to destination node is checked in the current nodes cache, if the path exists then this new path is used and the packet is forwarded. The node marks this packet as a salvaged packet.
- Ring-zero-search – This is also called as expanding ring search. The basic ides is to limit the propagation of route request packets. First, the TTL field is set 1 for the first route request packet and if no route reply is received after some time period, the TTL field is set to maximum for next route request packet.

The effect of these optimizations is evident from the latencies and the overheads shown in figure 32 & 34 respectively. No optimization has been applied to the VBR implementation in NS2, thus we are comparing an optimized code (DSR) with non-optimized code (VBR). The advantages of this optimization will become clearer in the medium mobility scenarios.

5.3.2 Medium Mobility Scenarios

Figure 35 shows the affect of medium mobility on the network with 100% local traffic. In this scenario some set of node move to a new destination, pause there fore some time and move back to the original poisons. Even though the initial traffic is local, since the nodes are moving it's possible that the local path will break and form global paths, this phenomenon can be seen in figure 35 (time interval 40 -70). Since all the traffic is local initially there is no latency in finding the route. Since the local routes are broken new route have to be found, till new route are found the data packets are dropped, thus reducing the throughput of the network (indicated by the fall in throughput between intervals 40 - 70). Once the routes have been established again the throughput increases. The node movements can also result in breaking of global route to form new local routes if the nodes move into each other's neighborhood. When the nodes that have moved, return to their original positions the local routes are reestablished. Therefore the VBR throughput increases and comes closer to DSR throughput in figure 35.

When the nodes move, the virtual backbone of the network might get disturbed (if a VB node is moving then the virtual backbone has to be rebuilt to cover nodes which

are set to panic state due to its movement), therefore in addition to finding new routes the virtual backbone also has to be constructed.

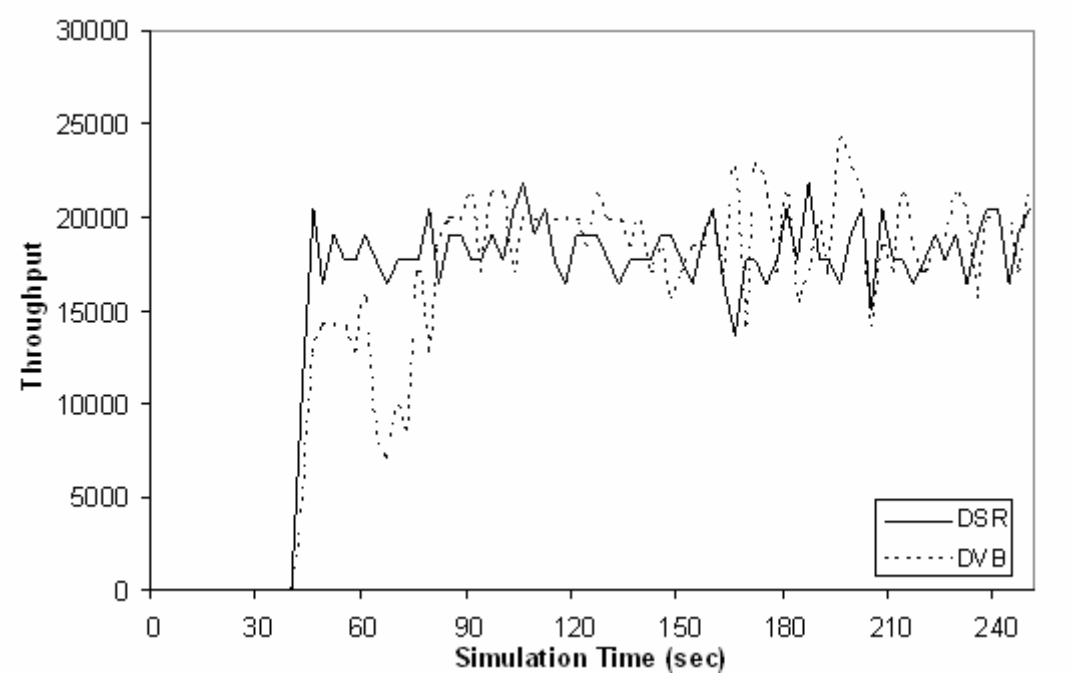

Figure 35: VBR network throughput for medium mobility and 100% local traffic in 20-node network

The effects of movement on VB construction can be seen from figure 36 and 37. In figure 37 the majority of the traffic is global therefore the latency to find the route can be seen. The throughput of VBR increases between time intervals 60 – 70 but reduces thereafter; this because the nodes move close to each other breaking the global and forming new local path. During this interval the throughput of VBR is lower than DSR because since nodes come close to each other, the proactive traffic increases and also new VB nodes have to be assigned to cover the panic nodes induced by the movement of nodes.

