Query-Localized Route Repair Mechanism for Ad-hoc On-demand Distance Vector Routing Algorithm

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Arunkumar Thippur Jayakeerthy

Certificate of Approval:

Drew Hamilton Associate Professor Computer Science and Software Engineering	Alvin S. Lim, Chair Associate Professor Computer Science and Software Engineering
Xiao Qin Assistant Professor Computer Science and Software Engineering	George Flowers Dean Graduate School

Query-Localized Route Repair Mechanism for Ad-hoc On-demand Distance Vector Routing Algorithm

Arunkumar Thippur Jayakeerthy

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
May 09, 2009

Query-Localized Route Repair Mechanism for Ad-hoc On-demand Distance Vector Routing Algorithm

Arunkumar Thippur Jayakeerthy

Signature of Author

Date of Graduation

Vita

Arunkumar Thippur Jayakeerthy, was born February 17th, 1983, in Bangalore, India. He graduated from M.E.S College with distinction in June 2000. He earned his Bachelors degree in Computer Science Engineering from Visveswaraya Technological University, Bangalore, India in 2004. He then joined work at Siemens Communications Software in the same year and has worked in the field of network management for fixed and mobile networks. He joined the masters program at Auburn University's Department of Computer Science and Software Engineering in Spring 2007. Since then he has worked under the guidance of Dr.Alvin Lim in the area of mobile computing.

Thesis Abstract

Query-Localized Route Repair Mechanism for Ad-hoc On-demand Distance Vector Routing Algorithm

Arunkumar Thippur Jayakeerthy

Master of Science, May 09, 2009
(B.E., Rashtreeya Vidyalaya College of Engineering, 2004)

85 Typed Pages

Directed by Alvin S. Lim

Ad-hoc networking is a concept in computer communications, which allows applications to communicate with each other by forming a temporary network, without any form of centralized administration. With the increasing popularity of mobile ad-hoc networking, the need to connect large numbers of wireless devices is becoming more prevalent. Each node participating in the network has both communication and computation capabilities and acts both as host and a router. The proliferation of such mobile devices in recent years has in turn given a boost to the amount of attention to mobile ad hoc networks (MANET) due to their potential applications.

On-demand routing is an important aspect of the current ad-hoc networking protocols, in which a route between a communicating node pair is discovered only on demand. However, links of MANETs are dynamic in the sense that they often experience breakage and changes as they move in the network, and link breakages severely deteriorate network throughput and routing performance. Existing ad-hoc routing schemes such as AODV

propose end-to-end route repair schemes which often entail additional route maintenance overhead with little or no improvement in network performance.

We propose a more reasonable route error handling scheme in keeping network performance goals. We call it Query Localized Route Repair (QLRR) where in the upstream neighbor that discovers the link failure tries to recover the route locally by discovering a route to the destination with itself as the source. While doing so the upstream nodes tries to limit the flooding to the nodes that are located in the vicinity of the original route. In this work we have incorporated our routing handling strategy into AODV's route maintenance scheme for our study. Our strategy is able to recover from link failures based on local interactions instead of performing a global route re-discovery. To demonstrate the effectiveness of our approach we compare our scheme with the default route error mechanism of AODV. For evaluating our approach in terms of reliability, the average delivery ratio was measured. To estimate the network routing overhead the number of protocol packets, and application packets transmitted were measured. Results show that QLRR makes a substantial improvement in the protocol routing overhead which is crucial for scalability of ad-hoc networks while the delivery ratio is also improved.

Acknowledgments

I would like to express my sincere gratitude to Dr. Alvin Lim for the guidance he has provided throughout my study at Auburn University. I would also like to express my gratitude to the advisory committee members, Dr. Drew Hamilton and Dr. Xiao Qin for taking time to review my work. Several fellow students have made significant contributions to this research including Raghu Neelisetti, Qing Yang, Santosh Kulkarni, and Brandon Maharrey. I am thankful for their support and their friendship. I am thankful to my new found love, Rashmi, for her patience and support during the final days of my research.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliography follows van Leunen's *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networking is an emerging technology that enables users to access a broad range of information and services while they are mobile. Wireless networks offer a great deal of flexibility and cost effectiveness making them a worthy alternative when compared to wired networks. Apart from this, the cost effectiveness of wireless networks makes them a preferred choice for large scale implementation in many organizations.

There are fundamentally two types of wireless networks: infrastructure networks and ad hoc networks. Infrastructure networks including cellular networks and WiFi networks let the existing wired network infrastructure carry data as well as voice. Although such networks have become immensely popular and practical, the dependence of such such networks on existing network infrastructure remains as a main limitation of networks and restricts users and applications to places from where the infrastructure is accessible.

On the other hand the advent of mobile ad hoc networks has provided an efficient, and most importantly cost effective way of exploiting the presence of mobile hosts when no infrastructure is available [1]. Mobile Ad hoc Networks are formed by autonomous system of mobile hosts connected by wireless links with no supporting fixed infrastructure or central administration. Such a shift in networking paradigm is effected by increasing capabilities of networking devices such as higher data rates and diminishing prices of such devices. Each of the nodes has a wireless interface and communicate with each other over either radio or infrared. Laptop computers and personal digital assistants that communicate directly with each other are some examples of nodes in an ad-hoc network. Nodes in the ad-hoc

network are often mobile, but can also consist of stationary nodes, such as access points to the Internet. Semi mobile nodes can be used to deploy relay points in areas where relay points might be needed temporarily.

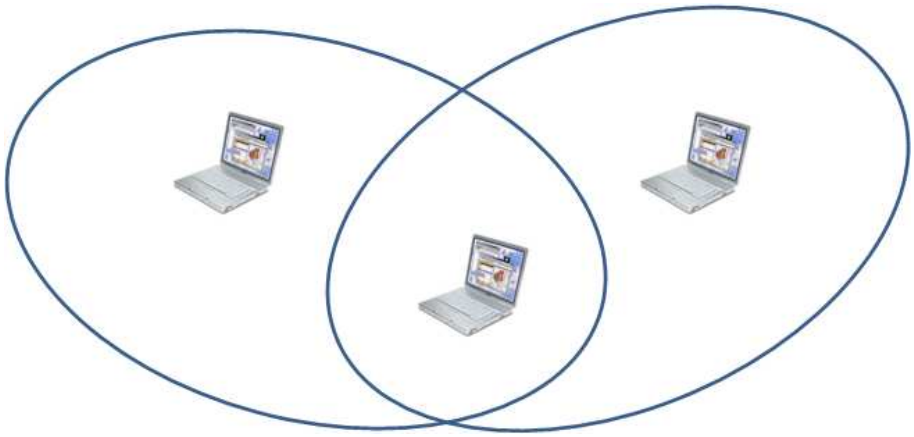

Figure 1.1: Example of Ad-Hoc Network with three nodes

Figure 1.1 shows a simple ad-hoc network with three nodes. The outermost nodes are not within transmission range of each other. However the middle node can be used to forward packets between the outermost nodes. The middle node is acting as a router and the three nodes have formed an ad-hoc network.

The advantages of such an ad hoc network include rapid deployment,robustness, flexibility and inherent support for mobility. In some application environments, such as battlefield communications, national crises, disaster recovery (fire, flood, earth quake) etc., the wired network is not available and ad hoc networks provide a feasible means for communications and information access. Also ad hoc networks are now playing important roles in civilian forums such as campus recreations, conferences, electronic classrooms, etc.

The successful implementation of ad hoc wireless networking technology presents a unique set of challenges that differ from traditional wireless systems and wired networks [2]. Wireless networks are prone to route breaks result from different sources such as: node mobility, signal interference, high error rates, fading environment, signal attenuation, Doppler effect and packet collision. Although a lot of research has been done toward the vision and the ultimate goal of ad hoc networks (by which users will be able to move anywhere anytime and still remaining connected with the rest of the world), many of the problems faced in ad hoc networks either remain partially solved or involve trade-offs amongst end user requirements. Numerous are the challenges that this class of wireless networks set to the research community [3]. For instance mobility in a network produces an actual route breaks while other sources produce a factious route breaks, but the MAC protocol, IEEE 802.11 [4], translates unsuccessful packet transmission as link failure. It has no ability to distinguish whether unsuccessful transmission has occurred due to mobility or something else. Therefore successful transmission rate in wireless environments, therefore, is much lower than that in wired environments.

Apart from this, low transmission power, and low available bandwidth are also the major challenges for routing in MANETs [5]. The volatile nature of links make routing an important aspect in managing a mobile ad-hoc network. Routing protocols should adjust to network topology changes, yet should not incur too much overheads in terms of the transmission of control messages. A considerable number of these challenges have been met through the development of sophisticated routing protocols which through their simplicity can provide stable solutions in these environments [6], [7] and [8]. Normal link state routing, established in wired Internet routing, has failed to fulfill the special requirements of ad hoc

routing [9]. This challenge has instigated extensive research on routing in ad hoc networks in recent years.

With the increase in the size and average route length, scalability becomes an issue for the current ad hoc routing protocols. Table-driven proactive routing protocols [10] that require periodic advertisement and global dissemination of connectivity information are not suitable for large networks. On-demand routing protocols are efficient for routing in large ad hoc networks because they maintain the routes that are currently needed, initiating a path discovery process whenever a route is needed for message transfer. AODV [11] and DSR [12] are two prominent ad hoc routing protocols that have used this approach. In AODV, the routing table at the nodes cache the next hop router information for a destination and use it as long as the next hop router remains active (originates or relays at least one packet for that destination within a specified time-out period). In DSR, which is a source-based routing, identity of all the intermediate nodes are included in the packet header. In large ad hoc networks, the header length could become very long. A survey of several routing protocols and their performance comparisons have been reported in [13] and [7], respectively.

As efficient as they are, on-demand ad-hoc protocols rely on some sort of broadcast mechanism to set up end-to-end routes. This tends to reduce their efficiency as a routing protocol while increasing the routing overhead at the same time. This thesis explores strategies by which ad-hoc routing protocols can be made more efficient and fault tolerant thereby improving overall reliability of the ad-hoc network. Specifically we investigate AODV protocol simply because it is one of the leading on-demand routing protocols [11] for ad-hoc networks.

Chapter 2 gives background information about AODV and an overview of research related to our area of concentration. In Chapter 3, we discuss the motivations for our work and describe relevant applications of MANETs. We describe the architecture of the proposed system in Chapter 4 and the design principles in Chapter 5. In Chapter 6, we give an overview of implementation of the mechanism from simulation point of view. Chapter 6 details our work. We conclude in Chapter 7 with a summary of our contributions and areas for future work.

Chapter 2

Background

This chapter gives background information on Ad-hoc on demand Distance Vector (AODV) routing protocol from a functional point of view. We then comment on shortcomings of the route repair mechanism of AODV. Finally we discuss literature dealing with improving reliability and fault tolerance of ad-hoc network routing protocols and AODV in particular.

2.1 AODV routing protocol

The Ad hoc On-demand Distance Vector (AODV) routing protocol [11], [14] is a reactive protocol designed for routing in ad hoc mobile networks. In this paper an implementation of AODV is utilized to determine the effectiveness of hello messages for determining local connectivity. A variety of approaches for improving the accuracy of hello messages as an indicator of local connectivity are examined.

2.1.1 AODV Architecture

Protocol Overview

The AODV protocol is a reactive routing protocol; routes are determined only as needed. When a route is required, AODV uses a route discovery process to learn the route. Once a route is established, it is maintained as long as it is needed through a maintenance procedure. These two operations are described in detail in subsequent sections. AODV

maintains routes using a soft state approach; if a route is not used it is expired after a specified time and is learned newly.

Protocol Details

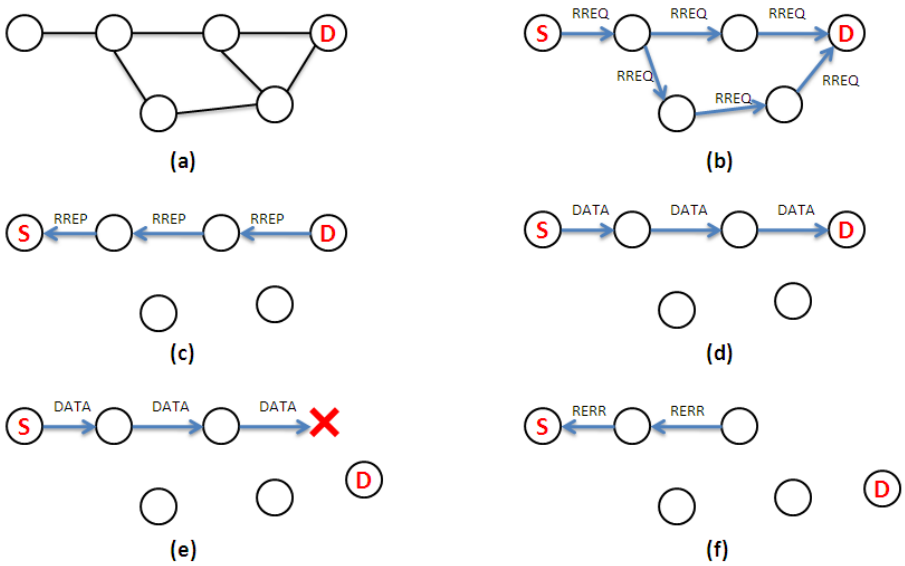

Figure 2.1: AODV Operation

Route Discovery When a source needs to send packets to a destination, it must first determine a path for communication. The source node begins route discovery by broadcasting a route request (RREQ) message containing the IP address of the destination. When an intermediate node receives the RREQ, it records the reverse route toward the source and checks whether it has a route to the destination. If a route to the destination is not known, the intermediate node rebroadcasts the RREQ. RREQ propagation is illustrated in Figure 2.1(b). When the destination, or an intermediate node with recent information

about a route to the destination, receives the RREQ, a route reply (RREP) is generated Figure 2.1(c). The RREP is unicast back to the source using the reverse route created by the RREQ. As the RREP propagates toward the source, a forward route to the destination is created at each intermediate hop. When a RREP reaches the source, the source records the route to the destination and begins sending data packets to the destination along the discovered path, as illustrated in Figure 2.1(d). If more than one RREP is received by the source, the route with the one with the latest sequence number followed by lowest hop count to the destination is selected.

