M.Tech. Dissertation

Route Repair in Mobile Adhoc Networks

Submitted in partial fulfillment of requirements
for the degree of
Master of Technology

By

Abhilash P

Roll No. : 00329003

Under the guidance of

Dr. Sridhar Iyer

Kanwal Rekhi School of Information Technology
Indian Institute of Technology, Bombay
Mumbai, 400 076
January 28, 2002

Dissertation Approval Sheet

This is to certify that the dissertation titled

Route Repair in Mobile Adhoc Networks

By

Abhilash P

(00329003)

is approved for the degree of **Master of Technology**.

Dr. Sridhar Iyer

(Guide)

Internal Examiner

External Examiner

Chairman

Date : _____

Acknowledgment

January 28, 2002

I would like to thank my guide, **Dr. Sridhar Iyer**, for his constant advice, encouragement and support. I would also like to thank my senior **Srinath Perur**, my colleagues **Praveen Kumar S.** and **Ajay K. Singh** for their feedback and support.

Abhilash P

Abstract

Mobile Adhoc Networks (MANET) are distributed, mobile, wireless, multihop networks that operate without pre-existing communication infrastructure, except for the mobile devices themselves. Several routing protocols both reactive and pro-active have been proposed to provide the self starting behavior needed for adhoc networks.

The nature of the network coupled with the mobility of the devices, result in a large number of route breakages. The current approach in case of broken routes is to flag an error and re-initiate route discovery either at the source or at the intermediate node. Repairing these broken links is a costly affair in terms of routing overhead and delay involved.

In this report, we propose a proactive approach called Routing Handoff, to repair broken routes, using the mobile devices in the vicinity of the broken link. The idea is incorporated into the AODV routing protocol. The results of the simulation indicate an increase in throughput under certain conditions. The improvement is a result of smaller overhead and delay. The approach may also be applied to other routing protocols with appropriate modification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the proliferation of mobile devices such as cell phone, laptop, and palm-top the demand for continuous network connectivity regardless of the physical location has spurred interest in mobile networks. Mobile networks can be broadly classified as *Cellular* or *Adhoc*. A cellular network provides mobility support to a region by dividing it into smaller well defined regions called cells. Each cell has a fixed base station that is in charge of providing network services to mobile devices within its range. The base station is also responsible for handoff of mobile devices to other base stations when mobile devices change cells. The cellular network is a single hop network because the one hop (i.e. base station to mobile device and vice versa) is wireless, the rest being part of the wired network. In India mobile phone/cell phone connectivity is achieved using cellular networks.

1.1 Mobile Adhoc Network

Mobile Adhoc Networks (MANET) [2] are a new paradigm for mobile devices. It is a cooperative engagement of a collection of mobile devices, herein referred to as *nodes*, without the required intervention of any centralized access point or existing infrastructure. Each node is equipped with a wireless transmitter and a receiver. In order to facilitate communication within the network, each node acts as a router and a routing protocol is used to discover routes between nodes. If only two nodes, located close to each other, are involved in the adhoc network, no real routing protocol or routing decisions are necessary. In many adhoc networks, two nodes that want to communicate may not be in the wireless transmission range of each other, but could communicate if nodes physically located between them are willing to forward packets.

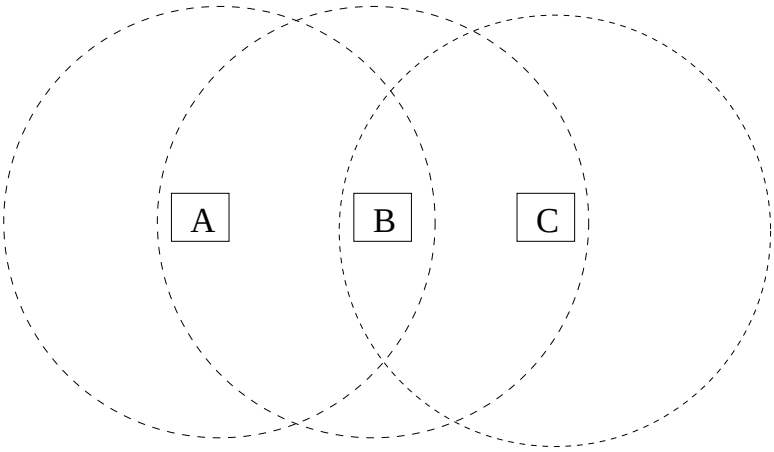

Figure 1.1: A simple Adhoc network of three wireless mobile nodes

For example in the network illustrated in Figure 1.1 mobile node C is not within the range of node A's wireless transmitter (indicated by a dashed circle around A) and node A is not within the range of node C's wireless transmitter. If A and C wish to exchange packets, they will have to enlist the service of node B to forward packets for them, since B is within the overlap range of node A and node C. The implicit assumption here is that node B will be on during the entire time of operation and is willing to forward packets from node A to node C. Highly dynamic nature of the network coupled with the fact that power and bandwidth in the nodes are constrained and should be utilized optimally, makes routing in mobile adhoc networks an important issue.

To make the concept of an adhoc network more concrete, such a network might be used in the following situations:

- Military Operations: Need to communicate in enemy territory bereft of communication infrastructure.

- Disaster Relief: Communication in disaster affected areas where traditional communication infrastructure has broken down.

- Conference: People attending conferences wanting to network their laptops together to exchange data, drafts etc. The size of the network and type of nodes brought may not be known beforehand.

1.2 Routing

Traditional IP routing does not mean much in adhoc network, unless connected to the Internet itself. Even so, since the network is a multihop network, the node will need to discover how to send packets to another node.

Several routing protocols both reactive [7, 3] and pro-active [6] have been proposed to provide the self starting behavior needed for adhoc networks. *Destination Sequenced Distance Vector Routing* (DSDV) [6] is a pro-active protocol which is an enhancement over the Distance Vector Routing. *Optimized Link State Routing* (OLSR) [8] is another pro-active protocol which is an enhancement over the Link State Routing Protocol. *Dynamic Source Routing* (DSR) [3] and *Adhoc On Demand Distance-Vector* (AODV) [7] routing protocol are two reactive protocols known for their simplicity and performance. In addition there are hybrid routing protocols which takes advantage of the best of reactive and pro-active routing protocols. *Kelpi* [9] is a hybrid routing protocol.

Pro-active routing protocols continuously update the nodes in the network when the topology changes, resulting in routes with no route discovery latency. Reactive routing protocols on the other hand discover routes as and when required. Though reactive routing protocols result in route discovery latency, it significantly reduces the routing overhead associated with pro-active routing protocols.

1.3 Problem definition

The performance of the reactive protocols and its enhancements have been affected by *high routing overheads* and *delays* in repairing broken routes. The current approach is to flag an error and re-initiate a route discovery either at the source or at the intermediate node. *Location Aided Routing* (LAR) [4] makes use of location information to reduce routing overheads. *Virtual Wire Messages* [11] and *Spine Routing* [10] make use of a Virtual Dynamic Backbone to reduce routing overheads. Several other approaches like caching of learned routes have also helped reduce routing overheads. But in all these approaches re-initiation of route discovery seems inevitable. The scope of the work is to investigate techniques that can locally repair the routes.

1.4 Solution Overview

We present an approach called *Routing Handoff* to repair broken routes, using mobile nodes in the vicinity of the broken link. We have incorporated this idea into the AODV routing protocol. We also present a theoretical analysis of this approach with respect to AODV, AODV with Local Route Repair (LRR). The analysis show remarkable improvement both in terms of reduction in routing overhead and delay. The simulations validate our claims with an increase in throughput as a result of smaller overhead and delay. This is possible because of the reduction in routing overhead and delay in repairing broken routes. This approach gives us better performance than the approach of rediscovering the routes. The approach may also be applied to other routing protocols with appropriate modification.

1.5 Organization of the Report

The report is structured as follows: In the second chapter we look at routing protocols in MANET and problems associated with them. It also highlights the need for an approach with smaller delay and overhead in repairing broken routes. Chapter three describes Local Route Repair and introduces the concept of Routing Handoff. Chapter four presents the theoretical analysis of AODV, Local Route Repair and Routing Handoff. Chapter five presents the results of the simulation. Chapter six provides the conclusion and future work.

