6-2006

# Evaluation of the Effects of Predicted Associativity on the Reliability and Performance of Mobile Ad Hoc Networks

Esteban Francisco Sanchez

## Recommended Citation

Sanchez, Esteban Francisco, "Evaluation of the Effects of Predicted Associativity on the Reliability and Performance of Mobile Ad Hoc Networks" (2006). *Theses and Dissertations*. 3456.
https://scholar.afit.edu/etd/3456

# Evaluation of the Effects of Predicted Associativity On the Reliability and Performance Of Mobile Ad Hoc Networks

THESIS

Esteban Francisco Sanchez, Captain, USAF

AFIT/GCE/ENG/06-05

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

AFIT/GCE/ENG/06-05

# EVALUATION OF THE EFFECTS OF PREDICTED ASSOCIATIVITY ON THE RELIABILITY AND PERFORMANCE OF MOBILE AD HOC NETWORKS

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Engineering

Esteban Francisco Sanchez, B.S.C.E.
Captain, USAF

June 2006

AFIT/GCE/ENG/06-05

# EVALUATION OF THE EFFECTS OF PREDICTED ASSOCIATIVITY ON THE RELIABILITY AND PERFORMANCE OF MOBILE AD HOC NETWORKS

Esteban Francisco Sanchez, B.S.C.E.
Captain, USAF

Approved:

| | |
|---|---|
| /signed/ | 1 Jun 2006 |
| Dr. Barry E. Mullins (Chairman) | date |
| /signed/ | 1 Jun 2006 |
| Dr. Rusty O. Baldwin (Member) | date |
| /signed/ | 1 Jun 2006 |
| Dr. Richard A. Raines (Member) | date |

AFIT/GCE/ENG/06-05

# *Abstract*

Routing in Mobile Ad Hoc Networks (MANETs) presents unique challenges not encountered in conventional networks. Limitations in bandwidth and power as well as a dynamic network topology must all be addressed in MANET routing protocols. Predicted Associativity Routing (PAR) is a custom routing protocol designed to address reliability in MANETs. By collecting associativity information on links, PAR calculates the expected lifetime of neighboring links. During route discovery, nodes use this expected lifetime, and their neighbor's connectivity to determine a residual lifetime. The routes are selected from those with the longest remaining lifetimes. Thus, PAR attempts to extend the duration routes are active, thereby improving their reliability.

PAR is compared to Ad Hoc On-Demand Distance Vector Routing (AODV) using a variety of reliability and performance metrics. Despite its focus on reliability, PAR does not provide more reliable routes. Rather, AODV produces routes which last as much as three times longer than PAR. However PAR, even with shorter lasting routes, delivers more data and has greater throughput. Both protocols are affected most by the node density of the networks. Node density accounts for 48.62% of the variation in route lifetime in AODV, and 70.66% of the variation in PAR. As node density increases from 25 to 75 nodes route lifetimes are halved, while throughput increases drastically with the increased routing overhead. Furthermore, PAR increases end-to-end delay, while AODV displays better efficiency.

# *Acknowledgements*

There are a number of people who deserve thanks for the research presented in this document. First, I thank Dr. Barry E. Mullins, my thesis advisor. Without his expert guidance, sage advice, and steadfast support this work would not have been possible. Additionally, I offer my thanks to Dr. Rusty O. Baldwin and Dr. Richard A. Raines. Their support and expertise shaped the direction of this research, and ensured its success through their insight.

Furthermore, I offer my deepest thanks to my family. The love, support, infinite patience, and understanding my wife has shown made this work possible. Her sacrifice kept things together while permitting me to accomplish this research. To her, I extend my utmost gratitude. I also offer my appreciation to my newborn son. He provides me with inspiration, joy, and a reprieve from the stress of my work. Finally, a debt of gratitude is owed my parents. They instilled me with an appreciation of hard work and a thirst for knowledge. Without these values none of this would be possible.

To those mentioned here, and the countless others who have supported me and my efforts, I extend my sincerest gratitude.

Esteban Francisco Sanchez

# *Table of Contents*

Page

Bibliography . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128

# *List of Figures*

# *List of Tables*

## *List of Abbreviations*

| Abbreviation | | Page |
|---|---|---|
| PAR | Predicted Associativity Routing | 2 |
| AODV | Ad Hoc On-Demand Distance Vector | 2 |
| DSDV | Destination Sequenced Distance Vector | 8 |
| NDPU | Network Data Packet Unit (in DSDV) | 8 |
| WRP | Wireless Routing Protocol | 9 |
| DT | Distance Table (in WRP) | 10 |
| RT | Routing Table (in WRP) | 10 |
| LCT | Link Cost Table (in WRP) | 10 |
| MRL | Message Retransmission List (in WRP) | 10 |
| GSR | Global State Routing | 11 |
| $A$ | Neighbor List (in GSR) | 11 |
| $TT$ | Topology Table (in GSR) | 11 |
| $NEXT_i$ | Next Hop Table (in GSR) | 11 |
| $D$ | Distance Table (in GSR) | 11 |
| FSR | Fisheye State Routing | 12 |
| CGSR | Clusterhead Gateway Switch Routing | 12 |
| STAR | Source-Tree Adaptive Routing | 13 |
| LORA | Least Overhead Routing Approach | 13 |
| ORA | Optimal Routing Approach | 13 |
| DSR | Dynamic Source Routing | 14 |
| RREQ | Route Request | 17 |
| RREP | Route Reply | 18 |
| ABR | Associativity Based Routing | 19 |
| BQ | Broadcast Query (in ABR) | 19 |
| RN | Route Notification (in ABR) | 21 |

| Abbreviation | | Page |
|---|---|---|
| LQ | Localized Query (in ABR) | 21 |
| FORP | Flow-Oriented Routing Protocol | 22 |
| LAR | Location-Aided Routing | 22 |
| SSR | Signal Stability Routing | 22 |
| TORA | Temporally Ordered Routing Algorithm | 23 |
| ZRP | Zone Routing Protocol | 24 |
| IARP | Intrazone Routing Protocol (in ZRP) | 24 |
| IERP | Interzone Routing Protocol (in ZRP) | 24 |
| MAC | Media Access Control | 24 |
| NDP | Neighbor Discovery Protocol (in ZRP) | 24 |
| TTL | Time To Live | 25 |
| CEDAR | Core Extraction Distributed Ad Hoc Routing | 26 |
| QoS | Quality of Service | 26 |
| RPGM | Reference Point Group Mobility | 33 |
| AOMDV | Ad Hoc On-Demand Multipath Distance Vector | 35 |
| RERR | Route Error (in PAR) | 47 |
| SUT | System Under Test | 63 |
| CUT | Component Under Test | 63 |
| pps | Packets Per Second | 72 |

# Evaluation of the Effects of Predicted Associativity On the Reliability and Performance Of Mobile Ad Hoc Networks

## I. Introduction

### *1.1 Background and Motivation*

Advances in wireless network technology have freed users from the tethers of conventional wired networks. Wireless networks permit users to move anywhere within transmission range of an access point. Even so, there are still limitations associated with this type of network. First, their range is limited to the transmission range of the nodes. All nodes in infrastructure networks are tied to an access point. So, while movement is supported in infrastructured wireless networks, it is not limitless. Additionally, traditional wireless networks require a supporting infrastructure be in place to facilitate communication. This places an additional burden when implementing the network.

Mobile ad hoc networks (MANETs) solve many of the limitations of infrastructured wireless networks. MANETs extend freedom of motion by not requiring nodes be within range of an access point. Rather, nodes in a MANET communicate directly with one another thereby allowing full freedom of motion, provided another node in the network can be reached. Furthermore, MANETs require no infrastructure which means MANETs can be established as needed and at minimal cost since all that is required is the nodes themselves.

MANETs are not without their own limitations. MANETs have a smaller bandwidth than wired networks and MANET nodes have limited computational power and energy. Additionally, the nature of the radio communications channel is a challenge to MANETs. Finally, the mobility of MANET nodes complicates communication.

These complications place a particular burden on routing in MANETs. Since there is no infrastructure, each node in the network serves as a router. Nodes cooperate to facilitate communication to distant nodes by forwarding messages from source to destination. The limitations of MANETs make developing and maintaining routes in these networks difficult. In particular, the mobility of the nodes in the network makes establishing reliable routes especially challenging. Frequent changes in topology mean routes fail regularly. Thus, reliable routes can be difficult to discover in MANETs.

Reliable routing is the motivation for this research. This research proposes a reliable routing protocol called Predicted Associativity Routing (PAR). This protocol estimates link lifetimes, and selects routes with the greatest expected remaining lifetime. To determine the effectiveness of this strategy, the reliability and performance of PAR is compared to the frequently used Ad Hoc On-Demand Distance Vector (AODV) routing protocol.

## *1.2 Objectives*

There are three primary objectives of this research. The first is to evaluate the reliability of PAR compared to AODV. Second, to determine the performance of both protocols in a variety of network configurations. Finally, the various factor levels of the experiment are changed to determine their effect on the reliability and performance of the protocols. The factors which have the greatest impact on the protocols are identified.

## *1.3 Approach*

Reliability in MANET routing can be defined in several ways. This research defines reliable routes as routes which, once established, can be depended on to remain active. Defined in this way, route lifetime becomes the metric by which routing protocols are evaluated and compared. Route lifetimes are the time interval between a route being entered into a node's routing table, and the failure of that route. These statistics are compared for each routing protocol to determine which protocol performs more reliably, and to evaluate how the various factors of the research impact reliability.

### *1.4 Summary*

This research compares the reliability of the custom PAR routing protocol to the AODV routing protocol. The performance of this new protocol is evaluated to determine the effectiveness of PAR's residual lifetime prediction method. Finally, this research determines the impact of a variety of factors on the performance and reliability of both the PAR and AODV routing protocols.

The remainder of this document is organized in the following way. Chapter 2 provides an overview of MANET routing protocols, methods for modeling mobility in MANETs, and previous research into reliable routing in MANETs. Chapter 3 describes the detailed implementation of the Predicted Associativity Routing protocol, and outlines the results of a pilot study to evaluate its parameters. Chapter 4 outlines the methodology used to conduct the experiments in this research. Chapter 5 provides the results of the research, provides analysis of those results, and draws conclusions about the protocols studied. Finally, Chapter 6 summarizes the research and its results, describes its impact, and suggests some possibilities for follow-on research.

# II. Literature Review

## *2.1 Introduction*

This chapter provides an introduction to several aspects of MANET routing. Section 2.2 introduces and discusses unique challenges of MANET routing. Section 2.3 presents a variety of Table-Driven, On-Demand, and Hybrid routing protocols, and describes their approach to the challenge of MANET routing. Section 2.4 introduces entity and group mobility models, and provides examples of each. Section 2.5 is an overview of several research efforts addressing improving the reliability of routing in MANETs.

## *2.2 Routing Challenges In Mobile Ad-Hoc Networks*

The limitations associated with nodes in mobile ad hoc networks present significant challenges to these networks. In particular, routing is problematic. Restrictions in resources and available bandwidth, the dynamic nature of the network's topology, and the properties of the transmission channel all impact routing in ad hoc wireless networks [Joh94]. Routing schemes must address these issues to ensure network hosts can communicate effectively.

The first of these challenges are the resource constraints of wireless networks [Joh94]. Nodes in an ad hoc network must be self sufficient, providing their own power sources. To achieve a level of portability, wireless devices sacrifice resources. Key among these is power. Batteries only provide the host with a limited supply of electrical power. Thus, power must be conserved. Taking the task of the routing from dedicated routers and imposing it on network nodes further taxes the battery. Nodes must now perform additional computations to determine routes. Additional transmissions must be sent to update routing tables or create routes. Nodes must not only send their own traffic, but the traffic of any other host using them as an intermediate hop. All of these additional burdens require power. The routing scheme employed in ad hoc wireless networks must limit this added routing overhead to preserve the available power.

Wireless nodes are often limited in other resources as well. Computational power and storage can be exhausted with the added burden of routing [MuM04]. In a large network with a significant amount of traffic, it is possible for a host to be inundated by route construction requests, route table updates, and packet relays for other nodes in the network. Storage is another resource that is limited in wireless nodes. Depending upon the protocol, certain data concerning the network topology must be stored to efficiently route traffic. In large networks, this can be a significant amount of data. Again, this is a resource consumed by routing overhead that could be used by the node itself. The routing scheme used by the node must make judicious use of these limited resources as well.

Bandwidth in wireless networks is limited [Joh94]. Since wireless networks use RF as their transmission mechanism, the available bandwidth is restricted. Nodes in these networks operate within a certain range of frequencies. The physical properties of radio transmission establish a ceiling on the bandwidth available in this range [MuM04]. In wired networks, this issue is of less concern; the use of fiber optics, copper, and multiplexing techniques afford wired networks greater data rates. Wireless networks, on the other hand, must operate within the constraints of the radio spectrum. With restricted bandwidth, ad hoc networks must ensure that they use this resource appropriately. If the routing protocol produces large or frequent routing messages, bandwidth is wasted and is an inefficient use of the network potentially lowering data throughput. Routing protocols, then, must also be designed to facilitate efficient use of available bandwidth.

The use of radio as a transmission mechanism brings with it further complications. Links using radio transmissions are subject to changes in capacity and an increased probability of error due to the RF channel. Additionally, since it is a broadcast medium collisions occur when nodes transmit simultaneously. One particular example of this issue is the hidden terminal problem, shown in Figure 2.1 [MuM04]. Node B is in the transmission range of both nodes A and C, but A and C are not within transmission range of each other. If node A is transmitting to node B, node C is unable to detect this transmission because it is out of node A's transmission range. Therefore, node C senses the medium is free, and transmits. This results in a collision at node B.

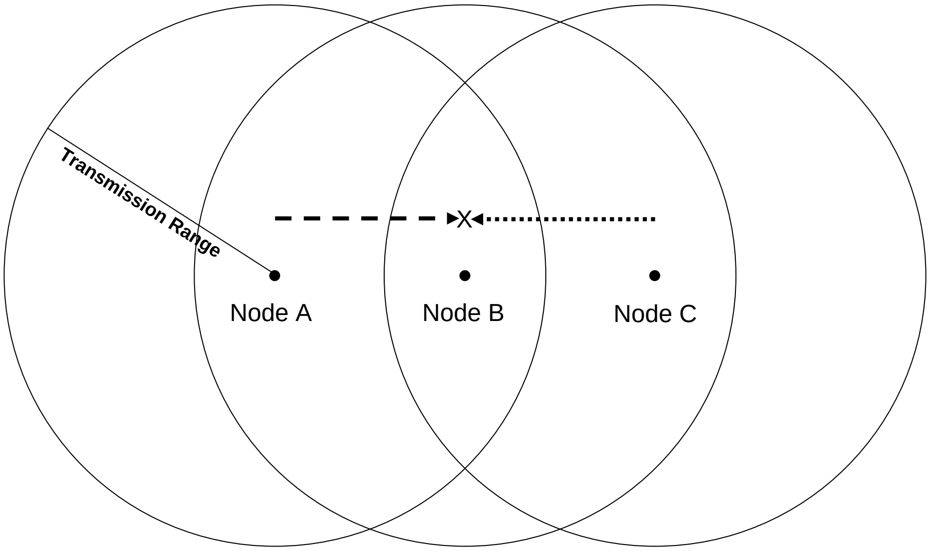


Figure 2.1: Hidden terminal problem [MuM04]

A further example of the complications of a radio transmission medium can be seen in the exposed terminal problem depicted in Figure 2.2 [MuM04]. Nodes B and C are within transmission range of each other. If node B is transmitting, to node A for example, node C will not transmit to any other node, even if that node is outside the range of node B since node C's broadcast would result in a collision with node B's ongoing transmission. However, node C's transmission to D would not interfere with that since B is transmitting to A. Thus, channel capacity is wasted. To establish reliable paths, routing protocols used in ad hoc networks must detect changes in link capacity, quality, and congestion. As these factors change, the protocol must adapt to ensure reliable communication between nodes.

Finally, the mobility of the nodes presents a challenge to routing in ad hoc wireless networks [Joh94]. Since nodes not only manage their own traffic but act as routers for other nodes, the mobility of a node can affect the entire network. When nodes are mobile the topology changes frequently. As hosts move within the network,

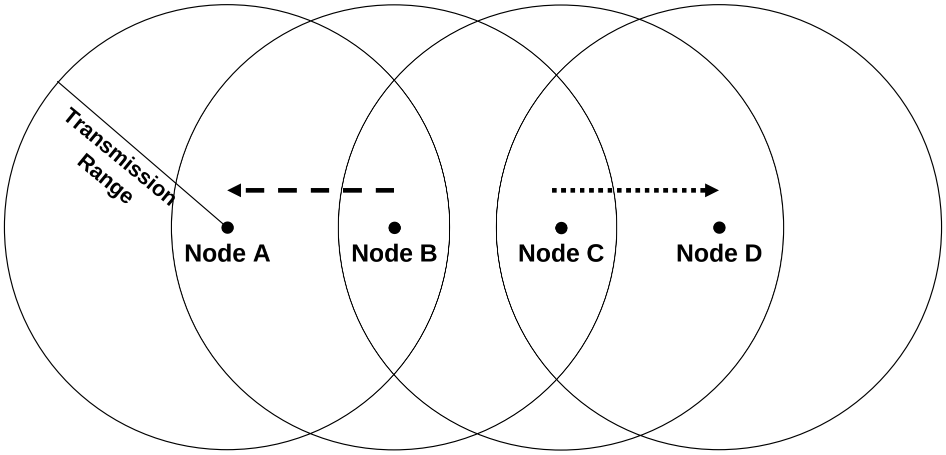


Figure 2.2: Exposed terminal problem [MuM04]

existing routes could be invalidated, thereby forcing nodes to send route updates, or to discover new routes to a desired destination. This generates routing control and update information that must be transmitted on the network. Routing protocols in mobile ad hoc networks must be capable of handling the mobility of the nodes without consuming excessive amounts of the limited resources. Mechanisms must ensure that paths broken by mobility can be repaired, and that paths to the destination can be reestablished quickly, whenever possible.

## *2.3 Routing Protocols*

There are numerous routing protocols in wireless ad hoc networks. The ultimate goal of all of these protocols is to efficiently discover and maintain routes for data transfer between pairs of nodes in the network. Each protocol performs these activities in unique ways. Protocols can be classified, based on their methodology for accomplishing these goals, into three categories: 1) Table Driven Protocols; 2) On Demand Protocols; and 3) Hybrid Protocols [MuM04].

*2.3.1 Table Driven Protocols.* Table driven protocols are proactive in their approach to routing. Each node in the network maintains information about the topology of the network [Mis99]. From this information, hosts can determine the optimal path to the desired destination. Protocols discover and maintain their routing information by broadcasting update messages periodically or when the network topology changes. When nodes receive update packets, the information is compared to the information currently possessed by the node. If the route to a destination has changed or is stale, the node will update its information. The following are examples of table driven routing protocols.

#### *2.3.1.1 Destination Sequenced Distance Vector Routing Protocol.*

The Destination Sequenced Distance Vector (DSDV) routing protocol is an extension of the distributed Bellman-Ford algorithm [Ily03]. All nodes maintain a route table with entries for every reachable destination in the network. For each entry, the protocol maintains information on the next hop along the path to the destination, as well as the hop count of the route. Additionally, destination sequence numbers preventing routing loops from developing, and ensures that only the most recent route information is retained in the route table.

Since the route table maintained by each node contains route information to every available destination node, route discovery is straightforward. Route maintenance is performed through a series of table update messages. Updates are sent periodically or are triggered by significant changes in the topology of the network [PeB94]. Destination nodes initiate route updates, incrementing their sequence number to a greater even value. Even sequence numbers identify this as a periodic update, rather than an update forced by a link failure. There are two varieties of update, incremental and full. Incremental updates occur when there are no significant changes in the node's local topology and the updated information can be transmitted in a single Network Data Packet Unit (NDPU). Full updates, on the other hand, transmit all routing information. They are used when significant changes to the network topology occur, or when the updated information requires multiple NDPUs [PeB94].

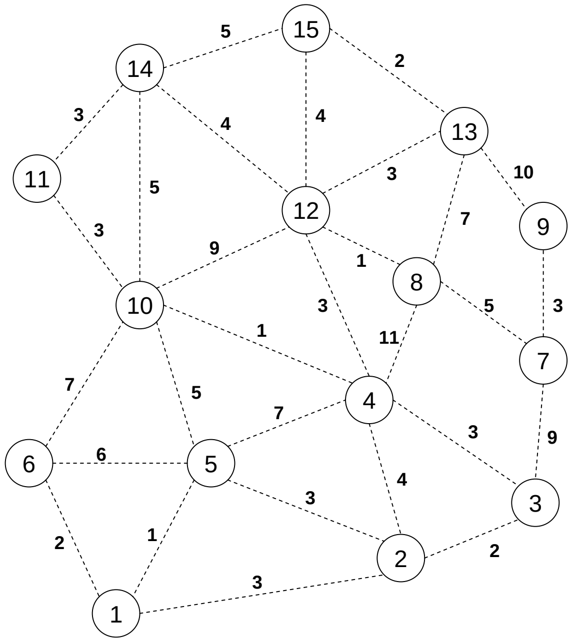
Nodes initiating updates, broadcast the appropriate update message to their neighbors. Upon receiving the update message, neighboring nodes either update their routing tables, or, if configured to do so, wait a predetermined period of time, allowing updates from several neighbors to arrive [PeB94]. Using the latter method allows a node to select routing information with the best metric, for example the shortest hop count. The node updates its tables based on the sequence number associated with the update. If the sequence number of the update is greater than the number stored in the table, signifying more current route information, the route table is updated. If, however, the destination sequence number is less than the current number, the update is rejected. The metric, in this case hop count, determines the best route between updates with identical destination numbers. These updated routes are propagated to neighboring nodes through periodic updates [PeB94].

Link failures occurring during transmission are handled in a slightly different manner. The node detecting the breakage updates its table to indicate a hop count of $\infty$ for the failed path and initiates a route update message reflecting this information with an odd sequence number greater than that stored in its routing table. Upon receiving this message, a node forwards the message to propagate this information throughout the network [PeB94]. Likewise, nodes receiving a message with an $\infty$ metric, which have a greater sequence number and a finite metric, immediately forward an update throughout the network reflecting this new route to the affected destination. In this way, when the node on the downstream end of the broken link broadcasts a route update, the path to this destination is updated and reestablished.

*2.3.1.2 Wireless Routing Protocol.* Like DSDV, the Wireless Routing Protocol (WRP) is an extension of the distributed Bellman Ford routing algorithm [MuM04]. However, while DSDV uses destination sequence numbers to prevent routing loops from developing, WRP uses the shortest path to each node and the penultimate hop on these paths to eliminate loops. WRP further distinguishes itself from DSDV by means of its route update and maintenance process.

Rather than maintain a single table containing routing data, WRP maintains a series of several tables to gather more accurate routing information [MuM04]. The first is the distance table (DT). This accumulates the distance and penultimate hop on the path to a particular destination reported by each of the node's neighbors. The routing table (RT) keeps track of up-to-date routes to all known destinations. In this table, the shortest route distance, the penultimate hop on the route to the destination, the next hop along the path to the destination, as well as a status flag indicating whether the particular route is a simple path, loop, or unmarked is stored. Figure 2.3 depicts the values that would populate the corresponding Routing Table fields, given the network topology and link costs. Next, the link cost table (LCT) records the number of hops to reach a destination when relaying a packet through each available link. To indicate broken links, $\infty$ is entered as the cost. Additionally, as a mechanism to detect link failures, the LCT maintains the number of intervals since the last successful update was received over a particular link [MuM04]. Finally, the message retransmission list (MRL) contains the list of all update messages that must be retransmitted, as well as a counter for each. Nodes acknowledge each update message. In the absence of an acknowledgement, a node retransmits an update message after a predetermined interval. With each subsequent transmission of a message, the counter is decremented until zero is reached. At this point the message is resent, and the MRL entry is deleted. This further aids the protocol in determining the viability of links. By maintaining this set of tables, nodes can perform consistency checks on the data when route updates are received from its neighbors [MuM04]. Additionally, tracking the predecessor node information for each destination node enables a node to converge to a viable route quickly.