Route Maintenance Route maintenance constitutes the task of detecting stale routes and marking them as invalid for further usage. Route maintenance is triggered whenever a link failure is detected. In order to be able to detect link failures each forwarding node to has to keep track of its continued connectivity to its active next hops. A node can maintain accurate information about its continued connectivity to the active next hops, using one or more of the available link or network layer mechanisms, as described below [14].

- Any suitable link layer notification, such as those provided by IEEE 802.11, can be used to determine connectivity, each time a packet is transmitted to an active next hop. For example, absence of a link layer ACK or failure to get a CTS after sending RTS, even after the maximum number of retransmission attempts, indicates loss of the link to this active next hop.

- If layer-2 notification is not available, passive acknowledgment will be used be used when the next hop is expected to forward the packet, by listening to the channel for a transmission attempt made by the next hop. If transmission is not detected within a

predefined time duration or the next hop is the destination (and thus is not supposed to forward the packet) one of the following methods SHOULD be used to determine connectivity:

- Receiving any packet (including a Hello message) from the next hop.
- A RREQ unicast to the next hop, asking for a route to the next hop.

In the default AODV protocol when ever a link along an active path, the node upstream of the break detects the break (see Figure 2.1(e)) and creates a route error (RERR) message. The RERR message lists all destinations that are now unreachable, due to the link break. The node then sends the RERR message toward the source. Each intermediate hop deletes any broken routes and forwards the RERR packet toward the source, as shown in Figure 2.1(f). When the source receives the RERR packet it determines whether it still needs the route to the destination. If so, the source creates a RREQ and begins the route discovery process again.

Hello Messages Network connectivity may be determined through the reception of broadcast control messages. Any broadcast control message also serves as a hello message, indicating the presence of a neighbor. When a node receives a hello message from its neighbor, it creates or refreshes the routing table entry to the neighbor. To maintain connectivity, if a node has not sent any broadcast control message within a specified interval, a hello message is locally broadcast. This results in at least one hello message transmission during every time period. Failure to receive any hello message from a neighbor for several time intervals indicates that neighbor is no longer within transmission range, and connectivity has been lost. Two variables control the determination of connectivity using hello

messages: HELLO INTERVAL and ALLOWED HELLO LOSS. HELLO INTERVAL specifies the maximum time interval between the transmission of hello messages. ALLOWED HELLO LOSS specifies the maximum number of periods of HELLO INTERVAL to wait without receiving a hello message before detecting a loss of connectivity to a neighbor. The recommended value for HELLO INTERVAL is one second and for ALLOWED HELLO LOSS is two [11]. In other words, if a hello message is not received from a neighbor within two seconds of the last message, a loss of connectivity to that neighbor is determined.

2.1.2 Strengths of AODV

On-demand Nature

As discussed above, route discovery in AODV happens on an ad-hoc basis. Because the protocol does not require global periodic routing advertisements, the demand on the overall bandwidth available to the mobile nodes is substantially less than in those protocols that do necessitate such advertisements. The reactive nature of ADOV makes it quite suitable for a dynamic self-starting network, as required by users wishing to utilize ad-hoc networks. It is also relevant to senor networks since the sensors usually remain inactive for long periods of time waiting for occurance of events of interest.

Localized Interactions

One of the most notable characteristics of AODV is its fully localized nature. Nodes only require knowledge about their 1-hop neighbors to establish forward and reverse routes. No global information, whether end-to-end or multi-hop, is needed for routing. This means that the routing table scales with the number of neighbors as opposed to the total number

of nodes in the network. The localized nature of AODV also simplifies the routing protocol since global information is not required to make routing decisions.

Links Inherently Serve Multiple Endpoints

Finally AODV allows various types of communication paradigms. Unlike other network protocols each link in AODV is capable of serving (N→M) end to end paths. This makes it possible to have request-receive which requires (1→N) and publish-subscribe (N→1) communication paradigms as well.

2.1.3 Weakness of AODV

Flooding

In AODV, route learning is limited only to the source of any routing packets being forwarded. That way, nodes in AODV can gather only a very limited amount of routing information. This causes AODV to rely on a route discovery flood more often, which may carry significant network overhead. An uncontrolled flood generates many redundant transmissions which may cause so-called broadcast storm problem [15].

Scalability

The cost of flooding increases with the increase in size of the network. The performance of the AODV protocol without any misbehaving nodes is poor in larger networks mainly because the average path length is higher in larger networks. A long path is more vulnerable to link breakages causing more flooding and resulting in high control overhead for maintenance. Furthermore, as a size of a network grows, various performance metrics

begin decreasing because of increased route maintenance work. Therefore the AODV is suited only for small and medium size networks, the scalability limit is about 1000 nodes. Simulations of Perkins [11] group show that at 1000 nodes the goodput ratio is about 70.53% as opposed to 98.75% and above for a network with 50 nodes seen in our simulations.

Security

Functioning of AODV routing protocol is based on the assumption that all nodes will cooperate. Without this cooperation no route can be established and no packet can be forwarded [16],[17]. This makes it vulnerable to various kinds of attacks given the possibility of existence of un-cooperative nodes. There are two main types of uncooperative nodes: malicious and selfish. Malicious nodes are either faulty and cannot follow the protocol, or are intentionally malicious and try to attack the network. Selfishness is noncooperation in certain network operations, for example dropping of packets which may affect the performance, but can save the battery power.

Weak Route Repair mechanism

Because the default route maintenance mechanism of AODV does not locally repair a broken link, the whole route is reconstructed, even though the route error has been caused by just one node. Route reconstruction between the endpoint nodes of a path adversely affects the network performance especially in a large network. Not only long latency for repairing the broken route but also excessive flooding of route request messages are induced by the inefficient route repair mechanism of AODV. Therefore, even though the protocol performs well in static and low mobility ad hoc environments, the performance degrades rapidly with increasing mobility. This further leads to the transport layer protocols leading

to low goodput levels. Although retransmission strategies can be adopted to improve on reliability, there is still room for improvement in the underlying AODV routing protocol.

In the rest of this chapter we discuss related research work that has taken place on this front and analyse their findings and shortcomings there by establishing a motivation for this thesis.

2.2 Related Work

Due to their inherent mobile nature, nodes of a MANET lead to a changing network topology. From individual links point of view this means possible link breakages. In MANETs, any link breakage along established routing paths will cause all those paths to fail, leading to traffic backlogs at upstream nodes, and more costs due to path re-establishments efforts. Therefore it is quite important to have a routing mechanisms that try to work around the issue of link breakages in MANETs. Several researchers have tried to approach the problem from different angles. After a literature review of published work, proposed solutions can be broadly classified into two types. The first category of solutions try to proactively predict the mobility pattern of nodes and make use of this information to adapt suitably. Another class of solutions are reactive in the sense that they act upon a link breakage. This section discusses some of the solutions of both types.

2.2.1 Associativity based routing protocol

[18] Associativity based routing protocol defines a new routing metric for ad-hoc mobile networks, called as degree of association stability. In ABR route is selected based on the degree of association stability of mobile nodes.Each node periodically generates a beacon to

signify its existence. This beaconing causes their associativity tables updated. Association stability is defined by connection stability of one node with respect to another node over time and space. According to [19] however, this approach is known to generate paths that are longer in length and therefore entail greater end-to-end delays. Besides every route discovery packet carries with it a associativity information of all the intermediate nodes that it passed. This approach does not scale very well as the size of the packets increase with increasing node densities.

2.2.2 SSA routing protocol

[20] The Signal Strength Adaptive (SSA) routing protocol performs on-demand route discovery selecting longer-lived routes based on signal strength and location stability. The signal strength criterion allows the protocol to differentiate between strong and weak channels. Each channel is characterized as strong or weak by the average signal strength at which packets are exchanged between the hosts at either end of the channel. The location stability criterion biases the protocol toward choosing a channel which has existed for a longer period of time. Together, these two concepts form the signal stability criterion that chooses strong channels which have been in existence for a time greater than some threshold. Although (SSA) routing protocol assumes signal strength to be an indicator of location stability. However it is well known that signal strength in wireless networks can vary to large extent over a period of time. This makes SSA quite sensitive to the changes in wireless medium.

2.2.3 Proactive Route Maintenance

[21] Proactive Route Maintenance in wireless ad hoc networks combines reactive route discovery and proactive route maintenance (PRM) of active routes, which adapts well to highly dynamic networks and reduces the frequency of costly route recoveries. In PRM all optimal paths are active in forwarding data packets. Sub-optimal paths are backup and activated only when all optimal paths have failed. PRM introduces a notion of watermarks in order to maintain loop free routes allowing data to propagate only from a node with higher watermark to a lower watermark. Each node broadcasts its watermark proactively to keep its neighbors informed. Preliminary results show that there is very little improvement in delivery ratios in a typical networks of 50 nodes in an area of 670X670. In fact local repair proposed for AODV itself yields better delivery ratios than PRM.

Routing protocols such as the one discussed above can solve the problem for a select class of MANETs and cannot be applied generically to all scenarios. With this argument it makes sense to focus our research on a reactive approach. Following are some of the existing approaches to improve the handling of errors due to link breakages in MANETs.

2.2.4 PATCH

[22] Proximity Approach To Connection Healing (PATCH) is a local recovery mechanism, which aims to reduce the control overhead and achieve fast recovery when route breakage happens. PATCH has an error recovery mechanism similar to AODV in that a request packet is sent out by the upstream node of the failed link to find the original next hop or other node which is at the further part of the original route with a time-to-live of 2 hops. It claims that the possibility of repairing the current route should be high and the

overhead should be much lower than using end-to-end global recovery. Although the idea is appealing there is no analytical model as to why a TTL of 2 is used. Also there are no results for other values of TTL indicating that optimal performance is obtained for TTL of 2.

2.2.5 AODV-BR

AODV-BR [23] is a modified protocol from AODV literally. The crux of AODV-BR is providing multiple alternate routes. The goal of the algorithm is to establish multiple routes without transmitting any extra control messages. This requires every node in the network to operate in promiscuous mode. When a RREP packet comes back to the source node, a neighboring node which is not part of the route overhears a RREP packet. And then it records the node which transmitted a RREP packet as the next hop to the destination in its alternate route table.

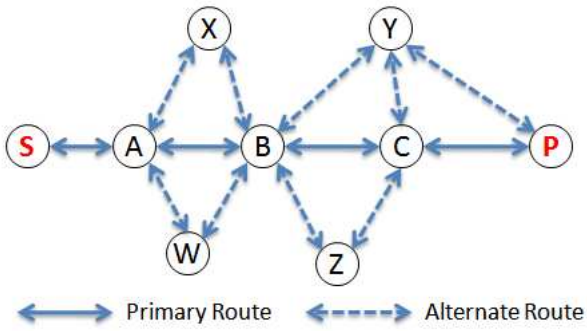

Figure 2.2: AODV-BR Operation

After these operations, the members of the route and neighboring nodes form a mesh structure. Within these mesh structures, a data packet is delivered via an alternate route

when the primary route is disconnected as depicted in Figure 2.2. While AODV-BR is quick to react towards route errors, it is not quite energy efficient as it requires all the nodes in the network to operate in promiscuous mode. Operating in promiscuous mode entails high rate of energy dissipation.

2.2.6 WAR

Witness-Aided Routing (WAR) [24] has a similar concept compared to AODV-BR. Like AODV-BR, WAR allows nodes to operate in promiscuous mode. A witness is defined as a host which can overhear a transmission that is not destined to it.