Chapter 2

Routing in MANET

The characteristics of mobile adhoc network, like highly dynamic and bandwidth constrained network and energy constrained nodes, makes routing a challenging issue. The traditional routing protocols like Distance Vector routing protocol and Link State routing protocol, have not been designed specifically to provide the kind of dynamic, self-starting behavior needed for adhoc networks. Most protocols exhibit their least desirable behavior when presented with a highly dynamic interconnection topology. These protocols also place too heavy a computational burden on each node. Moreover, the delay in propagation of the common view of the network of these protocols results in poor convergence characteristics.

2.1 Extending Wired Routing Protocol

In Link State routing protocol, each node maintains a view of the network topology with a cost for each link. To keep the views consistent, each node periodically broadcasts the link cost of its outgoing links to all other nodes in the network. Periodic updates even when the topology does not change causes transmission overhead. Some of the link cost in the node's view can be incorrect because of the highly dynamic nature of the network. Such inconsistent views of network topologies might lead to formation of routing loops. These loops are short lived, because they disappear in the time it takes a message to traverse the diameter of the network.

Distance vector routing algorithm is computationally more efficient, but can cause formation of both short-lived and long-lived loops. The primary cause for formation of routing loops is that nodes choose their next-hops in a completely distributed fashion based on information which can possibly be

stale and, therefore, incorrect. It also suffers from *counting-to-infinity* problem. Furthermore, the techniques of *split-horizon* and *poisoned-reverse* are not useful within the wireless environment due to the broadcast nature of the transmission medium.

The problems mentioned above associated with link state routing protocol and distance vector routing protocol discourages its use in adhoc networks. But enhancements of these routing protocols have been quite a success.

2.2 Routing Protocols in MANET

The issue of routing in MANET deals with finding paths between nodes in a constantly changing network topology, ensuring at the same time that minimal bandwidth and power are consumed for routing, and maximizing the degree of reliability such a network can offer. The primary attributes of such a routing protocol are loop free routes, quick convergence, minimum storage overhead, small computational and transmission overhead.

Adhoc routing protocols can be broadly classified as pro-active or reactive. Some routing protocols that have both pro-active and reactive components are categorized as hybrid routing protocols. Despite being designed for the same type of underlying network the characteristics of each class of protocols are quite distinct. The following section reviews pro-active, reactive and hybrid routing protocols.

2.3 Pro-active routing protocols

In pro-active routing protocols routes are maintained to all potential destinations (to all nodes in the network) all the time, whether or not all such routes are actually used. Pro-active routing protocols in adhoc networks are extensions of distance vector or link state routing protocols to adapt to the high dynamic nature of the such networks. In the next section we look at Destination Sequenced Distance Vector Routing protocol.

2.3.1 Destination Sequenced Distance-Vector Routing (DSDV)

The *Destination Sequenced Distance Vector* [6] approach is a modification of the routing algorithm used earlier in ARPANET. In DSDV, each node maintains a distance vector that contains entries for each destination. The entry indicates

the distance estimate and the next hop to be taken by a packet to reach a destination.

Each entry has a sequence number associated with it, indicating its freshness. If a destination is unreachable distance metric is set to infinity. Periodically a node's distance estimates are diffused to neighbors. Any change in routing information is also propagated across the network. Complete information is propagated when the network traffic is low. The rest of the time an incremental update is propagated.

DSDV provides with loop free routes. But on the other hand, increases network load due to periodic routing advertisement messages. Several parameters need to be considered for optimal performance using DSDV and optimal values of the parameters vary across networks.

2.4 Reactive routing protocols

Reactive routing protocols create and maintain routes as and when required. They are also know as on demand routing protocols. Thus when a route is needed some sort of a global search procedure is employed. The family of flooding algorithms belong to this group. Two examples of reactive routing protocols are Dynamic Source Routing and Adhoc On Demand Distance Vector Routing protocol.

2.4.1 Dynamic Source Routing (DSR)

In *Dynamic Source Routing* [3] routes are discovered on demand, when a node has packets to send to some destination node. Each mobile host participating in the adhoc network maintains a *route cache* in which it caches source routes that it has learned.The protocol consists of two major phases: route discovery and route maintenance.

In route discovery phase the source node broadcasts a *route request* packet. The route request packet in addition to the address of the original initiator of the request and the target of the request contains a *route record*, which records the sequence of hops taken by the route request packet as it is propagated through the adhoc network during route discovery. Each intermediate node receiving this packet broadcasts it till some node that has a route to the destination receives it. This node sends back a *route reply* packet with the route record appended with the path to the destination from it. Route maintenance monitors the correct operation of a route in use. When route maintenance detects a

problem with a route in use, it re-initiates a route discovery phase.

DSR uses no periodic routing advertisement messages, thereby reducing network bandwidth and conserving battery power. The disadvantages are that the packet size tends to be large because they record the entire route. This causes DSR to assume the network diameter to be small and prevents scaling to large networks

2.4.2 Adhoc On Demand Distance Vector Routing (AODV)

The *Adhoc On Demand Distance Vector*[7] (AODV) routing protocol builds on the *Destination Sequenced Distance Vector*[6] (DSDV) routing protocol. AODV creates routes on a on-demand basis. When a source node **S** desires to send a message to some destination node **D** and does not already have a valid route, it initiates a route discovery process. It broadcasts a route request **RREQ** packet to its neighbors, which in turn forwards the request to their neighbors, and so on, until either the destination or an intermediate node with a *fresh enough* route to the destination is located.

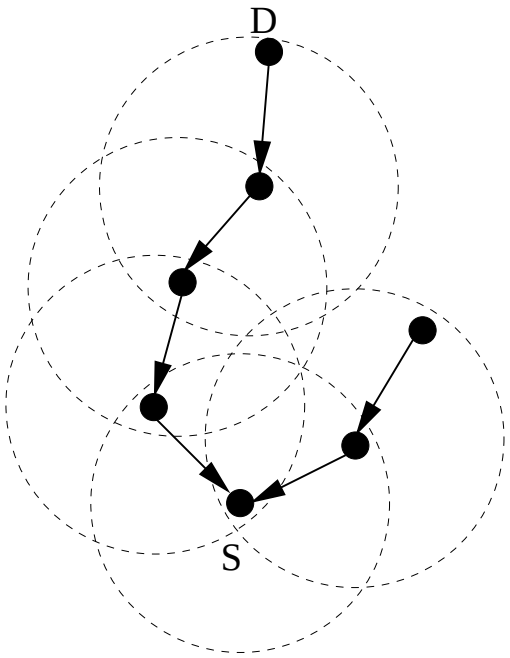

Figure 2.1: Reverse path formation

During the process of the forwarding the route request, intermediate nodes record the address of the neighbor from which first copy of the broadcast packet is received, thereby establishing a *reverse path* as illustrated in figure 2.1 [7].

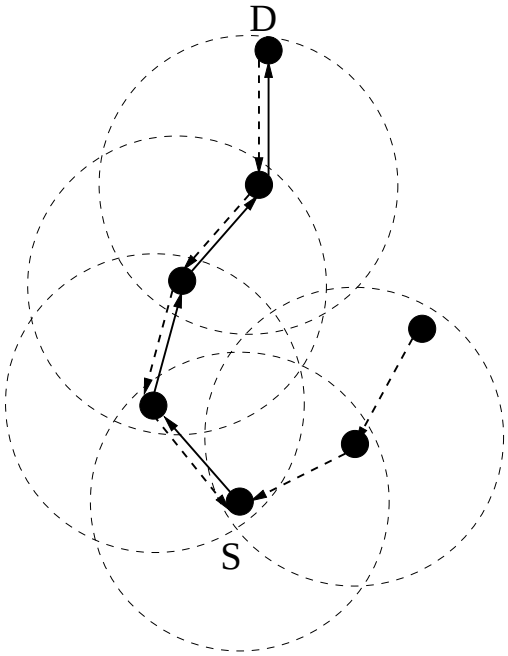

Figure 2.2: Forward path formation

The destination/intermediate node responds by unicasting a route reply **RREP** packet back to the neighbor from which it first received the route request. As route reply is routed along the reverse path, nodes along this path set up forward route entries in their route tables, thereby establishing a *forward path* as illustrated in figure 2.2 [7].