Using the mechanisms described above, nodes will detect link breakages. In response, nodes send an update message setting the minimum cost of the failed link to $\infty$ [MuM04]. Based on the information contained in its DT, the initiating node attempts to find an alternate path to the destination. If found, the node distributes an update. Nodes receiving updates reflecting a new route to the destination only accept it if it is shorter than the existing routes. In this manner viable routes are distributed throughout the network, overcoming the failure of a link.



| Node | NextNode | Pred | Cost |
|---|---|---|---|
| 15 | 15 | 15 | 0 |
| 14 | 15 | 14 | 5 |
| 13 | 15 | 13 | 2 |
| 12 | 15 | 12 | 4 |
| 11 | 14 | 14 | 8 |
| 10 | 4 | 12 | 8 |
| 9 | 13 | 13 | 12 |
| 8 | 12 | 12 | 5 |
| 7 | 8 | 12 | 10 |
| 6 | 10 | 12 | 15 |
| 5 | 10 | 12 | 13 |
| 4 | 12 | 12 | 10 |
| 3 | 4 | 12 | 7 |
| 2 | 4 | 12 | 11 |
| 1 | 2 | 12 | 14 |

Figure 2.3: Routing table entries for each node for destination node 15 [MuM04]

*2.3.1.3 Global State Routing Protocol.* The Global State Routing (GSR) protocol merges qualities of traditional link state protocols and extensions of the distributed Bellman Ford algorithm, as seen in DSDV routing [Mis99]. Like link state protocols, GSR establishes routes based on the exchange of state information between the nodes in the network. However, wired link state protocols do this by flooding the network. GSR improves on this by distributing route information to neighbors on a periodic basis. This is similar to the method used by DSDV for route information dissemination [Mis99].

Each node in the network maintains route information in a set of four lists and tables: a neighbor list ($A$), a topology table ($TT$), a next hop table ($NEXT_i$) and a distance table ($D$) [ChG98]. The list $A$ identifies all nodes adjacent to the given node. Adjacent is defined as the set of nodes that can be heard by a node. Table $TT$ has an entry for each reachable destination in the network. Each of these entries contain two components. $TT.LS$ contains link state information provided by a given destination timestamped with $TT.SEQ$. This timestamp allows nodes to evaluate the freshness of the routing information contained in the table entry. Tables $NEXT$ and $D$, similarly, contains an entry for each destination. $NEXT$ identifies the node to which packets for a particular destination should be forwarded along the shortest path. $D$ indicates the distance of this shortest path [ChG98].

Route information update messages are only distributed to the neighbors of a node. Upon receiving a message from a neighbor, a node examines the sequence number, or timestamp, associated with a particular piece of link state information and compares it to the link state data in table $TT$. If the new data is more recent than the $TT$ entry, $TT$ is updated with latest information. Otherwise, it is discarded. Once the topology table is updated, the node recomputes the routes based on the updated link state information. The resulting routes are updated in the $NEXT$ and $D$ tables and broadcast to the nodes neighbors. This process is repeated periodically [ChG98].

An extension of this protocol is the Fisheye State Routing (FSR) protocol [PGC00] which uses the same scheme of route maintenance. However, FSR modifies the frequency with which link state update information is sent based on a destination nodes distance, as seen in Figure 2.4. A node frequently sends updates about nearby nodes, represented by the darkest ring. Information about distant nodes is sent infrequently, indicated with the lightest ring. In doing so, FSR reduces the overhead required for sending updates, since updates are smaller. The consequence of this is nodes have accurate route information for nodes that are close by, while the link state information of distant nodes can be inaccurate. However, as a packet moves to the destination node, the route information becomes progressively more accurate [PGC00].

*2.3.1.4 Other Table Driven Protocols.* In addition to the protocols described above, there are many other table driven protocols. One such protocol is the Clusterhead Gateway Switch Routing (CGSR) protocol [Mis99]. In this protocol, nodes are organized into clusters, each with an elected cluster-head. Nodes belong to a given cluster if they are in range of the cluster-head. Gateway nodes belong to

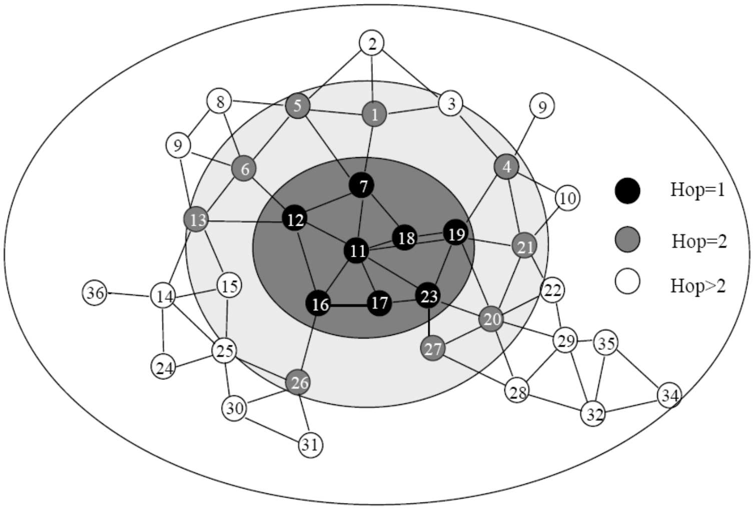


Figure 2.4: FSR update information regions [PGC00]

multiple clusters. Sources transmit packets by sending the packet to their cluster-head which forwards the packet to the gateway of the next cluster-head in the route. The gateway propagates the packet to the next cluster-head, and the process repeats until the cluster-head of the destination is reached which delivers the packet to the destination.

The Source-Tree Adaptive Routing (STAR) Protocol is another example of a table driven routing protocol [MuM04]. STAR reduces the overhead associated with table driven routing protocols by using a least overhead routing approach (LORA) rather than the optimal routing approach (ORA). The goal is to provide viable, though potentially suboptimal, paths. By doing so, the amount of control overhead required can be drastically reduced [MuM04]. In the STAR protocol, each node maintains and broadcasts source-tree information, consisting of the links in the preferred route to destination nodes. Through the receipt of this information from its neighbors, nodes generate partial graphs of the network. Updates are broadcast at initialization, and upon discovery of new destinations, nodes construct routes to every destination [MuM04].

#### *2.3.2 On Demand Protocols.*

On demand protocols generate routes as needed. Nodes without current routing information invoke a route discovery process to determine the path to a destination. Nodes using these protocols may use tables or caches to maintain information about the routes discovered, but, unlike table driven protocols, do not keep route information for all possible destination nodes. As such, these protocols do not exchange periodic information updates. On demand protocols, instead, perform route maintenance by monitoring the status of links in active routes. Topological changes that affect these routes initiate a route maintenance procedure that can reconstruct the route, invalidate the route, or rediscover the route, depending on the protocol. Several examples of on demand protocols follow.

##### *2.3.2.1 Dynamic Source Routing Protocol.*

In the Dynamic Source Routing (DSR) protocol, nodes with packets to send to another network node must construct a source route to that destination [Ily03]. A source route identifies each node on the path to the destination. The source route is contained in the packet's header. Intermediate nodes simply forward the packet to the next node identified in the packet header. This continues until the packet reaches its destination.

Nodes in networks using DSR maintain route caches of known routes. If a node has a packet for a particular destination, it checks its route cache to determine if a source route is available [JoM96]. If no route is available it must be constructed by means of a route discovery process. A source node initiates the route discovery by broadcasting a route request packet to its neighbors. The request contains the source and destination, a unique request id, and a route record. Each node maintains a list of the source address and request id of route requests recently received. Upon receipt of a route request the packet is checked against this list. If the request has already been received, it is discarded. Furthermore, if the receiving node is already in the route record, the request packet is discarded. These checks ensure the resulting route is loop free. If these conditions are met, a node appends its address to the route record and forward it to its neighbors. This process continues until the destination is reached. Figure 2.5 demonstrates this process. Route requests travel by three unique paths to the destination, node 15. Additionally, network links that do not have a corresponding route request reflect links over which duplicate requests are received. These requests are ignored, and the links do not appear in any route. [JoM96].

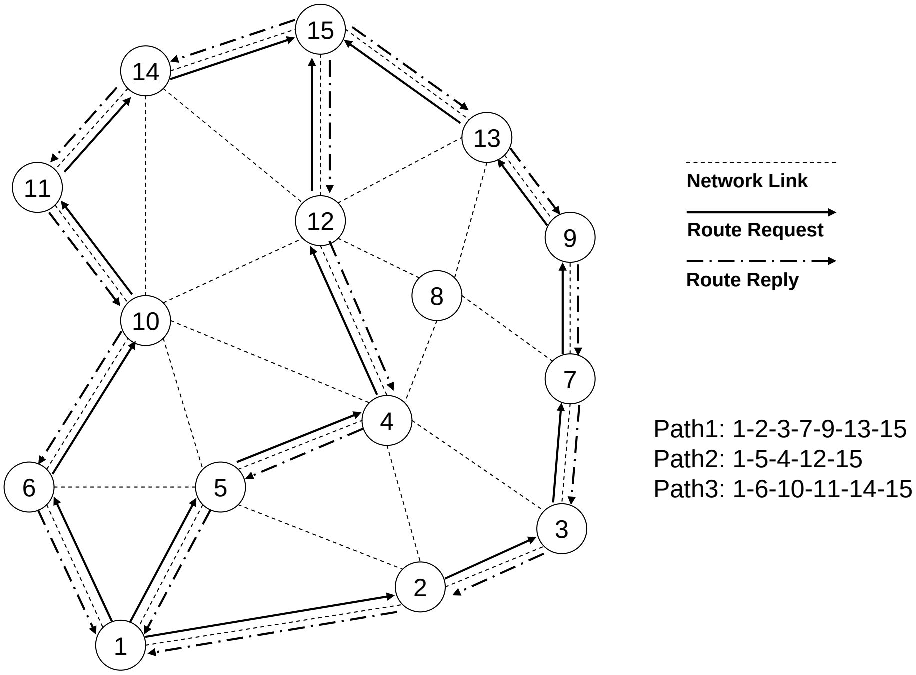


Figure 2.5: DSR route request/route reply process [MuM04]

Upon receiving a route request packet destined for itself, a target node will generate a route reply packet. DSR does not assume symmetric links [JoM96]. Thus, to return a route reply packet to the source, the destination node must have a route to the source. If a route exists in the destination node's route cache it is used. Additionally, if a network has symmetric links, the destination node can use the reverse of the route record for the route reply. Alternatively, the route reply message is "piggybacked" on a route request packet destined for the originator of the initial route request. Figure 2.5 also shows the reply process. Requests arrive at the target destination over three unique routes. The destination replies to each, sending the route reply along the reverse path.

Unlike other on-demand routing protocols, DSR does not use beacon or *hello* messages to detect failures in the links. Rather, a variety of methods are used by the route maintenance mechanism to detect link failures. One method relies on data link level reports to determine transmission problems [JoM96]. Another method, if link level acknowledgements are not available, requests the next hop send explicit acknowledgements of packets received. A final approach is through passive acknowledgement. In this method, a node determines its successor has successfully received a packet if it hears it rebroadcast the packet to the next node in the route. This particular method requires the node operate in a promiscuous mode, examining all packets it is able to hear even if it is not the destination [JoM96].

Regardless of the method used to detect link failures, the route maintenance process remains unchanged. Upon detecting a link failure, the upstream node examines its cache, truncating any routes that contain the failed link. The node then generates a route error packet which identifies the nodes at either end of the broken link. This packet is forwarded to the source of the packet (who's failure detected the link break) using any of the methods of determining routes for sending route reply messages. Upon receipt of a route error packet, a node truncates routes in its cache containing the failed link. To repair the broken route, the source initiates another route discovery process [JoM96].

There are many optimizations that can be applied to DSR to make it more efficient. The first of these makes further use of route caches [JoM96]. Nodes can examine their cache and learn routes to other nodes based on partial paths in previously discovered routes. Additionally, routes can be discovered and cached when forwarding packets for other nodes. If promiscuous receive mode is used, nodes can discover routes from packets overheard on the transmission channel. All of these methods reduce the need to perform independent route discoveries to determine routes. Having intermediate nodes reply to route requests is another optimization [JoM96]. Suppose a valid route request packet is received by an intermediate node and the route is in the intermediate node's cache. The intermediate node can generate a route reply packet, with the route record from the request packet concatenated to the node's cached route to the destination. This reduces routing overhead by limiting the retransmission of requests. One final cache eliminates stale routes by limiting the lifetimes of the routes in the cache thus preventing stale routes from being propagated to other nodes in the network [Ily03].

Optimizations of the route maintenance procedures include caching negative route information when links fail [Ily03]. By caching a particular link has failed, a node will not accept routes that contain this link. Additionally, the protocol is more efficient if it widens the distribution of route error messages [JoM96]. Nodes not on the immediately impacted route can update their caches to reflect the link failure. Finally, when a path from the source to the failed link does not match the path traveled by the route error message, the source, upon receiving the error message, can forward the message along the original path. This ensures that all nodes along this route update their caches, eliminating the failed link [JoM96].

*2.3.2.2 Ad Hoc On-Demand Distance-Vector Routing Protocol.* Ad Hoc On-Demand Distance-Vector (AODV) routing leverages properties of both the DSDV and DSR protocols to provide a scalable efficient routing protocol for ad hoc networks [Ily03]. To provide the freshest routes and prevent routing loops, AODV uses destination sequence numbers in a similar manner as DSDV. Furthermore, AODV's route discovery method is similar to DSR's. These mechanisms are combined to reduce the control overhead required in either of these other protocols. AODV does not retain the unnecessary route information and eliminates periodic route updates of DSDV and the overhead of DSR is reduced by maintaining only next hop route information, rather than storing and transmitting full paths to a destination.

Route discovery in AODV is initiated by broadcasting a route request (RREQ) packet to its neighbors [PeR97]. These packets contain the source address and sequence number, a unique broadcast ID, destination address and sequence number, and the hop count of the route. The source increments the broadcast ID for each

RREQ it generates. The combination of the source address and broadcast ID uniquely identify each route discovery. Nodes receiving a RREQ packet compare the source address and broadcast ID of the packet to those requests already processed. If the ID matches one that has already been forwarded, the request is dropped. In doing so, the protocol prevents routing loops from developing [PeR97]. However, if this is a new request and the receiving node cannot service the request itself, the packet is rebroadcast. The intermediate node maintains the source and destination address, broadcast ID, expiration time of the reverse path, the address of the neighbor from which the RREQ was received, and the source sequence number. The address of the previous node establishes a reverse path to the source [PeR97].

This process continues until the destination is reached, or until an intermediate node with an active route to the destination is reached. Such intermediate nodes can service route requests only if they have a stored destination sequence number greater than that identified by the source in the RREQ packet. Therefore, nodes are prohibited from replying to route requests with stale routes.

The responding node, be it the destination or an intermediate node, establishes the route by sending a unicast route reply (RREP) along the reverse path [PeR97]. The RREP contains the source and destination addresses, the destination sequence number for this path, the hop count, and the lifetime of the path. Upon receiving a RREP packet, nodes update their routing table to reflect the new route. Each entry in the routing table contains the destination address and sequence number, as well as next hop and hop count information. Additionally, this table contains a list of neighbors who are actively using the node to forward packets to a given destination. Active neighbors are those who have transmitted at least one packet within a predetermined timeout period. This information is used in the event of a downstream link failure. Finally, a expiration time is established for this entry. If the route is not used within the expiration time, the route entry is invalidated. Furthermore, reverse routes are invalidated when no RREP packet is received for the route within a timeout interval [PeR97].

The AODV routing protocol only performs route maintenance on links with active neighbors [PeR97]. Therefore, changes in local connectivity that do not impact active routes result in no action from the protocol. Changes in network topology are detected through the use of *hello* messages which are periodically broadcast by each node. Failure to receive a predetermined consecutive number of these messages from a neighbor indicates its link with that neighbor has failed. Upon detecting such a failure, the hop count is set to $\infty$, and a destination sequence number is incremented for any routes that include the failed link. The node then broadcasts an unsolicited RREP packet with the updated hop count and destination sequence number to any active neighbors using this link. Intermediate nodes update their routing tables and forward the packet to their active neighbors. The message will continue to propagate until all upstream nodes using the broken link have been notified. If a route to the particular destination is still required, the source node reinitiates a route discovery with a RREQ packet. The destination sequence number for this packet is incremented to ensure a viable route is discovered [PeR97].

#### *2.3.2.3 Associativity-Based Routing Protocol.*

The associativity-based routing (ABR) protocol is a distributed routing protocol that selects routes based on the stability of the wireless links [Mis99]. Each node in the network periodically broadcasts a beacon signal. Nodes monitor the beacons received from their neighbors, incrementing associativity counters for each beacon signal received from each neighbor. Counts higher than a pre-determined threshold, $A_{threshold}$, imply a link that is stable and long-lived. Conversely, associativity counts lower than $A_{threshold}$ signify high mobility, and consequently lower link stability [Toh97].

To establish a communication route with the desired destination, source nodes initiate a route discovery process by broadcasting a broadcast query (BQ) packet [Toh97]. BQ packets propagate through the network until the destination node is reached. Unique sequence numbers are used to identify each BQ packet and each packet is only broadcast once. Additionally, the source's associativity counts with each of its neighbors is included in the packet, as well as other metrics that aid in the selection of a route [Toh97].

Intermediate nodes receiving the BQ packet check to ensure the packet has not been processed before. If it has, the packet is discarded which ensures no loops develop in the path between source and destination [Toh97]. If not discarded, the node compares its ID with the packet destination ID to see if it is the desired destination. If it is not, node appends its ID to the packet and deletes all of its upstream node's associativity counts, except the one corresponding to itself. This process is demonstrated in Figure 2.6. The source BQ packet in the figure contains all of the source neighbors and their corresponding associativities. At the first intermediate node, IS1, all entries in the BQ packet are removed except IS1, and IS1 adds entries to the packet for each of its neighbors. Counts for each of the intermediate node's neighbors are added to the packet which captures both the route taken, and the associativity levels of nodes in that route. The BQ packet is then rebroadcast.

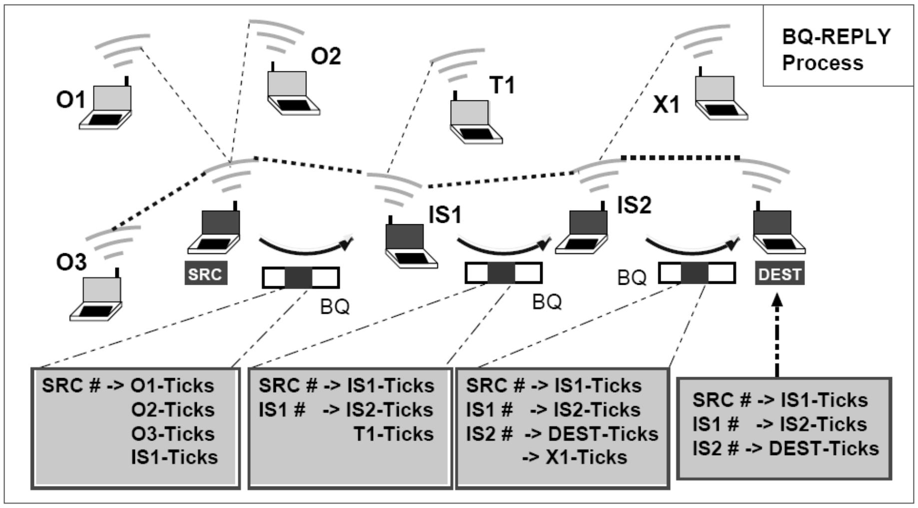


Figure 2.6: ABR broadcast query process [Toh97]

When the BQ packet reaches the destination, the destination node waits for a predetermined period of time in case BQ packets from alternate routes arrive at the destination [Toh97]. At the expiration of the waiting period the destination node selects the best route from those BQ packets received. Routes with highest associativity are selected, with shorter paths being selected in the case of multiple routes with equal associativity. If both of these metrics are equal, an arbitrary route selection is made.

Having selected a route, the destination node sends a REPLY packet along the determined route to the source. This packet contains the complete path, as well as metrics about this route. Intermediate nodes along this path mark the route as valid. Alternate paths remain invalid, preventing duplicate packets from arriving at the destination [Toh97].

Once a route is established, the route maintenance process responds to movements by nodes that invalidate the path. ABR uses a localized query mechanism to reconstruct routes that break due to mobility [Toh97]. The immediate upstream (towards the source) and downstream (towards the destination) node will detect a topology change based on the associativity beacons. In response, the downstream node sends a route notification (RN) packet toward the destination which causes all subsequent nodes in the path to invalidate the route.

The upstream node begins a localized query (LQ) process to discover a partial route to the destination, thus repairing the broken path. The LQ process is similar to the BQ process; however, the LQ process searches for partial routes that are, at most, the same number of hops as the invalidated route [Toh97]. Upon receiving LQ packets, the destination selects the best partial route, and generates a REPLY packet. This packet is sent to the upstream node initiating the LQ packet, with intermediate nodes validating the new route as the REPLY packet progresses along this new path. If, however, the LQ packet fails to reach the destination, implying no partial path of at most the desired length exists, the upstream node completely invalidates its route, and the process backtracks to the next upstream node. This node performs then repeats the LQ process. The process continues to backtrack until the node performing the LQ is more than half of the original route's hop count away from the destination [Toh97]. Rather than perform an LQ, this node sends an RN packet to the source which causes all upstream intermediate nodes to invalidate their routes, and causes the source to initiate a new BQ process.

*2.3.2.4 Other On Demand Protocols.* Besides the three protocols outlined above, there are numerous other on demand routing protocols. The Flow-Oriented Routing Protocol (FORP) [MuM04] uses GPS to predict route handoff. Using GPS location, velocity, and direction information nodes compute the expected lifetime of links. The expected lifetime of a given multi-hop route is the minimum of the lifetimes of the included links. When a node detects a route is about to fail, a handoff process is initiated. This process attempts to find an alternate path to transmit packets so a source node can use this new path without suffering a link failure.

Location-Aided Routing (LAR) [Ily03]also uses GPS. Nodes wishing to establish a route to a destination calculate the destination's expected location based on past information about the node's location and motion, thus defining two zones. The expected zone is the geographic area where the destination is anticipated to be [Ily03]. The request zone establishes a boundary that limits the scope of the route discovery. Once these zones are established, route requests are sent from the source. Depending on the particular algorithm used, requests are discarded by intermediate nodes who are either outside the request zone, or are further from the destination than the source. Otherwise the request is forwarded to the intermediate nodes neighbors. This process continues until the destination is reached and a reply packet is sent to the source establishing the route.

Like ABR, Signal Stability Routing (SSR) [Mis99] attempts to create reliable routes by preferring links with stronger signal strengths. Nodes broadcast periodic beacons to their neighbors. Based on these beacons links are characterized as either strong or weak. Routes are initiated by flooding route requests in the network. Requests received by intermediate nodes over weak links are discarded. Otherwise, the packet is forwarded by the node. Upon reaching the destination node, a route reply is sent back to the source establishing the route. Since request packets are only forwarded over strong links, the route is known to contain only strong links. When no such path exists, the route request times out. The source reinitiates the route discovery, setting a flag in the request packet permitting weak links to be considered in the path. This allows routes to be generated when no strong path exists.

Temporally Ordered Routing Algorithm (TORA) [MuM04] establishes routes by constructing a directed acyclic graph between the source and destination. By developing routes in this manner, TORA is able to simultaneously maintain multiple paths to a given destination. TORA distinguishes itself by its method for handling link failures. Upon the failure of a link, the node detecting the failure reverses the direction of the link to its immediate upstream node. Intermediate nodes continue to reverse their upstream links until the source is reached. Since traffic flows along the directed graph, and the path containing the failed link now points to the source, this path has been effectively invalidated. This mechanism allows the protocol to minimize the impact of link failures by containing the scope of control messages to a small section of the network. Furthermore, it also means the protocol can detect partitions in the network. If a node, which has already reversed its link in response to a link failure, is triggered to again reverse the link's direction, a partition has developed since conflicting information about the proper direction for the link has been received [MuM04].

*2.3.3 Hybrid Protocols.* Hybrid protocols combine the benefits of both table driven and on demand routing protocols. These protocols maintain local topology information in tables reducing the latency associated with route discovery. Routes to distant nodes are established using a route discovery mechanism, as found in on demand protocols. Thus, a hybrid protocol minimizes the number of large route table updates that occur. The combination of these mechanisms is intended to make better use of network resources. The following are examples of hybrid routing protocols.

*2.3.3.1 Zone Routing Protocol.* Zone Routing Protocol (ZRP) establishes routing zones based on the distance to neighboring nodes [Ily03]. A zone radius parameter dictates the size of zones. A node's routing zone consists of the nodes whose distance from a "central" node is at most equal to the zone radius. This is depicted in Figure 2.7 for the central node, S. The nodes inside the circle are in a zone of radius two. Node L is not in the zone because the distance to S is three hops. Routing zones are established relative to each node in the network. As such, the routing zone of each node tends to overlap with the zones of other nodes. ZRP uses an Intrazone Routing Protocol (IARP) to communicate with nodes within a zone. An Interzone Routing Protocol (IERP) establishes routes for transmitting to nodes outside the routing zone [HaP01].