After 70 seconds the node starts moving back to their original positions, thus breaking the local route and disturbing the virtual backbone. Once the nodes reach their original positions, first the virtual backbone has to be setup only then the route discovery process can start. During all this time the packets are dropped, thus reducing the throughput of the network. Once the new virtual backbone has been setup, the redundancy check is performed on the entire network, thus eliminating

redundant VBs, this again causes node to panic and go through the VB construction & maintenance phase.

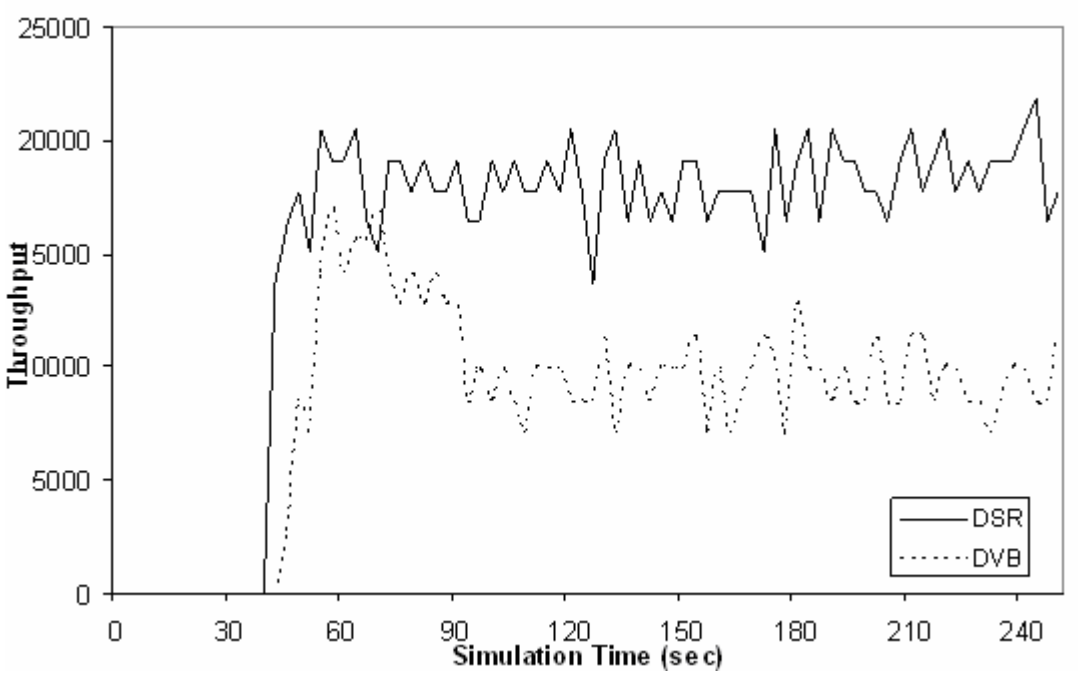

Figure 36: VBR network throughput for medium mobility and 20% local 80% global traffic in 20-node network