Route Maintenance monitors the correct operation of a route in use. Movement of nodes within the adhoc network affects only the routes containing those nodes; such a path is called *active path.* When either the destination or some intermediate node moves, a Route Error **RERR** packet is sent to the affected source nodes. When a source node receives the RERR, it can re-initiate route discovery if the route is still needed.

2.5 Hybrid routing protocols

Hybrid routing protocols are a mixture of pro-active and reactive concepts. An example of such a routing protocol is the Kelpi routing protocol

2.5.1 Kelpi Routing Protocol

Kelpi [9] is a location based, hierarchical, hybrid routing protocol designed to provide stable, long lived routes. Kelpi imposes a cellular structure on the MANET. It uses a concept of *router handoff*, resulting in retention of routing information in the vicinity of its use, to provide long lived routes.

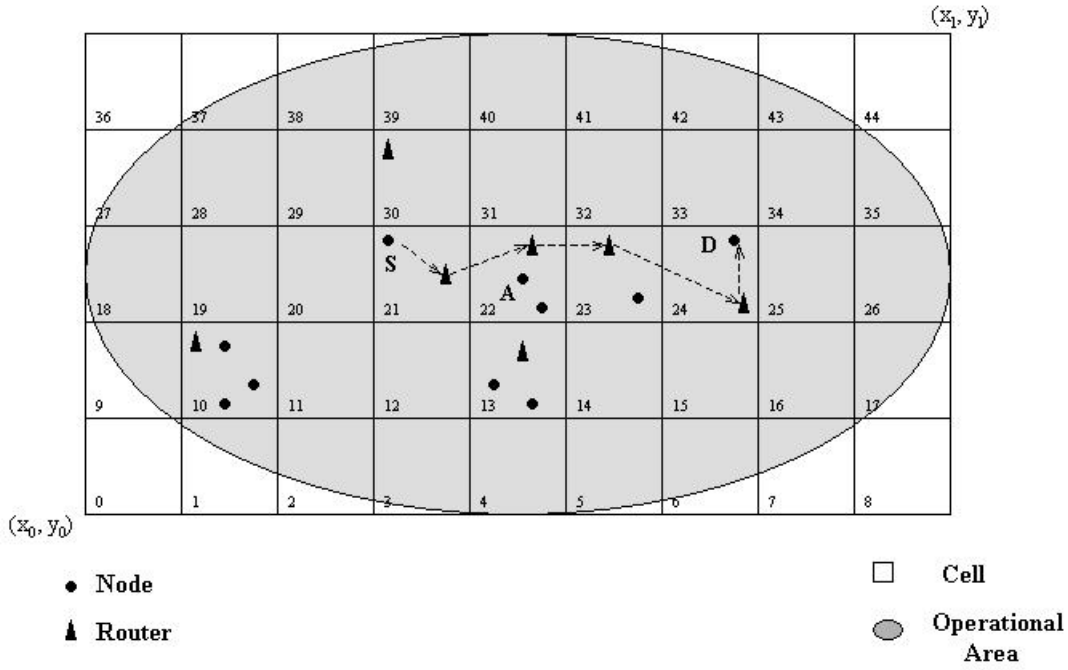

Figure 2.3: Routing in Kelpi

Kelpi assumes that the area of operation of the network is know before deployment, and divides it into a static grid with square cells as shown in figure 2.3 [9]. Prior to deployment, each node is initialized with sufficient information to compute the grid. This coupled with a positioning system, such as Global Positioning System (GPS), allows each node to be aware of its current cell. One of the nodes in the cell is distinguished as a router. Nodes register with the router when they enter the cell.

The router in each cell is associated with the *Cell Router Address* (CRA). The CRA is fixed for a given cell and is a function of the cell number. Since all nodes are aware of the CRAs of cells, communication with cell router can occur without the current router's address being known. Communication between nodes in different cells is through their respective cell's router's, which may in

turn use intermediate router's.

Distance Vector updates, computes the next hop router for each cell, and stores this information in a routing table. This constitutes the *proactive* component of Kelpi. Potential senders flood router's with messages and learn the destination nodes cell from a reply sent by the destination cell's router. This forms the *reactive* component of Kelpi.

2.6 Pro-active vs Reactive routing protocols

As pro-active routing protocols maintains routes to every other node in the network, regular routing updates imposes large overheads. Most of the effort is wasted because a few routes are used for a short duration of time. Pro-active protocols can be taxed to the full extent only in high traffic, low mobility networks. Moreover less mobility in such networks would help reduce the routing overheads to some extent. On the other hand, as reactive routing protocols maintain routes to only those nodes which are needed, they incur small routing overhead. But cost of finding routes is expensive since flooding is involved. Such a protocol is good in a small, medium traffic networks.

2.7 Problem

The performance of reactive routing protocols is affected by *routing overheads* and *delay* in repairing broken routes. Routing overheads are the result of error broadcasts followed by flooding in the route discovery phase. Delay in repairing routes is due to its inability to find an alternative route without re-initiating a route discovery phase. The need is for an approach that will repair broken routes with small overhead and delay.

2.8 Related Work

In this section we look at Preemptive routing. The central idea of Preemptive Routing [1] is to find an alternate path before the current path breaks. Preemptive outing is an enhancement over the basic AODV routing protocol. When a path is likely to be broken, a preemptive warning (RERR) is sent to the source indicating the likelihood of a disconnection. The source can then initiate a path discovery early, potentially avoiding the disconnection altogether.

A path is considered likely to break when the received packet power becomes close to the preemptive threshold. The preemptive threshold is assigned to a

value such that, by the time the path actually breaks an alternative path is available. Though preemptive routing reduces the delay in repairing broken routes it still does little to reduce routing overheads.

2.9 Focus of the work

Our focus here will be on reactive routing protocols and in specific AODV routing protocol. We have chosen AODV routing protocol because it is easily amenable to the solution we provide. The solution may be applied to other reactive routing protocols with appropriate modifications.

Chapter 3

Routing Handoff in AODV

It is evident from the earlier chapters that reactive routing protocols are affected by routing overheads and delays, in repairing broken routes. This is due to the re-initiation of route discovery. Our investigation into techniques which repair broken routes locally has resulted in the conception of *Routing Handoff*

But before we look at Routing Handoff we present the current solutions for repairing broken routes. In AODV when the destination or intermediate node moves breaking the route, a Route Error (RERR) packet is sent to the affected nodes. The source node when it receives the RERR packet re-initiates route discovery process. This results in large number of RERR packets followed by Route Request (RREQ) packets. An optimization to this is the Local Route Repair.

3.1 Local Route Repair

AODV with Local Route Repair (LRR) differs from AODV in route maintenance. In LRR, a link failure causes the intermediate node to initiate a route discovery. In the event of failure of route discovery (i.e. Local Route Repair Timer times out), a RERR is broadcast to the sources affected. In the event of success, a RERR with 'N' flag set is broadcast to the sources affected. This is to update the hop lengths along the path to the source. Besides only one route can be repaired at a time.

The problem with this approach is that the delay in fixing broken routes is still large. This is because we will have to wait till the route discovery succeeds or till the local route repair times out. In this process we also incur the flooding of RREQ packets involved in the route discovery process. In case of failure to find a route there is overhead of RERR broadcasts. This is also followed by

re-initiation of route discovery which further adds to the routing overhead

3.2 Routing Handoff

Routing handoff is a pro-active approach of dealing with route breaks. In routing handoff, each node makes use of its *Neighbor Information Table* (NIT). The central idea of routing handoff is to find a node in the neighborhood to take the task of routing the packets routed through a link which is about to break. The link is about is about to break when the ratio between received power and threshold power is less than *Handoff THreshold* (HTH).