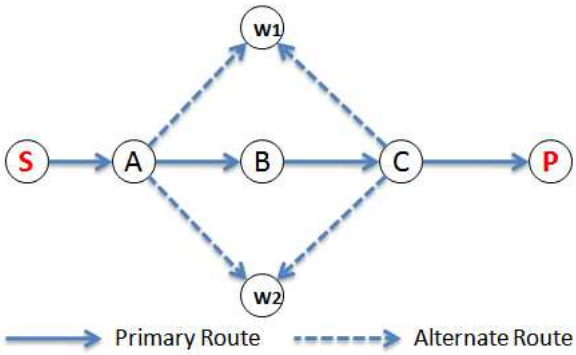

Figure 2.3: WAR Operation

Figure 2.3 shows how witnesses participate in the routing process. Both node W1 and W2 hear node As transmission to node B, which makes them potential active witnesses of node A with respect to the packets sent from node A to node B. At this point, they will wait to see if node B attempts to deliver the packet to node C, which would mean that node B received it from node A. If that is the case, their role with respect to the packets

sent from node A to node B reduces to sending an acknowledgment to node A (to avoid an error in case node A could not hear node B

s

transmission to node C). If neither W1 nor W2 hear node Bs transmission to node C, they conclude that the packets from node A to node B failed to reach node B. In this case, they will both attempt to deliver the packet directly to node C, although, indirectly, they target node B as well. Since node W1 and node W2 do not necessarily have a way to communicate with each other and avoid contention, they will ask node C for arbitration before sending the packet. If node C rejects their request, it means that it has already received the packets from node B and their role reduces to sending the acknowledgment to node A.

2.2.7 Other Local repair schemes

A Path Reliable Routing Protocol in Mobile Ad-hoc Networks

As mentioned in [25] this protocol works by maintaining a Path_Reliability metric in the route request packet. Route setup phase consists of request and reply phases. When a source needs to send data to a destination, it first broadcasts a route request (Sequence_ID, Source_ID, Destination_ID, Path_Reliability, Time_Span), where Sequence_ID is the request sequence number, which is used to identify a request and prevent a node from forwarding duplicated requests. Path_Reliability is set to 1.0 initially, and updated when the request is received. Time_Span is a user specified time span that the established route is expected to be reliable within this time span.

Link Lifetime Management - similar to the approaches discussed so far, this protocol also requires each node to send out a link layer beacon to its neighbors once every time quanta. Every node receiving this beacon records the link lifetime in a table. If link layer does not provide such function, then a network layer hello message is used. Hello message is a small message which is broadcast to every neighbor periodically.

2.2.8 QLRS with APM

Quick Local Repair Scheme using Adaptive Promiscuous Mode in Mobile Ad Hoc Networks [26] proposes a local repair scheme mainly composed of two parts: adaptive promiscuous mode and quick local repair scheme. In adaptive promiscuous mode a node repeatedly switches between promiscuous mode and non promiscuous mode to overcome energy dissipation caused by using promiscuous mode in overall time and quick local repair scheme is to quickly perform the local re-route discovery process with the information of the active connection in the local area acquired by promiscuous mode. The main condition for a node to stay in promiscuous mode is that it has to have at least three neighbors, to ensure that it can effectively act as a backup for the routes passing through the neighboring links that are within transmission range of current node. When a link fails, the upstream node sends a HELP message similar to AODV operating in local repair mode. Then the node in promiscuous mode which can effectively replace the broken link replies with a REPLY message and that completes local route repair. It has to be noted however that in most of the typical scenarios for an ad hoc network to perform reasonably, the average number of neighbors to a given node is usually greater than 3. This means with APM, most nodes

end up operating in promiscuous mode and we may not achieve as much energy efficiency as claimed by the authors.

Yet another approach to handling link failures in MANETs to let transport layer services take care of end to end connections.

2.2.9 Path reliability at transport layer protocol

Argyriou, et.al. [27] introduce an end-to-end approach for achieving the dual goal of enhanced reliability under path failures, and multi-path load balancing in mobile ad hoc networks (MANETs). It relies on building a disjoint-path identification mechanism for maintaining multiple routes between two endpoints on top of the Stream Control Transmission Protocol (SCTP), and the Dynamic Source Routing (DSR) protocol. The proposed approach differs from previously work in the sense that it is entirely an end-to-end scheme built on top of a transport layer protocol.

Chapter 3

Motivations, Objectives, and Applications

In this chapter we discuss the motivations for our improvements to AODV. We also present several existing and potential applications of mobile ad-hoc networks. Our protocols are particularly relevant to such systems since they require robust and reliable network services.

3.1 Motivations

Although AODV is generally a well established ad-hoc routing protocol, it has several apparent weaknesses. Its reliance on flooding of route requests incurs a significant penalty on communication and energy efficiency. Furthermore, the default AODV has a poor mechanism of route repair that can lead to periodic flooding. We propose an efficient localized flooding scheme to reduce overall routing overhead and increase scalability of the network. We also introduce a route repair algorithm that reconfigures the network after node failure faster than the default mechanism.

Because of the dynamic nature of mobile ad-hoc networks, node and link failures are expected occurrences. When links fail, the routing protocol may attempt to repair from the breakage in one of two ways: global route repair or local route repair. Global route repair protocols initiate the recovery process by either implicitly informing the source with the absence of a positive acknowledgment or by explicitly alerting the source node of the problem with a negative acknowledgment. In the former case, a negative acknowledgment will be sent from the intermediate node to the source node, reporting the link failure. In

the latter case, the node which detects a break will take no action so that the sender will timeout while waiting for a positive acknowledgment. AODV essentially uses the explicit approach (negative acknowledgment) in that the protocol defines an explicit route error (RERR) message to initiate re-discovery at the source node to overcome broken links. The source node begins route discovery again by broadcasting a route request (RREQ) message containing the IP address of the destination.

The problem with end-to-end repair is the high cost of network-wide flooding. This has serious implications on the performance of a system in terms of scalability, energy consumption, and latency. Protocols which depend on global error-recovery mechanisms do not scale well with network size. Moreover, since every node must forward the flooded packet, each node consumes energy repairing a route which is possibly very distant. Latency of end-to-end repair mechanisms also suffers since routes are completely rediscovered from the source to the sink.

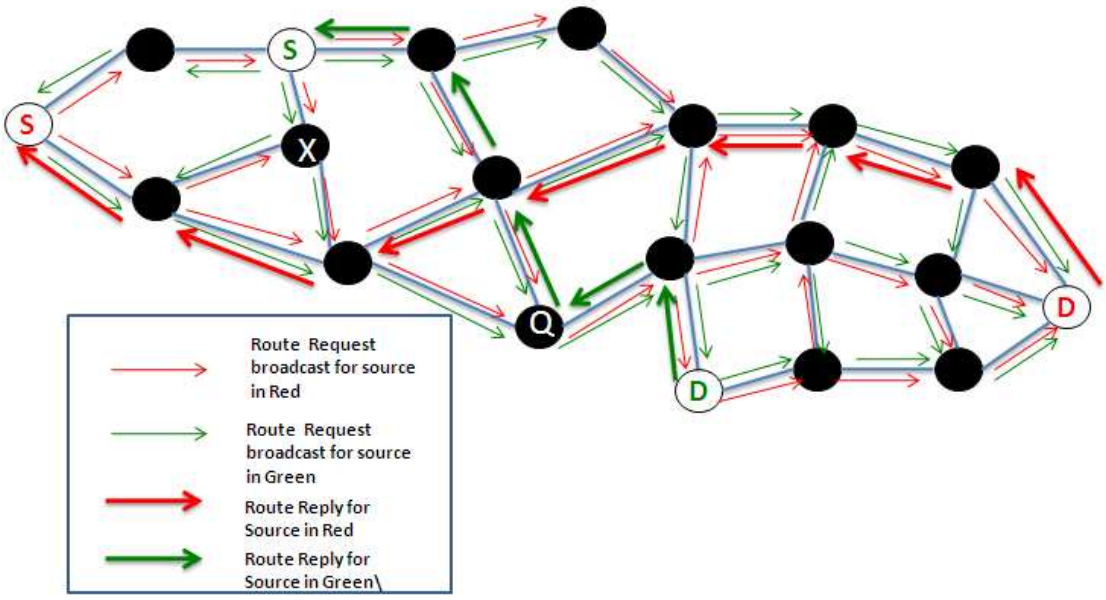

Figure 3.1: Global Route Repair

Figure 3.1 graphically illustrates the flooding of route request packets in AODV. Note that the diagram depicts only two source nodes. With more number of sources, the overhead caused by flooding increases at a fast rate. When node Q moves relatively far away from node X, it causes X to initiate an error-recovery process. In an ideal case, only nodes around the link failure should be involved in the repair process. the localized approach greatly reduces the overhead associated with repairs. Figure 3.2 shows an ideal case for local route repair where distant nodes are not involved in the recovery process at all. Nodes in the immediate vicinity of the break participate in the repair algorithm. Note that the source is not involved in the recovery process. The destination node may or may not be involved depending on whether routing information can be obtained from other nodes to that destination.

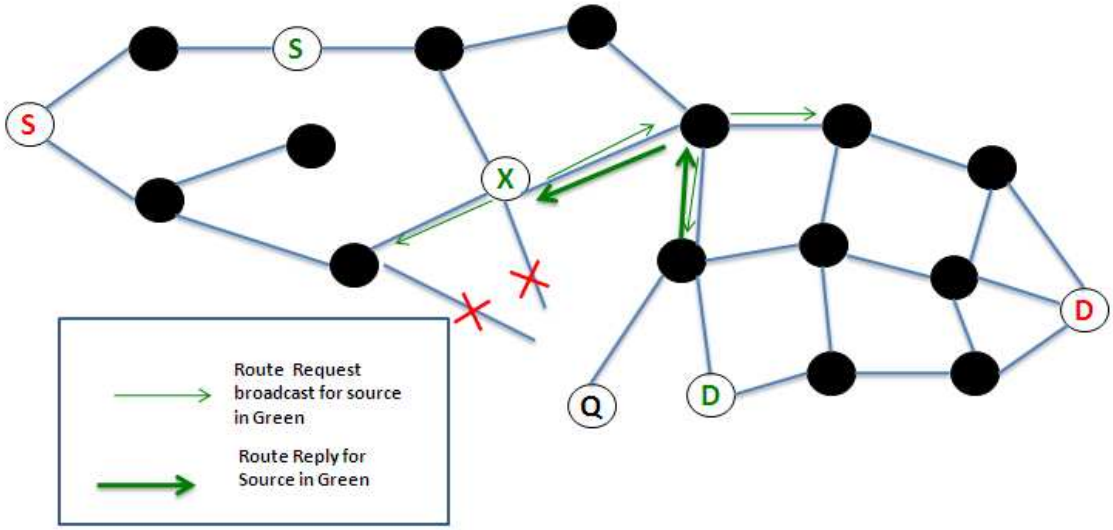

Figure 3.2: Ideal Local Route Repair

The advantages of the local repair approach include increased scalability, increased flooding efficiency, and decreased latency of repair. Since only a portion of the nodes are involved in local repair, the protocol scales more gracefully with network size and consumes less overall system energy. The localized nature of the recovery also lends itself to faster route repair since the complete (source to sink) route does not have to be traversed. In their analytical comparison of local and end-to-end recovery mechanisms, Aron and Gupta [24] show that with end-to-end recovery the probability of successfully delivering a packet on the first attempt rapidly degrades with increasing network size. In summary, local repair improves the scalability, efficiency, and latency of a network protocol. Additionally, the amount of resources consumed per packet is several orders of magnitude larger for end-to-end repair than for local repair. In summary, local repair improves the scalability, efficiency, and latency of a network protocol.

3.2 Objectives

In this section, we discuss the specific objectives of route error handling scheme. We explain our overall goal and highlight the improvement that will be achieved by our design.

3.2.1 Efficient Route Repair

The goal of our route repair approach is to quickly repair broken data paths in a highly localized fashion. Reactive repair is vital so that the network can quickly recover from failure. Repair time should be kept to a minimum so that latency and delivery effectiveness are within permissible levels for many types of applications. Localized repair is essential in order to minimize the overhead associated with path repair. Thus, our route repair protocol aims to achieve localized and timely repair without impeding the efficiency and improving scalability.

3.3 Applications

The emergence of pervasive computing has expanded the frontier for mobile ad-hoc systems. Although the MANET paradigm like sense and response systems is broadly applicable to various fields, its potential utility has not been fully realized. The applications of these networks can be in several areas, some of which are mentioned below:

- military[2], i.e. it is possible to equip soldiers with devices in enemy environments so that they can communicate each other.
- personal area network, i.e. printers, PDA, mobile phones
- business indoor application, i.e. meetings, symposium, demos

- civilian outdoor application, i.e. taxis, cars, sport stadiums [3]
- emergency application, i.e. emergency rescue operations, police, earthquakes
- home intelligence devices.

We present a summary of the applications of previous ad-hoc systems and propose several new areas well-suited to this paradigm.

3.3.1 Military applications

In most of the cases, small-scale military operations are often spontaneous i.e. with little or no fixed network infrastructure. These operations require a communications solution which is spontaneous too. In other words, the soldiers should be able to form a network when and where it is needed. In [28], the authors outline the requirements of a planning and decision aid to support small unit operations in urban terrain. This presents a realistic application scenario in which it is required to equip soldiers with devices in enemy environments so they can communicate each other, for example, acquiring GPS coordinates. Geographical location is one of the most important factors of any military operation which makes the current geographical positioning systems less efficient to support such operations. As the soldiers in a combat operation cannot afford exposing themselves in a battlefield to acquire GPS coordinates. One of the other limitations of the current geographical positioning system is that the satellite signals cannot penetrate through caves, underground bunkers or inside shielded buildings. In comparison with geographical positioning systems, mobile ad-hoc networks can support the built-in geographical location by using an extremely accurate form of triangulation. This feature enables soldiers in a military operation to triangulate its position based on the mobile enabled vehicles or other devices. In mobile ad-hoc

networks, readings are faster than the geographical positioning systems because the soldiers don't have to wait for multiple satellites to acquire a centralized security information. The devices used in combat operations must be able to address both communications security and a way to secure the network from unauthorized use. Mobile ad-hoc networks also allow devices to transmit at a lower output power to their neighbors which benefits the overall network by lowering the probability of detection and by increasing the battery Therefore if the device is captured, the soldiers can list that device to maintain the integrity of the network.