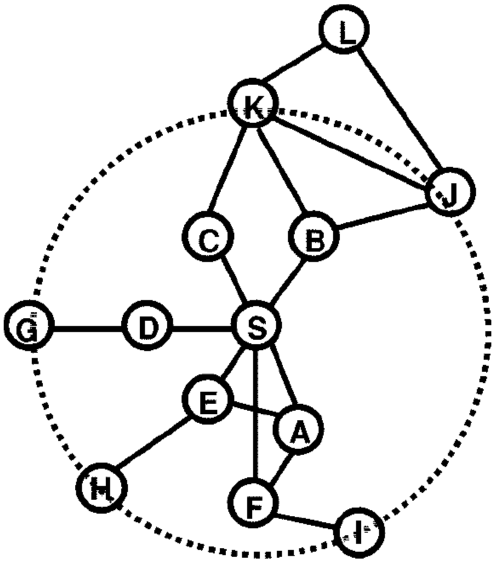


Figure 2.7: ZRP routing zones, defined by hops from central node [HaP01]

To establish routing zones, nodes must first discover which nodes are within their own zone radius. This is accomplished through a variety of methods. The protocol can leverage media access control (MAC) layer information to directly determine a node's neighbors [HaP01]. Alternatively, the Neighbor Discovery Protocol (NDP), through the periodic broadcasting of *hello* messages, can determine active nodes within its zone radius. IARP uses this information to generate routing tables to communicate within the node's zone. IARP is based on traditional link state routing protocols, modified to accommodate the use of zones by limiting the scope of route updates through a time to live (TTL) on route update packets [HaP01].

To communicate with a destination node, a source node checks if the destination is within the routing zone. If it is, IARP routes the packet to the destination [HaP01]. If the node is outside the routing zone, a route to the destination is determined using IERP. To discover routes, IERP uses bordercasting to locate the desired destination node [HaP01]. If the destination node is not within the routing zone, the source generates a route request, and bordercasts it to its peripheral nodes. Peripheral nodes are those whose distance from the source is exactly equal to the zone radius [HaP01]. These nodes check their routing zones to determine if the destination is present. If not, the peripheral node appends its identification and also bordercasts the request. This continues until a node finds the destination within its zone. This node appends the route to the destination to the route record found in the request. A route reply, containing this complete route, is sent to the originator of the request by way of the reverse route. Similar to other protocols described, duplicate route requests are discarded, thus avoiding route loops. Upon receiving any route replies, a source chooses the best route according to a route selection criteria, such as shortest path [HaP01].

This route discovery mechanism has several characteristics that must be addressed [HaP01]. For example, peripheral nodes may be multiple hops from the source. Thus, the source may not have a direct route to these nodes and intermediate nodes in the routing zone may deliver the route request to the zone border. To accommodate this, the bordercasting tree must be appended to request packets [HaP01]. This tree holds the path from the source to the border of each zone. In doing so, it is possible to construct complete routes from source to destination. Additionally, during the route discovery process, each bordercast covers an entire routing zone. However, subsequent bordercasts may overlap this zone, generating redundant and unnecessary route request packets from this zone. There are a number methods to control these query packets [HaP01]. One allows intermediate nodes in a zone to operate in a promiscuous mode. When forwarding or overhearing a route request packet, a node will mark the request as seen. Therefore, when a future bordercast reaches this node, the request can be discarded, limiting the number of redundant route requests in the network.

*2.3.3.2 Core Extraction Distributed Ad Hoc Routing Protocol.* The Core Extraction Distributed Ad Hoc Routing (CEDAR) protocol is a hybrid routing protocol that integrates quality of service (QoS) into a routing protocol [MuM04]. The protocol defines a core set of network nodes in a process called core extraction. This core set approximates the minimal dominating set of the network which is the set of least cardinality in which every node in the network is either in the dominating set, or a neighbor of a node in the dominating set [SSB99]. Core nodes are selected such that there is a core node within three hops of any other core node. Nodes not in the core set choose a core node as their dominating node, and are considered core members for that core node [SSB99]. The core nodes maintain local state information about core members. Furthermore, the path between any two core nodes is considered a virtual path.

Nodes find routes to a particular destination by polling their dominating node, providing the destination and the required QoS. If the destination is also a core member of the dominating node, the route is immediately established. Otherwise, the source initiates route discovery by sending a route request to the dominating node [SSB99]. A core broadcast mechanism is used to establish a core path which consists of a route of core nodes from the dominator node of the source, to that of the destination. In the core broadcast process, the dominator of the source broadcasts a route request to each of its neighbor core nodes. These nodes evaluate their member nodes to determine if the destination is present. If it is not, the request is rebroadcast. Each core node appends its address to the path contained in the request. Once the dominator node of the destination is found, that core node initiates a acknowledgement message containing the core path discovered [SSB99].

CEDAR attempts to find a path that meets the QoS requirements specified by the source node by using a path to the farthest possible core node along the core path that meets the QoS requirement [SSB99]. If this path does not reach the dominator node, the intermediate node performs a similar QoS path discovery. This process repeats until a path is created to the destination that meets the stated requirement. If multiple paths are discovered, the core node for the source chooses the shortest path with the most available bandwidth as the route [SSB99].

Link failures along active paths are handled with a route reconstruction process. Upon detecting a broken link along a path, a node sends a notification message to the source of the packets. The node then begins a local route repair procedure to find a partial route to the destination node. Having received the link failure notification, the source node begins route reconstruction itself [SSB99]. If a partial route cannot be established from the error detecting node, the source establishes a new route itself. This has the potential for packet loss [SSB99]. Nodes detecting link errors drop packets that arrive after the error has been detected. Sources, on the other hand, continue to transmit packets until notification of the failed route arrives. Because of this, packets transmitted after link failure are lost.

### *2.4 Mobility Models*

The routing protocols discussed previously are all designed for wireless ad hoc networks. The performance of each particular protocol varies based on a number of factors. One such factor, critical to the evaluation of protocols in wireless ad hoc networks, is mobility. Research on mobile ad hoc networks must consider the mobility of the network nodes and must model that motion appropriately. Mobility models can be classified into two groups: 1)Entity Models and 2) Group Models. Entity models represent the independent motion of mobile nodes in the network [CBD02]. Each node acts as an autonomous entity, free to move throughout the simulation area in a unique manner. Group models, on the other hand, represent the movement of groups of mobile nodes within a given simulation area. Clusters of nodes move together in the area [CBD02]. Individual nodes in a group move independently, based on an entity model, but are restricted to the vicinity of the group.

### *2.4.1 Entity Models.*

*2.4.1.1 Random Walk Mobility Model.* The Random Walk Mobility Model emulates erratic and unpredictable movements of entities [CBD02]. This model determines the motion of nodes by selecting a random speed and direction from a predefined, uniformly-distributed range of values defined by [*speedmin*, *speedmax*] and $[0, 2\pi]$, respectively. The node continues in motion at the selected speed and along the selected path for a predetermined duration specified as either a set distance, $d$, or a set period of time, $t$. If a border is reached, the node "bounces" off the boundary of the area at an angle determined by the incident direction. Figure 2.8 shows a motion pattern generated using the Random Walk Mobility Model when nodes travel 60 seconds before changing direction. This model has the property of being memoryless. The result of this memoryless quality is that nodes can behave unrealistically. As shown in Figure 2.8, nodes make extremely sharp turns and sudden, and instantaneous, stops.

The motion of nodes under this model is a function of the parameters $d$ and $t$, whichever is used in a particular simulation. These variables must be assigned appropriate values to ensure the model represents the desired mobility. If these parameters are assigned too small a value, nodes are restricted to a small portion of the simulation area which results in a network with a relatively static topology [CBD02]. Thus, small values of $d$ and $t$ should be used to model networks with slower rates of topological change, whereas larger values provide the mobility encountered in more dynamic networks.

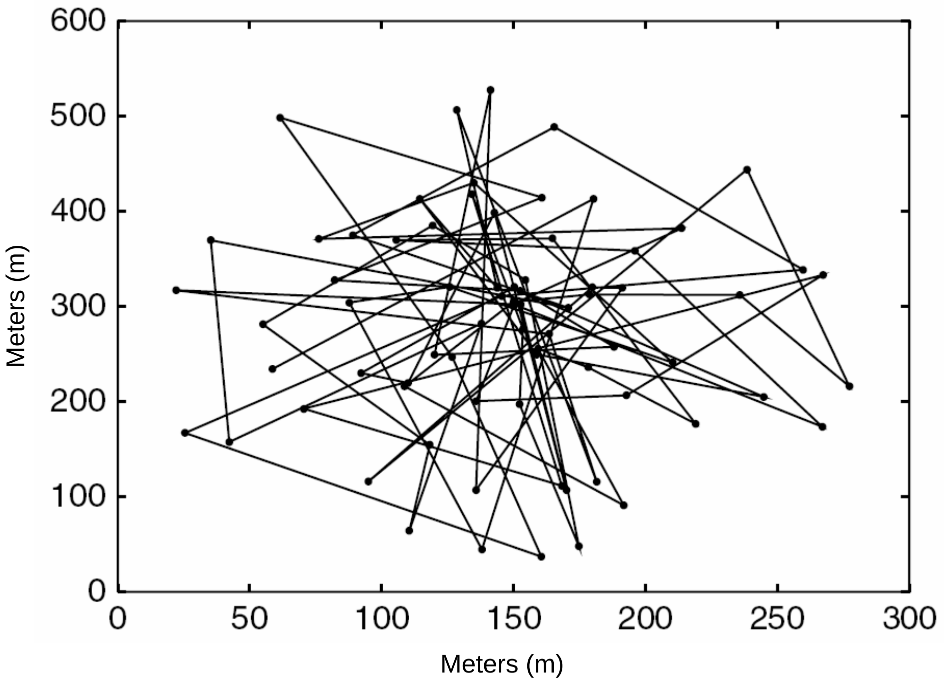


Figure 2.8: Movement pattern of a node using the Random Walk Mobility Model [CBD02]

*2.4.1.2 Random Waypoint Mobility Model.* The Random Waypoint Mobility Model operates in a similar fashion to the Random Walk Mobility Model [CBD02]. Nodes choose, at random, a speed in the uniformly distributed range [*speedmin*, *speedmax*]. However, unlike the Random Walk Model, nodes in the Random Waypoint Model randomly select a destination within the simulation area. The node then moves directly to the selected destination at the determined speed. Upon arriving at this location, the node pauses for a specified period of time and then selects a new speed and destination. The traveling pattern that this generates can be seen in Figure 2.9.

Under this model nodes tend to cluster, particularly in the center of the simulation area [CBD02]. There is high probability that a node's selected destination is either in the center of the simulation area, or requires the node to travel through this area. Node density tends to converge to the center of the simulation area, disperse, and converge again.

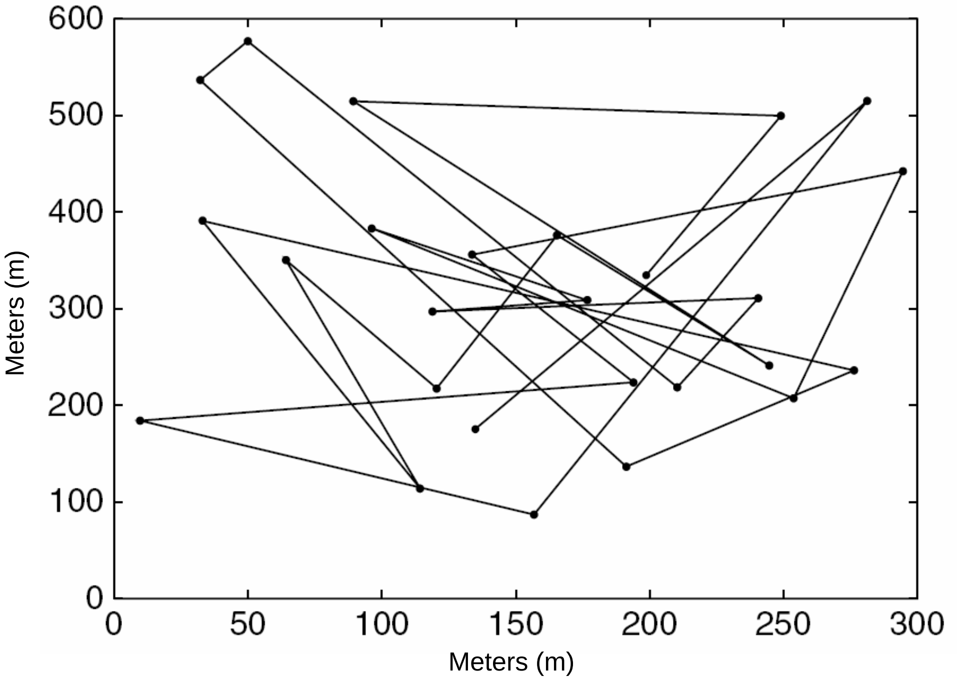


Figure 2.9: Movement pattern of a node using the Random Waypoint Mobility Model [CBD02]

The Random Waypoint Mobility Model can experience some anomalous behavior upon initialization. In most evaluations using this model, nodes are randomly dispersed throughout the simulation area [CBD02] which results in an initial high variability in the percentage of nodes which neighbor each other. The high variability experienced can impact the results of performance evaluations, especially when simulations are of short duration. To overcome this, a simulation can be run beyond the point where variation is high. The locations of nodes at the end of this simulation run can be preserved and used as starting locations for nodes in subsequent simulation runs [CBD02]. By doing so, an initial node distribution is created which does not have such variability in the percentage of neighboring nodes. Alternatively, rather than dispersing nodes randomly, a distribution that more accurately represents the initial distribution of nodes in a system can be used. Finally, simulations can be run for extended periods with the data collected during the period of high variability being discarded. This eliminates the undesirable initial high variance, and results in random initial placement of nodes in the simulation area.

Finally, similar to the Random Walk Mobility Model, the stability of the network is affected by the choice of pause time. If the pause time is long, the network is stable, even with nodes moving with high speeds [CBD02]. Furthermore, a network combining high speed nodes with long pause times results in a topology that is more stable than a network with low speeds and short pause times [CBD02].

*2.4.1.3 Gauss-Markov Mobility Model.* The Gauss-Markov Mobility Model differs from the previous models by using previous information to make mobility decisions [CBD02]. Nodes are initially assigned a speed and direction and continue to move using these parameters for a fixed period of time. When this period ends, the node computes a new speed and direction for the $n^{th}$ iteration based on the values from the $(n-1)^{st}$ iteration. The updated values are computed using:

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)s_{x_{n-1}}} \tag{2.1}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)d_{x_{n-1}}} \tag{2.2}$$

where $\alpha$ is the randomness parameter; $\bar{s}$ and $\bar{d}$ are the mean values of speed and direction as $n \to \infty$, and $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variates from a Gaussian distribution. Furthermore, at each time iteration the next position is calculated using:

$$x_n = x_{n-1} + s_{n-1}\cos d_{n-1} \tag{2.3}$$

$$y_n = y_{n-1} + s_{n-1}\sin d_{n-1} \tag{2.4}$$

Thus this model is not memoryless. The previous values for speed, direction, and position affect future values. This alleviates much of the unrealistic motion produced by the Random Walk and Random Waypoint Mobility Models. Unlike these models, the Gauss Markov Mobility Model does not result in sharp turns or sudden stops as can be seen in Figure 2.10. The movement of nodes under this model is gradual and results in more realistic node movements throughout the simulation area.

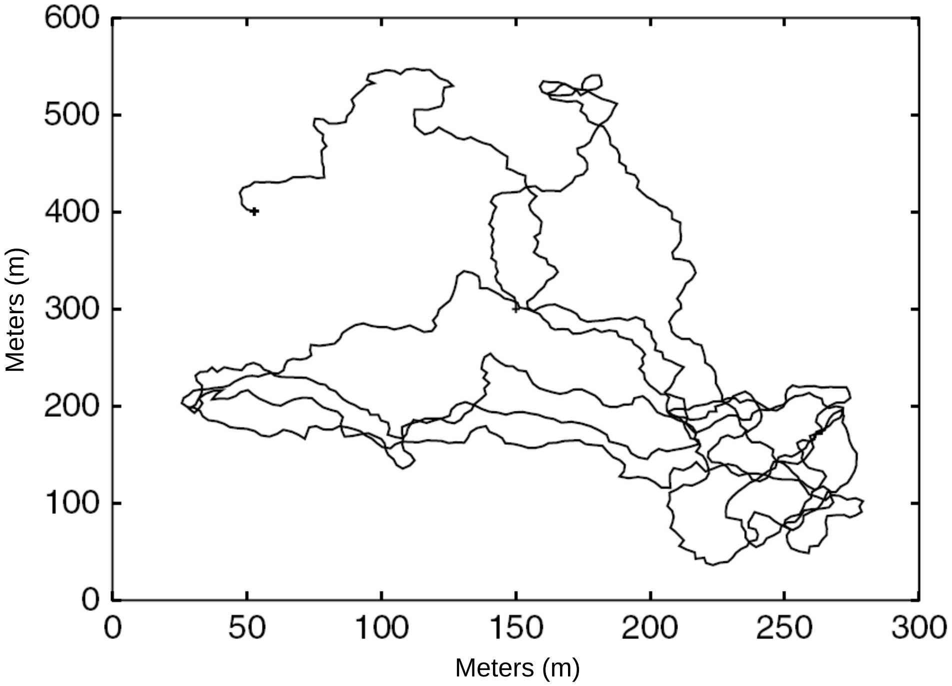


Figure 2.10: Movement patter of a node using the Gauss-Markov Mobility Model [CBD02]

While this method produces more realistic movement, it also has an unintended consequence. Since changes in motion are more subtle, nodes using this model may remain near an edge for extended periods of time [CBD02]. Nodes can be directed away from an edge when they move too close by altering the value of $\bar{d}$ to move away from the edge of the simulation area, preventing the node from loitering near this border [CBD02].

### *2.4.2 Group Models.*

*2.4.2.1 Exponential Correlated Random Mobility Model.* The Exponential Correlated Random Mobility Model produces patterns of motion for both individual mobile nodes and groups of nodes, based on the motion function [CBD02]:

$$b(t+1) = b(t)e^{-\frac{1}{\tau}} + \left(\sigma\sqrt{1-\left(e^{-\frac{1}{\tau}}\right)^2}\right) r \qquad (2.5)$$

where $\tau$ is a Gaussian Random Variable with variance $\sigma$. Small values of $\tau$ correspond to large changes in motion. While inherently simple to calculate, it is difficult to produce a particular motion pattern by selecting appropriate values of $\tau$ and $\sigma$ [CBD02].

*2.4.2.2 Reference Point Group Mobility Model.* The Reference Point Group Mobility (RPGM) model represents the motion of a group of mobile nodes [CBD02]. The motion of the individual nodes is determined by the movements of the group's center. To determine the motion of this central reference point, a group motion vector, $\vec{GM}$, is selected, either randomly or from a predefined value. The combination of the central point and $\vec{GM}$ is used to calculate the motion of the group. A representative traveling pattern for groups of various sizes is shown in Figure 2.11 [CBD02]. In this figure, each line represents the motion of an individual mobile node, while each style of line represents a group of the stated size. The figure demonstrates that while each node has a unique motion pattern, the nodes in a group follow approximately the same path.

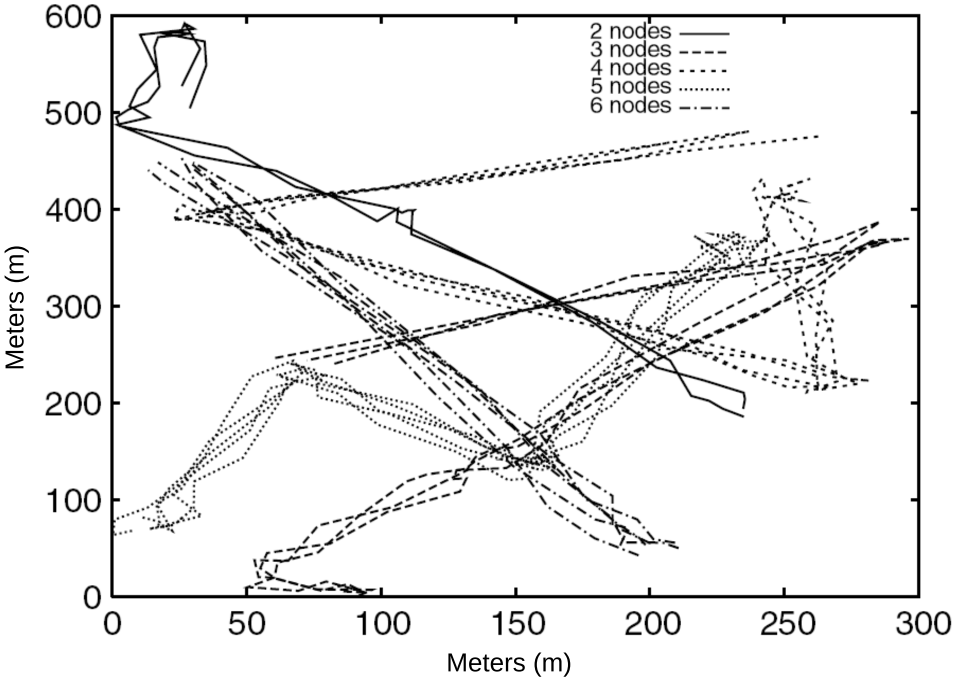


Figure 2.11: Movement pattern of 5 groups of various sizes using the Reference Point Group Mobility Model [CBD02]

The model further captures the random motion of individual nodes within group. Each node within the group moves about an individual reference point [CBD02]. The locations of these reference points for a given time interval is dependent on the motion of the logical center of the group. To calculate the motion of the individual nodes, a random motion vector, $\vec{RM}$, is selected. This vector has a uniformly distributed length within a certain radius of the new reference point [CBD02]. Similarly, the direction of the vector is uniformly distributed between 0 and $2\pi$. $\vec{RM}$ is then summed with the reference point of a node to compute the node's new position [CBD02].

A number of extensions to the RPGM model are possible. With careful selection of initial group positions and group paths, this model can be used for an assortment of applications. One such extension is the In-place Mobility Model [CBD02]. This model divides the simulation area into subsets, assigning each to a group. Each group then operates exclusively in its assigned subset. Additionally, the Overlap Mobility Model [CBD02] simulates different groups, with unique motion characteristics, operating in a given geographic area. Finally, the Convention Mobility Model [CBD02] represents an application of the RPGM model. In this application, the simulation area is divided into subsets, with groups moving throughout the subsets with similar patterns. Each group in this model can have unique motion characteristics.

Furthermore, the RPGM model can implement several other mobility models. The Column Mobility Model [CBD02] uses reference points located linearly on a reference grid. As this grid moves, the nodes of the groups move appropriately to adjust to the motion of their predefined reference nodes. The motion of the nodes in this model can be parallel to the reference grid, similar to a single file line of students, or perpendicular to the grid, comparable to a rank of soldiers marching side-by-side in formation. Similarly, the Nomadic Community Mobility Model [CBD02] can be implemented using the RPGM model. All nodes in RPGM share a reference point about which they move independently. Finally, the Pursue Mobility Model [CBD02] identifies one of the group's nodes as the target. All other nodes will track, or pursue, the target node.

## *2.5 Research in Reliable Routing for Ad Hoc Networks*

There are a myriad of methods to address the challenge of routing in ad hoc wireless networks. Each attempts to provide an efficient method to route packets in an environment where power and other resources are constrained. When evaluating these protocols, one important characteristic to consider is the reliability of the routes produced. Reliability, in this sense, refers to the fault tolerance, or the robustness, of routes developed by the protocol. Significant research has been undertaken to develop constructs for achieving reliable routing. This section examines several examples of such research.

*2.5.1 Ad Hoc On-Demand Multipath Distance Vector Routing.* In [MaD01], an approach is presented to improve the reliability of the standard AODV routing protocol by modifying the AODV protocol to compute multiple link-disjoint paths for each route discovery. To be considered link-disjoint, paths cannot have any common links. The revised protocol, known as Ad Hoc On-Demand Multipath Distance Vector (AOMDV) Routing [MaD01], discovers and maintains multiple independent paths between source and destination thereby improving reliability by substituting a redundant path in the event of the failure of the primary route. Thus, the protocol is expected to eliminate the need for repeated route discoveries, and the increased delay associated with them.

The performance of this modified protocol is evaluated by comparing it to the traditional AODV protocol [MaD01]. In order to vary the mobility of the nodes using the Random Waypoint Mobility Model, the maximum speed is varied. Under this variation, AOMDV outperforms AODV in every metric. Figure 2.12 shows the ratio of packets received to packets sent. While the packet delivery ratio for both protocols decreases as mobility increases, AOMDV loses 3-5% fewer packets than AODV. Similarly, AOMDV drastically reduces the average end-to-end delay almost always yielding a 100% improvement over AODV [MaD01]. This is expected since AOMDV supplies alternate routes upon link failures without delay while AODV must incur the cost of a route discovery with each link break. The frequency of these route discoveries is shown in Figure 2.13. AOMDV provides approximately a 20% reduction in the frequency of route discoveries. Similar, improvements are noted in the ratio of routing control packets [MaD01]. Again, these improvements are expected since the maintenance of multiple routes allows nodes to access new routes without initiating a route discovery. Consequently, the number of control packets required is reduced.