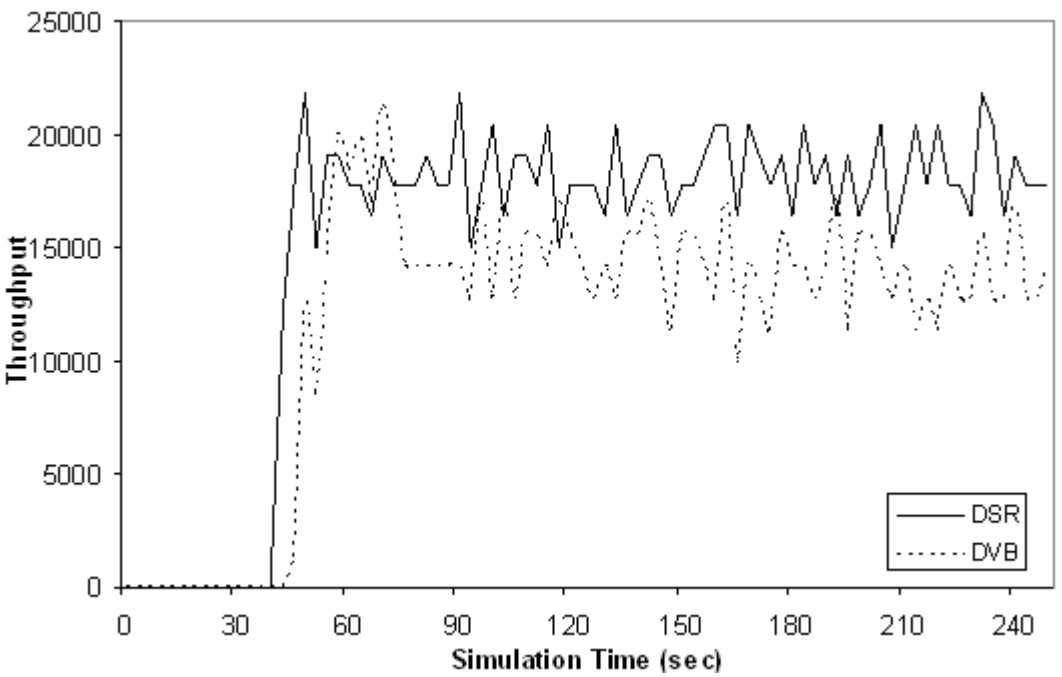

Figure 37: VBR network throughput for medium mobility and 100% global traffic in 20-node network

When ever the network is disturbed or a network is new (i.e. if the virtual backbone is disturbed due to movement of nodes or there is no virtual backbone), some time has to be spent on setting up or repairing the virtual backbone, this is the reason why no data is send in the first 40 seconds of the simulation for a new network and the throughput is low after node movements.

In each run of the VBR protocol the virtual backbone is setup automatically, and the nodes which form the virtual backbone are different for each run of the algorithm. In figure 36 most of the nodes which where selected for movement were the virtual backbone nodes; there the throughput for VBR was severely affected. In case of figure 37 out of all the nodes chosen for movement only few of them were VB nodes thus the effect of movement on the network was not very bad.

In all the cases the throughput of DSR was not adversely affected by the movement of nodes, because the overhead in DSR is much less compared to VBR, the only overhead is the route discovery process when new route have to be found.

The overheads for VBR protocol is more in medium mobility than in no mobility, in case of DSR the overhead is almost similar, but in both the cases the overhead for VBR is always more than DSR. The overhead of VBR for all traffic scenarios is almost same; one would think that the overhead should increase for the 20%local and 80%global traffic scenario because the throughput for VBR in this scenario is very less. This increase in overhead is not seen here because figure 38 shows the total overhead, and in the above case one component (VB construction & maintenance) overhead increases but the route discovery overhead reduces because the nodes are in panic as the virtual backbone is not setup, thus the overall overhead remains almost the same.

Number of data packets received and send in each traffic scenario is shown in figure 39. When the percentage of local traffic is high 100% - 80% the statistics are almost similar for DSR and VBR, but when the global routes starts increasing the VBR values are affected, the number of packet received in the 20%local 80% global traffic is less than the 100% global traffic, this is because of the overheads explained earlier in this section.