3.3.2 Crisis Management and Emergency Medical Response

A mobile ad-hoc network can also be used to provide crisis management services. As an example, in a disaster recovery situation where the entire communication infrastructure could possibly be destroyed, resorting to communication quickly is crucial. By using a mobile ad-hoc network, an infrastructure could be set up in hours instead of weeks, as is required in the case of wired communication. [29] presents an interesting application scenario where a group of mobile robots networked using MANET have the objective of isolating a contaminator thought to be in an unexplored facility. The scenario is typical of MANET environment in that individual nodes have to cope with Network partition due to mobility, limited transmission range and security risks, needless to mention that reliable routing is one of the crucial challenges for successful realization of such an application.

3.3.3 Commercial Applications

Although still in its infancy, ad-hoc network technology has already been used to develop some prototype applications. The National Central University of Taiwan [30] has

realized a prototype for a tourists guide system for its own campus based on the knowledge of the context. The system is composed of a GPS receiver which is connected to a notebooks port; the tourists guide system software, implemented in java, is composed of three parts:

- 2D map:it shows the users position in the campus
- 3D virtual world: to make virtual tours and web information (in which all the info about the campus are stored). By this ad-hoc technology, the visitor can interact with other visitors.

[31] presents a set of MANET application services related to image acquisition or processing. They include File transfer, Remote controller, Remote viewfinder, Video stream Remote processing and Message oriented applications. Other commercial scenarios include e.g. ship-to-ship ad-hoc mobile communication, law enforcement, etc. VANETs are a special class of MANETs as applied to vehicles on the road [32]. Vehicular ad-hoc network is composed of cars with GPS devices and, in more advanced systems, with video cameras. Cameras may acquire information about the environment, such as traffic signs, congestions, traffic accidents, and road merging information, and either report them to the base station or broadcast them to their neighbors. A number of applications have been envisioned for these networks:

- vehicle collision warning
- security distance warning
- driver assistance
- dissemination of road information

- map location

3.3.4 Applications in Local and Personal Environments

[33] Ad-hoc networks can autonomously link an instant and temporary multimedia network using notebook computers or palmtop computers to spread and share information among participants at, for example, a conference or a classroom. Another appropriate local level application might be in home networks where devices can communicate directly to exchange information. Similarly in other civilian environments like sports stadium, boat and small aircraft, mobile ad-hoc communications will have many applications.

[33] Short-range MANET can simplify the intercommunication between various mobile devices (such as a PDA, a laptop, and a cellular phone) and form a Personal Area Network (PAN). Tedious wired cables are replaced with wireless connections. Such an ad-hoc network can also extend the access to the Internet or other networks by mechanisms e.g. Wireless LAN (WLAN), GPRS, and UMTS. The PAN is potentially a promising application field of MANET in the future pervasive computing context.

Chapter 4

Query Localized Route Repair - Architecture

We have used Ad-hoc On-demand Distance Vector (AODV) as the baseline network routing protocol for our study. We have developed an improved route repair strategy for AODV to overcome its drawbacks. We describe in detail the mechanism of (QLRR) for AODV which emphasizes localization of repair. Flooding reduction is especially relevant in the context of energy efficiency and network latency. QLRR is crucial for reliable operation of the system in the face of node failures. It provides an efficient method of route healing to cope with node failure.

4.1 Repair Mechanism (QLRR)

To deal with the shortcomings in ADOV's ability to efficiently adapt to failure and mobility, we have developed a mechanism to efficiently handle route repair. We call our local repair protocol for AODV, QLRR. Our solution emphasizes localized repair in order to reduce recovery latency and routing overhead. The local repair problem can be divided into two phases: break detection, and localized gradient repair. We will describe how QLRR handles each of these phases in this section.

4.1.1 Break Detection

The first step in adapting to node failure or mobility is detecting the link breakage. We assume that appropriate algorithms may be used to reliably detect a link breakage. Our focus is on handling the break after it is detected, not adaptively detecting path breakage.

AODV [11] protocol suggests one of the following two mechanisms for detecting broken links. First is the use of periodic hello messages (unsolicited special RREP packets) the absence of which indicates link failures. Secondly and with far less control overhead, such failures could be detected by using link-layer acknowledgments. A link failure is also indicated if attempts to forward a packet to the next hop fail. We use the link-layer feedback supported by 802.11 to lower the amount of broadcasts compared to hello messages.

4.1.2 Localized Route Repair

Once link breakage has been detected by an upstream node, it handles the erroneous situation locally. Further maintenance action to be taken by the node depends on whether or not the broken link was part of an end-to-end route. Both the situations are further explained below with reference to Figure 4.1.

The Figure 4.1(a) shows an ad-hoc network with one source S and two sinks P and Q. There exists an active route carrying data packets from S to Q but no active routes from S to P. Figure 4.1(b) shows a scenario where nodes X and W have moved out of range of V there by disconnecting routes V-P and V-Q. This in turn, breaks the end-to-end path from S to Q and S to P. The up stream node V detects the link breakage via LLACKS and initiates Query Localized Route Repair. The specific actions taken by V differs for each end to end path. The two cases are when:

- The broken link is not used by any active route: The actions taken in this situation are no different from the default protocol. As no routes get affected the upstream updates its list of neighbors, removing the node at the other end of the broken link. Any further route requests to the nodes downstream the broken link will now be

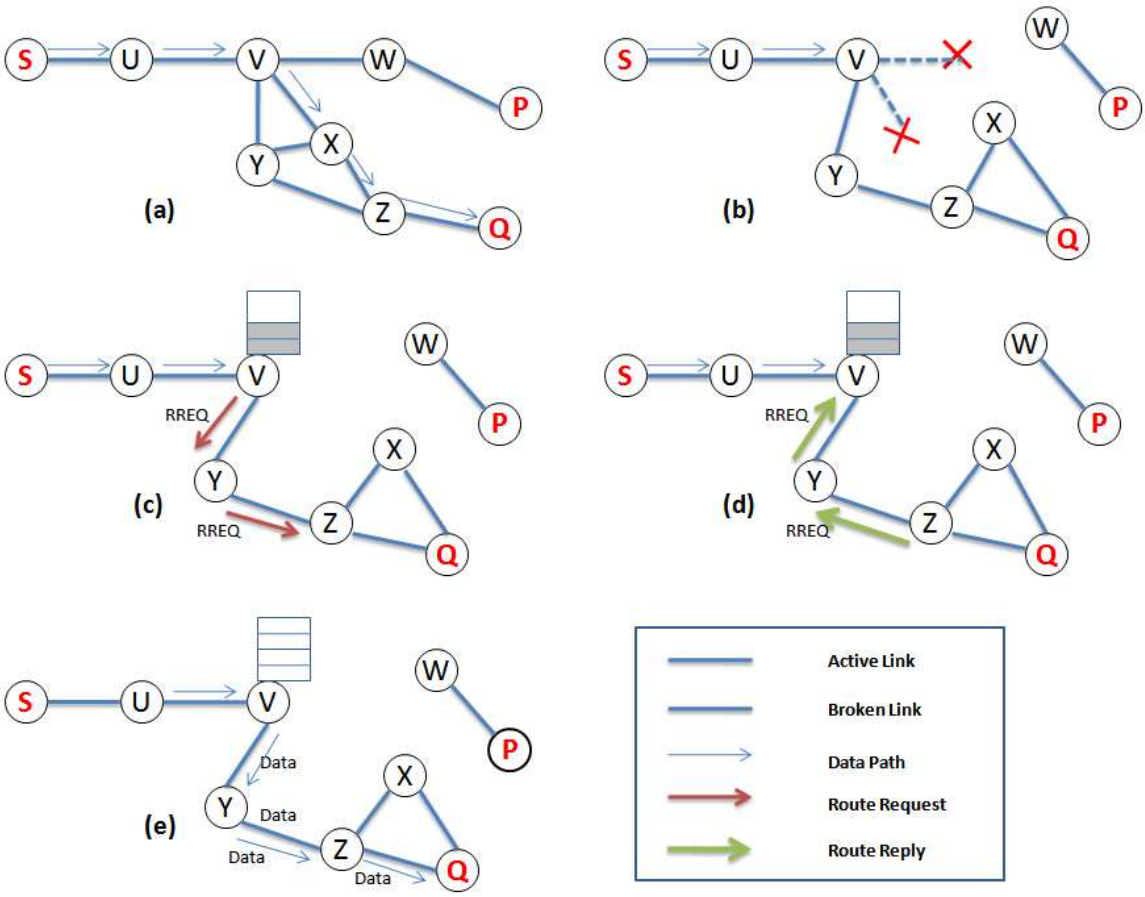

Figure 4.1: Example Scenario for QLRR

unreachable through this link. This situation is the same as if the link never existed. This is depicted by Figures [11](c) and (d).

- The broken link is used by at least one active route: Since transport-level reliability is the main goal of QLRR, the upstream neighbor buffers the incoming packets so they can be dispatched in order once the route is recovered. Dropping them reduces the overall delivery ratio making the protocol unreliable. However since the nodes might

have limited buffer capacity, the size of the buffer is made a configurable parameter which can be used to fine-tune on a case-to-case basis. Figure [11] (c) depicts buffering the incoming payload while it initiates QLRR. Node V constructs a new Route Request (RREQ) packet with itself as the source and the destination same as the destination of the route that uses the broken link. In this case the destination is Q. In general this action is taken for each affected route in the routing table of node V.

Optimized Route Request Flooding

In order to restrict the flooding required to find new paths to the destination, we limit the packet forwarding by hop-limited flooding. In this approach, the time to live field of new RREQs is incremented at each hop. When the hop count exceeds a threshold value, the RREQs are dropped. As noted in [22] the possibility of repairing the current broken route should be reasonably high in most cases and the resulting overhead should be much lower than using end-to-end global recovery.

Reconnecting Affected Routes

Once the RREQs are dispatched the upstream node waits for a timeout period as predefined in AODV protocol. As the RREQ travels from an upstream node V to various destinations, it automatically sets up the reverse path from all nodes back to that node. These reverse path route entries are maintained for at least enough time for the RREQ to traverse the network and produce a reply to the sender. This process is similar to the Route Discovery phase of the default AODV protocol [11]. Eventually when the RREQ packets either reach the destination or a node that contains a valid route to the destination, a RREP is generated and is sent back to the source i.e. the upstream node V of the broken

link. In Figure 4.1(d), node Z, which is the immediate neighbor of the destination Q, generates a RREP back to V. The upstream node V then sends a gratuitous reply back to the original source. That completes the route repair process. With the end-to-end route now repaired, the payload data packets that were buffered will now be routed successfully to the destination.

Chapter 5

Query Localised Route Recovery - Design

5.1 Design Principles

In designing the network services, we were guided by several principles. Our primary design goals were scalability, localized and distributed nature, and on-demand behavior. In this section, we describe each of these design goals and explain their implications on our design decisions.

5.1.1 Scalable

A general goal of our system and any MANET is scalability. Routing protocols should scale up to networks with large numbers of nodes. As the network size increases, these unneeded communications lead to network congestion. In sufficiently large networks, flooding-induced congestion can completely cripple the network. Hence, efficient flooding is vital to the robust operation of large-scale networks. The default global route repair mechanism of AODV involves global flooding of RREQ packets which greatly limits the scalability of the protocol since flooding is so costly for large networks. QLRR addresses the goal of scalability by reducing unnecessary transmissions.

5.1.2 Localized

Localized protocols maintain information about one-hop neighbors. This significantly reduces the amount of state information that must be stored and hence, improves the scalability and simplicity of the algorithm. By decreasing the complexity of an algorithm,

the principle of localization simplifies the design and implementation. QLRR supports the goal of localized interactions by localizing its repair work to the area immediately surrounding the failed link. One of the best properties of AODV is that the nodes maintain information about their neighbors by means of HELLO messages which are localized in nature. Therefore, the advantages of localized interactions are preserved.

5.1.3 Distributed

The distributed nature of sensor networks is one of their most challenging and powerful characteristics. The goal is to fully distribute algorithms over all the nodes in the network, in contrast to typical algorithms which utilize the client/server model. Our protocols address this challenge by distributing tasks among all the nodes. In QLRR, any node may potentially detect and repair a breakage. In this sense, all nodes act as adaptation servers forming a completely distributed adaptation service. QLRR distributes the responsibility for repair among all the nodes.