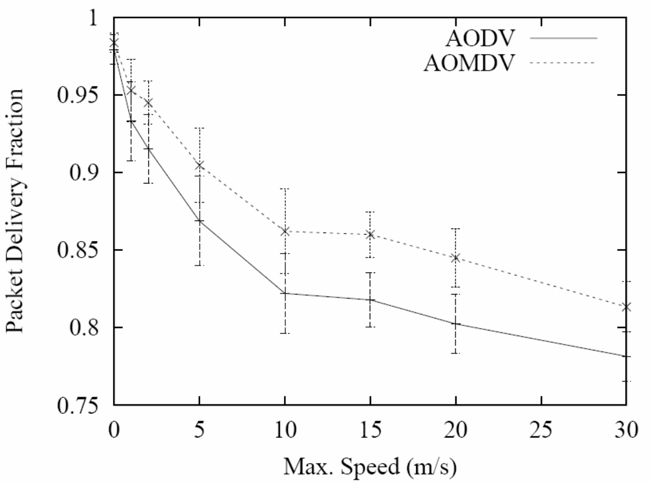


Figure 2.12: Packet Delivery Ratio - AODV vs AOMDV [MaD01]

The impact of varying load on these protocols is also evaluated [MaD01]. AOMDV provides substantial improvements in end-to-end delay. For both protocols, routing control load increases with the number of sessions, with AOMDV demonstrating an improvement as the number of sessions increases. However, at high packet rates AOMDV yields a higher routing load than AODV. This is because AOMDV requires multiple route reply packets in response to a route discovery. At high packet rates, the number of discoveries increases and as the network saturates packets are dropped. Thus, AOMDV can cause higher routing overhead at high packet rates. However, this multipath protocol can be used to increase reliability in dynamic networks where link failures are common. In general, AOMDV outperforms AODV in every metric, pro-

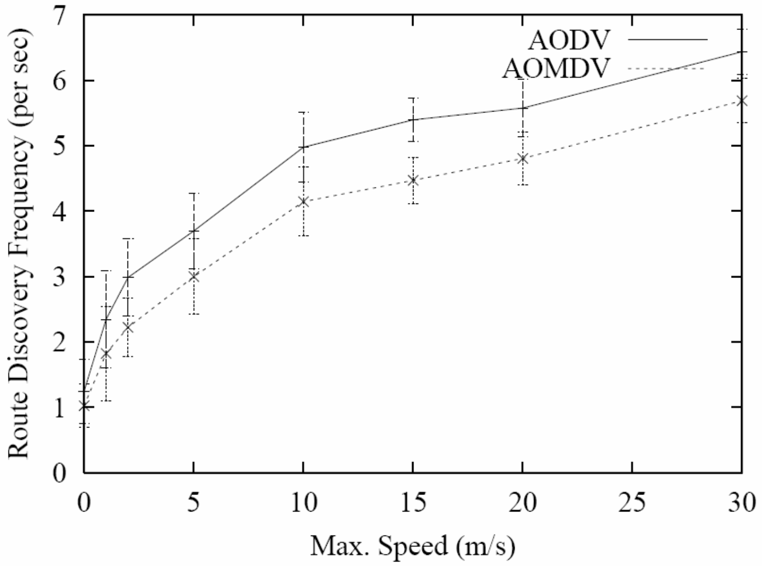


Figure 2.13: Route Discovery Frequency - AODV vs AOMDV [MaD01]

viding substantial improvements in end-to-end delay, lower packet loss, and a lower frequency of route discoveries [MaD01].

*2.5.2 Multipath AODV With Reliable Nodes.* In [YKT03], the concept of a multipath protocol is examined further. Similar to the previous research, [YKT03] uses a multipath modified AODV protocol as the basis for a framework for reliable routing. AODV is modified to incorporate the more stringent requirement of node-disjoint paths. Paths are node-disjoint if the paths share no common nodes. Simulation results of this protocol, using three different node densities, indicate that AOMDV can find at least 80% of the node-disjoint paths found by an ideal search of the network when the network is dense. In a less dense network, the protocol found at least 70% of those found in the ideal case. Furthermore, Figure 2.14 is the probability of finding at least three node-disjoint paths. While high for networks with high density and short paths, this probability decreases quickly when the network has fewer nodes. Therefore, even with moderate node density such as Case 1 in Figure 2.14, the probability of finding a sufficient number of node-disjoint paths to provide the required reliability is low [YKT03].

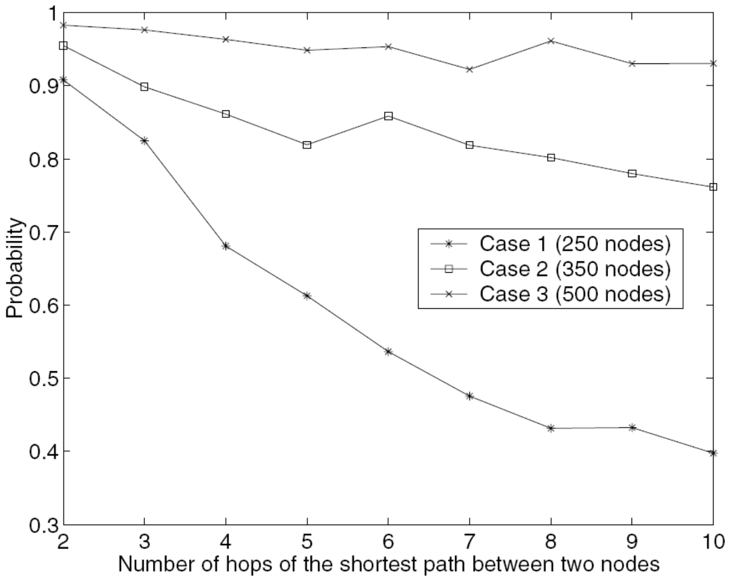


Figure 2.14: Probability that the number of node-disjoint paths is ≥3 [YKT03]

To compensate for a lack of node-disjoint paths, the notion of R-nodes, or reliable nodes, is introduced [YKT03]. R-nodes are unique network hosts with increased reliability, that is, a more capable node, perhaps with regard to power capacity. These nodes are placed strategically throughout the network space to increase the probability of establishing a reliable path. A reliable path is a path between source and destination comprised entirely of reliable segments. A reliable segment is defined as either: 1) a segment comprised entirely of R-nodes, or 2) a path for which there are at least $\kappa$ node disjoint paths between its origin and its termination where $\kappa$ specifies the level of reliability required [YKT03].

R-nodes must be distributed in the network judiciously to increase the probability of finding a reliable path [YKT03]. The min-cut algorithm determines locations in the network that are vulnerable to partitioning and places R-nodes in these locations. R-nodes can also be placed in the vicinity of nodes with low degrees of connectivity, since it is likely that these nodes will be bottlenecks when forming node-disjoint paths. R-nodes placed near neighbors of the low degree nodes attempt to make the links of nodes on the network periphery reliable [YKT03].

Through the use of simulations, the effectiveness of each of these strategies is evaluated [YKT03]. R-nodes constitute 10% of all nodes, and the reliability parameter, $\kappa$, is set to three. Figure 2.15 shows that the min-cut based strategy is a significant improvement over a network with no R-nodes. Furthermore, a random distribution of R-nodes is ineffective, yielding approximately the same probability of finding a reliable path as no R-nodes. This shows that the combination of a multipath routing protocol, and the judicious use and deployment of R-nodes can provide significant improvements in the reliability of routing in a network [YKT03].

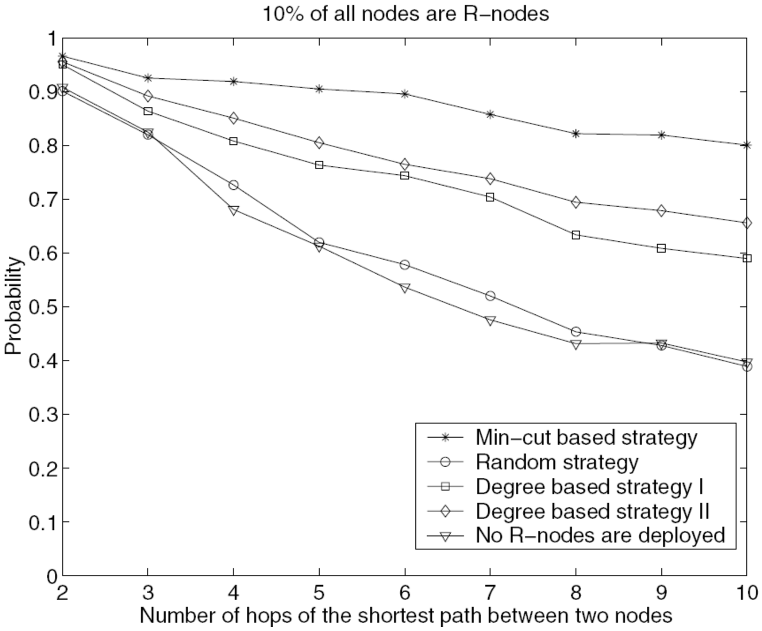


Figure 2.15: Probability of finding a reliable path using various R-node deployment strategies [YKT03]

*2.5.3 AODV Using Link Lifetimes.* An alternative to multipath routing uses links that are stable, as determined by the expected behavior of links based on past behavior [BKS03]. That is, links that have been long-lived, are expected to remain for a long period of time. This is similar to the basic principle of ABR, preferring links with long durations of uninterrupted connectivity. In [BKS03], AODV-REL modifies the AODV protocol to use link longevity as the metric for determining routes. Further modifications add query restriction mechanisms to reduce routing overhead by limiting the scope of route request flooding. AODV-GOD, on the other hand, estimate the residual, or remaining, lifetime of a link by taking the difference of the expected lifetime of the link and the duration of its connectivity. In doing so, the protocol establishes an omniscient knowledge of the perceived link lifetimes.

Simulations of the three protocols (AODV, AODV-REL, and AODV-GOD) shows that for low traffic level, AODV-GOD protocol performs best [BKS03]. Since the extent to which packets flood on a route query is limited routing overhead for the modified protocols is lower, with AODV-GOD performing better than AODV-REL. Similarly, as the number of packets dropped due to the failure of active connections is minimized. AODV-GOD is the best performer. As the level of mobility increases, the number of dropped packets increases, as can be seen in Figure 2.16. The two modified protocols require a greater number of route discoveries, since they are searching for reliable routes based on expected link longevity. Thus, given this requirement, these protocols may reject paths that are acceptable to the standard AODV protocol, resulting in more failed route discoveries [BKS03] as shown in Figure 2.17. Similarly, AODV outperforms the others with regard to end-to-end delay as expected. AODV selects optimal paths, while the others select the path with the highest level of reliability. Finally, AODV-GOD yields the greatest throughput, with AODV-REL outperforming AODV. This is the anticipated result, verifying the improved reliability of these protocols [BKS03].

High load results are similar to the low load. The modified protocols show improved performance in routing control overhead as anticipated since the modified protocols restrict the extent to which routing packets are flooded. Similarly, the modified protocols improve the number of packets dropped due to route failures since the modified protocols' reliable routes fail less often [BKS03]. AODV excels when measuring end-to-end delay since it seeks optimal, shortest path routes and is less selective in its route discovery. AODV also outperforms the modified protocols in the number of failed route discoveries, again because it is less selective in the routes it discovers. There is some unique behavior with regard to throughput. At high levels

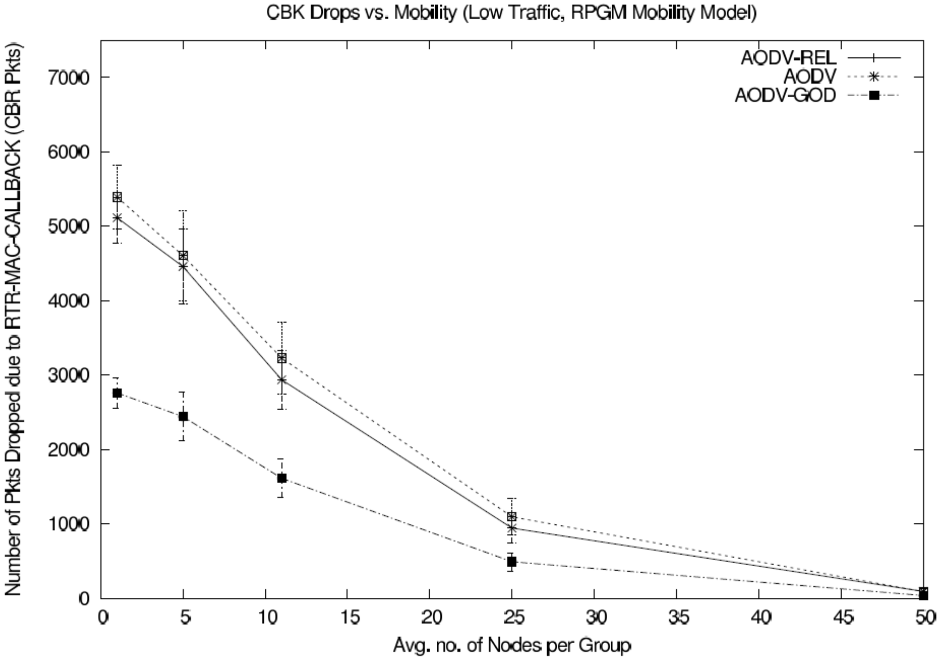


Figure 2.16: Throughput versus Percent mobility [BKS03]

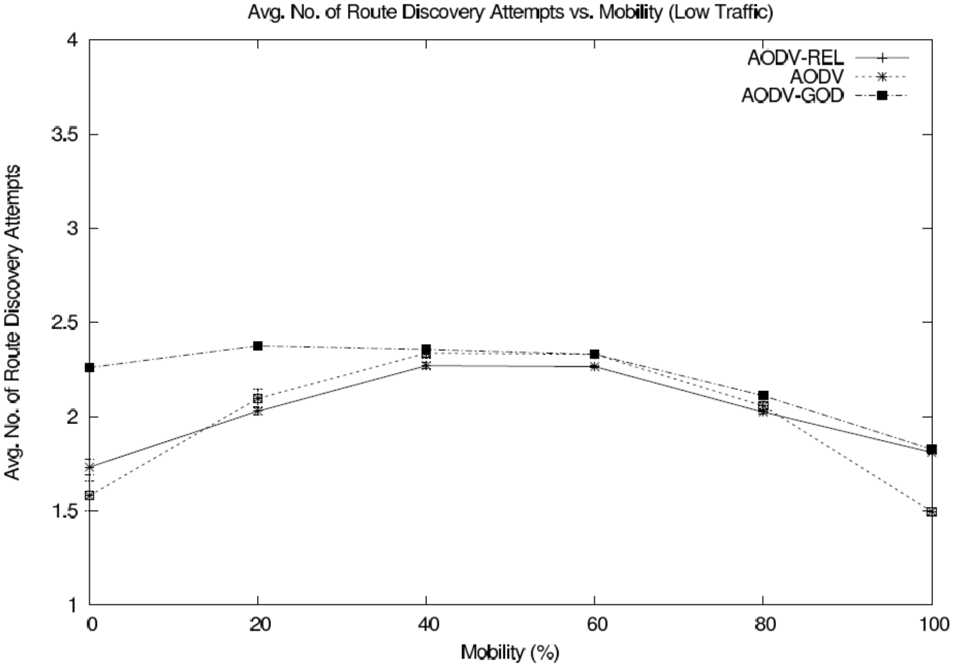


Figure 2.17: Packets lost due to link failure versus Group associativity [BKS03]

of group associativity, implying low mobility, AODV-REL outperforms AODV-GOD. However, since mobility is minimized, it is reasonable that the less restrictive protocol, AODV-REL, would perform better. Thus, these results are consistent with expectations [BKS03]. In general, the significant improvements in performance displayed by AODV-GOD lends credence to the use of link stability as a metric for determining reliability. To achieve its full benefit, however, the reliability metric must be tuned to yield a more accurate approximation of the residual link lifetime [BKS03].

## *2.6 Summary*

This chapter introduces the routing challenges encountered in MANETs. Several unique Table-Driven, On-Demand, and Hybrid routing protocols are identified, and their operation explained. Examples of Entity and Group mobility models offer insight into the ways motion in MANETs can be captured. Finally, this chapter identifies research efforts that have investigated mechanisms to improve the reliability of MANET routing protocols.

# III. Predicted Associativity Routing

## *3.1 Introduction*

This chapter presents the Predicted Associativity Routing protocol. First, the general distinguishing characteristics of the protocol are described. Then, the operation of the protocol is presented in detail. The chapter concludes with a discussion of a pilot study which evaluated the various parameters of Predicted Associativity Routing.

## *3.2 General Description*

Predicted Associativity Routing (PAR) is an on-demand MANET routing protocol. It is specifically designed to address route reliability in ad hoc networks. Using Ad Hoc On-Demand Distance Vector (AODV) Routing as its basis, PAR makes route selection decisions using residual link lifetime information to establish the longest lasting routes possible. In doing so, application layer data can be more dependably delivered. The end result of this is improved throughput. Additionally, the overhead associated with route failures and maintenance can be avoided.

PAR leverages several unique concepts. First, PAR uses associativity to determine link lifetimes. The concept of associativity, introduced in [Toh97], is simply a count of consecutively received HELLO packets from a neighboring node. By maintaining this information for each adjacent node, hosts can determine how long a link has been active, and, given an expected value for that link, how long the link will remain active.

Another distinguishing characteristic of PAR is its use of residual link lifetimes as a metric for selecting routes. Similar to the work in [BKS03], PAR uses the expected remaining lifetime of links to select the best route. In so doing, the protocol uses link performance to establish routes that will remain active for extended durations. This produces more dependable routes, and increases the reliability of the protocol.

While similar to [BKS03] in its use of residual lifetime, the method by which this information is collected and the best route is determined are unique in PAR. By collecting samples of experienced route lifetimes, PAR computes a mean, or expected, lifetime for links. This expected lifetime is particular to each node, and adapts as the conditions of the network around the node change. PAR establishes a zone of associativities about the mean called the No Go Zone. The No Go Zone is a confidence interval on the expected lifetime, and delimits the reliability required by the network to deem a link reliable. If the associativity of a particular link falls in this zone it is either expected to fail soon, or has just passed its expected lifetime and may fail soon. Thus, these links are not considered reliable based on the configuration of the network. In this way, PAR only accepts routes that provide acceptable residual route lifetimes. This is the distinguishing characteristic of the PAR protocol.

## *3.3 Operation of PAR*

*3.3.1 Key PAR Structures.* To operate effectively, PAR maintains several structures to hold information about routes, route requests, and neighbor connectivity. These structures are:

- Route Table: The route table maintains information about active and recently expired routes. Like AODV, PAR populates the common IP route table. Thus, this route table serves only to support the discovery and maintenance of routes. Packets are forwarded according to the common IP route table. Each entry in the route table contains the following information, as well as other data supporting routing:
  - Destination Address: Destination of route.
  - Destination Sequence Number: This information helps ensure that route information is fresh, and that routes are free of loops.
  - Next Hop Address: This is the next node packets are sent to when being forwarded to a destination node.

  - Minimum Residual Associativity: The minimum residual associativity is the residual associativity of the shortest lifetime link on the route to the destination.
  - Reliable Route Flag: Indicates whether the route entry is a reliable route.
  - Route Insertion Time: The time the entry was placed into the table.
  - Route Expiry Time: The time the route is no longer considered valid.
  - Hop Count: The length of the route.
  - Precursor List: A list of upstream neighboring nodes that use the route to a destination.
  - Route Entry State: This field determines if a route entry is valid or expired.

- Route Request Table: The route request table is two lists. The first holds information about route requests that originate at a node. The second tracks information about route requests from other nodes. These lists contain the following information:
  - Destination Address (Originator List Only): This is the destination for the initiated route request.
  - Request ID: The request ID is a unique identifier for the route request.
  - Insertion Time: The time the route request was inserted into the table.
  - Expiry Time: This entry provides two functions. In the originator list, this item is the time a route request times out if no reply has been received. In the forward list, it is the time by which a destination node should reply to a route request.
  - Source Address (Forward List Only): The source address is the originator of the route request packet.
  - Number of Retries (Originator List Only): This is the number of route request retries available before the request is discarded.

- Neighbor Connectivity Table: This table maintains information about neighboring nodes which currently have connectivity with a given node. Each entry in this table contains the following information:
    - Neighbor Address: The address of the neighboring node.
    - Last HELLO Time: This is the time the last HELLO packet was received from this neighbor.
    - Connectivity Expiry Time: If no further HELLO packets are received, connectivity is deemed to have been lost at this time.
    - Neighbor Associativity: This is a count of consecutive HELLO packets received during the current connection with the neighbor.

*3.3.2 Neighbor Connectivity Maintenance.* Nodes in MANETs using PAR evaluate neighbor connectivity using periodic HELLO messages. After broadcasting a HELLO message, the node randomly selects a new HELLO period from the uniform distribution in the range $[d - 0.05, d + 0.05]$, where $d$ represents the desired period for HELLO packets. Thus, collisions from the simultaneous transmission of HELLO packets are avoided. When this period is reached, the node broadcasts a HELLO packet and selects a new period. This process continues indefinitely.

When a node receives a HELLO packet, it updates the appropriate entries in the Route and Neighbor Connectivity Table or adds them if they do not exist. If an entry was added, the Reliable Route flag is set to TRUE and the Minimum Residual Associativity is set to the expected lifetime of the link. For an existing entry, the associativity is incremented to reflect the successful reception of a HELLO packet and the the Connectivity Expiry Time is set to the current time plus the product of a HELLO period (selected using the method described above) and the allowed HELLO loss. The allowed HELLO loss is a parameter of the network intended to give nodes some tolerance for packet loss. The Route Table entry is updated to reflect the most current information, with the Minimum Residual Associativity reset to the expected lifetime.

When the Connectivity Expiry Time for a neighbor is reached, the node deems connectivity with that node has been lost. In this event, the node must add the associativity information in the Neighbor Connectivity Table entry for that neighbor to its knowledge base for calculating expected link lifetimes. The associativity is added to the node's set of lifetime samples. The mean of this set is computed, as well as the standard deviation. This information is then used to re-compute the No Go Zone. The confidence interval is calculated using the t-distribution values [MiA03] for the corresponding size of the No Go Zone. After the number of samples reaches 100, the Normal distribution, Z value, is used in computing the No Go Zone. Once these calculations are made the entry is removed from the Neighbor Connectivity Table and the Route Table Entry is removed. The Route Error Process, discussed in detail below, is initiated to notify other nodes of the failed link.

### *3.3.3 Route Request Process.*

Application layer packets are sent to the routing protocol process when no route is available to the destination in the common IP route table. If these packets are from another node, they are dropped and a Route Error (RERR) is sent, since PAR does not support local route repairs. If, however, the packets are received from the node's own application layer, a route discovery is initiated. Route discoveries consist of broadcasting route requests (RREQ), and the subsequent route reply (RREP) of the selected route.

To transmit an application layer packet a node sends a RREQ packet and adds the request to the Originator List in the Route Request Table. Any application layer packets for the destination are queued until the route discovery is resolved. Figure 3.1 shows the format of RREQ packets. The packet has the following fields:

- Type: Indicates the type of packet.
- U: Flag that indicates that the destination sequence number is unknown.

- I: Flag that indicates that the route discovered thus far is reliable.
- R: Flag that indicates that a reliable route is required.
- Hop Count: Indicates the hop count from the source to the current node.
- RREQ ID: A unique identifier for the request.
- Destination Address: The destination for which a route is sought.
- Destination Sequence Number: The last sequence number the source has on record. This ensures the route provided is not stale.
- Source Address: The address of the originator of the request.
- Source Sequence Number: The current sequence number for the source. This is used in establishing the reverse route.
- Minimum Residual Associativity: The minimum positive residual associativity of the route thus far. This is the remaining lifetime of the shortest lived link in the route.