When movement of the intermediate node or destination may cause a link to break, the node which uses the link as the next hop broadcasts a *Handoff REQuest* (HREQ). HREQ is a single hop packet. HREQ indicates the next hop node and all the previous hop nodes that use the link. When a neighbor node which receives the HREQ is in a position to route packets form some of the previous hop nodes to the next hop node, a decision made by using NIT, it sends the *Handoff REPly* (HREP). The node also updates its routing table. The previous hop nodes, which receive the HREP updates the routing table to make the node which sent the HREP as the next hop, thereby avoiding the broken link. The HREP is a single hop packet. The frequency of HREQ packet sent and HREP packet received is restricted by timers.

The advantage of this approach is that the broken routes are repaired with just two packets HREQ and HREP. Since it locally tries to find a alternative route, the delay is also less. Moreover more than one route can be fixed at a time. An added advantage of routing handoff is that in some cases route discovery becomes redundant.

3.2.1 Algorithm

The algorithm followed by each node in the network to perform routing handoff is outlined below. The details regarding the timers are ignored here. Here `Received Packet` refers to data, routing or Hello Message packets. Hello Messages are used by nodes to discover neighbors and maintain the neighbor information table. For each node in the network:

```
Begin
                        ⋮
    if((Power of Received Packet/threshold Power) < HTH)
    {
```

```
            Create Handoff Request Packet;
            Send Handoff Request Packet;
        }
        if(Received Packet == Handoff Request)
        {
            Check Neighbor Information Table;
            if(Next Hop Node in HREQ is a Neighbor)
            {
                if(Any Previous Hop Node in HREQ is a Neighbor)
                {
                    Update Routing Table;
                    Create Handoff Reply Packet;
                    Send Handoff Reply Packet;
                }
            }
        }
        if(Received Packet == Handoff Reply)
        {
            if(Handoff Reply is for this Node)
            {
                Update Routing Table;
            }
        }
                        ⋮
    End
```

3.2.2 Example

To make the concept of routing handoff more concrete consider the situation in figure 3.1. Let route from source $S1$ to destination $D1$ pass through A, B, C, D and the route from source $S2$ to destination $D2$ pass through E, C, D. At some point of time, the movement of node C causes either link BC or CD to break.

Figure 3.2 shows the scenario when the movement of C causes link CD to break. Before the the link CD is about to break, since node C has D as the next hop for some of the routes, initiates a HREQ (refer figure 3.4). HREQ invites responses from nodes that is the range of D and is in the range of either B or E. The hop count of HREQ is 1. Node F refers its neighborhood information

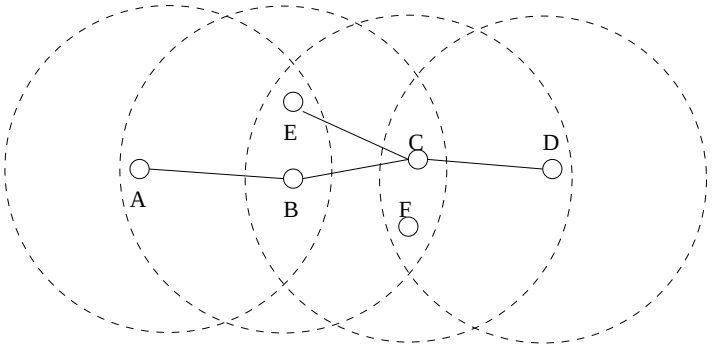

Figure 3.1: AODV with Routing Handoff

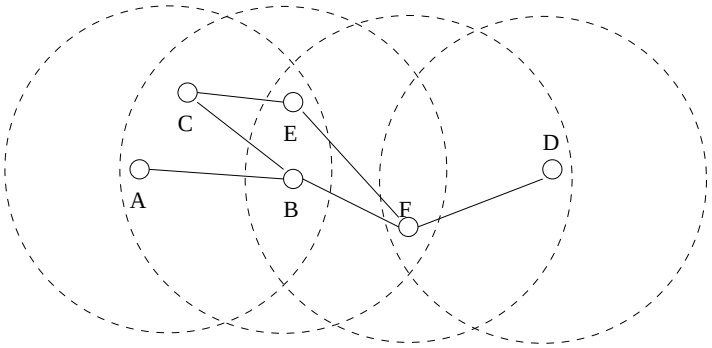

Figure 3.2: Route Repair when the link CD breaks

table and sends a HREP (refer figure 3.5). HREP from node F indicates that it can route packets from B and E to D. The hop count of HREP is 1. B and E on receiving the HREP updates its routing table so as to make node F as the next hop for packets form source $S1$ and source $S2$.

Figure 3.3 shows the scenario when the movement of C causes the link BC to break. Before the link BC is about to break, since node B has C as the next hop for some of the routes, initiates a HREQ. HREQ invites responses from nodes that is in the range of C and in the range of A. Node E refers its neighborhood information table and sends a HREP. HREP from node E indicates that it can route packets from A to C. A on receiving the HREP updates its routing table so as to make node E as the next hop for the packets from source $S1$.

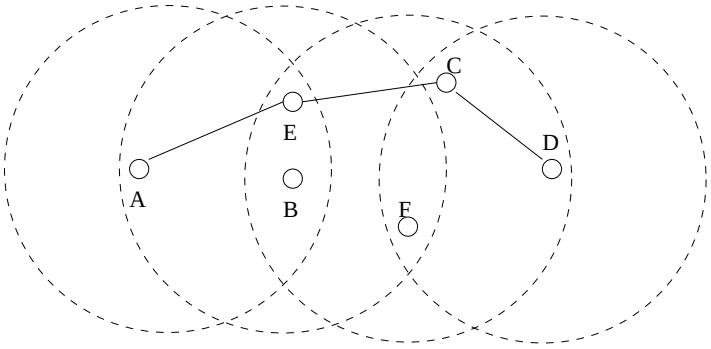

Figure 3.3: Route Repair when the link BC breaks

3.2.3 Computation of Handoff Threshold (HTH)

The idea in routing handoff is to hand over the routes before the link breaks. This is done by performing handoff when the ratio of the received power (RxPr) and the threshold power (RxThresh) of the received packet is less than Handoff Threshold (HTH).

$$\frac{\text{RxPr}}{\text{RxThresh}} \leq \text{HTH} \tag{3.1}$$

Let t be the time required for routing handoff, s be the maximum speed of the node and d be the distance to be covered during which the handoff is to take place (refer figure 3.6). But

$$\text{Received Power} \propto \frac{1}{\text{distance}^4}$$

$$\text{RxThresh} \propto \frac{1}{\text{R}^4} \tag{3.2}$$

$$\text{RxPr} \propto \frac{1}{(\text{R - d})^4} \tag{3.3}$$

Substituting 3.2 and 3.3 in equation 3.1 we get

$$\frac{\text{R}^4}{(\text{R - d})^4} \leq \text{HTH} \tag{3.4}$$

Substituting for d in equation 3.4 we have

$$\frac{\text{R}^4}{(\text{R - (s} * \text{t))}^4} \leq \text{HTH} \tag{3.5}$$

To get a clearer picture of the concept of routing handoff and how it fares with respect to other approaches, we need to analyze them. The next chapter introduces the underlying network model. The network model forms the basis of the theoretical analysis which is also presented in the next chapter.