5.1.4 On-demand

Our final design goal is to preserve the reactive nature of AODV protocol. AODV being a reactive protocols is more suitable to ad-hoc networks than proactive protocols since they are more energy efficient. Reactive protocols respond to events instead of proactively maintaining state information ahead of time. QLRR responds to repair broken links by reactively repairing them.

5.2 Design Decisions

5.2.1 Repair Mechanism (QLRR)

HELLO Messages versus MAC layer feedback

AODV with only MAC layer support will not get the routes to the neighbors installed in the routing tables, neither will it update the routes to the neighbors who forwarded packets to the current node. On the contrary, with HELLO messages, nodes will have up-to-date information about neighbors and can keep track of connectivity to neighbors. Without the HELLO messages, a node which receives a MAC layer failure feedback will have to buffer the data packets while it initiates route rediscovery. This will increase the delivery latency for the end-to-end connection. With the HELLO messages, however, the route rediscovery process is initiated irrespective of whether there is a packet waiting to be transmitted. This is done for all the active connections through the upstream node of the broken link. Although buffers are still required, the delivery latency will be reduced. However using HELLO messages increases the load on the network substantially [34]. Since we know that wireless transmissions are expensive we wanted to reduce such periodic transmissions. Therefore we have used link layer feedback our mechanism.

Local Flooding versus Full fledged Route Discovery

In order to speed up route repair, the upstream node initiates a route discovery procedure by itself, instead of banking on the originating node to repair the route. This saves time involved in generating route error packets and sending them to the source node. Another serious problem with this approach stems from the possibility that the new route

may comprise entirely of a different set of nodes. This means all the data packets that are enroute the failed path will be dropped, therefore decreasing the overall delivery ratio. Also if link failure happens at the end of a long route, then the possibility of the RREQ packet getting delivered itslef might pose a problem eventually timing out route entries. But this will cause a definite loss of data since during this window of time the source node is still pumping in data packets into the route. On the other hand in our approach, by buffering data packets that are en route, at the upstream node, delivery of all the packets that were dispatched from the source node is ensured.

Chapter 6

Implementation and Simulation Study

6.1 AODV-QLRR Implementation

In this section we discuss the implementation details of AODV-QLRR protocol. The default version of AODV that is shipped with ns-2 is not compliant with the latest IETF draft of AODV protocol, RFC 3561. AODV-UU (Ad-hoc On-demand Distance Vector Routing, from Uppsala University) [35] is a routing protocol under investigation by the IETF for use in ad-hoc networks, where both end-users and routers are mobile. This implementation supports IPv6 and multicasting (with the appropriate patch) and is compliant with RFC 3561. Performance of AODV-UU in ns-2 is very similar to the existing ns-2 AODV implementation. The main difference is that AODV-UU provides bug-free implementation of HELLO messages and expanding ring search.

QLRR Extensions to AODVUU

We extended the code of AODVUU to add QLRR functionality as per algorithm mentioned below. The extension mainly involves introduction of a new state for routing entries and appropriate handling of data and control packets belonging to those routing entries.

Routing State REPAIR By default AODV-UU defines the following two routing states for each route in the routing table of every participating node in the ad-hoc network:

- VALID - This state marks the route as a valid route. Data packets using this route will be forwarded to the next-hop node as indicated by the routing entry.

- INVALID - This state marks the route as an invalid route. When data packets using this route arrive at the current node, they get dropped.

According to default AODV protocol a transition from VALID to INVALID state triggers sending of RREQ packets back to the source of route. The problem with this approach is that all the packets that were induced into the network in the time interval between sending of RREQ packets and reception of the same by the source node, are dropped by the upstream nodes in that route. This hampers the overall delivery ratio from the applications point of view. We solve the issue effectively by introducing a new routing state called REPAIR. A node enters this state when it receives a link-layer-send failure feedback. Following Algorithm 1 depicts the actions that are taken when a node receives a delivery failure feedback from the link layer.

Algorithm 1: QLRR Initiation

if *Link Layer Failure* **then**
    Change routing entry from VALID TO REPIAR
    Initiate recovery timer
    Initiate Route Recovery for every route with broken link as the next hop
    Initialize packet ttl to threshold value
    Create a routing recovery buffer for each destination that requires use of broken link

Following Algorithm 2 depicts the actions taken when a node receives a route repair packet.

Algorithm 2: QLRR Propagation

```
if Action = Forward Packet then
    if Current node is final destination then
        Send Route Reply back to the source
    else if Next hop routing entry state is VALID then
        Send Route Reply back to the source
    else if Packet TTL = 0 then
        Drop Route Request packet
    else if Next hop routing entry is either invalid Or not found then
        Decrement TTL by 1
        Forward Routing Request packet
```

6.2 Simulation Environment

The simulator we have used to simulate the ad-hoc routing protocols is the Network simulator 2 (ns) [36] from Berkeley. To simulate the mobile wireless radio environment we have used mobility extensions to ns that is developed by the CMU Monarch project at Carnegie Mellon university.

6.2.1 Network Simulator

Network simulator 2 is the result of an on-going effort of research and development that is administrated by researchers at Berkeley. It is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols. The simulator is written in c++ and a script language called OTcl [1]. Ns uses an OTcl interpreter towards the user. This means that the user writes an OTcl script that defines the network (number of nodes, links), the traffic in the network (sources,

[1] Object Tool Command Language

destinations, type of traffic) and which protocols it will use. This script is then used by ns during the simulations. The result of the simulations is an output trace file that can be used to do data processing (calculate delay, throughput etc) and to visualize the simulation with a program called Network Animator (NAM). An overview of how a simulation is done in ns is shown in Figure 6.1.

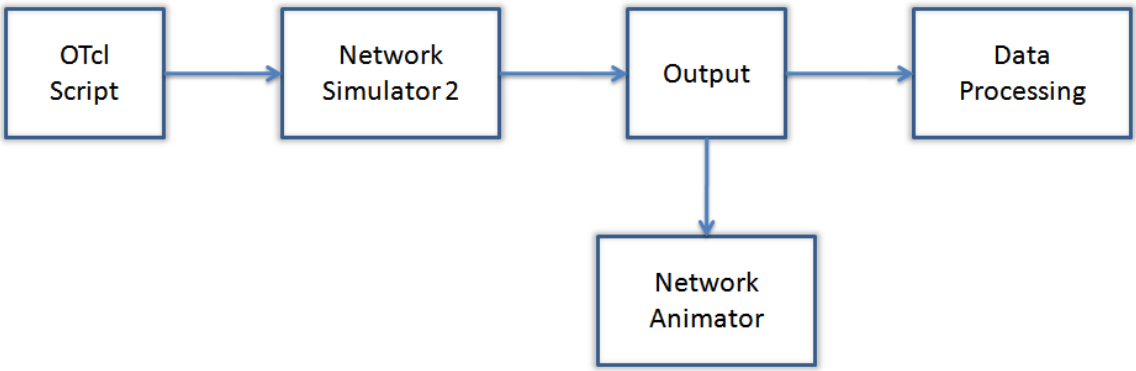

Figure 6.1: Network Simulator 2

The version 2.29 of the Network simulator supports simulation of mobile wireless environments. The Network simulator alone was initially only intended for stationary networks with wired links. Researchers at CMU's Monarch group realized the need for mobility models in ns and therefore started to design and implement a mobility model that would extend the simulator. The CMU Monarch extensions to ns provide new elements at the physical, link, and routing layers of the simulation environment. Using these elements, it is possible to construct detailed and accurate simulations of wireless subnets, LANs, or multi-hop ad-hoc networks. Recent versions of ns contain further extensions to this model to allow combined simulation of wired and wireless networks. The following section provide an overview of the extensions added to ns.

6.2.2 Mobility Extensions

Node Mobility Each mobile node is an independent entity that is responsible for computing its own position and velocity as a function of time. Nodes move around according to a movement pattern specified at the beginning of the simulation.

Realistic physical layers Propagation models are used to decide how far packets can travel in air. These models also consider propagation delays, capture effects and carrier sense [2].

MAC 802.11 An implementation of the IEEE 802.11 Media Access Protocol (MAC) [37] protocol was included in the extension. The MAC layer handles collision detection, fragmentation and acknowledgments. This protocol may also be used to detect transmission errors. 802.11 is a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol. It avoids collisions by checking the channel before using it. If the channel is free, it can start sending, if not, it must wait a random amount of time before checking again. For each retry an exponential back off algorithm will be used. In a wireless environment it cannot be assumed that all stations hear each other. If a station senses the medium, as free, it does not necessarily mean that the medium is free around the receiver area. This problem is known as the hidden terminal problem and to overcome these problems the collision avoidance mechanism together with a positive acknowledgment scheme is used. The positive acknowledgment scheme means that the receiver sends an acknowledgment when it receives a packet. The sender will try to retransmit this packet until it receives the acknowledgment or the number of retransmits exceeds the maximum number of retransmits. 802.11 also support power saving and security. Power saving allows packets to be buffered

even if the system is asleep. security is provided by an algorithm called Wired Equivalent Privacy (WEP). It supports authentication and encryption. WEP is a Pseudo Random Number Generator (PRNG) and is based on RSA's RC4. One of the most important features of 802.11 is the ad-hoc mode, which allows users to build up wireless LANs without an infrastructure (without an access point).

Address Resolution Protocol The Address Resolution Protocol, ARP [38] is implemented. ARP translates IP-addresses to hardware MAC addresses. This takes place before the packets are sent down to the MAC layer.

Radio network interfaces This is a model of the hardware that actually transmits the packet onto the channel with a certain power and modulation scheme [2].

Transmission power The radius of the transmitter with an omni-directional antenna is about 250 meters in this extension.

Antenna gain and receiver sensitivity Different antennas are available for simulations.

Shared Media

The wireless extensions in ns2 are based on a shared media model (Ethernet in the air). This means that all mobile nodes have one or more network interfaces that are connected to a channel (see Figure 6.2). A channel represents a particular radio frequency with a particular modulation and coding scheme. Channels are orthogonal, meaning that packets sent on one channel do not interfere with the transmission and reception of packets on another channel. The basic operation is as follows, every packet that is sent (i.e. put on the

channel) is received (i.e. copied to all mobile nodes) connected to the same channel. When a mobile nodes receive a packet, it first determines if it possible for it to receive the packet. This is determined by the radio propagation model, based on the transmitter range, the distance that the packet has traveled and the amount of bit errors.

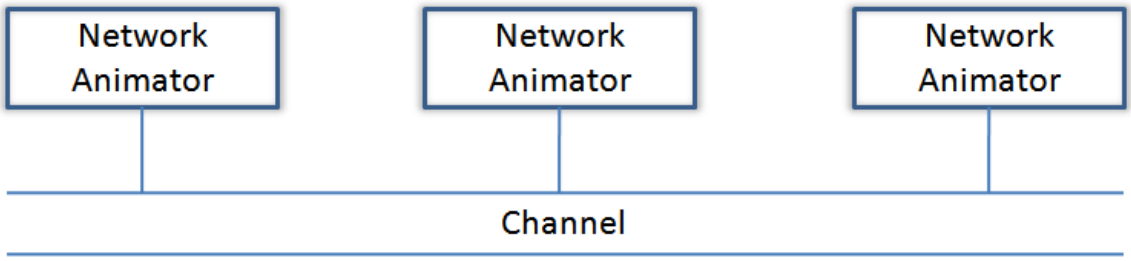

Figure 6.2: Shared Media Model in ns2

Mobile Node

Each mobile node (Figure 6.3) makes use of a routing agent for the purpose of calculating routes to other nodes in the ad-hoc network. Packets are sent from the application and are received by the routing agent. The agent decides a path that the packet must travel in order to reach its destination and stamps it with this information. It then sends the packet down to the link layer. The link layer level uses an Address Resolution Protocol (ARP) to decide the hardware addresses of neighboring nodes and map IP addresses to their correct interfaces. When this information is known, the packet is sent down to the interface queue and awaits a signal from the Multiple Access Control (MAC) protocol. When the MAC layer decides it is ok to send it onto the channel, it fetches the packet from the queue and hands it over to the network interface which in turn sends the packet onto the radio channel. This packet is copied and is delivered to all network interfaces at the time at which the first

bit of the packet would begin arriving at the interface in a physical system. Each network interface stamps the packet with the receiving interfaces properties and then invokes the propagation model. The propagation model uses the transmit and receive stamps to determine the power with which the interface will receive the packet. The receiving network interfaces then use their properties to determine if they actually successfully received the packet, and sends it to the MAC layer if appropriate. If the MAC layer receives the packet error-free and collision-free, it passes the packet to the mobiles entry point. From there it reaches a demultiplexer, which decides if the packet should be forwarded again, or if it has reached its destination node. If the destination node is reached, the packet is sent to a port demultiplexer, which decides to what application the packet should be delivered. If the packet should be forwarded again the routing agent will be called and the procedure will be repeated.