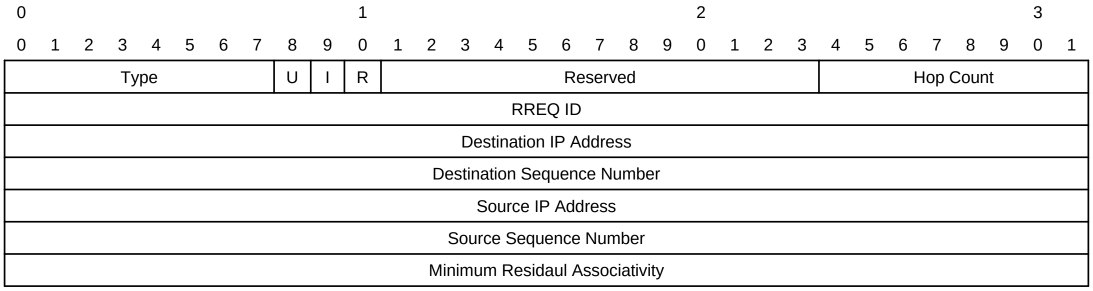


Figure 3.1: RREQ Packet Format for PAR

PAR supports two types of RREQs distinguished by the *R* flag in the packet. If this field is set to TRUE, a reliable route is required. This means all hops in the route must be deemed reliable. The second type of RREQ, where *R* is FALSE, does not require reliable routes. Usually reliable routes will be requested; however, there are two conditions under which non-reliable requests are sent. The first is when a source node has fewer than two associativity samples. In this case, the originator does not have sufficient information to determine the reliability of a link and cannot assume any node has such information. Thus, non-reliable requests are sent to ensure a route, any route, is discovered in a timely manner. Second, if no reliable route exists between source and destination no route will be discovered. The source then attempts to retry the discovery for any available route using a non-reliable request. If no route is found, the discovery is terminated and application layer packets are dropped.

RREQ are dropped if the TTL is exceeded. They are also ignored if the RREQ requires a reliable route, and the node has fewer than two associativity samples since the node does not have enough information to determine if the previous hop is reliable. Similarly, the RREQ is dropped if a reliable route is required and the nodes associativity with the previous node falls in the No Go Zone. This indicates residual lifetime of the link is short and is thus unreliable. RREQ packets are ignored if the receiving node is not the destination, and a request with the same source address and request ID is in the Forward List of the Route Request Table. Such an entry indicates the request has already been processed by the node. If the receiving node is the destination of the RREQ, the packet is dropped if the Route Reply Backoff has expired, meaning a RREP has already been sent. Finally, and perhaps most obvious, the packet is dropped if the receiving node is the originator of the request.

If none of the above conditions hold, the receiving node processes the RREQ. The request is added to the Route Request Table in the Forward List. If the node is the destination of the request, the Route Reply Backoff is set which allows the destination to receive several route requests, from various paths. The destination selects the best, most reliable, route from the received requests. Additionally, the node updates the appropriate values in the RREQ packet. The hop count is incremented to reflect the latest link added to the route and the node examines the residual associativity of the last hop. The node subtracts the associativity in the Neighbor Connectivity Table for the previous node in the path, from the expected link lifetime. If the difference is positive and less than the Minimum Residual Associativity indicated in the packet, the packet is updated to reflect the new associativity. Otherwise, the existing value is preserved.

The node then must establish a reverse route to the source of the RREQ so that the RREP has a route by which to return to the source. If no route is in the Route Table, an entry is added reflecting the information from the RREQ packet. However, if an entry does exist for the source, it may require an update. The current route table entry is updated if the RREQ contains more current information than the Route Table. That is, if the source sequence number of the RREQ is greater than the sequence number for the source found in the Route Table, the entry is updated. Additionally, if the entry in the Route Table is invalid, indicating that it has expired but has not yet been deleted, but the sequence numbers are equal, the entry is updated. Furthermore, if the RREQ packet reflects a reliable route while the table entry is unreliable, but the sequence numbers are equal, the Route Table is updated to reflect the reliable route. Finally, if the sequence numbers are equal, and the Minimum Residual Associativity of the RREQ is greater than the remaining lifetime of the route entry, the route is updated. Reverse routes are allowed to expire if no RREP is received. Thus, only the selected path, both forward and reverse, is preserved by the nodes in the network.

With the reverse route updated, non-destination nodes complete the processing of a RREQ by rebroadcasting the updated RREQ packet to its neighbors. Destination nodes, however, must process the RREQ further. If this is the first RREQ to reach the destination, it has established the Route Reply Backoff when it is added to the Route Request Table. The node retains this RREQ packet. As subsequent RREQ packets are received, their Minimum Residual Associativities are compared against the retained packet. If the subsequent packet has a greater associativity, it is determined to be more reliable. This subsequent packet is retained and the previous packet dropped. In doing so, the node keeps track of the best route received, while ignoring inferior routes. At the expiration of the Route Reply Backoff, the node sends a RREP for the request currently being retained. Thus, the most reliable route of those discovered is used.

*3.3.4 Route Reply Process.* Route Replies are sent by the destination to indicate to the source, and intermediate nodes, the preferred path between source and destination. Using the information from the RREQ packet, the destination creates a RREP packet. Figure 3.2 contains the format of the RREP packets. The fields contained in these packets are:

- Type: Indicates the type of packet.
- I: Flag indicating whether the route is reliable or not.
- Hop Count: Represents the hop count of the discovered route.
- Destination Address: The address of the destination of the discovered route.
- Destination Sequence Number: The most recent sequence number for the destination of the discovered route.
- Source Address: The originator of the route request.
- Lifetime: The duration for which the route is accepted as valid.
- Minimum Residual Associativity: The minimum residual associativity of the route discovered.

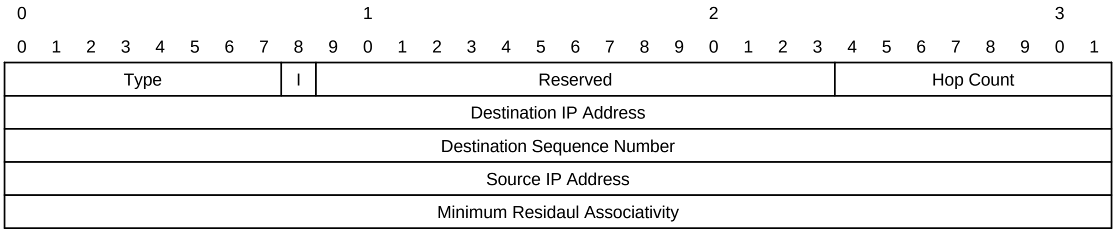


Figure 3.2: RREP Packet Format for PAR

Nodes receiving the RREP packet process it in much the same way as a RREQ. First, a determination is made whether the packet should be handled by the node. Unlike the RREQ process, there are only two conditions in which a node ignores RREP packets. The first is if a destination node receives its own RREP. Secondly, if no route to the source exists, and the node is not the source itself, the node ignores the RREP packet.

If not discarded or ignored, the RREP packet must be updated. Like the RREQ process, the node compares the Minimum Residual Associativity in the RREP packet with that for the previous hop. If the residual associativity of the previous hop is less than that of the packet, the packet is updated to reflect the new associativity. Additionally the hop count is incremented to indicate an additional hop in the route.

Each node establishes the forward path to the destination of the route request. If no entry to the destination exists in the Route Table, the entry is added with the Reliable Route Flag and Minimum Residual Associativity field set to match the respective fields in the RREP packet. However, if the entry does exist in the Route Table, the node uses the criteria outlined in the RREQ Process determines if the entry should be updated. When nodes update or add Route Table entries during the RREP process, they also update the precursor lists for both the forward and reverse paths. That is to say, the node adds the upstream node on the route to the Precursor List for the destination node. Similarly, the downstream node is added to the Precursor List for the source node to ensure the node is aware of all neighbors depending upon it to provide routes to these destinations.

Once the Route Tables are updated to reflect the new route information, the RREP packet is forwarded to the next hop along the reverse path to the originator of the route discovery. Nodes continue to process the RREP in this same way until the source is reached. The originator of the discovery also processes the route packet in this way. However, the source node takes an additional step and deletes the entry in the Originator List of the Requst Table corresponding to the received reply. Additionally, the source node cancels the Route Request Expiry Timer since a route has been discovered.

*3.3.5 Route Error Process.* Route Error packets are sent for two reasons. First, if a node receives a application layer packet, but has no route to the destination of that packet, a RERR is sent. Secondly, a node will send a RERR packet if it detects a broken link. RERR packets notify upstream nodes of failures in routes. The format of these packets is shown in Figure 3.3. The fields of a RERR message are:

- Type: Indicates the type of packet.
- Dest Count: A count of the number of unreachable destinations.
- Dest Address: Address of the unreachable destination (This field is repeated for each unreachable destination).
- Dest Sequence Number: The greatest known sequence number for the unreachable node (This field is repeated for each unreachable destination).

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td colspan="16">Reserved</td><td colspan="8">Dest Count</td></tr>
<tr><td colspan="32">Unreachable Destination IP Address</td></tr>
<tr><td colspan="32">Unreachable Destination Sequence Number</td></tr>
<tr><td colspan="32">Additional Unreachable Destination IP Address (if needed)</td></tr>
<tr><td colspan="32">Additional Unreachable Destination Sequence Number (if needed)</td></tr>
</table>

Figure 3.3: RERR Packet Format for PAR

Nodes process both types of Route Errors in approximately the same way. If the error is generated in response to an application layer packet, the node adds the destination of that packet to the list of unreachable destinations in the RERR packet. This process is slightly more complex if a node detects a link failure. In these circumstances a node searches its Route Table. Any destination in the table for which the failed link is the next hop is added to the unreachable destination list.

With all unreachable destinations identified, the node must determine which neighbors rely on it for routes to these unreachable destinations. By examining the Precursor Lists, a node can determine which nodes must be notified of the failed route. If no such precursor nodes exist, the node simply invalidates the entries in its own Route Table. Alternatively, if a single node must be notified, the RERR packet is unicast to that neighbor. Otherwise, the RERR is broadcast. Once the RERR is sent the Route Table entries for each unreachable destination are invalidated and their Precursor Lists are destroyed.

Upon receiving a RERR packet, a node processes it in the same way the originating node created the packet. The node examines the Route Table entries for each listed unreachable destination. If there are precursors for the unreachable destination, the RERR packet must be forwarded. As before, if there is a single precursor the RERR packet is unicast to that neighbor, otherwise it is broadcast. The node invalidates its route entry, and destroys the Precursor Lists for the unreachable nodes. This process continues until all nodes comprising the failed routes have been notified, and the routes invalidated.

### *3.4 Results of Pilot Study*

The Predicted Associativity Routing protocol uses several new parameters, and places increased emphasis on others. A pilot study is conducted to examine these parameters and their effect on the routing protocol. In particular, the pilot study determines the protocol configuration used in the main experiment.

There are three factors in this pilot study. The first is the Hello Periodicity. This factor has three levels: one packet per second, two packets per second, and five packets per second. Additionally, the size of the No Go Zone, corresponding to the confidence level of the interval, is varied among three levels: 20%, 50%, and 80%. Finally, the duration of the RREP Backoff assumes values from two levels: 0.1 s and 0.05 s. The pilot study is a full factorial experiment of these factors. Each network configuration experiment is run five times for a total of 90 experiments. These experiments are run on a network of 50 nodes. Each node uses Random Waypoint Mobility with speeds zero to five meters per second. Finally, each node produces application layer traffic at a rate of two packets per second.

Since the focus of the main research is the reliability of the protocol, this pilot study primarily examines the route lifetimes of the different configurations to find the best configuration to compare against AODV. The results of the pilot study for route lifetime are depicted in Figures 3.4 and 3.5. Visual analysis of these figures, indicates that PAR achieves its best route lifetimes when a HELLO periodicity of one packet per second is used. This is contrary to the expectation that higher HELLO periodicities would yield greater route lifetimes due to the higher precision of link lifetime samples. However, as is explained in detail in Chapter V, frequent link failures occur in PAR so the longer HELLO periods mean nodes won't detect link for longer periods of time, thereby increasing route lifetime.

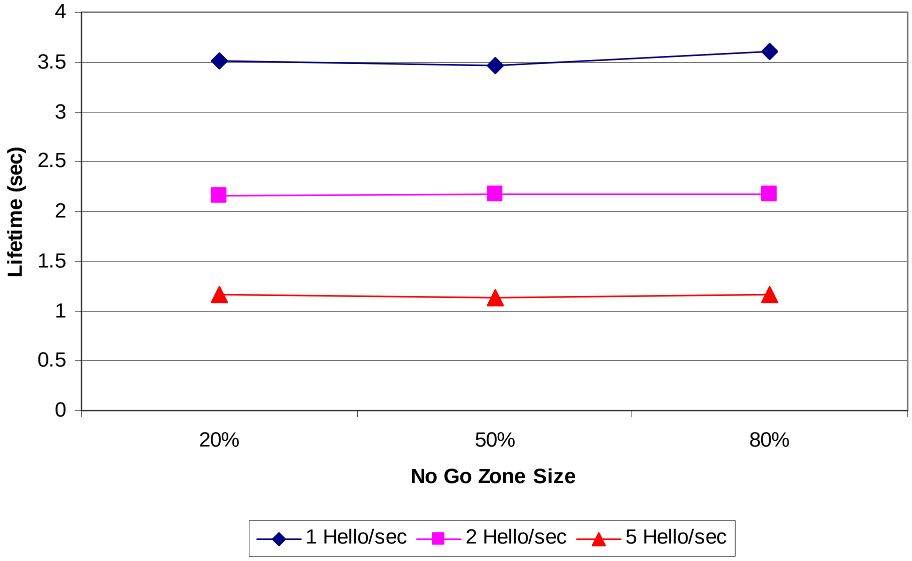


Figure 3.4: Route Lifetime versus No Go Zone Size for RREP Backoff of 0.05 s

Further inspection of Figures 3.4[1] and 3.5 reveal interesting observations about the other factors of the experiment. The line for each HELLO periodicity is almost horizontal, with only minor variations. This implies the size of the No Go Zone has little impact on the route lifetime. Similarly, comparing the lines for the each periodicity in the two figures demonstrates that PAR produces routes with similar route lifetimes, regardless of the value of the RREP Backoff. Therefore, the factor that has the only significant impact on route lifetime is HELLO periodicity.

---

[1] Confidence intervals for this, and similar figures, are generally to tight to be seen on the figures. Information regarding confidence intervals for these figures can be found in Appendix A.

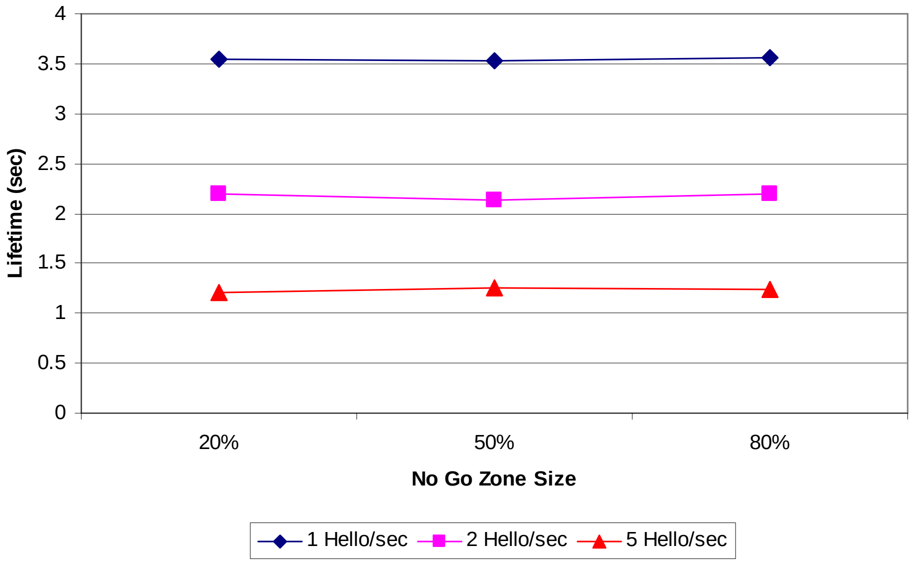


Figure 3.5: Route Lifetime versus No Go Zone Size for RREP Backoff of 0.1 s

Based on these observations the configuration that produces the greatest route lifetime is selected. It is clear a HELLO periodicity of one packet per second should be used. However, since the remaining factors have little impact on route lifetimes, the configuration of HELLO periodicity of one packet per second, RREP Backoff of 0.05 seconds, and No Go Zone Size of 80% is selected. This configuration, referred to hereafter as the research configuration, is used to compare with AODV.

Analysis of the other results of this pilot study support the selection of the research configuration. Figure 3.6[2] represents the end-to-end delay produced under the various configurations of PAR. Examination of the figure shows that delay is minimized when the HELLO periodicity is one packet per second. Thus, while it does not achieve the minimum delay, the research configuration provides near minimum results for end-to-end delay.

Similar results occur for the amount of application layer data received. Figure 3.7 depicts the application data received for a RREP Backoff of 0.05 seconds.

---

[2] With few exceptions, the factors have similar effects on the other metrics collected in the pilot study as they did on route lifetime. Therefore, only the figures for RREP Backoff of 0.05 are presented here. The remaining figures can be found in Appendix A.

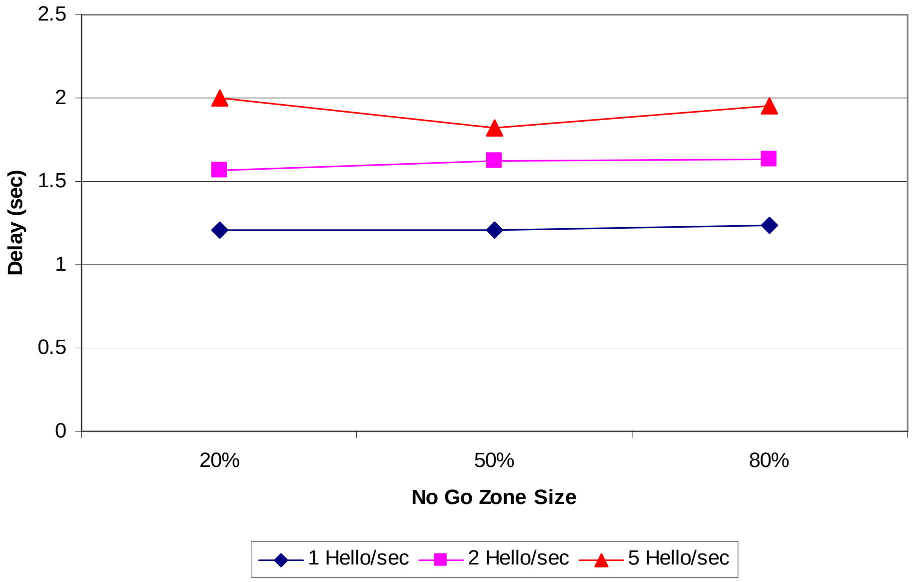


Figure 3.6: End-to-end Delay versus No Go Zone Size for RREP Backoff of 0.05 s

The figure shows that, once again, HELLO periodicity of one performs the best. This is to be expected, since it is known that this periodicity produces the longest lasting routes. These long lasting routes permit greater amounts of data to be transmitted without being interrupted by failed routes. Again, the research configuration does not maximize this statistic. However, it performs at near maximum levels, while producing the longest lasting routes.

Likewise, similar conclusions are drawn for routing overhead, depicted in Figure 3.8. In this case, HELLO periodicity of one packet per second produces the least routing overhead. This stands to reason, as a periodicity of one packet per second reduces the number of HELLO packets that are sent. While route discoveries and route errors also contribute to this statistic, the reduction in HELLO messages significantly reduces the overall overhead of the protocol, more than cutting it in half as compared to a periodicity of five packets per second. This further supports the selection of the research configuration.

When the amount of data successfully received is examined compared to the routing overhead, a measure of efficiency can be determined. Figure 3.9 shows the

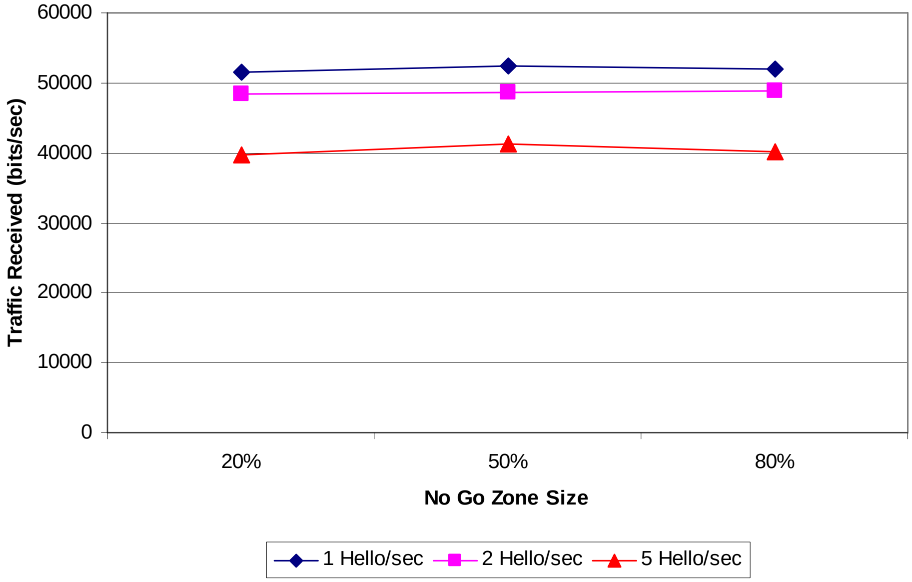


Figure 3.7: Data Traffic Received versus No Go Zone Size for RREP Backoff of 0.05 s

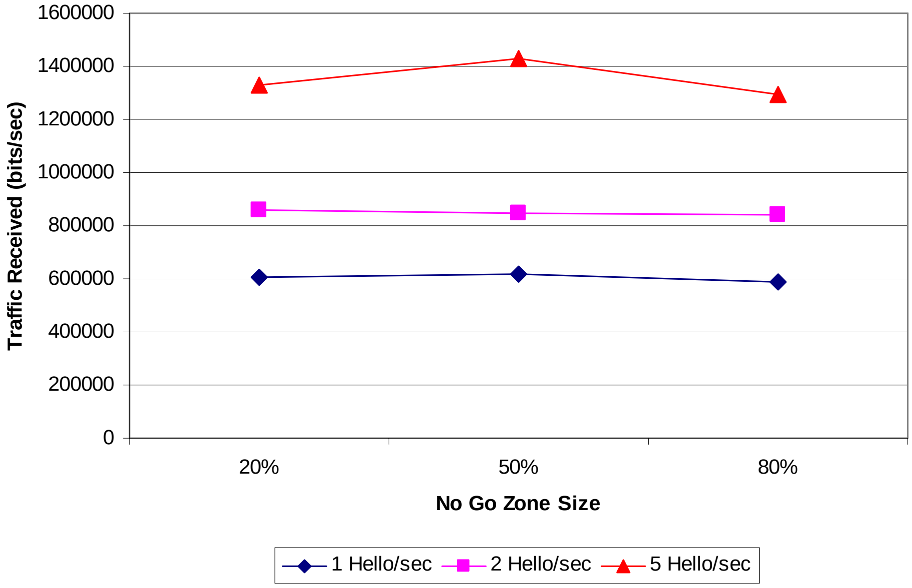


Figure 3.8: Route Traffic Received versus No Go Zone Size for RREP Backoff of 0.05 s

efficiency of the network. Once again HELLO periodicity has the largest effect on the metric. And, as with the other statistics collected, a periodicity of one produces the greatest efficiency. Taken in the context of previously examined statistics, this is expected. Periodicity of one delivers greater amounts of application layer data, while minimizing routing overhead. Thus, the research configuration is efficient, further validating it as a sound choice for comparison to AODV.

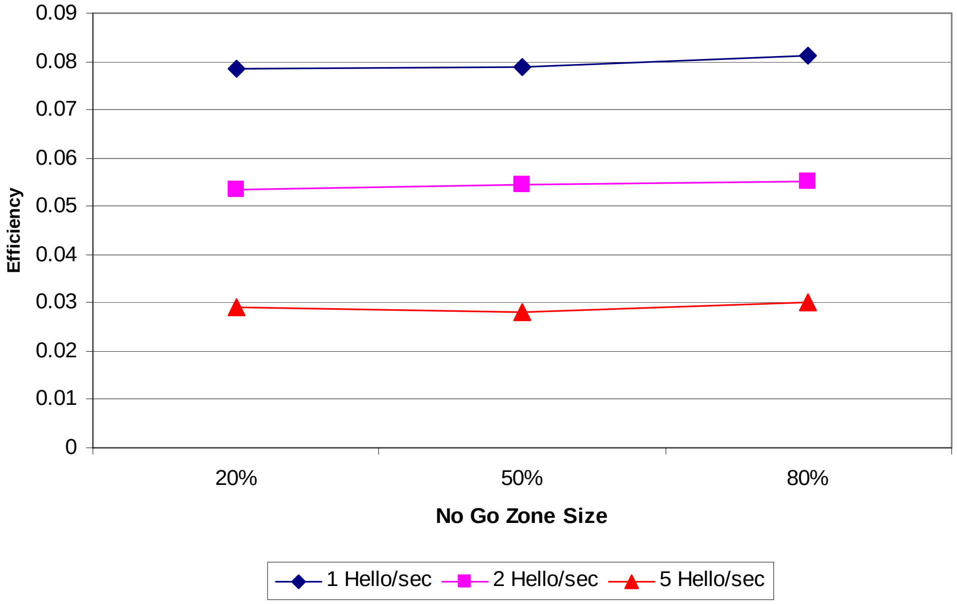


Figure 3.9: Efficiency versus No Go Zone Size for RREP Backoff of 0.05 s

The number of reliable routes discovered stand in contrast to the previous results. While previous statistics show that PAR favors a HELLO periodicity of one, this statistic shows that more reliable routes are discovered using higher periodicities. This result, while contrary to other results, is expected. Higher periodicities produce associativities that more precisely represent the actual lifetimes of the links. Thus, the protocol can distinguish, at high periodicities, two associativities that would be indistinguishable at low periodicities. This means that more links fall in the No Go Zone at low periodicities, resulting in fewer reliable routes discovered. Hence, a periodicity of one is least favorable for producing large quantities of reliable routes. However, ultimately, the routes produced by PAR with a periodicity of one last longer than those discovered when a periodicity of two or five is used.