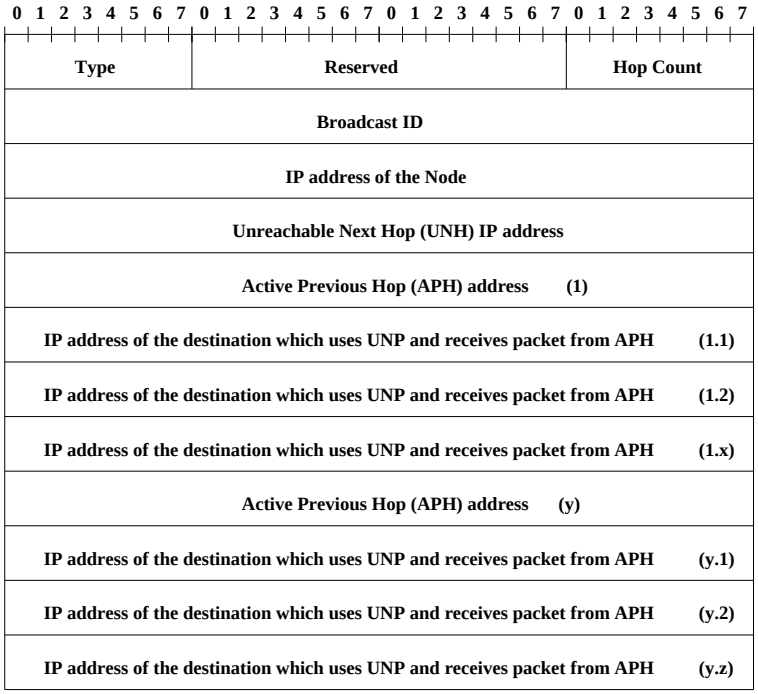

Figure 3.4: Handoff REQuest Packet Format

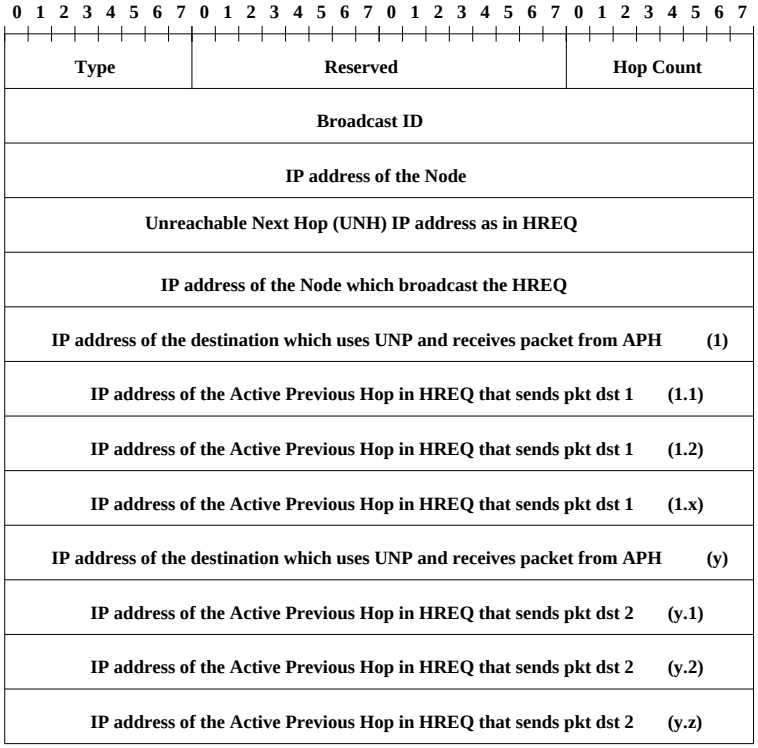

Figure 3.5: Handoff REPly Packet Format

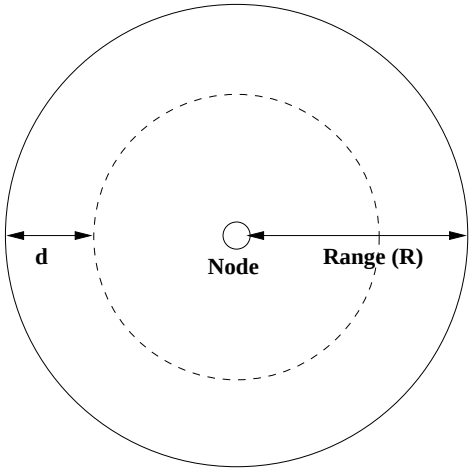

Figure 3.6: A node in the annulus performs routing handoff

Chapter 4

Theoretical Analysis

Here we introduce our network model. The network model forms the basis for analysis of the approaches presented in earlier chapters. There is not much work on analyzing any routing protocol because of the dynamic nature of the network. Our analysis has been made easier because of the fact that we are concentrating on route repair.

4.1 Network Model

Let A be the area of the network under consideration. N is the number of nodes uniformly distributed over the network. Assuming each node in the network has the same power, the range of transmission is R. We assume a random traffic pattern: each source node initiates packets to randomly chosen destinations in the network. The expected length $\overline{L}$ for such traffic

$$\overline{L} = \frac{2\sqrt{A}}{3} \tag{4.1}$$

The above result is derived in section 4.1 of Jingyang Li [5]. The result can be intuitively understood as follows. Since the nodes are uniformly distributed over the network, the number of nodes at a distance x from any node is proportional to x and the number of nodes within a circular radius of x is proportional to x^2. This means the plot of the radius x versus number of nodes at distance x is a straight line passing through origin. The slope of the straight line is immaterial for the analysis. The maximum distance is $\sqrt{A}$ for a square network with area A. Though strictly speaking the maximum distance is $\sqrt{2A}$, the results from the simulation concurs with the former value. Any node in the network can communicate with any other node in the network with equal probability. Then

the probability of a node communicating with another node at a distance x is

$$p(x) = \frac{x}{\int_0^{\sqrt{A}} x dx}$$

Therefore the expected path length for a random traffic pattern is

$$\overline{L} = \int_0^{\sqrt{A}} x p(x) dx = \frac{2\sqrt{A}}{3}$$

When a link breaks let ϕ be the number of routes effected. Between any source destination pair, each of the links can break with equal probability. For our analysis, we do not take cached entries into account during route discovery. We also assume that that time is proportional to the number of hops involved, i.e more the hops involved, more the time required.

4.2 Parameters

We analyze AODV, Local Route Repair and Routing Handoff with respect to the following parameters.

1. Number of packets involved in repairing a broken link (PKT)
2. Delay involved in repairing a broken link (DEL)

4.3 Basic Results

Here we show some basic results which forms the basis of our analysis. The results are based on the model stated in 4.1.

1. Number of packets involved in flooding the network:
   Here we assume RREQ broadcast reaches all over the network i.e. TTL of RREQ is equal to the diameter of the network. Since each node forwards the RREQ packets only once, the number of broadcasts required is N. Hence number of packets involved in flooding is N.

2. Number of hops H on the average to reach the destination:
   Since the expected path length is $\overline{L}$ and the transmission range is R, the number of hops required is

$$\begin{aligned} H &= \frac{\overline{L}}{R} \\ H &= \frac{2\sqrt{A}}{3R} \end{aligned} \tag{4.2}$$

3. Average hop/time to discover a route:
   From the above result the average hop/time to discover a route is $2H$.

4. Number of RERR broadcasts involved when a link breaks:
   Since our model assumes each of the H hops can break with equal probability, a link break closer to the source results in 1 RERR broadcast and a link break closer to the destination results in $H-1$ broadcasts, provided only one route is effected by the link breakage. Then the average number of RERR packets, per affected route.

$$\begin{aligned} k &= \frac{1+2+3+\ldots+H-1}{H} \\ &= \frac{H-1}{2} \\ &\approx \frac{H}{2} \\ k &= \frac{\sqrt{A}}{3R} \end{aligned} \tag{4.3}$$

   But if ϕ routes are affected by a link breakage and the path from these sources to the point of link breakage do not overlap, maximum number of RERR broadcast required is

$$\begin{aligned} K &= \phi k \\ K &= \frac{\phi\sqrt{A}}{3R} \qquad K < N \end{aligned} \tag{4.4}$$

   Note the the value of K is bounded by number of nodes in the network N. Such a scenario arises when a RERR broadcast ends up flooding the network.

4.4 Analysis of AODV

In AODV, a link failure causes a RERR broadcast to the sources affected. The source on receiving a RERR initiates a route discovery. The source may initiate a route discovery if it has a packet to send, here we assume it will.