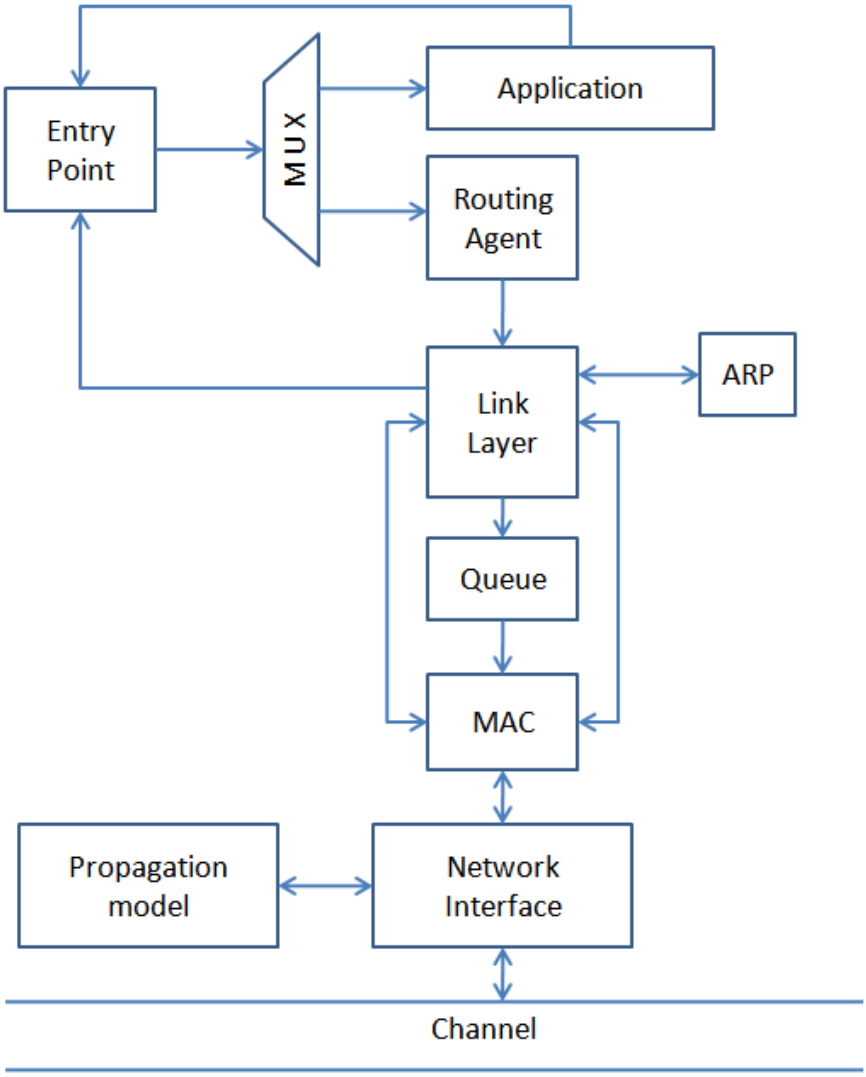

Figure 6.3: Mobile Node

6.2.3 Simulation Overview

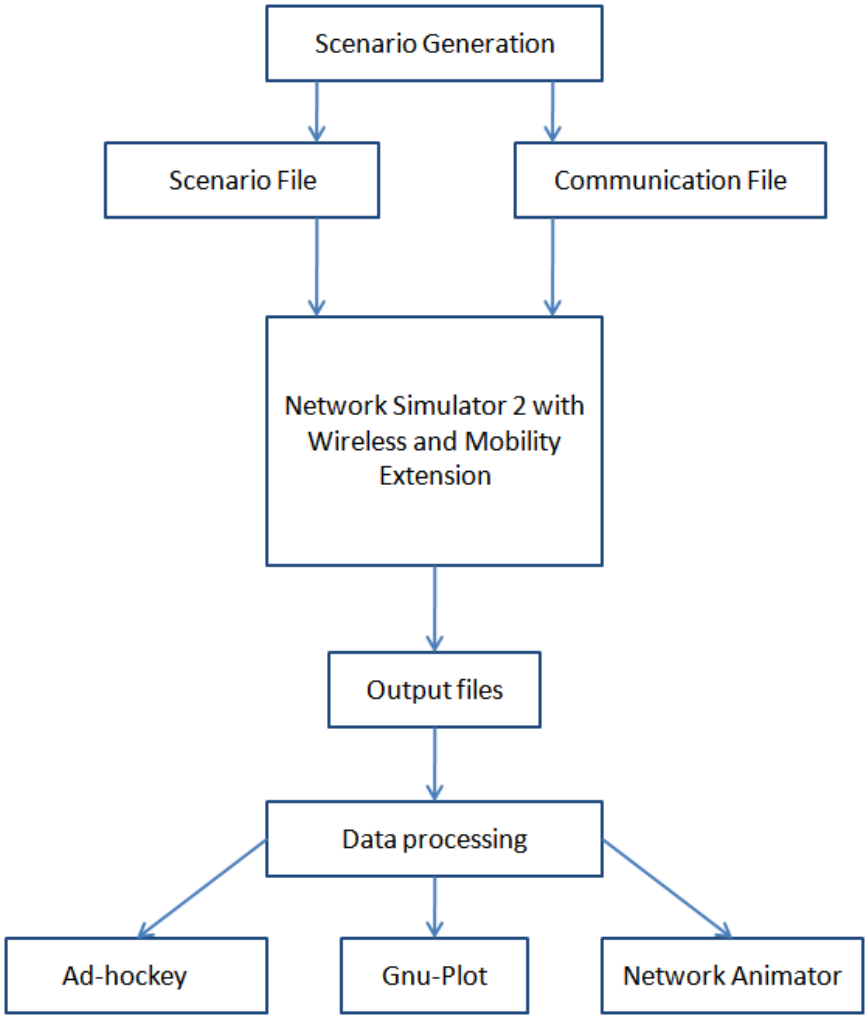

Figure 6.4: NS Simulation Overview

A typical simulation with ns and the mobility extension is shown in Figure 6.4. Basically it consists of generating the following input files to ns:

- A scenario file that describes the movement pattern of the nodes.
- A communication file that describes the traffic in the network.

These files can be generated by drawing them by hand using the visualization tool Ad-hockey or by generating completely randomized movement and communication patterns with a script. We used the scripts provided as part of ns2 since we found this method faster. These files are then used for the simulation and as a result from this, a trace file is generated as output. Prior to the simulation, the parameters that are going to be traced during the simulation must be selected. The trace file can then be scanned and analyzed for the various parameters that we want to measure. This can be used as data for plots with for instance Gnuplot. The trace file can also be used to visualize the simulation run with either Ad-hockey or Network animator.

6.3 Simulation Study

We simulated AODV-QLRR and default AODV protocol as per rfc3561 from here on referred to as AODV-RFC3561. We did so to get a comparison of how much better/worse the QLRR performs than the protocol with default route repair mechanism. The simulations were conducted on an Intel(R) PC with a Xeon(TM) processor at 1700 MHz, 512 Mbytes of RAM running Fedora Core 8.0.

6.3.1 Measurements

Before we go into the actual simulations, we will discuss the parameters [39] that are interesting to measure when studying routing protocols in an ad-hoc network. There are two main performance measures that are of substantial interest as far as efficient repair mechanisms in routing protocols are concerned: the average end-to-end delivery-ratio and the average normalized routing overhead.

Quantitative Metrics

The measurements that we have conducted can be seen from two angles: externally and internally. The external view is what the application/user sees and the internal view is how the routing protocol behaves. The external measurement is basically the end-to-end delivery ratio. The internal behavior can be gauged by measuring routing efficiency i.e normalized routing overheard – it is the number of routing packets transmitted per data packet sent to destination. Each forwarded packet is counted as one transmission. This metric is highly correlated with the number of route changes occured in the simulation.

Parameters

The metrics has to be measured against some parameter that describes the characteristic behavior of an ad-hoc network and can be varied in a controlled way. One of the advantages of simulation is that it allows us to do so. The parameters that we have chosen to simulate with are:

- Mobility, which probably is one of the most important characteristics of an ad-hoc network. This will affect the dynamic topology, links will go up and down.
- Offered network load. The load that we actually offer the network. This can be characterized by three parameters: packet size, number of connections and the rate that we are sending the packets with. For our simulations we vary the number of connections.
- Network size (number of nodes and the geographical size of the area that the nodes are moving within). The network size basically determines the connectivity. Fewer nodes

in the same area mean fewer neighbors to send requests to, but also smaller probability for collisions. Since routes in AODV are based on hop counts, it makes sense to test it for different average route lengths than with increasing size of the geographical area. Therefore we have used varying number of nodes in a given geographical area as the network size parameter.

6.3.2 Simulation Setup

In this chapter we will describe how the simulations were done. We have done 3 different types of simulations:

- Mobility simulations: we vary the mobility to see how it affects the different metrics that we are measuring. In order to test performance thoroughly we conducted simulations with varying speed as well as varying pause-time between movements.
- Offered load simulations: we vary the load that we offer the network, to see how the protocols behave when for instance the load is high.
- Network size simulations: we vary the number of nodes in the network. The geographical area itself was fixed in order to make sure the density of nodes is varied.

The choice of AODV with link layer support over hello messages was made because first of all, link layer support is probably a necessity to achieve a performance that is good enough and secondly because the removal of hello messages somewhat changes the overall functionality of AODV. Removal of hello messages would not only save us from the overhead of the hello messages, but also makes the protocol completely on-demand. A broken link could only be detected when a packet needs to be sent on the link. In all simulations we

have used randomized scenarios. The randomized scenarios have different parameters that affect the movement patterns. The parameters that can be changed are:

- Maximum speed: Every time a speed is going to be randomized, it is randomized in the interval [0,maximum speed].
- Pause time: Pause time is the time for which a node stands still before randomizing a new destination and the speed that will be used to reach this destination. We have used a pause time for 1 second in all simulations.
- Number of nodes: This was constant during the simulations. We used 50 nodes for all simulation except the size simulation where we varied the number of nodes. This parameter determines the density of mobile nodes in a given area of dimensions 1000 x 1000 meters.
- Simulation time: The time for which the simulations will be run. We have used a simulation time of 900 seconds for all simulations. However the first 200 seconds of each simulation run were discounted from measurement of metrics. This is to allow the network queues to stabilize before we can get reliable values for metrics.

Randomizing of scenarios . First of all every node stands still for pause time seconds. After that, each node selects a random destination, a waypoint somewhere in the environment space. Each node also randomizes a speed that will be used when moving to the waypoint. This speed is randomized uniformly in the interval 0 to maximum speed. Every time a node reaches a waypoint, this procedure will be repeated. Note that a factor that we have not taken into consideration with the scenarios is the fact that a real person is not likely to stand on the same place if the connection goes down. A real person is more likely to find

a place where the reception is good enough. The system would be too complex if this factor were included too. We have assumed bi-directional links during all our simulations, i.e. the links work equally well in both directions. However it is questionable whether unidirectional links are desirable when using the IEEE 802.11 MAC protocol, because bi-directional links are necessary if 802.11 acknowledgments are supposed to be used.

6.4 Mobility Simulations

6.4.1 Setup

The simulations where we varied the mobility were done by taking input from scenario files generated using scenario generation tools provided by ns2. In order to simulate varying mobility pattern we generated scenario files for ns2 first by controlling the maximum speed parameter and then by varying the pause time between successive movements of a node. The simulation parameters that have been used for the mobility simulations with varying speed are shown in Table 6.1.

Table 6.1: Mobility Simulation Parameters

Parameter	**Value**
Transmitter range	250.10 meters
Bandwidth	2 Mbps
simulation time	900 s
Number of nodes	50
Pause time	0 ms
Environment size	1000m X 1000m
Traffic type	Constant Bit Rate
Packet rate	4 packets/sec
Packet size	64 KB
Number of flows	15

The simulation parameters that have been used for the mobility simulations with varying pause time are the same as shown in Table 6.1.

The scenarios are a very crucial part of the simulation. We have therefore collected 10 measurements for each wanted value of both maximum speed and pause time. The scenarios that where created were then analyzed in terms of the metrics mentioned in Section 6.3.1. By increasing the value for maximum speed in the scenario generation scripts, the amount of mobility is also increased thereby increasing chances of disconnection. We have varied the maximum speed from 1 m/s to 25 m/s. A speed of 25 m/s corresponds the speed of a vehicle, which will lead to a high mobility. We used the same communication pattern for all mobility simulations. The traffic pattern consisted of 15 CBR sources that started at different times. We did not use TCP for the simulations, because we did not want to

investigate TCP, which uses flow control, retransmit features and so on. We wanted to get a general view of how AODV-QLRR behaves. The communication pattern was again randomly created using tools provided by ns-2. The parameters that were specified when randomizing the communication pattern were the number of wanted sources, the packet size, the rate at which they were sending and the simulation time. In these simulations, we wanted to investigate how the mobility affected the protocols, so the load that we offer is not high. We only use 15 CBR sources sending 64-byte large packets with a rate of 4 packets/s. The bandwidth of the links are 2 Mbit.

6.4.2 Simulation Results

In this section we discuss the results for network size simulations with parameters shown in Table 6.1.