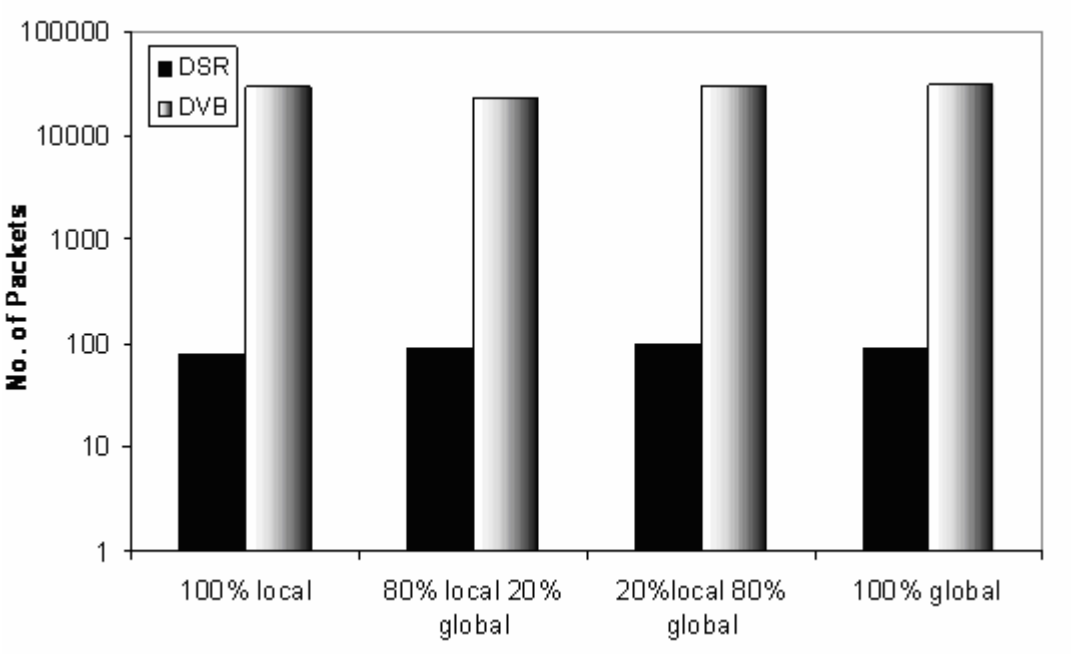

Figure 38: VBR routing overhead for medium mobility (the y axis represents log10 values) in 20-node network

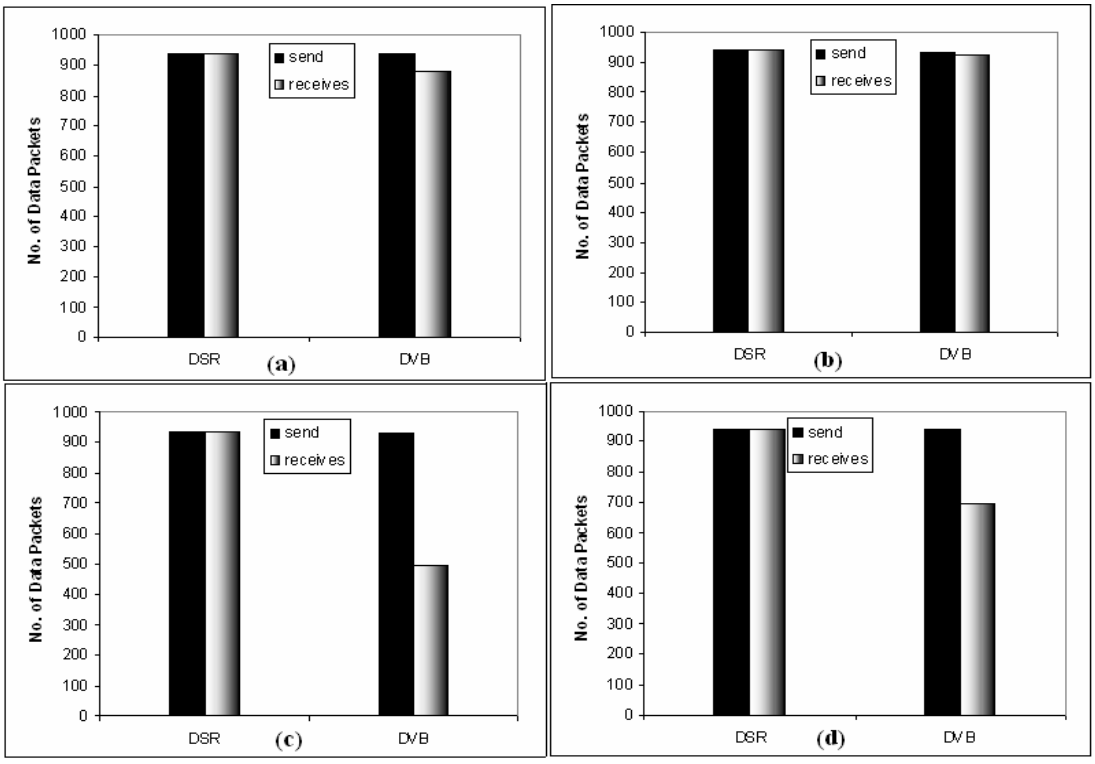

Figure 39: packets send and received for medium mobility and (a) 100%local, (b) 80%local 20% global, (c) 20%local 80% global (d) 100% global traffic

Chapter 6

Conclusions

The goal of this thesis was to judge the performance of a hybrid routing protocol with a currently implemented protocol. The dynamic virtual backbone routing (VBR) was selected for implementation and compared with dynamic source routing (DSR) protocol.

The results obtained from the comparison showed DSR as a superior routing protocol compared to VBR for mobility scenarios. The main reason for this is the range of optimizations which has been implemented in DSR. In the current state this thesis has tried to compare something optimized (DSR) with something that is not optimized (VBR). As this issue was identified late in the project, there was no scope to optimize VBR to the level done to DSR.