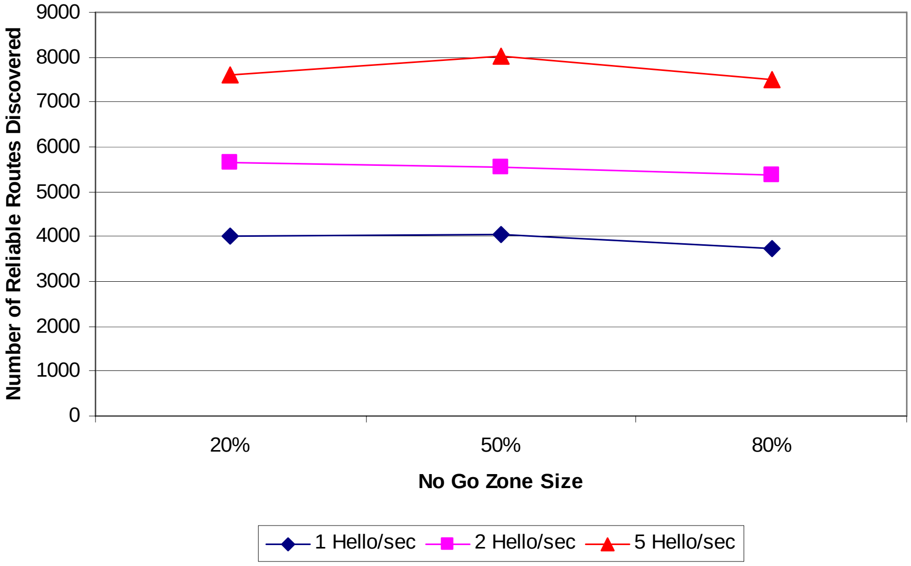


Figure 3.10: Reliable Routes Discovered versus No Go Zone Size for RREP Backoff of 0.05 s

### *3.5 Summary*

This chapter examines the details of the Predicted Associativity Routing protocol. PARs general distinguishing characteristics are discussed. Furthermore, the detailed operation of the protocol and its structures are outlined. Finally, the results of a pilot study on the protocol are analyzed. These results are used to select a configuration of the PAR protocol to experiment with in the primary research of this effort.

# IV. Methodology

## *4.1 Introduction*

This chapter outlines the methodology used to experiment with, and evaluate the performance and reliability of, MANET routing protocols. The problem is defined, laying out the goals and hypotheses of this research in Section 4.2. Section 4.3 defines the system under test. Section 4.4 outlines the system's services. Sections 4.5, 4.6, and 4.7 describe the system workload, performance metrics, and system parameters, respectively. The experiment's factors are listed in Section 4.8. Section 4.9 discusses the evaluation techniques. Finally, Section 4.10 addresses the experimental design.

## *4.2 Problem Definition*

*4.2.1 Goals and Hypothesis.* MANET routing protocols often focus on finding optimal paths between a source and a destination. However, in many cases reliable paths are preferable to optimal ones. Reliable routes can provide higher quality of service, and in some cases, improve the overall performance of the network. Many paradigms have been devised to provide MANETs with reliable routing schemes. The primary goal of this research is to improve the reliability of the routes produced by the network. More specifically, this research evaluates the impact of modifying the AODV protocol to collect link lifetime information, and to use this information as the metric for selecting routes. The research addresses not only the impact of the modified protocol on the reliability of the network, but also its performance.

Modifying the AODV protocol in this manner is likely to improve the overall reliability of the routes produced. Increasing route reliability has ramifications for several areas of system performance. First, it is expected that fewer active routes will experience route failures compared to the AODV protocol. Since the modified protocol, known as Predicted Associativity Routing (PAR), tracks the expected lifetime of the links composing a route, it is less likely a route near the end of its lifetime is chosen and route lifetimes are expected to increase.

This increase has some consequences on other aspects of system performance. One such implication is a greater amount of application data will be successfully transmitted. Greater lifetimes imply fewer route failures. Thus, routes have to be repaired less frequently. Since, the route is active for a greater portion of time, the amount of data transmitted is expected to increase.

The improved reliability also carries some liabilities. First, additional information must be collected to estimate link lifetimes. As outlined in the description of the PAR protocol in Chapter III, this is done through HELLO packets. This periodic information increases routing overhead. Similarly, selecting reliable routes implies other, unreliable routes, are ignored. Thus, route discovery is expected to fail more often further adding to routing overhead.

End-to-end delay is expected to increase since route discoveries are expected to fail more often. Thus, application packets are forced to queue for extended periods during route discovery. Furthermore, PAR prescribes a delay before responding to received route requests which further increases the end-to-end delay. Therefore, the delay of PAR is expected to exceed that of AODV.

Finally, the throughput of the PAR protocol is expected to increase compared to AODV since both the amount of application layer traffic transmitted and the amount of routing overhead are anticipated to rise. Thus, it is hypothesized that the throughput of PAR is greater than that of AODV.

*4.2.2 Approach.* To achieve the goals outlined above, the AODV protocol gathers information about the link lifetimes experienced by nodes in the network and uses it to compute expected link lifetimes. The difference of the expected value for link lifetime and the experienced lifetime yields the residual lifetime which PAR uses to select routes. Using a series of simulations, reliability and performance metrics are observed. These metrics show the impact of the modifications on the reliability of the routing protocol. In particular, it is possible to conclude if the reliability of the modified protocol is greater than the base protocol, thus confirming the research hypothesis. Furthermore, the collected performance metrics allow conclusions to be drawn about the impact reliability has on overall system performance.

### *4.3 System Boundaries*

The system under test (SUT) consists of the components that comprise a MANET, and is depicted in Figure 4.1. First, the system includes a collection of mobile hosts which serve as communicators. Additionally, the area of operation is included in the system because the size of this region can have a significant impact on system performance. Next, the system includes node mobility. The mobility model defines the movement of the nodes in the area of operation. The system also includes the MAC layer protocol, in this case IEEE 802.11. The final component in the system is the routing algorithm being used which is also the component under test (CUT). Since a goal of this research is to evaluate the effects of the selected routing protocols on the system, it is clear that the routing protocol is the appropriate CUT.

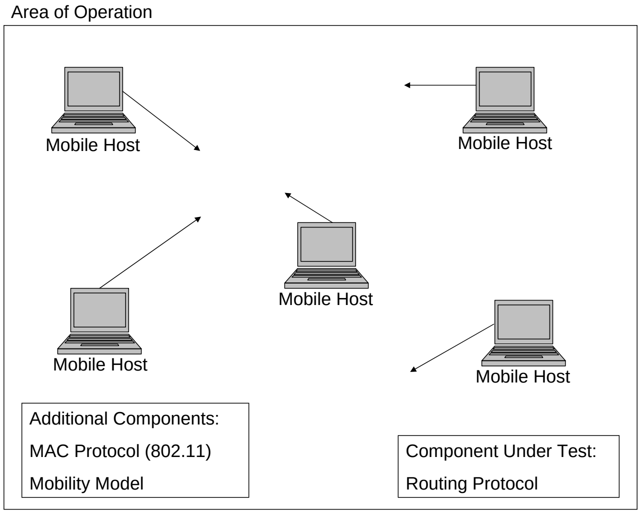


Figure 4.1: System Under Test

This study evaluates the reliability and performance of routing protocols. As such, the impact of the MAC layer protocols and wireless radio are not considered. The effect of group mobility is not considered. Rather, the study is limited to individual node motion using an entity mobility model. Finally, power consumption is ignored.

## *4.4 System Services*

One of the most fundamental services a network provides is a data transfer service. This service transmits data from a sending node to a destination node. A data transfer is considered successful when the destination node receives the data. There are a variety of failure modes for this system service. The failure modes considered as part of this research are failures due to failed routes. While modeled in the simulations, failures due to congestion or interference are not considered in the scope of this study.

A second system service is route discovery. This is the process of finding a viable path between a sender and a receiver. For this process there are two possible outcomes. The first is success, defined as the establishment of a complete route. Alternatively, the second outcome is failure. This second outcome may be due to partitions in the network. In the PAR protocol, failures may be caused by lack of a reliable route to the destination.

A final system service is route repair. This service repairs failed routes by finding alternate paths between source and destination. A successful outcome results in the restoration of service through an alternate route. A failure occurs when there is no alternate route through which service can be restored. This may happen when the destination is unreachable, again, perhaps due to network partitions, or lack of a reliable route.

## *4.5 Workload*

The workload submitted to the system is the data to be transferred over the network. This workload affects the system in a number of ways. First, data transfers from a source node triggers the route discovery process. To evaluate the reliability of routes generated by a protocol, the routes must first be established. Providing data to the system to transfer ensures routes are available to evaluate. The frequency of data transfers determines the number of routes a system must generate. Furthermore, by transferring data over active links, routes are kept active (i.e., in use) allowing an accurate appraisal of reliability.

Additionally, data transfers impact the performance of the system. Transfers can increase the congestion on the network which can increase the amount of time required to transmit a packet. As such, end-to-end delays are impacted by data transfers. Furthermore, network loading can contribute to errors in packet delivery which directly impacts the performance and reliability of the network.

## *4.6 Performance Metrics*

To evaluate the system, several reliability and performance metrics are collected. These metrics are:

- Route Lifetime: The lifetime of the route is the time between the route's establishment (i.e., the transmission of a route reply by the destination) to its failure. This metric is used to evaluate the reliability of the protocols. Longer route lifetimes indicate a protocol produces more reliable routes.
- Throughput: Throughput is the number of successfully transmitted bits divided by the elapsed time for the network. This metric is establishes a correlation between reliability and system performance. In particular, this metric demonstrates that more reliable routes yield a higher throughput.
- Efficiency: Efficiency is defined as:

$$Efficiency = \frac{Data_{Rx}}{(Data_{Rx} + Route_{Rx})} \tag{4.1}$$

  where $Data_{Rx}$ is the number of data bits per second received, and $Route_{Rx}$ is the number of routing protocol bits per second received. This metric is included as a measure of the efficiency of the protocols. It is expected that the PAR protocol is more efficient than AODV.

- End-to-End Delay: This metric is defined as the average time required to transfer a data packet from the source node to the destination node. This delay metric is used to establish the hypothesis that while increasing reliability, the PAR protocol also increases average end-to-end delay.

## *4.7 Parameters*

*4.7.1 System Parameters.* The list below are parameters that affect the performance of the system.

- Number of Nodes: The number of nodes in the system, along with the size of the area of operation, affects the node density of the system. This, in turn, affects the level of connectivity of the network. With higher levels of connectivity, more routes are available between source and destination, making it more likely that a reliable route can be found. Thus, the number of nodes in the network can greatly impact reliability of the system.
- Size of Area of Operation: This parameter, coupled with the number of nodes, determines node density. An operating area of 500m by 500m is chosen to provide nodes ample room to move. Furthermore, these dimensions force routes to be selected that are of significant length.
- Mobility Model: The mobility model determines the way mobile nodes move. Since routes fail due to the mobility of the nodes, movement patterns are important to consider. The Random Waypoint Model is used in a variety of research including [BKS03], [YKT03], and [MaD01]. As such, it is selected as a reasonable mobility model for this research.
- Node Speed: Node speed affects the level of mobility of the nodes. The mobility of the nodes directly impacts the lifespans of links in the network. High levels

of mobility result in shorter link lifetimes and so node speed can greatly impact route reliability.

- Routing Protocol: This parameter is the focus of the research. The routing protocol determines how the routes are chosen. Thus, the protocol is ultimately responsible for the reliability of the routes. Furthermore, the manner in which these routes are determined has a significant impact on system performance. Therefore, the protocol used must be considered when performing performance evaluations of networks.
- Reliability Thresholds: Reliability thresholds determine the level of route reliability required for a route to be selected by the routing protocol. This level is determined by the amount of anticipated lifetime a route must have for the protocol to deem the route reliable. Since this parameter is involved in the selection of routes, it has great influence over the reliability of the routes.
- Transmission Range: The transmission range of the nodes determines the maximum distance of communication. If the distance between nodes exceeds this parameter, nodes cannot communicate. Transmission range impacts the level of connectivity in the network. The level of connectivity impacts the reliability of the routes in the network by varying the number of routes possible. This value is set to 100 meters, a reasonable range for 802.11 MAC protocols [KaO02].

*4.7.2 Workload Parameters.* The following list describes the workload parameters that impact the performance of the system:

- Number of Sources/Destination Pairs: This parameter can also be considered the number of conversations occurring in the network. Each of these conversations requires a route first be established. By increasing the number of conversations, more routes are established. Thus, this parameter can greatly impact the reliability and performance metrics. In this research, all nodes generate traffic, selecting random destination nodes from the network.

- Packet Interarrival Rate: The packet interarrival rate determines how frequently data packets are sent. By varying this parameter, varying levels of network loading can be achieved. Loading leads to contention in the network which subsequently affects the performance of the system.
- Packet Sizes: Packet size is also a contributor to the loading experienced by a network. Similar to the previous parameter, packet sizes can impact the delays and throughputs experienced on the network. This value is set to 1024 bits.

## *4.8 Factors*

This list specifies the factors, and corresponding levels, used:

- Node Density:
  - Low Density: To represent a network with low node density, and a corresponding low degree of connectedness, 25 nodes are used. By selecting a small number of nodes, the possible routes between source and destination are limited and reliable routes will be more difficult to establish.
  - Medium Density: 50 nodes provide a system configuration with medium levels of connectivity. As such it is expected that some reliable routes are available, although not for every desired route.
  - High Density: The high node density level is 75 nodes making a variety of routes available for a source. Thus, it is more likely that a reliable route can be found.
- Node Speed
  - Low Speed: A range of 0 to 5 m/s is assigned to this level. The Random Waypoint model selects a speed from this range and moves the node at the selected speed. At this speed, links are likely to remain valid for longer periods of time, resulting in higher route stability.

    - High Speed: For this level, 0 and 20 m/s is used. High levels of mobility reduce link lifetimes and route reliability will decrease.

- Routing Protocol
    - AODV: This protocol is the base system. AODV uses hop count to make route selection decisions.
    - PAR: This protocol is a modification of AODV, designed to determine an expected value for link lifetimes, and use this knowledge to select long-lived routes.
- Offered Load
    - Low: At this level, application layer packets arrive at a rate of 2 packets per second. This provides a low offered load to the network and provides smaller numbers of route discoveries.
    - High: The high level corresponds to 4 packets per second. With the higher levels of application layer packets more route discoveries are initiated, and the network experiences greater levels of loading.

### *4.9 Evaluation Technique*

This research uses simulations to evaluate the reliability and performance of the routing protocols. Simulation is selected as the evaluation technique for a number of reasons. Direct measurements are impractical and the cost associated with establishing a network on which measurements can be made is prohibitive. Additionally, controlling the various parameters of the system in a live testbed would be problematic. Therefore, simulation is the best technique for this evaluation.

The simulations are run using OPNET version 10.5.A which includes models for the AODV protocol. These models serve as a basis for the custom PAR models that must be created. The code base for AODV is modified to include estimation of link lifetimes, and the use of residual lifetimes to select routes.

For valid results, the models must accurately represent the system. Since the AODV models are provided by the manufacturer and comply with the draft standard for AODV, they are assumed to be valid. Therefore, they are used to validate the PAR models. In order to validate the PAR models, the protocol is configured as closely to AODV as possible. Simulations are run on both protocols and the results compared.

### *4.10 Experimental Design*

To ensure a complete evaluation of main effects and interactions, a full factorial experimental design is used. There are four factors, three having two levels and the fourth having three levels. This results in $3 * 2 * 2 * 2 = 24$ experiments.

The inclusion of mobility in these simulations is expected to increase the variance in collected results. As such, it is necessary to run several iterations of each experiment to ensure the variance is at an acceptable level. Thus, ten iterations of each experiment are conducted. The result is 240 experiments must be run. The data collected is evaluated using 90% confidence intervals.

### *4.11 Summary*

There are a number of protocols to address the problem of routing in Mobile Ad Hoc Networks. Many, if not most, of these protocols select routes based on the length of the route, favoring shorter routes. However, it may be preferential to select routes which are expected to be long lasting. Predicted Associativity Routing uses an estimation of link lifetimes to determine expected residual lifetimes. Using these residual lifetimes as the basis for selecting routes, PAR attempts to improve the reliability of the routes in the system. This research compares to the PAR protocol to a standard MANET routing protocol, AODV, in terms of both reliability and performance.

There are several expected results from this research. PAR possesses a knowledge of link lifetimes and uses this knowledge to select routes. As such the routes selected are anticipated to have longer durations. Therefore, the lifetime of routes selected by PAR should be greater than those selected by AODV. For this same reason, it is expected that PAR will deliver a larger amount of application layer data and thus demonstrate an improvement in throughput over AODV. Similarly, with more data packets delivered, efficiency is expected to improve with the use of the reliable protocol.

While reliability is expected to improve with the use of PAR, there are certain costs associated with this improvement. First, PAR must gather information about the performance of links which requires additional routing overhead. As such, the amount of routing traffic is expected to be greater with PAR. Additionally, since PAR selects routes based on reliability, and not shortest path, the end-to-end delay is expected to be greater.

This chapter outlines the experimental methodology used to test the PAR and AODV routing protocols, and analyze their performance and reliability. The system is defined, its services outlined, and its parameters described. Furthermore, the factors of the experiment are detailed, and reliability and performance metrics are identified. Finally, the experimental design is explained in detail.

# V. Experiments, Data, and Analysis

## *5.1 Introduction*

This chapter presents the results and analysis evaluating the reliability and performance of the AODV and PAR routing protocols. Section 5.2 describes the validation of the custom routing protocol, PAR. Section 5.3 outlines the methods of experimentation and data collection. Sections 5.4, 5.5, 5.6, and 5.7 discuss the results of the research for route lifetime, throughput, delay, and efficiency, respectively. The analysis examines the effects of each factor on the reliability and performance of the protocols.

## *5.2 Routing Protocol Validation*

To validate the models created for the Predicted Associativity Routing Protocol a comparison is made between the custom models, and the Ad Hoc On-Demand Distance Vector Routing Protocol models provided with the OPNET 10.5.A wireless module. These AODV models are the basis for comparison throughout this research and are assumed to be valid.

In order to validate the PAR models, a series of experiments are conducted and the results compared to ensure similar trends are followed by each protocol. In these experiments, the protocols are configured to be, to the greatest extent possible, identical. Therefore, the PAR protocol is configured to operate without a No Go Zone or RREP Backoff, as AODV does not use these parameters. Furthermore, PAR is configured to accept any available route, rather than require searches for reliable routes, since this is AODV's mode of operation. The AODV protocol, under normal operation, uses an expanding ring search during route discovery. Since PAR does not operate in this manner, the starting time to live for AODV is set to the network diameter. All other parameters of the protocols are set identically.

The validation experiments are a full factorial design of four factors. The first is the protocol used, PAR and AODV. Next, the levels for node density used are, 25, 50, and 75 nodes. The offered load levels are two and four packets per second (pps), and the node speed levels are 0-5 and 0-20 meters per second. Each experiment is repeated ten times and the results for route lifetime, throughput, delay, and efficiency are collected and compared. Figures 5.1[3] through 5.4 are the results for node speed 0-5 m/s. Results for node speed 0-20 m/s are included in Appendix B, but omitted here as they repeat the trends seen in the results for node speed 0-5 m/s.

Figure 5.1 shows there is a significant difference in the route lifetimes of AODV and PAR. This difference can be explained by a fundamental, and unavoidable, difference in the way the protocols determine connectivity with their neighbors. AODV updates the connectivity expiry timer after the receipt of any packet. This includes RREQ, RREP, and application packets in addition to HELLO packets. PAR, in contrast, only updates the connectivity timer in response to HELLO packets. Thus, a missed HELLO packet may cause a loss of connection in PAR, where it may not in AODV. These losses in connectivity result in lower route lifetimes in PAR.

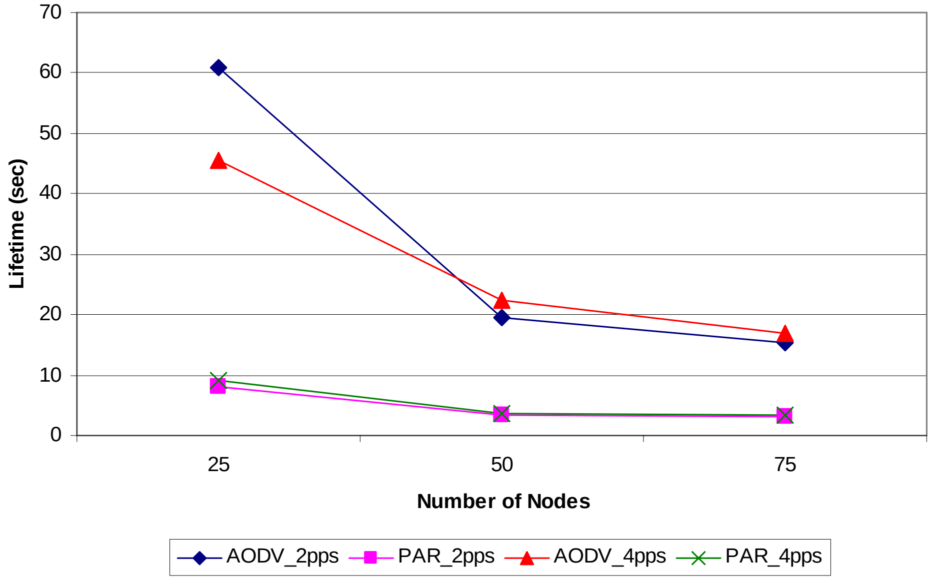


Figure 5.1: Route Lifetime versus Node Density at 0-5 m/s

---

[3] Confidence intervals for this, and similar figures, are generally to tight to be seen on the figures. Information regarding confidence intervals for these figures can be found in Appendix B.

Similarly, the disparities in throughput seen in Figure 5.2 can be explained. PAR's more frequent losses in connectivity result in higher routing overhead since a loss of connection with a neighbor can trigger the transmission of a RERR packet, as well as initiate a route discovery to repair failed routes. Increasing the amount of routing protocol packets sent increases the total amount of data sent, thus increasing overall throughput.

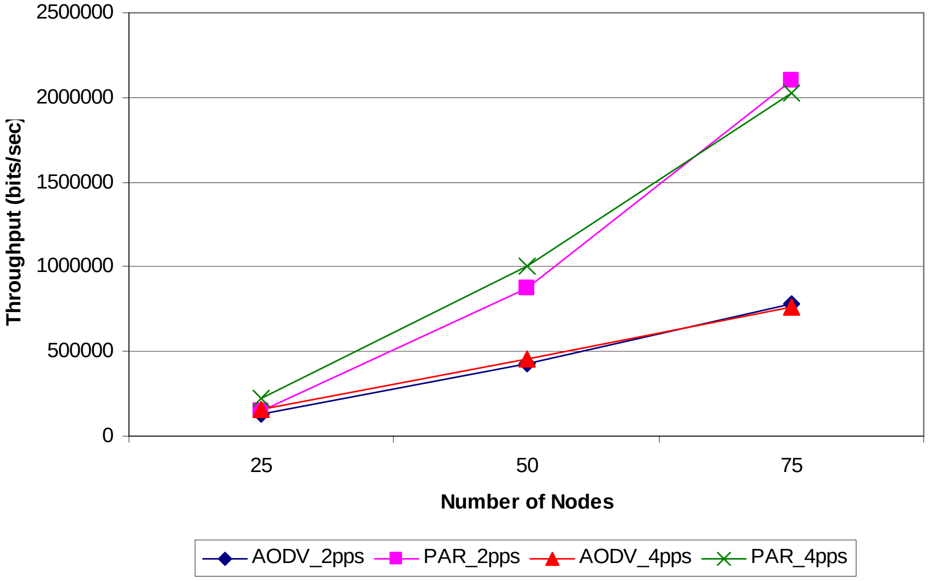


Figure 5.2: Throughput versus Node Density at 0-5 m/s

The characteristics of PAR explaining its higher throughput, can also explain the disparity seen in end-to-end delay in Figure 5.3. Route failures initiate new route discoveries to repair the failed routes. This forces application layer packets to queue while a route is found. The result is increased delay, particularly in larger networks, where routes may be substantial in length. Another reason for this disparity is a difference in the criteria for route selection in the two protocols. AODV is designed to find shortest path routes to a destination. PAR, on the other hand, is intended to seek longer lasting routes. Thus, routes discovered by AODV may not be accepted in PAR, even when the protocol is not searching for reliable routes exclusively. This more stringent route selection criteria can also increase delay.