Routing Overhead

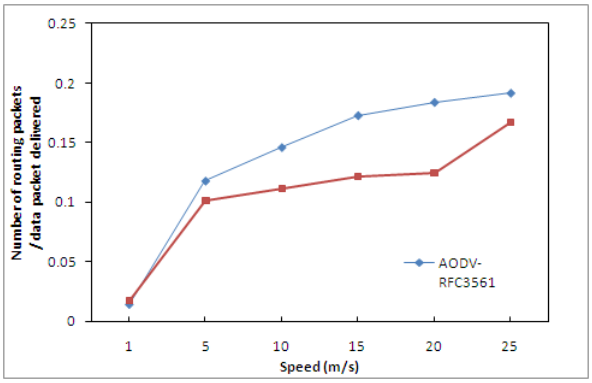

Figure 6.5: Speed versus Routing Overhead

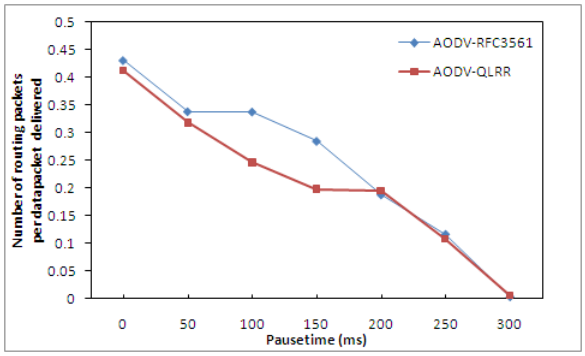

Figure 6.6: Pause time versus Routing Overhead

Figures 6.5 and 6.6 shows routing overhead of AODV-QLRR in comparison with the default AODV protocol for varying degrees of node mobility. We have tested the protocol first by varying the speed with which the nodes move and also by varying the pause time between node movements. One of the main issues in mobile ad-hoc networks is link breakage due to the mobile nature of the network nodes. Every time a link breakage occurs, local recovery kicks in to repair the end-to-end route. This amounts to overhead as no data gets delivered while the link is being repaired. We can see that the average routing overhead keeps increasing as speed increases or pause time between movements decreases. It can be noticed that at low density, AODV-QLRR does not show much advantage over default ADOV. When the speeds are less than 5 m/s and pause time more than 200 ms, the network links are less susceptible to breakage due to mobility and therefore chances of local recovery is fairly even for both default AODV and AODV-QLRR. Notice that this fact is reflected by high values for delivery ratio depicted in Figures 6.7 and 6.8 for both protocols. However, as the mobilility increases, the link failure rate also increases. In such a situation, the local recovery comes into play to a greater degree. We see that in such cases, routing overhead

is reduced by as much as 32.3%. Since AODV-QLRR broadcasts route repair requests to a threshold-limited hop count, chances of route repair by this mechanism are less leading to fewer delivered packets when compared to the default local recovery scheme of AODV. We also notice an improvement in delivery ratio for AODV-QLRR by about 7.4%.

Delivery Ratio

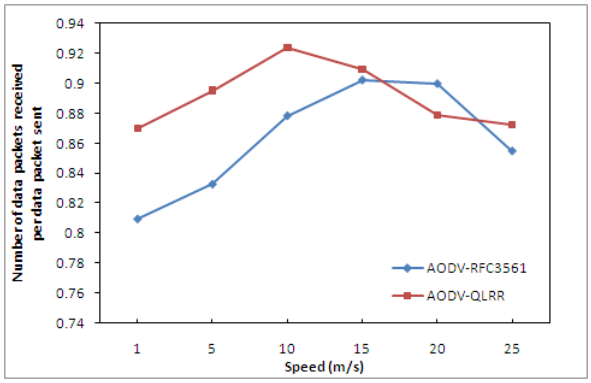

Figure 6.7: Speed versus Delivery Ratio

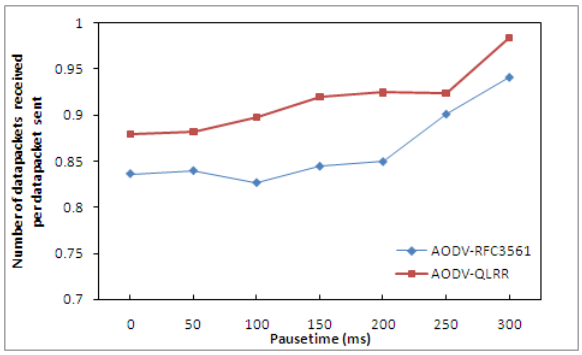

Figure 6.8: Pause time versus Delivery Ratio

Figures 6.7 and 6.8 show routing overhead of AODV-QLRR in comparison with the default AODV protocol. One of the main issues in mobile ad-hoc networks is link breakage due to the mobile nature of the network nodes. We have tested the protocol first by varying the speed with which the nodes move and also by varying the pause time between node movements.We notice that lower speeds of upto 5 m/s and pause times of above 250 ms do not have a significant degradation in delivery ratios for both protocols. For speeds above 20 m/s the delivery ratios drops by about 1.9% for AODV-QLRR and 2.6% for default AODV. We notice that on an average AODV-QLRR improves on delivery ratio by about 3.4%. This is quite acceptable considering that AODV-QLRR incurs 22% less routing overhead compared to the default AODV.

Route Length

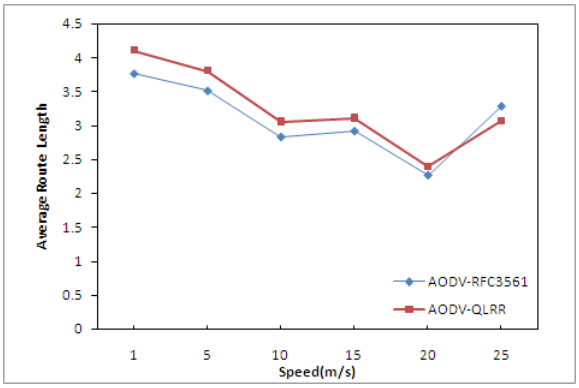

Figure 6.9: Speed versus Route Length

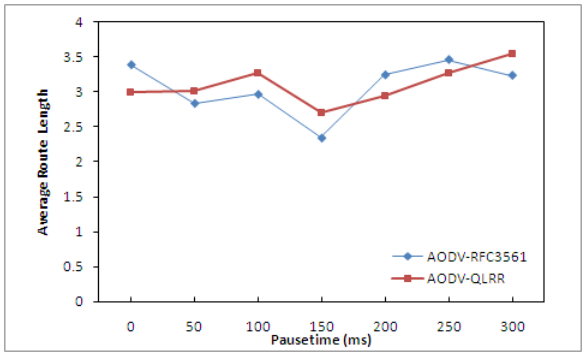

Figure 6.10: Pause time versus Route Length

Figures 6.9 and 6.10 show routing length of AODV-QLRR in comparison with the default AODV protocol. Routing length is an indicator of end-to-end delay experienced by transport and application higher layer entities. Although the average route length for AODV-QLRR is higher than default AODV, the values are still comparable.

6.5 Offered Load Simulations

6.5.1 Setup

The offered load simulations were done by varying the load that was offered to the network. We mainly focused on the total number of CBR flows to adjust the offered load. Th rate at which the flows are sending packets and the size of the packets used were kept constant. Since we used a fairly moderate offered load for mobility simulations, we wanted to investigate how the protocols behave when the load was increased. By varying the number of CBR flows, we could control the overall number of data packets that are in the ad-hoc network at any point in time. We did not study the behavior with varying packet sizes mainly because the packet as a unit that is of interest to the routing layer and not really

the size of the packet which is critical to the link layer. This is mainly because fragmentation and defreagmentation is generally a function of link layer while the routing layer is quite transparent to it. We have used four different offered load cases with the number of flows varying from 5 to 25. The packet size was held constant at 64 bytes and the flow rate as 4 packets/second. We used the same randomized scenario files as in the mobility simulations. The parameters that we used during the offered load simulation are shown in Table 6.2.

Table 6.2: Offered Load Simulation Parameters

Parameter	**Value**
Transmitter range	250.10 meters
Bandwidth	2 Mbps
simulation time	900 s
Number of nodes	50
Maximum Speed	20 m/s
Pause time	0.00 s
Environment size	1000m X 1000m
Traffic type	Constant Bit Rate
Packet rate	4 packets/sec
Packet size	64 KB

6.5.2 Simulation Results

In this section we discuss the results for network size simulations with parameters shown in 6.2.

Routing Overhead

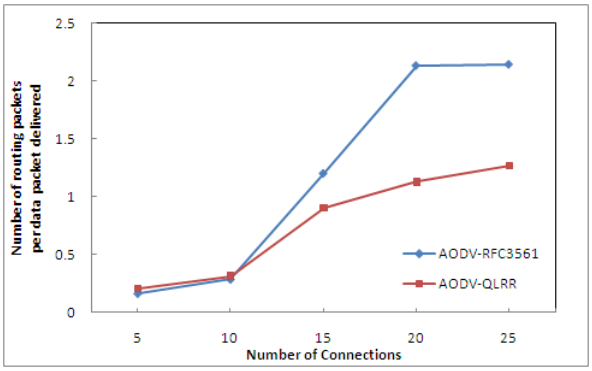

Figure 6.11: Offered Load versus Routing Overhead

Figure 6.11 shows routing overhead of AODV-QLRR in comparison with the default AODV protocol. We can see that the average overhead keeps increasing as the node density increases. It can be noticed that for low intensity traffic, AODV-QLRR does not show much advantage over default ADOV. When number of end-to-end flows is more than 15, there is a drastic increase in the number of packets that each node is trying to send out at any given point in time. This increases contention for the media leading to more collisions. This in turn is reflected within each node by increasing number of failure feedbacks by the link layer to the routing protocol. Since both the versions of AODV protocols that we have considered use link layer feedback for route repair, we see an increased number of route repair requests being initiated. In such a situation, the QLRR comes into play to a greater degree bringing in an obvious advantage over default recovery scheme, as the AODV-QLRR floods the request only in a small region while the default local recovery scheme in AODV floods the entire network. As load increases further, the overall savings in routing overhead

becomes very obvious, and reaches as high as 47% for 20 flows, compared with default AODV.

Delivery Ratio

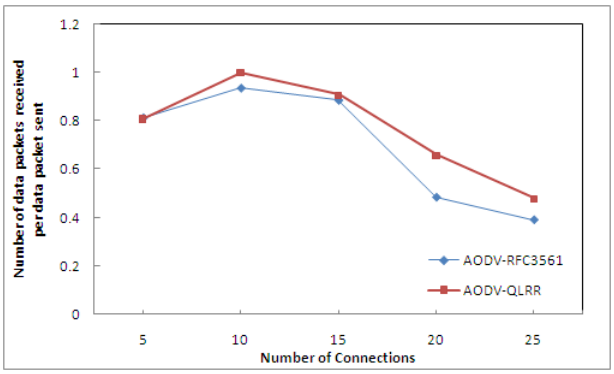

Figure 6.12: Offered Load versus Delivery Ratio

Figure 6.12 shows delivery ratio of AODV-QLRR in comparison with the default AODV protocol. For the reason mentioned in the previous section, the effectiveness of our protocol in delivering packets decreases with increasing number of connections. This is because as the number of flows increases, the number links that are used in the network increases. As more links are being used, the probability of a link breakage affecting the data flow also increases, leading to loss of data packets. We notice that for up-to 5 connections, the delivery ratio is still acceptable. However for more number of connections deliver ratio drops to much lower values. We notice that on an average AODV-QLRR sacrifices delivery ratio only by about 13.4%. This is quite acceptable considering that AODV-QLRR incurs 47% less routing overhead compared to the default AODV.

Route Length

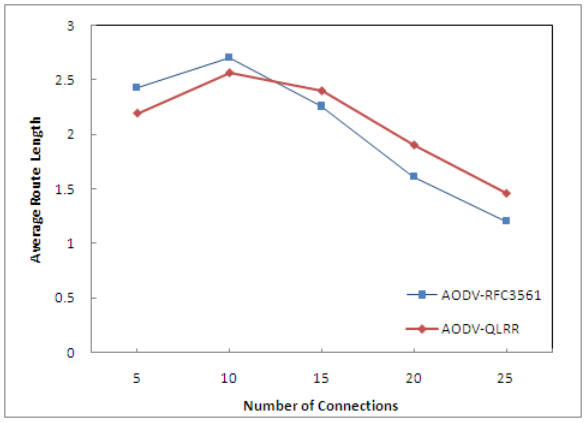

Figure 6.13: Offered Load versus Route Length

Figure 6.13 shows routing length of AODV-QLRR in comparison with the default AODV protocol. Routing length is an indicator of end-to-end delay experienced by transport and application layer entities. Although the average route length for AODV-QLRR is higher than default AODV, the values are still comparable.

6.6 Network Size Simulations

We did simulations by varying the number of nodes that participated in the network. The purpose of this simulation is to study the scalability of the routing protocol with QLRR. We increased the number of nodes to 5 and 25 nodes. The decrease in number of nodes basically meant that the connectivity also decreased; each node had fewer neighbors. Decreased connectivity meant of course that we did not get as many packets through the network as in the mobility simulations. The worst results for each simulation happened when the mobility was 0. The reason for the bad result when the nodes is standing still is

the randomized scenarios. If a randomized scenario has poor connectivity, this connectivity will be the same during the whole simulation if the nodes are standing still, especially since the same scenarios are used for all the protocol variations under study. The nodes are not moving and cannot therefore affect the connectivity. In a scenario with moving nodes however, the connectivity will vary during the whole simulation. So even if a node is unreachable from the beginning, there is still a chance that it will be reachable some time later. Increasing the number of nodes however increases the average hop count, therefore increasing the number of link breaks in an end-to-end route. This initiates route repairs and therefore the capability of the repair mechanism to recover becomes evident in this study. A drop in the delivery ratio implies that the routing protocol has an inefficient route repair mechanism.