1. Number of packets involved in repairing a broken route (PKT)
   Number of packets involved in repairing a broken route = RERR broadcast to the sources affected + flooding to discover the route for each route+ RREP unicast from the destination to the source

$$\begin{aligned} PKT &= K + \phi N + \phi H \\ &= \frac{\phi\sqrt{A}}{3R} + \phi N + \phi\frac{2\sqrt{A}}{3R} \end{aligned} \tag{4.5}$$

2. Delay involved in repairing a broken route (DEL)
   Delay involved in repairing a broken route = RERR broadcast to reach the source + RREQ to reach the destination + RREP to reach the source

$$\begin{aligned} DEL &= k + H + H \\ &= k + 2H \\ &= \frac{\sqrt{A}}{3R} + \frac{4\sqrt{A}}{3R} \\ &= \frac{5\sqrt{A}}{3R} \end{aligned} \tag{4.6}$$

4.5 Analysis of Local Route Repair

In AODV with local route repair we assume that the intermediate route discovery succeeds.

1. Number of packets involved in repairing a broken route (PKT)
   Number of packets involved in repairing a broken route = RERR broadcast + flooding to discover the route for each route + RREP unicast from destination to the intermediate node

$$\begin{aligned} PKT &= K + \phi N + \frac{\phi\sqrt{A}}{3R} \\ &= \frac{\phi\sqrt{A}}{3R} + \phi N + \phi\frac{\sqrt{A}}{3R} \end{aligned} \tag{4.7}$$

2. Delay involved in repairing a broken route (DEL)
   Delay involved in repairing a broken route = RREQ to reach the destination + RREP to reach the intermediate node

$$\begin{aligned} DEL &= \frac{H}{2} + \frac{H}{2} \\ &= H \\ &= \frac{2\sqrt{A}}{3R} \end{aligned} \tag{4.8}$$

4.6 Analysis of Routing Handoff

The question is how to ensure that there will be a node in the neighborhood which will takeover the routing of packets routed earlier on a broken link. Consider a section of the network as shown is Figure 4.1. When node C moves, it is not possible to find another node that will take the responsibility of routing packets from B to D (unless of course a new node moves to the original position of C). The point to note here is that certain amount of overlapping of

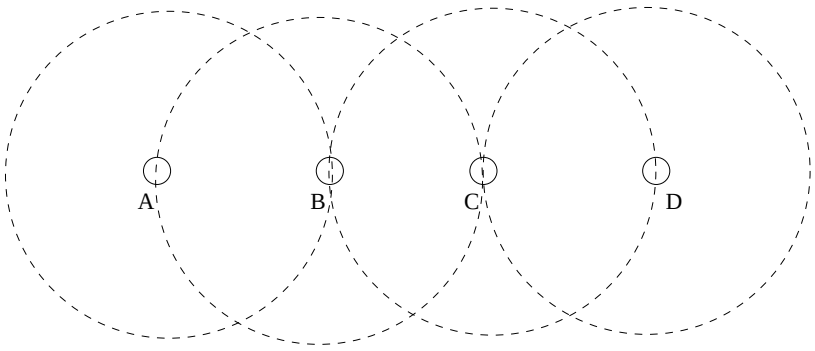

Figure 4.1: Non overlapping transmission ranges of nodes A, B, C, D

transmission ranges of B and D is required. This alone will not suffice. We also need nodes in the overlapping area that will take up the responsibility of routing packets from B to D as in Figure 4.2.

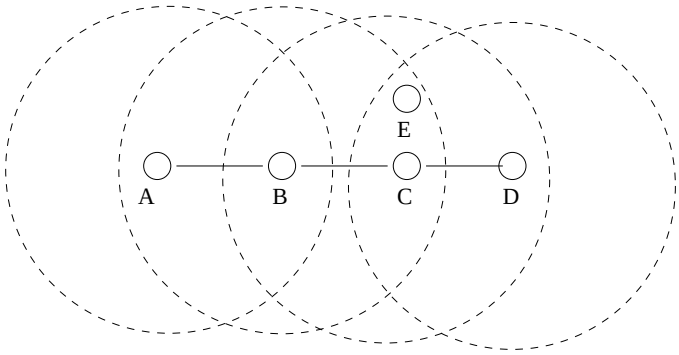

Figure 4.2: Overlapping transmission ranges of nodes A, B, C, D

The maximum extent of overlapping between nodes B and D, should be less than shown in Figure 4.3, or else we would not have used node C to relay the packets to D. The overlapping area will be less than $1.23R^2$. Since the nodes are uniformly distributed over the network, the number of nodes in the overlapping area

$$\eta \leq \frac{1.23R^2N}{A} \quad \text{and} \quad \eta \geq 2 \tag{4.9}$$

$$N \geq \frac{\eta A}{1.23R^2} \tag{4.10}$$

Equation 4.10 provides us the condition under which nodes will be present in the overlapping area. But how do ensure that the transmission ranges of

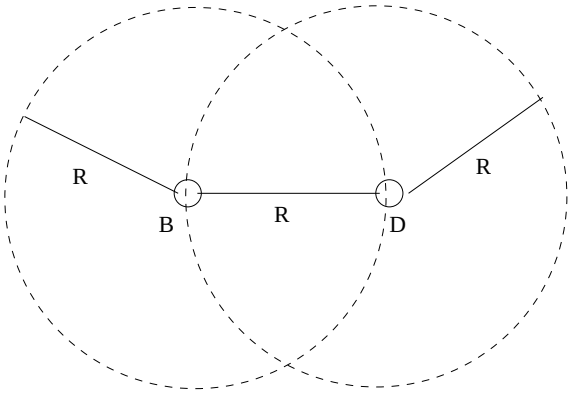

Figure 4.3: Maximum overlapping possible between node B and D

B and D overlap. Consider Figure 4.4 which is representative snapshot of our model. For the transmission range of B and D to overlap

$$\begin{aligned} R &> \frac{\sqrt{A}}{\sqrt{N}} \\ N &> \frac{A}{R^2} \end{aligned} \tag{4.11}$$

which essentially similar to equation 4.10

1. Number of packets involved in repairing a broken link (PKT)
   Number of packets involved in repairing a broken link = HREQ + HREP

$$\begin{aligned} PKT &= 1 + 1 \\ &= 2 \end{aligned} \tag{4.12}$$

2. Delay involved in repairing a broken link (DEL)
   Delay involved in repairing a broken link = HREQ + HREP

$$\begin{aligned} DEL &= 1 + 1 \\ &= 2 \end{aligned} \tag{4.13}$$

4.7 Discussion

Theoretically Routing Handoff is the most suitable approach for repairing broken routes. This is under the assumption that there is a node in the neighborhood of the broken link which is able to route the packets. If the condition derived in equation 4.10 is met then with high probability we will find a node in the neighborhood. Though this is not always possible. On the other hand, if

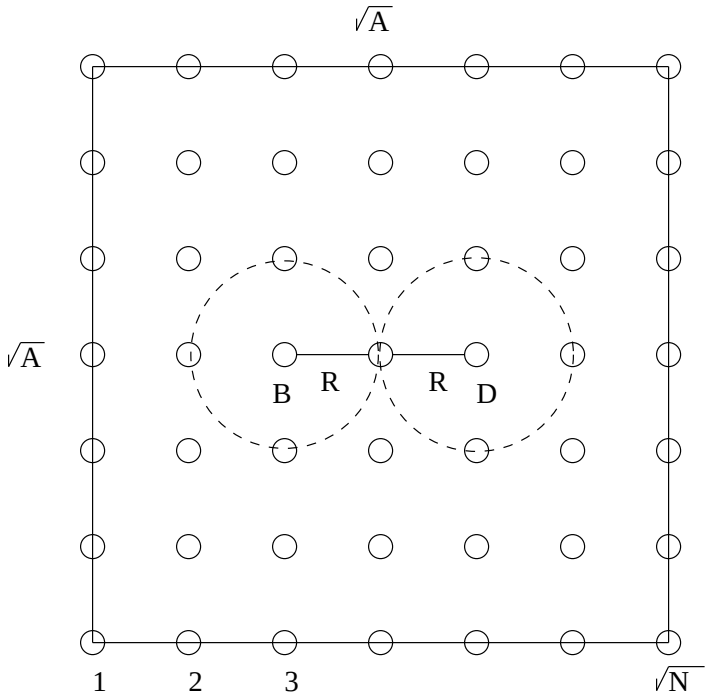

Figure 4.4: Snapshot of our Network Model

several nodes in the neighborhood are willing to route then we may have more than one Handoff Reply. Another point to note is that that for AODV and local route repair we have assumed worst case scenarios.