The efficiency of VBR is closely tied with the efficiency of the underlying proactive and reactive protocols. In this implementation of VBR a simple link state routing protocol and a flooding (VB assisted flooding) protocol has been used for the proactive and reactive components of the VBR implementation, the performance of VBR be improved further if more optimized, robust and efficient proactive and reactive protocols are implemented.

Another reason for the poor performance of the VBR protocol was due to the limitation of the implementation of the VBR protocol. The current implementation of the VBR can be used to simulate only small network scenarios, but according to [7] the performance gains for VBR can be observed only when the network size of 300-600 nodes is used.

An open line from this work is to optimize the VBR implementation, in order to compare the DSR and VBR protocols again under more fair circumstances.

REFERENCES

1. Nicklas Beijar (May 2002), *Zone Routing Protocol (ZRP)*, Helsinki University of Technology.
2. Prince Samar, Marc R. Pearlman and Zygmunt J. Haas (August 2004), *Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks*, IEEE/ACM Transactions on Networking, Vol. 12, No. 4.
3. Zhenhua Fan (November 2005), *Wireless Ad Hoc Network Protocol Analysis*, the University of Adelaide, Thesis Report.
4. Yu-Liang Chang and Ching-Chi Hsu (2000), *Routing in wireless/mobile ad-hoc networks via dynamic group construction*, National Taiwan University, Mobile Networks and Application 5, 27- 37.
5. Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella (September 2004), *AntHocNet: an Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks*, Istituto Dalle Molle sull'Intelligenza Artificiale (IDSIA), Technical Report No. IDSIA-27-04-2004.
6. Elizabeth, Royer, Chai-Keong, Toh (April 1999): *A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks*, IEEE Personal Communications.
7. Liang, B. and Haas, Z.J., *Hybrid Routing in Ad Hoc Networks with a Dynamic Virtual Backbone*, to appear in IEEE Transactions on Wireless Communications.
8. M. Joa-Ng and I.-T. Lu, *A peer-to-peer zone-based two-level link state routing for mobile Ad-hoc networks*, IEEE J. Select. Areas Commun., vol. 17, no. 8, pp. 1415-1425, 1999, Special Issue on ad-hoc networks.
9. Venugopalan Ramasubramanian, Zygmunt J. Haas, and Emin Gun Sirer. *SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks***.** In Proceedings of *MobiHoc 2003*, Anapolis, MD, 2003.
10. S. Roy and J.J. Garcia-Luna-Aceves, *Node-Centric Hybrid Routing for Ad Hoc Networks*, *Proc. MASCOTS Conf. 2002*, Oct. 2002.
11. D. Tang and M. Baker, "*Analysis of a Metropolitan-Area Wireless Network*," Proc. Int'l Conf. Mobile Computing and Networking (Mobicom), pp. 1-10, 1999.

12. K. Papagiannaki, M. Yarvis, and W. S. Conner, "*Experimental characterization of home wireless networks and design Implications*," in INFOCOM, 2006 (to be printed).
13. D. Tang and M. Baker, "Analysis of a Local-Area Wireless Network," *Proc. 6th Int'l Conf. Mobile Computing and Networking* (Mobicom 00), ACM Press, 2000, pp. 1–10.
14. Kotz D., Essein K., 2005, "*Analysis of a campus wide wireless network*", *Wireless Networks*, 11:115
15. A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. *Characterizing User Behavior and Network Performance in a Public Wireless LAN*. In Proc. ACM SIGEMTRICS'02, June 2002 (To Appear).
16. Zygmunt Haas and Marc Pearlman, "*Providing Ad-hoc connectivity with reconfigurable wireless networks*", In Charles Perkins, Editor, Ad-hoc Networks Addison-Wesley Longman, 2000
17. Ns manual - http://www.isi.edu/nsnam/ns/ns-documentation.html
18. F. J. Ros and P. M. Ruiz, "*Implementing a New Manet Unicast Routing Protocol in NS2*",December 2004
19. Mahadevan, P., Rodriguez, A., Becker, D., and Vahdat, A. 2006. "*MobiNet: A scalable emulation infrastructure for ad hoc and wireless networks.*" SIGMOBILE Mob. Comput. Commun. Rev. 10, 2 (Apr. 2006), 26-37. DOI= http://doi.acm.org/10.1145/1137975.1137979