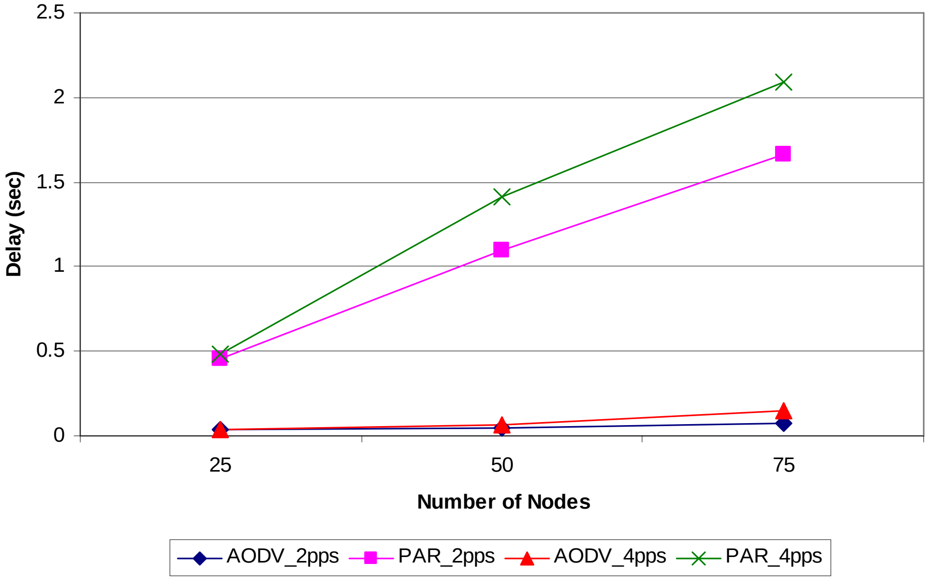


Figure 5.3: Delay versus Node Density at 0-5 m/s

PAR efficiency in Figure 5.4 is also impacted by the increase in routing overhead. Efficiency indicates the ratio of application data delivered to the total data delivered, both application and routing data. As the route overhead increases, the efficiency is driven down. Since PAR, by its nature, results in higher overhead, the efficiency for this protocol is less than that of AODV. This disparity is magnified as the node density increases, since more links fail.

It is clear there are differences between PAR and AODV in the reliability and performance metrics presented. As discussed, these disparities are the result of the fundamental differences in the protocols. However, if the trends are considered, the responses of both protocols to the factors of the experiments are similar for all metrics with the exception of Delay. These trends indicate that the factors of the experiment affect like configured instances of the PAR and AODV protocols in similar ways.

The observations made in this section lead to important conclusions about the PAR protocol. The differences in the magnitude of the protocols' responses to the factors of the validation experiments indicate that PAR and AODV are unique protocols. The fundamental, and unavoidable, properties of PAR make it a distinct protocol,

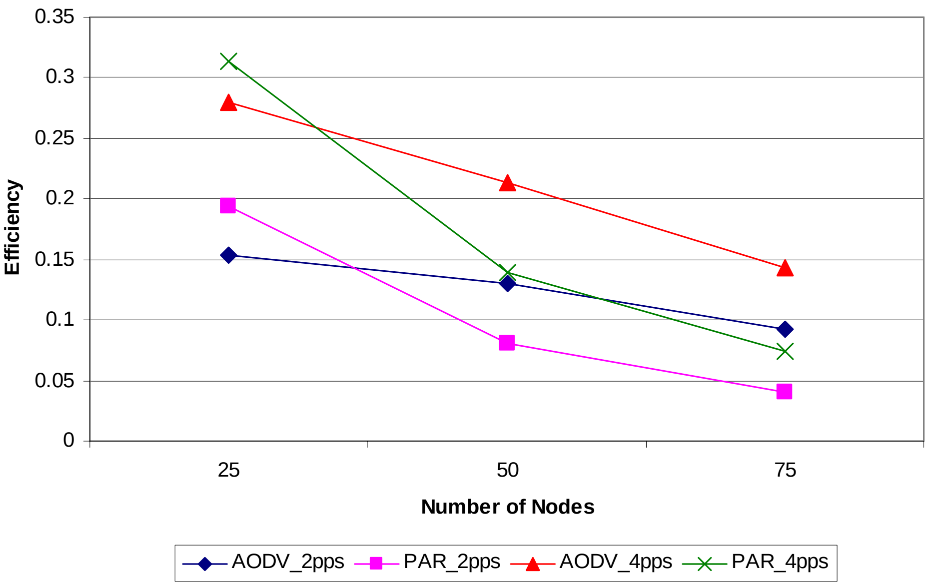


Figure 5.4: Efficiency versus Node Density at 0-5 m/s

despite its roots in AODV. However, the similarities in the trends shown by the protocols show that PAR can exhibit AODV-like behavior. Ultimately, the validation experiments conclude that PAR is, at its essence, unique from AODV, and, as such, can not be validated directly against AODV. The similar trends observed, however, permit valid comparisons to be drawn between the protocols, since PAR demonstrates similar behavior to AODV.

## *5.3 Data Collection Methods*

The experiments are run for a total of 16 simulation minutes. The first 60 seconds of each simulation are discarded to ensure that the network has reached a steady state. In doing so, the effects of initialization do not skew the results. Furthermore, since the Random Waypoint Mobility Model is used, discarding the first 60 seconds ensures a random starting network configuration.

The metrics are collected using OPNET statistics collection functions. Each metric, is collected in “bucket” mode, with each bucket set to five seconds. In bucket mode, individual statistics are collected for the specified duration and collected in the "bucket". At the end of this interval, a predetermined operation is performed on the bucket data. The metrics in this research are either averaged or summed. The result of this operation constitutes a single sample of the metric. In all cases, 90% confidence intervals are used.

### *5.4 Route Lifetime Analysis*

Route lifetime is the mean lifetime of active routes that have failed. In the context of this research, longer route lifetimes imply greater reliability. Both AODV and PAR are analyzed for each factor. Additionally, a computation of effects is included to determine the impact of the various levels of the factors on the route lifetime. Finally, an ANOVA is accomplished to determine the quantify the portion of total variance contributed by the factors.

*5.4.1 Routing Protocol.* It is expected that the route lifetime for PAR would be greater than that for AODV. However, Figure 5.5 indicates that the opposite is true. In fact, for each node density level PAR is inferior to AODV. The figure indicates at each level AODV has approximately three times the route lifetime.

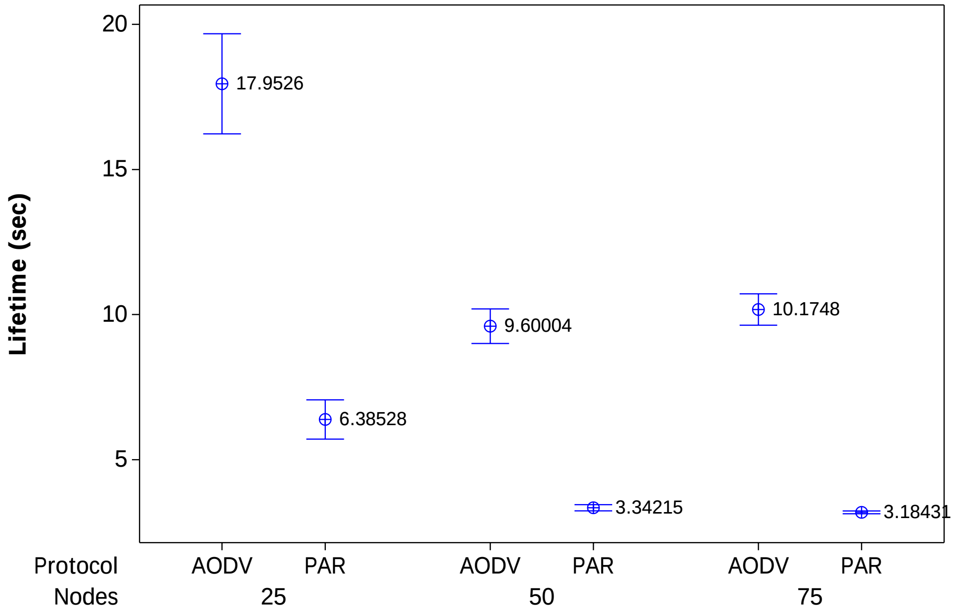


Figure 5.5: Plot of Route Lifetime by Protocol and Node Density

This unexpected result can be explained by examining the manner in which the protocols handle neighbor connectivity. AODV updates a neighbor connectivity expiry timer with every packet received by the routing protocol. PAR, on the other hand, only updates the expiry timer with HELLO packets. Thus, links fail more often with the PAR protocol. Figures 5.6[4] and 5.7 demonstrate that link breaks for PAR exceed those of AODV at each node density and at each speed and offered load. Thus, the route lifetimes for AODV are greater than those of PAR.

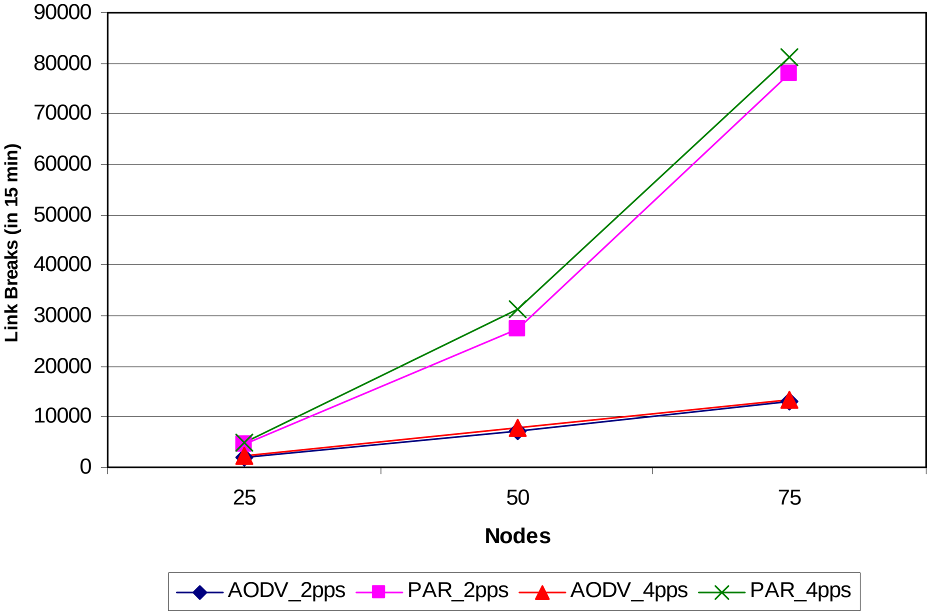


Figure 5.6: Link Breaks versus Node Density at 0-5 m/s

*5.4.2 Node Density.* Further analysis of Figure 5.5, reveals the effects of node density on the route lifetime. Increasing node density from 25 nodes to 50 nodes results in a halving of the route lifetime for both protocols. However, increasing from 50 nodes to 75 nodes has a smaller effect. This can be explained, once again, by examining Figures 5.6 and 5.7. In these figures there is a significant increase in the number of link breaks as the node density increases. These link breaks add a greater

---

[4] Confidence intervals for this, and similar figures, are generally to tight to be seen on the figures. Information regarding confidence intervals for these figures can be found in Appendix D.

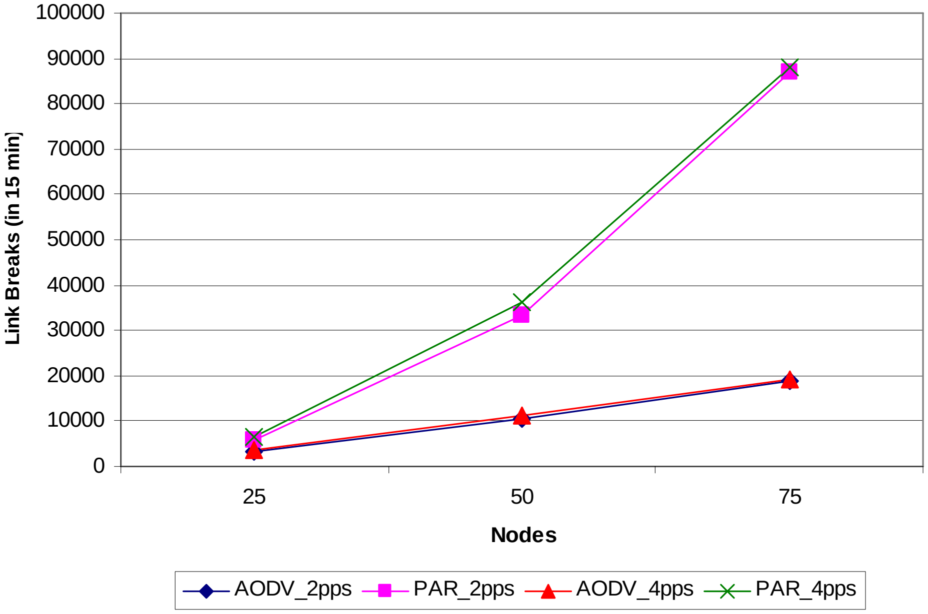


Figure 5.7: Link Breaks versus Node Density at 0-20 m/s

number of samples of low route lifetimes thereby forcing a decline in route lifetime for both protocols. However, as the node density increases even further, the level of connectivity also increases. Thus, while there are increased link breaks, there are also significantly more moderate and long lasting routes. The effects of both of these facts ultimately maintain the route lifetime at its current level.

*5.4.3 Node Speed.* As the speed at which the nodes move increases, the amount of time to leave the range of a neighbor decreases. With this in mind, it is expected that increasing node speed results in a decrease in the lifetime of routes. Figure 5.8 verifies this expectation. For both PAR and AODV, and for all node densities, route lifetimes for 0-5 m/s exceed those for 0-20 m/s. As was discovered in Section 5.2, regardless of the node speed, AODV provides more reliable routes than PAR. However, as Tables 5.1 and 5.2 indicate, PAR is less sensitive to changes in speed than AODV. Figure 5.8 verifies this; there is little difference between the route lifetimes for the two speed levels for the PAR protocol. This is particularly true for 50 and 75 node densities. The higher levels of connectivity achieved at high densities have a stabilizing effect on route lifetimes by providing significantly more routes over which to sample. Thus, the impact of short lived routes resulting from rapid link breaks is decreased.

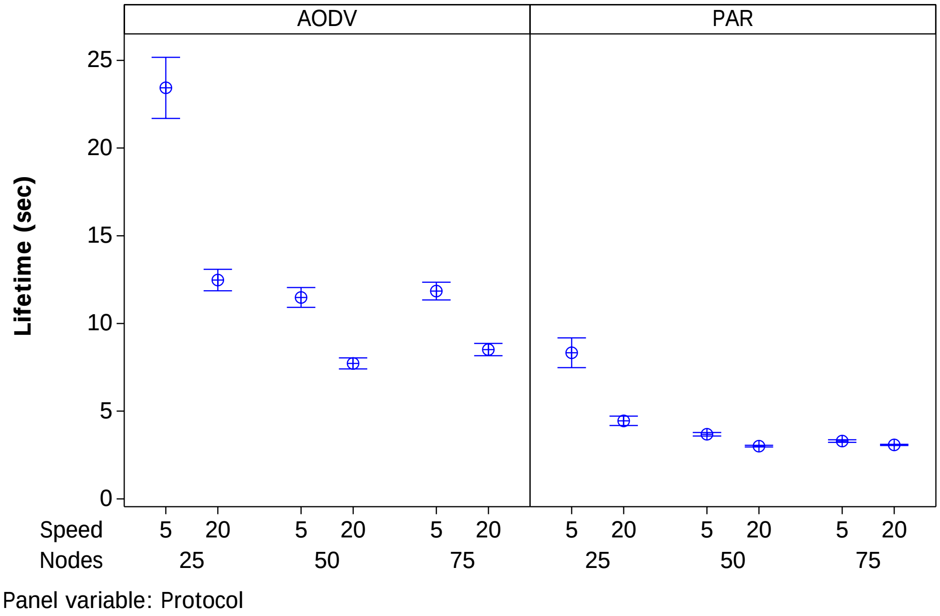


Figure 5.8: Plot of Route Lifetime by Speed and Node Density

*5.4.4 Offered Load.* The expectation is that route lifetime increases as the offered load increases. Figure 5.9 verifies this expectation. In all but one case, the higher offered load resulted in higher route lifetimes. The remaining instance, for AODV at 25 nodes, shows no statistically significant difference between the two loads. While the figure confirms the expectation, the effect of the offered load factor is seen to be small.

*5.4.5 Computation of Effects for Route Lifetime.* To evaluate the effects of each level of the factors a computation of effects for route lifetime was performed. Figures 5.10 and 5.11 plot the main effects of the factor levels for AODV and PAR respectively. Both protocols respond in a common manner to the levels of the factors, with route lifetimes dropping sharply between node densities of 25 and 50 nodes. Increasing node density from 50 to 75 nodes has less impact on route lifetime than

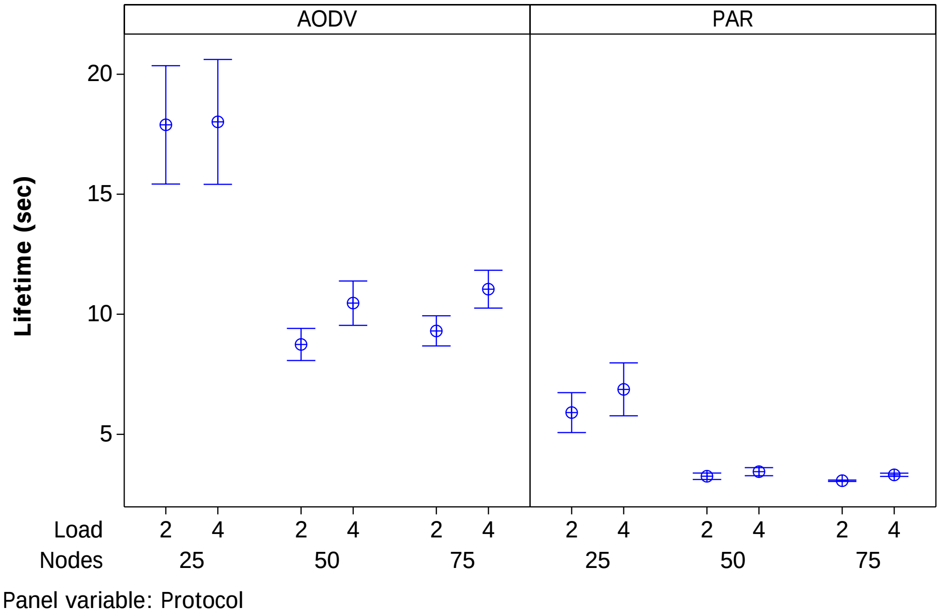


Figure 5.9: Plot of Route Lifetime by Offered Load and Node Density

increasing from 25 to 50 nodes. Similarly, as expected, the route lifetimes improve in response to an increase in offered load. Furthermore, the figures reinforce the expectation that as the speeds of the nodes increase, route lifetimes lessen.

Tables 5.1 and 5.2 summarize the effects of the factors. These tables show that, for both protocols, the node density has the greatest effect on the route lifetime. The nodes speed have the next greatest effect. Since none of the confidence intervals for the factor levels include 0, all of the levels are considered statistically different from the mean.

Table 5.1: Table of Effects for AODV Route Lifetime

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 12.58 | 2.07 | 12.26 | 12.89 |
| | 25 | 5.38 | 0.27 | 4.93 | 5.82 |
| Nodes | 50 | -2.98 | 0.27 | -3.42 | -2.53 |
| | 75 | -2.40 | 0.27 | -2.85 | -1.96 |
| Load | 2 pps | 0.60 | 0.19 | -0.91 | -0.28 |
| | 4 pps | -0.60 | 0.19 | 0.28 | 0.91 |
| Speed | 5 m/s | 3.01 | 0.19 | 2.70 | 3.32 |
| | 20 m/s | -3.01 | 0.19 | -3.32 | -2.70 |

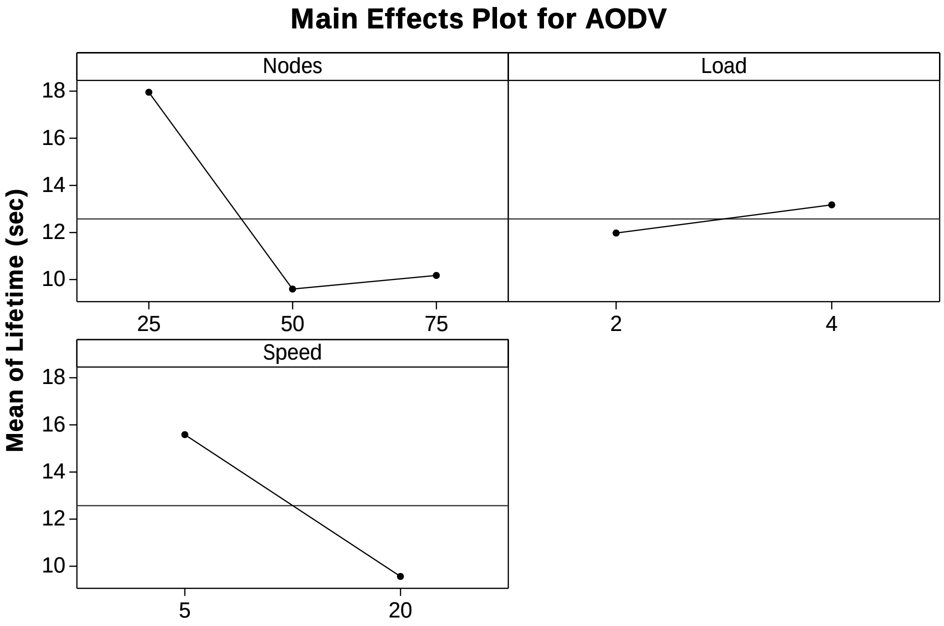


Figure 5.10: AODV Main Effects Plot of Route Lifetime

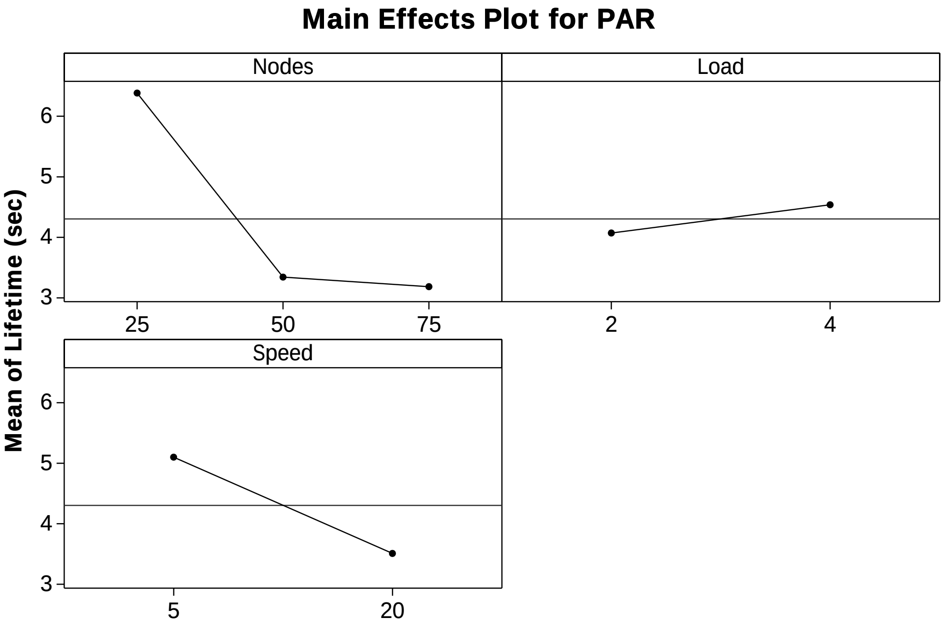


Figure 5.11: PAR Main Effects Plot of Route Lifetime

Table 5.2: Table of Effects for PAR Route Lifetime

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 4.30 | 0.91 | 4.17 | 4.44 |
| | 25 | 2.08 | 0.12 | 1.89 | 2.28 |
| Nodes | 50 | -0.96 | 0.12 | -1.16 | -0.77 |
| | 75 | -1.12 | 0.12 | -1.31 | -0.92 |
| Load | 2 pps | 0.23 | 0.08 | -0.37 | -0.10 |
| | 4 pps | -0.23 | 0.08 | 0.10 | 0.37 |
| Speed | 5 m/s | 0.80 | 0.08 | 0.66 | 0.93 |
| | 20 m/s | -0.80 | 0.08 | -0.93 | -0.66 |

*5.4.6 Route Lifetime ANOVA.* In order to determine the factors that have the greatest effect on the lifetime of routes discovered in the MANET, an ANOVA is used. This analysis determines the portion of the total variance for which each factor and interaction is responsible. For the conclusions drawn in the ANOVA to be valid, several assumptions must be met. First, the errors of the observations must be normally distributed. Second, the variance of the observations must be constant. Furthermore, the errors must be independent. Finally, it is assumed that the errors are independent of the factor levels. In order to validate these results, a series of visual tests are conducted. Appendix C contains several figures that accomplish these visual tests and validate the ANOVA assumptions, as well as an explanation of the data transforms used to validate these assumptions.