Chapter 5

Simulations

5.1 Network Simulator

For the purpose of simulation we use the Network Simulator [12] (NS). The simulator is written in C++; it uses OTcl as command and configuration interface. The reason for choosing ns simulator is its wide popularity coupled with the fact that the routing protocols to be modified are already implemented.

5.2 Implementation of Routing Handoff

The AODV routing protocol and its optimization Local Route Repair has already been implemented in NS. We have incorporated the routing handoff concept into the AODV routing protocol in NS. We created two new packet headers HREQ and HREP.
The structure of the HREQ is

```
struct src_list {
     nsaddr_t                src;       // previous hop node
     struct src_list         *next;
};

struct handoff_route
     nsaddr_t                dst;       // destination node
     u_int32_t               dst_seqno; // sequence no
     u_int8_t                hops       // hops to destination
     struct src_list         *src_list_hdr
     struct handoff_route    *next
};
```

```
struct hdr_aodv_hrequest {
     u_int8_t               hrq_type;
     u_int8_t               reserved[2];
     u_int8_t               hrq_hop_count; // set to 1
     u_int32_t              hrq_bcast_id;
     nsaddr_t               hrq_index;     // node which sends HREQ
     nsaddr_t               hrq_nexthop;   // next hop node
     struct handoff_route   *handoff_route_hdr;
};
```

The structure of HREP is

```
struct dst_list {
     nsaddr_t               dst;          // destination node
     struct dst_list        *next;
};

struct handoff_sources {
     nsaddr_t               src;          // previous hop node
     struct dst_list        *dst_list_hdr;
     struct handoff_sources *next;
};

struct hdr_aodv_hreply {
     u_int8_t               hrp_type;
     u_int8_t               hrp_flags;
     u_int8_t               reserved;
     u_int8_t               hrp_hop_count;// set to 1
     u_int32_t              hrp_bcast_id;
     nsaddr_t               hrp_index;    // node which sends HREP
     nsaddr_t               hrp_nexthop;  // next hop node
     nsaddr_t               hrp_node;     // node which sent HREQ
     struct handoff_sources *handoff_sources_hdr;
};
```

Routing handoff uses two timers `HandoffRequestTimer` and `HandoffReplyTimer`. The `HandoffRequestTimer` is used to restrict the number the of HREQ sent by

a node. It is controlled by the parameter `HRQ_ID`. Similarly `HandoffReplyTimer` is used to restrict the number of HREP received by a node. It is controlled by the parameter `HRP_ID`. The following functions form the core of the implementation of routing handoff

- `sendHandoffRequest(nsaddr_t nexthop)`: Creates the handoff request packet and broadcasts it.
- `recvHandoffRequest(Packet *p)`: Receives handoff request and checks if the node is in a position to route packets. If it is able to route the packets it calls sendHandoffReply.
- `sendHandoffReply(Packet *p)`: Updates routing table. Creates handoff Reply and broadcasts it.
- `recvHandoffReply(Packet *p)`: Receives handoff reply and updates routing table if appropriate.

5.3 Results

We now present the results of the simulations. The simulation results of AODV, Local Route Repair (LRR) and Handoff Routing (HR) are compared here. We conduct the simulations on network with 25 and 50 nodes. The transmission range of each node is 100 mts. The simulation time is 100 seconds. The throughput of the network is taken to be the ratio of the number of tcp data packets successfully received by all the nodes in the network to the number of tcp data packets sent by all the nodes in the network. Routing overhead for AODV and LRR is the sum total of RREQ, RREP and RERR. While routing overhead for routing handoff is the sum total of RREQ, RREP, RERR, HREQ and HREP. We subject each network to two different scenarios. One with low mobility and the other with high mobility. In each case the time required to perform routing handoff is 0.6 seconds.

In low mobility scenario, the minimum pause (minp) time is 5 seconds and maximum pause (maxp) time is 10 seconds. The speed of the node varies between 20 m/s to 40 m/s. The value of Handoff Threshold (HTH) is computed according to equation 3.5 and is set to value of 3.

In high mobility scenario, the minimum pause (minp) time is 1 seconds and maximum pause (maxp) time is 2 seconds. The speed of the node varies between 40 m/s to 60 m/s. The value of Handoff Threshold (HTH) is computed according to equation 3.5 and is set to value of 6.

5.3.1 25 Nodes

For a network with 25 Nodes and satisfying the criteria in equation 4.11 we set the area to 500 × 500 sq mts. The network is subjected to TCP traffic with 10, 15 and 20 tcp connections for both low and high mobility scenario. For low mobility HRQ_ID = 5 sec, HRP_ID = 0.1 sec and for high mobility HRQ_ID = 7 sec, HRP_ID = 0.1 sec provide better results.

TCP connections	AODV	LRR	HANDOFF
10	7331176	8163728	8624528
15	8115248	7381080	8819880
20	8419144	7784432	8453920

Table 5.1: TCP packets received for 25 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
10	39082	42038	41422
15	43347	43305	43841
20	44890	43335	44651

Table 5.2: Routing overhead (pkts) for 25 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
10	99.5944	99.3829	99.1753
15	98.8947	99.1456	99.0845
20	98.7148	99.2536	99.0334

Table 5.3: Throughput (%) for 25 Nodes (low mobility)

Low mobility: Table 5.1 shows the number of tcp packets successfully received/transmitted. Routing handoff is clearly able to successfully send more packets than AODV and LRR. Table 5.2 shows the routing overhead. A slightly higher overhead in routing handoff for connections 10 and 15 is justified by the higher number of tcp packets successfully sent. The throughput of routing handoff in table 5.3 is comparable with AODV and LRR.

TCP connections	AODV	LRR	HANDOFF
10	7497656	7090576	7723448
15	7679576	7709784	8686680
20	8094536	7716790	7973664

Table 5.4: TCP packets received for 25 Nodes (high mobility)

TCP connections	AODV	LRR	HANDOFF
10	39570	40895	39804
15	43484	43844	44536
20	42933	43419	44801

Table 5.5: Routing overhead (pkts) for 25 Nodes under (high mobility)

High mobility: Table 5.4 shows the number of tcp packets successfully received/transmitted. Routing handoff is clearly able to successfully send more packets than AODV and LRR in connection 10 and 15. In the case of 20 connections it does better than LRR and is comparable with AODV. Table 5.5 shows the routing overhead. A slightly higher overhead in routing handoff for connections 10 and 15 is justified by the higher number of tcp packets sent. But in case of connection 20 there is an increase in overhead without a corresponding increase in the number of tcp packets sent. This is because of less number of successful routing handoffs. The throughput of routing handoff in table 5.6 is comparable with AODV and LRR.

5.3.2 50 Nodes

For a network with 50 nodes and satisfying the criteria in equation 4.11 we set the area to 700 $\times$ 700 sq mts. The network is subjected to TCP traffic with 20, 30 and 40 tcp connections for both low and high mobility scenario. For low mobility HRQ_ID = 6 sec, HRP_ID = 0.1 sec and for high mobility HRQ_ID = 8 sec, HRP_ID = 0.1 sec provides better results.