6.6.1 Setup

Table 6.3 summarizes the simulation parameters used for network size simulations.

Table 6.3: Network Size Simulation Parameters

Parameter	**Value**
Transmitter range	250.10 meters
Bandwidth	2 Mbps
simulation time	900 s
Pause time	0.0 s
Maximum speed	20.0 m/s
Environment size	1000m X 1000m
Traffic type	Constant Bit Rate
Packet rate	4 packets/sec
Packet size	64 KB
Number of flows	15

6.6.2 Simulation Results

In this section we discuss the results for network size simulations with parameters shown in 6.3.

Routing Overhead

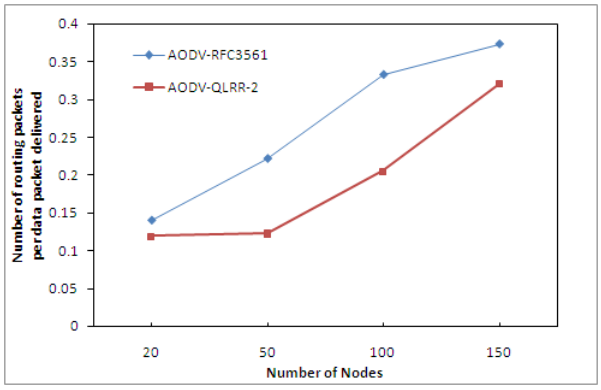

Figure 6.14: Network Size versus Routing Overhead

Figure 6.14 shows routing overhead of AODV-QLRR in comparison with the default AODV protocol. We can see that the average overhead keeps increasing as the node density increases. It can be noticed that at low density, AODV-QLRR does not show much advantage over default ADOV. When number of nodes is around 20, the connectivity of the whole network is not quite good and partitioning is severe. Notice that this fact is reflected by low values for delivery ratio depicted in Figure 6.15. Most of the transmission is successful only in small partitions with short route length. In such situation, the local recovery covers most portion of the whole partition anyway. There is no real scope for query localization, thus we cannot see obvious control packet saving at low density. However, as

the density goes higher, the connectivity of the network becomes higher, transmission with longer route length can be formed. In such a situation, the local recovery comes into play to a greater degree bringing in an obvious advantage over end-to-end recovery scheme, as the AODV-QLRR floods the request only in a small region while the default local recovery scheme in AODV floods the entire network. As successful local recovery ratio goes higher as the average degree of node goes higher, the overall savings in routing overhead becomes very obvious, and reaches as high as 44% at high density (for 150 nodes), compared with default AODV.

Delivery Ratio

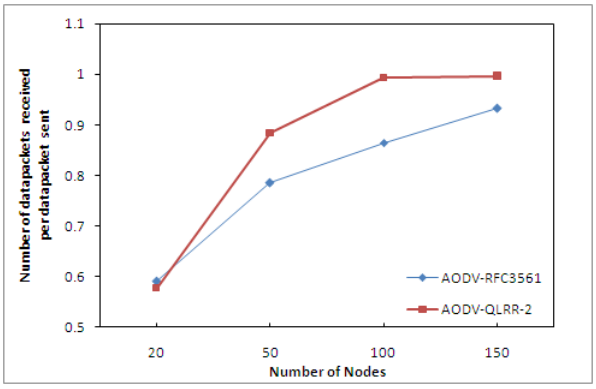

Figure 6.15: Network Size versus Delivery Ratio

Figure 6.15 shows routing overhead of AODV-QLRR in comparison with the default AODV protocol. For small networks of 20 nodes, given the high values of speed of 20 m/s it is expected that the delivery ratios values are quite low of around 60%. However we see a drastic improvement in the delivery ratios averaging more than 98% when the network is more thickly populated. This is mainly because there are fewer network partitions. Since

the local recovery scheme of default AODV protocol floods unconditionally, the chances of route repair is higher than AODV-QLRR which floods in a limited way. However this is not in the interest of AODV-QLRR as its main purpose to reduce routing overhead. It can be noticed that AODV-QLRR is outperformed only by about 2.2% which is not too much of a sacrifice given the fact that more than 44% saving is achieved in terms of routing overhead.

Route Length

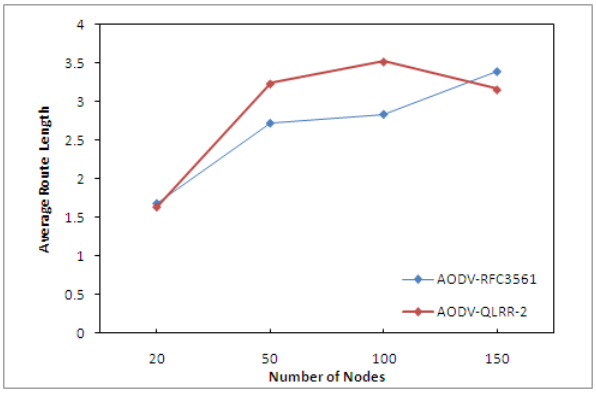

Figure 6.16: Network Size versus Route Length

Figure 6.16 shows routing length of AODV-QLRR in comparison with the default AODV protocol. Routing length is an indicator of end-to-end delay experienced by transport and application higher layer entities. Although the average route length for AODV-QLRR is higher than default AODV, the values are still comparable.

Chapter 7

Conclusions and Future Work

We have presented a scalable local recovery mechanism based on scheme called Query Localization of Route Request - QLRR, for Ad-hoc On-demand Distance Vector routing protocol. In contrast to the end-to-end recovery mechanism, QLRR relies on broadcast of route request within the region around link failure, to repair the broken route quickly, thus providing a way for fast recovery. As QLRR only floods a small region, which is a circle with the radius of 2 hops distance, it will considerably outperform end-to-end recovery especially in scenarios with higher node density, as it only relies on a small percentage of the whole network to achieve the route recovery process. Hence, it is a scalable solution for the route re-discovery process.

We compared AODV-QLRR with the version of AODV protocol compliant to RFC 3561 which also has a local recovery mechanism, but without query localization. We used a well-known implementation of AODV protocol by Uppasala University, called AODV-UU, to customize it to include QLRR. We performed several simulations using the well known and widely used Network Simulator. We created several simulated scenarios with varying network node densities, speed, pausetime and offered load and ran the simulations for sufficiently longer duration so as to obtain stable values for our measurements.

From the resulting graphical plots, we have seen that AODV-QLRR considerably reduces routing control overhead for all kind of loads, mobility and network sizes. We have seen that the improvement can be as high as 84% improvement over AODV-RFC3561 i.e. AODV with default local repair mechanism as per RFC 3561. Furthermore, we also expect

AODV-QLRR to reduce the end-to-end delay. It is interesting to note however that AODV-QLRR has consistently lower delivery ratios than the default protocol. The main purpose of query localization is to exploit node locality and reduce the number of routing message transmissions. Localizing the query, however, has the risk of not being able to establish the route. However we must also notice that the values are still comparable to its default counterpart.

Thus through this work we have shown by simulation that query localized schemes can be used beneficially for improving scalability in ad-hoc mobile networks. However verifying the same by actual experiments is inherently difficult due to the a variety of challenges posed by the phycial medium. It therefore requires a lot more effort and remains a candidate for future work. Coming up with routing protocols for mobile ad-hoc networks which experience minimal performance degradation when used in increasingly large networks is a challenge, and there remains a significant amount of work to reach this goal.

Bibliography

[1] Samba Sesay, Zongkai Yang, and Jianhua He. A survey on mobile ad hoc wireless network. Asian Network for Scientific Information, Pakistan, 2004.

[2] Theodore S. Rappaport and Theodore Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, December 2001.

[3] Suman Banerjee and Archan Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *In Proceedings of Mobihoc*, pages 146–156, 2002.

[4] IEEE Standards Department. Wireless lan medium access control (mac) and physical layer (phy) specifications, ieee standard 802.11,, 1997.

[5] Chao Gui and Prasant Mohapatra. A framework for self-healing and optimizing routing techniques for mobile ad hoc networks. *Wirel. Netw.*, 14(1):29–46, 2008.

[6] Timthoy X. Brown, Harold N. Gabow, and Qi Zhang. Maximum flow-life curve for a wireless ad hoc network. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 128–136, New York, NY, USA, 2001. ACM.

[7] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, New York, NY, USA, 1998. ACM.

[8] J. Broch, D.B. Johnson, and D.A. Maltz. The dynamic source routing protocol for mobile ad hoc networks, 1998.

[9] nez César A. Santivá Ram Ramanathan, and Ioannis Stavrakakis. Making link-state routing scale for ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 22–32, New York, NY, USA, 2001. ACM.

[10] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, 1994.

[11] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[12] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[13] E.M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2):46–55, Apr 1999.

[14] C. Perkins, Elizabeth M. BeldingRoyer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing protocol, 2002.

[15] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.

[16] Sasikanth Avancha, Jeffrey Undercoffer, Anupam Joshi, and John Pinkston. Security for wireless sensor networks. pages 253–275, 2004.

[17] Po-Wah Yau and C.J. Mitchell. Reputation methods for routing security for mobile ad hoc networks. *Mobile Future and Symposium on Trends in Communications, 2003. SympoTIC '03. Joint First Workshop on*, pages 130–137, Oct. 2003.

[18] Chai keong Toh. Associativity-based routing for ad-hoc mobile networks. *Wireless Personal Communications*, 4:103–139, 1997.

[19] Elizabeth M. Royer and C k Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6:46–55, 1999.

[20] Rohit Dube, Cynthia D. Rais, Kuang yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communications*, 4:36–45, 1997.

[21] Fei Dai and Jie Wu. Proactive route maintenance in wireless ad hoc networks. *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, 2:1236–1240 Vol. 2, May 2005.

[22] Genping Liu, Kai juan Wong, Bu sung Lee, Boon chong Seet, Chuan heng Foh, and Lijuan Zhu. Patch: A novel local recovery mechanism for mobile ad-hoc networks.

[23] Sung ju Lee and Mario Gerla. Aodv-br: Backup routing in ad hoc networks. In *In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC*, pages 1311–1316, 2000.

[24] D. Aron and Sandeep K. S. Gupta. Analytical comparison of local and end-to-end error recovery in reactive routing protocols for mobile ad hoc networks. In *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 69–76, New York, NY, USA, 2000. ACM Press.

[25] Wei Tang and Wei Guo. A path reliable routing protocol in mobile ad hoc networks. *Mobile Ad-hoc and Sensor Networks, 2008. MSN 2008. The 4th International Conference on*, pages 203–207, Dec. 2008.

[26] Joo sang Youn. Quick local repair scheme using adaptive promiscuous mode in mobile ad hoc networks.

[27] Antonios Argyriou and Vijay Madisetti. 3 realize load balancing in mobile ad hoc networks q, 2004.

[28] Austin Tate, John Levine, Peter Jarvis, and Jeff Dalton. Using ai planning technology for army small unit operations. In *Poster Paper in the Proceedings of the Artificial Intelligence Planning and Scheduling Systems Conference (AIPS-2000*, pages 379–386, 2000.

[29] Christine Cheng, Ravi Jain, and Eric van den Berg. Location Prediction Algorithms for Mobile Wireless Systems. In Borko Furht and Mohammad Ilyas, editors, *Wireless Internet Handbook. Technologies, Standards, and Applications*, pages 245–263. CRC Press, Florida, 2003.

[30] Yu-Chee Tseng, Shih-Lin Wu, Wen-Hwa Liao, and Chih-Min Chao. Location awareness in ad hoc wireless mobile networks. *Computer*, 34(6):46–52, 2001.

[31] M. Guarnera, M. Villari, A. Zaia, and A. Puliafito. Manet: possible applications with pda in wireless imaging environment. the. In *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2394–2398, 2002.

[32] Azzedine Boukerche, Horacio A. B. F. Oliveira, Eduardo F. Nakamura, and Antonio A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Comput. Commun.*, 31(12):2838–2849, 2008.

[33] Jun-Zhao Sun. Mobile ad hoc networking: an essential technology for pervasive computing. *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on*, 3:316–321 vol.3, 2001.

[34] Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins. Scalability study of the ad hoc on-demand distance vector routing protocol. *Int. J. Netw. Manag.*, 13(2):97–114, 2003.

[35] Erik Nordstrom. Aodv-uu. http://core.it.uu.se/core/index.php/AODV-UU.

[36] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[37] Ieee 802.11-2007, wireless lan medium access control (mac) and physical layer (phy) specifications, june 2007., June 2007.

[38] David C. Plummer. An ethernet address resolution protocol – or – converting network protocol addresses to 48.bit ethernet address for transmission on ethernet hardware. RFC 826 (Standard), November 1982.

[39] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. RFC 826 (Standard), 1999.