Low mobility: Table 5.7 shows the number of tcp packets successfully received/transmitted. Routing handoff is able to successfully send more packets than AODV and LRR for 20 and 30 connections and its performance is comparable with AODV for 40 connections. Table 5.8 shows the routing overhead. A slightly higher overhead in routing handoff for connections 20 and 30 is jus-

TCP connections	AODV	LRR	HANDOFF
10	98.9254	99.0406	99.4457
15	98.716	99.4093	98.6894
20	98.6199	98.375	98.5048

Table 5.6: Throughput (%) for 25 Nodes under (high mobility)

TCP connections	AODV	LRR	HANDOFF
20	6305528	5724408	7001456
30	7288416	6745776	7569112
40	7991400	6737080	7962256

Table 5.7: TCP packets received for 50 Nodes (low mobility)

tified by the higher number of tcp packets sent. The reason for more overhead without a corresponding increase in the number of tcp packets sent is due to less number of successful handoffs. The throughput of routing handoff in table 5.9 is comparable with AODV and LRR.

High mobility: Table 5.10 shows the number of tcp packets successfully received/transmitted. Routing handoff is clearly able to successfully send more packets than AODV and LRR. Table 5.11 shows the routing overhead. A slightly higher overhead in routing handoff for connections 30 and 40 is justified by its higher number of tcp packets sent. The throughput of routing handoff in table 5.12 is better than AODV and LRR.

5.3.3 50 Nodes with larger Area

The criteria in equation 4.11 restricts a network with 50 nodes to an area of 700 $\times$ 700 sq mts. We now examine the effects when criteria is violated. Here we consider a network with 50 nodes and area 850 $\times$ 850 sq mts.The network is subjected to TCP traffic with 20, 30 and 40 tcp connections for both low and high mobility scenario. For low mobility HRQ_ID = 6 sec, HRP_ID = 0.1 sec and for high mobility HRQ_ID = 8 sec, HRP_ID = 0.1 sec provides better results.

Low mobility Table 5.13 shows the number of tcp packets successfully received/transmitted. The performance of routing handoff is erratic with respect

TCP connections	AODV	LRR	HANDOFF
20	43125	45894	45389
30	48957	48691	49351
40	52061	52234	52326

Table 5.8: Routing overhead (pkts) for 50 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
20	98.6335	98.3769	98.6997
30	97.3548	98.2126	98.5361
40	98.1511	98.4879	98.1073

Table 5.9: Throughput (%) for 50 Nodes (low mobility)

to AODV and LRR. Sometimes it perform better sometimes worse. This is because of low probability of finding a node in the neighborhood to perform routing handoff. Table 5.14 shows the routing overhead. The throughput of routing handoff in table 5.15 is comparable with AODV and LRR.

High mobility: Table 5.16 shows the number of tcp packets successfully received/transmitted. The performance of routing handoff is erratic with respect to AODV and LRR. Sometimes it perform better sometimes worse. This is because of low probability of finding a node in the neighborhood to perform routing handoff. Table 5.17 shows the routing overhead. The throughput of routing handoff in table 5.18 is comparable with AODV and LRR.

We can conclude the following from the results of the simulation:

- Routing Handoff performance is better than local route repair when the network confirms to the criteria in equation 4.11.
- Routing Handoff performance is comparable or better than AODV when the network confirms to the criteria in equation 4.11.
- Routing Handoff performance becomes erratic with respect to AODV and LRR when the criteria in equation 4.11 is violated.
- Routing Handoff performance varies with parameters like HTH, HRQ_ID and HRP_ID.

TCP connections	AODV	LRR	HANDOFF
20	7085696	5631656	7343312
30	6949472	7080072	7585968
40	6898712	5927256	7544856

Table 5.10: TCP packets received for 50 Nodes (high mobility)

TCP connections	AODV	LRR	HANDOFF
20	46882	48924	45949
30	52136	54186	56408
40	53167	55037	56670

Table 5.11: Routing overhead (pkts) for 50 Nodes (high mobility)

- It is difficult to predict the values of HTH, HRQ_ID and HRP_ID for which routing handoff would provide the best performance.

TCP connections	AODV	LRR	HANDOFF
20	97.5571	97.0997	97.6469
30	96.7787	98.6673	97.6422
40	96.3518	97.4998	96.8943

Table 5.12: Throughput (%) for 50 Nodes (high mobility)

TCP connections	AODV	LRR	HANDOFF
20	5960912	6338336	6078240
30	8108168	7519288	7686200
40	7592632	7989944	7544896

Table 5.13: TCP packets received for 50 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
20	48572	53726	44473
30	54584	54146	53557
40	57792	63480	57407

Table 5.14: Routing overhead (pkts) for 50 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
20	97.8452	98.152	96.6441
30	98.1356	98.357	97.9166
40	97.533	98.0815	97.786

Table 5.15: Throughput (%) for 50 Nodes (low mobility)

TCP connections	AODV	LRR	HANDOFF
20	7088928	6647112	7508768
30	6901304	64328376	6328376
40	6845264	6749008	6519672

Table 5.16: TCP packets received for 50 Nodes (high mobility)

TCP connections	AODV	LRR	HANDOFF
20	48586	46438	47702
30	51642	53469	53590
40	54965	56271	52438

Table 5.17: Routing overhead (pkts) for 50 Nodes (high mobility)

TCP connections	AODV	LRR	HANDOFF
20	96.8709	97.6809	97.64
30	96.6593	96.7083	96.8429
40	96.2848	97.1465	96.8833

Table 5.18: Throughput (%) for 50 Nodes (high mobility)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The performance of reactive routing protocols is affected by routing overheads and delays in repairing broken routes. Routing overheads are a result of error broadcasts followed by flooding in the route discovery phase. Delay in repairing routes is due to its inability to find an alternative route without re-initiating a route discovery phase. The need was for an approach that will repair broken routes with small overhead and delay. This led us to investigate techniques that will be able to locally repair the routes

The outcome was the concept of Routing Handoff. It is a pro-active approach to repairing broken links. The central idea is to find a node in the neighborhood to take the task of routing the packets routed through a link which is about to break. Theoretical analysis of the approach has provided us with a routing handoff criteria, conformance of which leads to better routing handoff performance. The concept of Routing Handoff has been incorporated into the AODV routing protocol.

The results of the simulation show better performance of routing handoff over AODV and Local Route Repair, when the the network satisfies certain condition. The performance of routing handoff becomes erratic when the condition is violated. The performance of routing handoff varies with parameters like Handoff Threshold, HRQ_ID and HRP_ID. The reason for the better performance of routing handoff is small overhead and delay in repairing broken routes.

6.2 Future Work

Future work in this project involves estimation of parameters like Handoff Threshold, HRQ_ID and HRP_ID for which routing handoff performs best. The parameters will vary across different networks. But given a network we need to come up with some technique, theoretical or heuristic to estimate the parameters. We also need to investigate the benefits of routing handoff in other routing protocol.

Bibliography

[1] Tom Goff, Nael B Abu-Ghazaleh, Dhananjay S Phatak, and Ridvan Kahvecioglu. Preemptive routing in ad hoc networks, 2001.

[2] MANET Working Group. http://www.ietf.org/html.charters/manet-charter.html.

[3] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*. Kluwer Academic Publishers, Dordrecht, The Netherlands., 1996.

[4] Young-Bae Ko and Nitin H.Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Mobicom*, pages 66–75, Dallas, Texas, October 1998. ACM/IEEE.

[5] Jinyang Li, Charles Blake, Douglas S. J., De Couto, Hu lmm Lee, and Robert Morris. Capacity of ad hoc wireless networks, 2001.

[6] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing. In *SIGCOMM*. ACM, October 1994.

[7] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *WMCSA*. IEEE, New Orleans, LA, Feb. 1999.

[8] Charles E. Perkins, Elizabeth M. Royer, and Samir Das. *draft-ietf-manet-aodv-09.txt*. IETF, November 2001.

[9] Srinath Perur and Sridhar Iyer. Kelpi: A cellular approach for efficient routing in mobile ad hoc networks. 2001.

[10] Raghupathy, Sivakumar, Bevan Das, and Vaduvur Bharghavan. Spine routing in ad hoc networks, 1998.

[11] Bo Ryu, Jason Erickson, Jim Smallcomb, and Son Dao. Virtual wire for managing virtual dynamic backbone in wireless ad hoc networks.

[12] Network Simulator. http://www.isi.edu/nsnam/ns.