Each ANOVA computed allocates variance over all factors, as well as the interactions of these factors. However, not all factors or interactions have statistically significant effects on the response. The p-value of an ANOVA is an indicator of the significance of the factor. Furthermore, the p-value determines the probability that the variance attributed to the particular factor or interaction is, in fact, due to error. To make this determination the p-value for each element of the ANOVA is compared to the $\alpha$ value for the experiment. Since this research uses 90% confidence intervals, $\alpha = 0.10$. Any factor or interaction with a p-value greater than $\alpha$ is not statistically significant. These factors are pulled from consideration, and the ANOVA is recomputed, attributing the variance of these factors to error.

Table 5.3 shows the final ANOVA accomplished for the AODV protocol. As explained in Appendix C, the ANOVA is accomplished using data transformed with a natural log transform. The interaction of Offered Load and Node Speed, as well as the interaction of all factors are insignificant. Therefore, these factors are removed, and their variance attributed to error. About 48.62% of the total variance in ln(Route Lifetime) is due to the Node Density of the network which supports the conclusions drawn when examining the main effects previously. Node Speed has a significant impact on the response as well. For AODV, Node Speed is responsible for 37.66% of the variation in ln(Route Lifetime). While this factor is expected to have a significant effect, it is not expected that this factor would represent such a large percentage of the variation. Error also contributes significantly with 7.41% of the total variation. All other factors and interactions represent smaller percentages, as seen in Table 5.3

Table 5.3: ANOVA Table for AODV ln(Route Lifetime)

| Component | Sum of Squares | | Percentage Variation | DOF | Mean Square | F Comp | p |
|---|---|---|---|---|---|---|---|
| Total | SST | 14.7332 | 100.00% | 117 | | | |
| Node Density | SSA | 7.1637 | 48.62% | 2 | 3.5818 | 377.91 | 0.00 |
| Offered Load | SSB | 0.3862 | 2.63% | 1 | 0.3862 | 37.19 | 0.00 |
| Node Speed | SSC | 5.548 | 37.66% | 1 | 5.548 | 560.26 | 0.00 |
| Node Density*Offered Load | SSAB | 0.2061 | 1.40% | 2 | 0.1031 | 10.29 | 0.00 |
| Node Density*Node Speed | SSAC | 0.3372 | 2.29% | 2 | 0.1686 | 16.83 | 0.00 |
| Error | SSE | 1.0919 | 7.41% | 109 | 0.0100 | | |

Similar ANOVA computations are accomplished for PAR. In this case, an inverse transform is used. The initial ANOVA indicates that the interactions of Node Density and Offered Load, Offered Load and Node Speed, and of all factors are statistically not significant. Thus, these factors are removed from the final ANOVA, allocating their variance to error. Table 5.4 is the final computed ANOVA. For the inverse of Route Lifetime for the PAR protocol, Node Density is responsible for the greatest percentage of variation, accounting for 70.66%. While significantly less, Node Speed also contributes significantly with 17.05%. These results are expected, and confirm the conclusions drawn previously from computing and analyzing the main effects of the factors. The error in this ANOVA is allocated 5.23% of the variation.

Table 5.4: ANOVA Table for PAR (1/Route Lifetime)

| Component | Sum of Squares | | Percentage Variation | DOF | Mean Square | F Comp | p |
|---|---|---|---|---|---|---|---|
| Total | SST | 0.6404 | 100.00% | 116 | | | |
| Node Density | SSA | 0.4525 | 70.66% | 2 | 0.2263 | 690.05 | 0.00 |
| Offered Load | SSB | 0.0104 | 1.62% | 1 | 0.0104 | 37.8 | 0.00 |
| Node Speed | SSC | 0.1092 | 17.05% | 1 | 0.1092 | 372.34 | 0.00 |
| Node Density*Node Speed | SSAC | 0.0348 | 5.43% | 2 | 0.0174 | 57.09 | 0.00 |
| Error | SSE | 0.0335 | 5.23% | 110 | 0.0003 | | |

## ***5.5 Throughput Analysis***

Throughput measures the total bits per second the network transmits. This metric is one means of evaluating the performance of the network. In particular, this metric shows the impact of reliability on the performance of the network. It is expected that as reliability improves, so too does the amount of data successfully transmitted. Therefore, it is anticipated that throughput increases with increases in route lifetime.

*5.5.1 Routing Protocol.* As seen from the route lifetime metric in Section 5.4, AODV provides more reliable routes, on average. This leads to the expectation that AODV has greater throughput than PAR. Figure 5.12 shows the opposite trend. For all node densities, PAR has greater throughput. Furthermore, the difference between PAR and AODV is greater as node density increases. At 75 nodes, the throughput of PAR has grown to almost two and one half times that of AODV.

Routing overhead and data traffic received are two major contributing factors to the throughput experienced in these experiments. Figures 5.13 and 5.14 show these two factors. As expected and explained in Section 5.2, PAR has significantly more routing overhead. However, PAR also is superior in delivering application layer packets. Since PAR has greater rates for both data transmission and route overhead transmissions, its throughput is greater.

*5.5.2 Node Density.* Both AODV and PAR behave similarly as the node density factor is varied. Figure 5.12 clearly shows that throughput increases as node

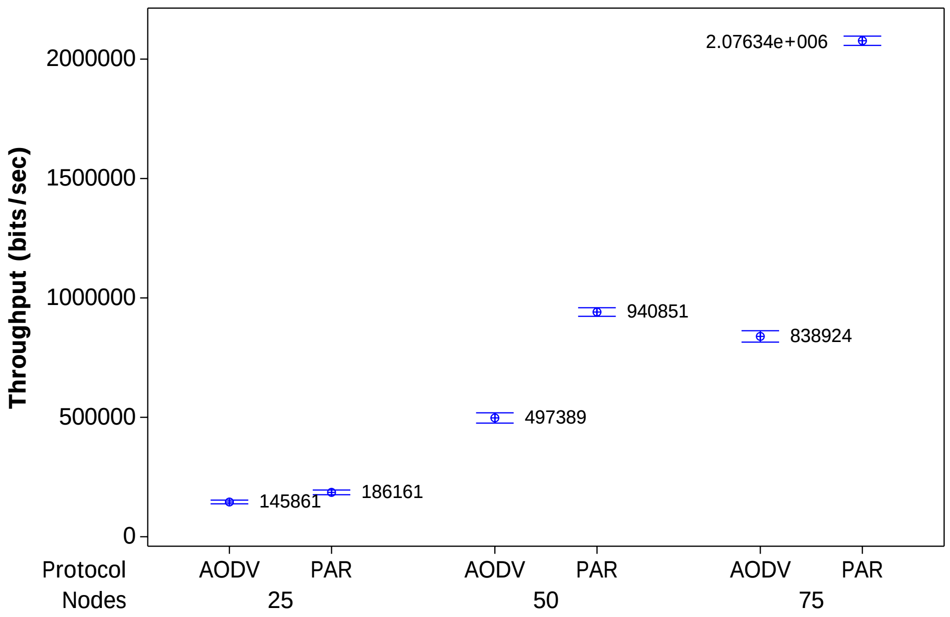


Figure 5.12: Plot of Throughput by Protocol and Node Density

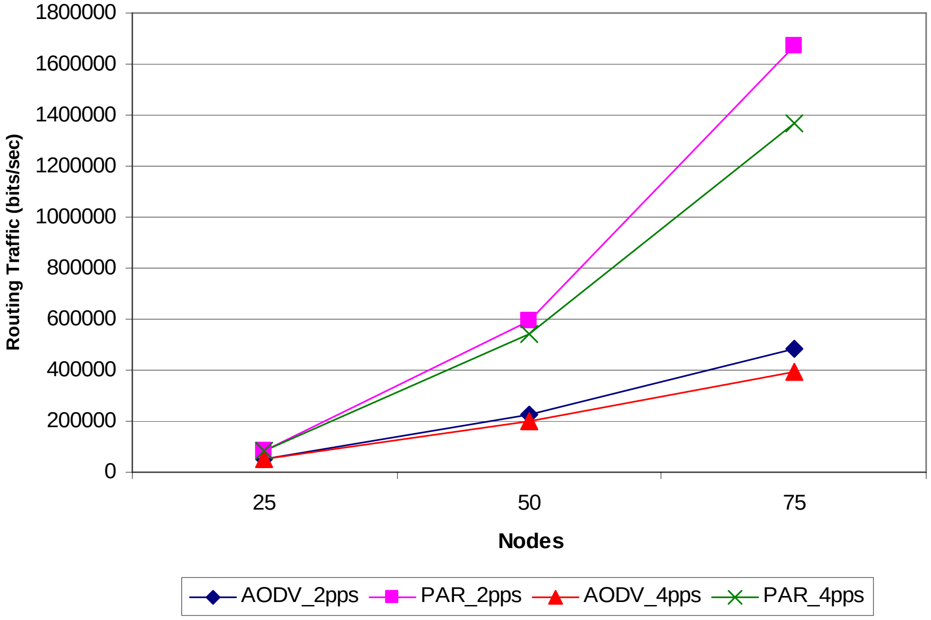


Figure 5.13: Route Traffic Received versus Node Density for 0-5 m/s

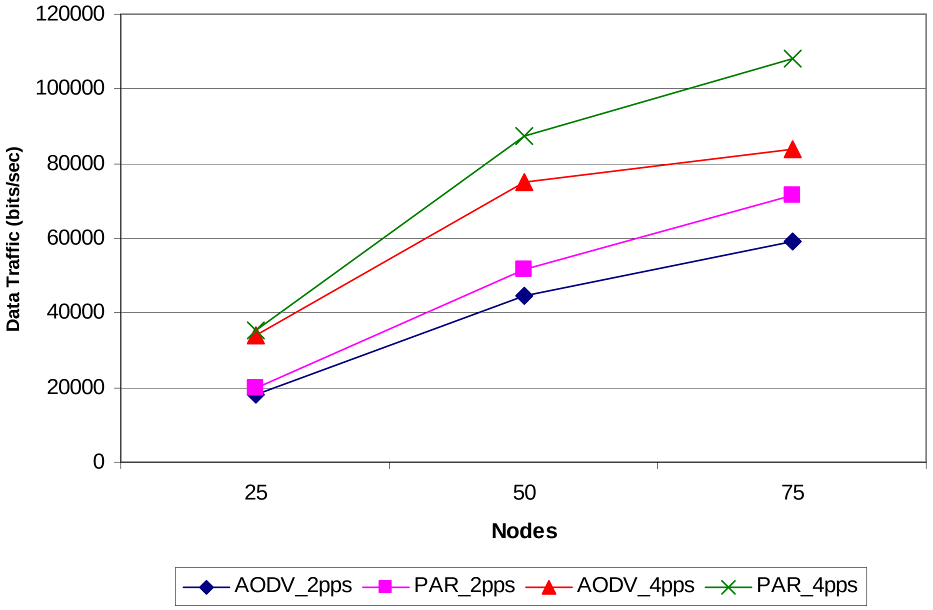


Figure 5.14: Data Traffic Received versus Node Density for 0-5 m/s

density increases. The most fundamental reason for this rise in throughput is increased routing overhead associated with the additional nodes. Nodes not only produce their own traffic, but are responsible for forwarding the overhead of other nodes. At higher densities, connectivity levels increase and overhead grows sharply which increases the throughput. As expected, this rise occurs more rapidly in the PAR protocol since PAR by design produces higher levels of overhead.

*5.5.3 Node Speed.* Lower values for node speed result in longer route lifetimes, and consequently, greater amounts of data to be transmitted. Therefore, it is anticipated that lower values for nodes' speeds will result in higher throughputs. Figure 5.15 shows the throughput values versus speed and node densities. This figure confirms the anticipated results for AODV. Additionally, the figure shows that the effect of the node speed factor is greater for AODV than for PAR. However, examining the plot for PAR indicates there is no statistically significant effect on throughput from the node speed factor.

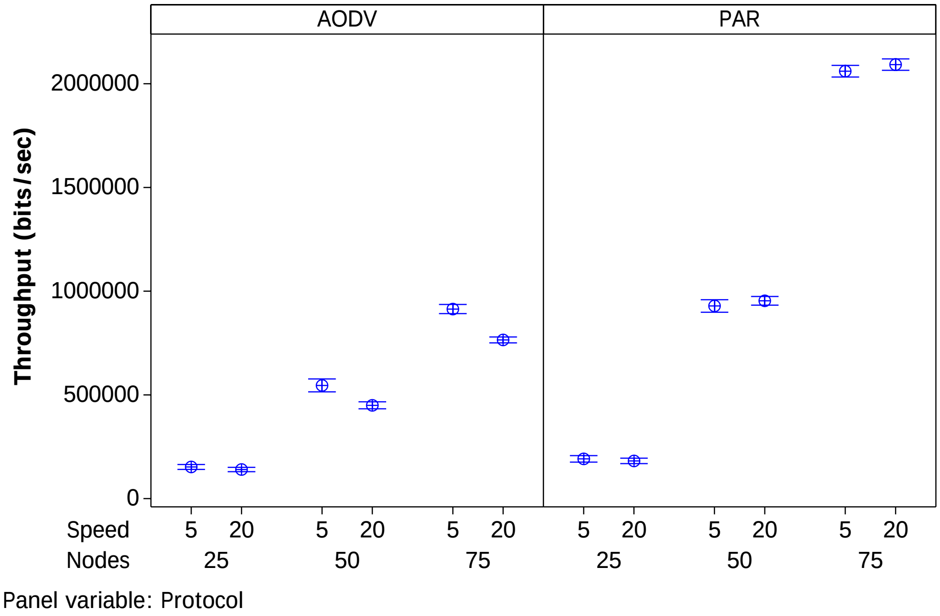


Figure 5.15: Plot of Throughput by Speed and Node Density

*5.5.4 Offered Load.* The offered load factor seems to have the most direct impact on throughput. Greater offered loads increase throughput in the network. Figure 5.16 verifies this expected result. The figure demonstrates that for both protocols increased offered loads yield greater throughputs, with the exception of PAR at 75 node density. In this instance, throughput is greater for lower offered load. This is due to greater routing overhead for this configuration. This extra overhead drives the throughput higher for this particular factor level.

*5.5.5 Computation of Effects for Throughput.* Figures 5.17 and 5.18 are plots of the main effects for each factor. The figure for AODV confirms that each factor has the anticipated effect on the throughput of the network. Increasing node density increases throughput as both the amount of routing overhead and data traffic increase. An increase in offered load increases throughput as well since there is more application layer traffic on the network. Node speed, however, reduces throughput since routes fail more frequently at high speeds, interrupting communications. The figure for PAR, however, reveals that the offered load and node speed factors have little effect on the throughput experienced by the network.

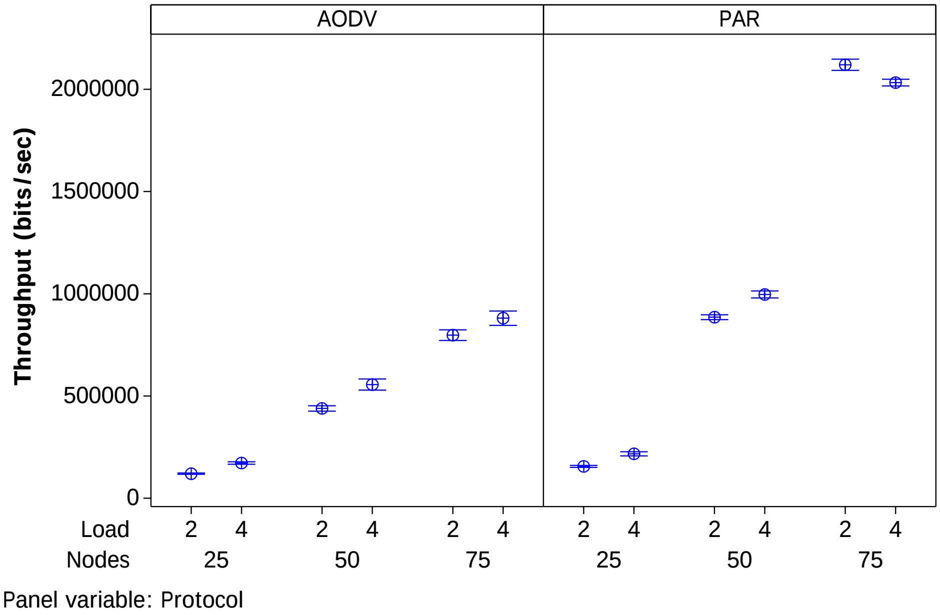


Figure 5.16: Plot of Throughput by Offered Load and Node Density

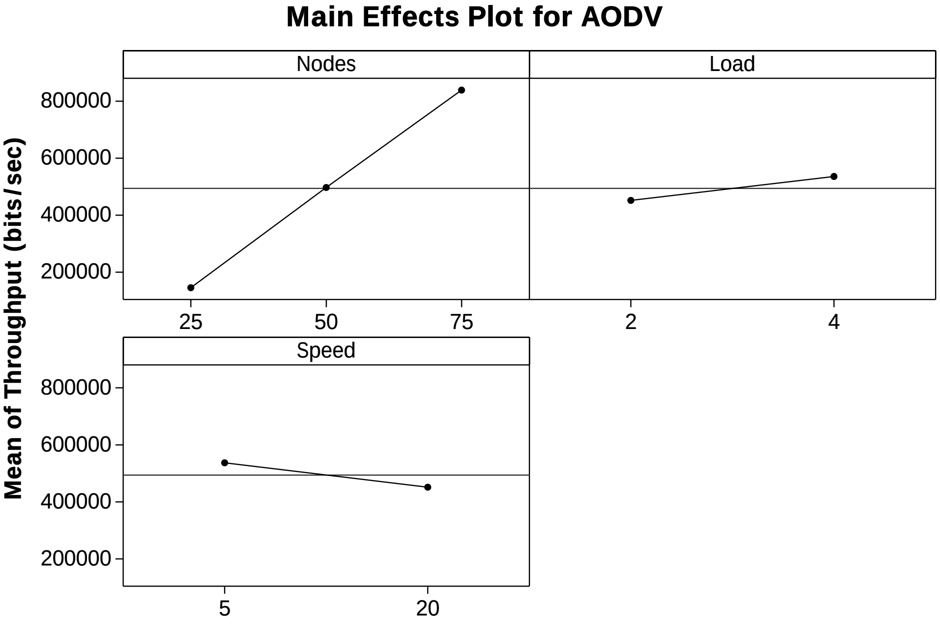


Figure 5.17: AODV Main Effects Plot of Throughput

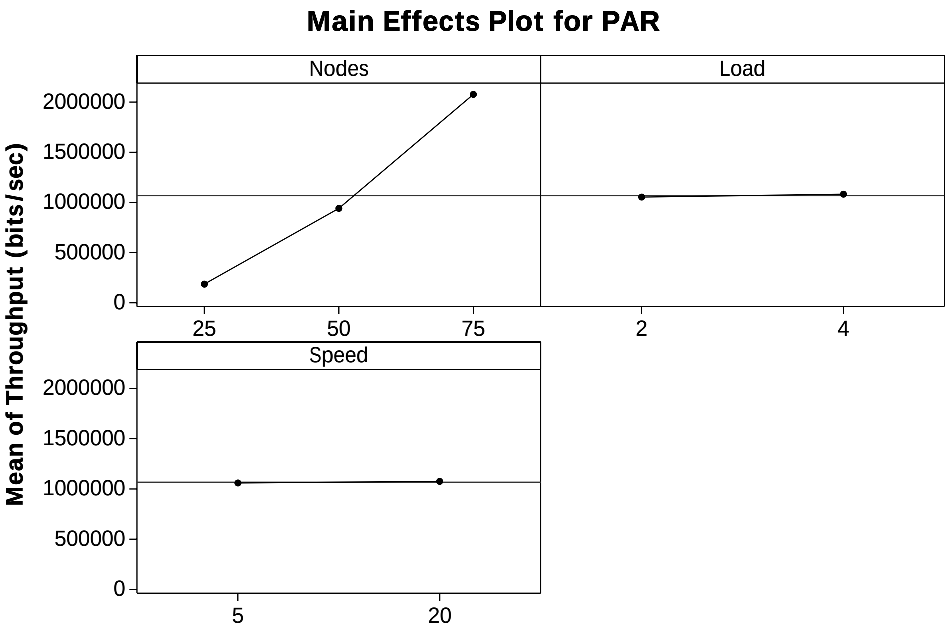


Figure 5.18: PAR Main Effects Plot of Throughput

Summaries of the effects of the various levels of the factors are presented in Tables 5.5 and 5.6 for AODV and PAR respectively. These tables indicate that for both protocols node density has the greatest effect on the throughput of the network. Furthermore, the effects of node speed and offered load are similar. These tables also indicate that for AODV there is no statistical difference between the mean and the node density level 50. Furthermore, for PAR the effect of the 0-20 m/s node speed is not statistically different than the mean.

Table 5.5: Table of Effects for AODV Throughput

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 494057.98 | 16844.02 | 491507.03 | 496608.93 |
| | 25 | -348197.22 | 2174.55 | -351804.80 | -344589.63 |
| Nodes | 50 | 3330.90 | 2174.55 | -276.68 | 6938.49 |
| | 75 | 344866.32 | 2174.55 | 341258.73 | 348473.90 |
| Load | 2 pps | 41972.93 | 1537.64 | -44523.88 | -39421.99 |
| | 4 pps | -41972.93 | 1537.64 | 39421.99 | 44523.88 |
| Speed | 5 m/s | 42902.03 | 1537.64 | 40351.09 | 45452.98 |
| | 20 m/s | -42902.03 | 1537.64 | -45452.98 | -40351.09 |

Table 5.6: Table of Effects for PAR Throughput

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 1067785.32 | 39946.90 | 1061735.55 | 1073835.09 |
| | 25 | -881624.32 | 5157.12 | -890179.98 | -873068.66 |
| Nodes | 50 | -126934.45 | 5157.12 | -135490.12 | -118378.79 |
| | 75 | 1008558.78 | 5157.12 | 1000003.11 | 1017114.44 |
| Load | 2 pps | 14265.49 | 3646.64 | -20315.26 | -8215.72 |
| | 4 pps | -14265.49 | 3646.64 | 8215.72 | 20315.26 |
| Speed | 5 m/s | -7879.22 | 3646.64 | -13928.99 | -1829.45 |
| | 20 m/s | 7879.22 | 3646.64 | 1829.45 | -13928.99 |

## *5.6 End-to-End Delay Analysis*

It is anticipated that end-to-end delay is greater for PAR since it is designed to discover reliable routes rather than short ones. Thus, route discovery can take longer periods of time, forcing data packets to queue. This results in longer end-to-end delays.

*5.6.1 Routing Protocol.* Figure 5.19 indicates that the delay associated with PAR is substantially greater than that of AODV. At its smallest difference, the delay of PAR is approximately six times greater than that of AODV. There are several reasons for this. First, AODV employs an expanding ring search during route discovery. This means a node sends out its request with a limited time to live (TTL). Subsequent retries of the request increment this value until it reaches a threshold. At this point the TTL is set to the diameter of the network. PAR, on the other hand, immediately sets the TTL to the network diameter. Therefore, it takes longer for PAR to detect a failed route request. This fact increases the end-to-end delay of the network. A second reason for the significant difference between AODV and PAR is PAR employs a backoff upon the receipt of a route request packet. This further increases the route discovery time. Thus, the end-to-end delay is significantly greater for PAR than for AODV.

*5.6.2 Node Density.* Figure 5.19 also demonstrates that the end-to-end delay grows as the node density of the network increases. This trend is particularly

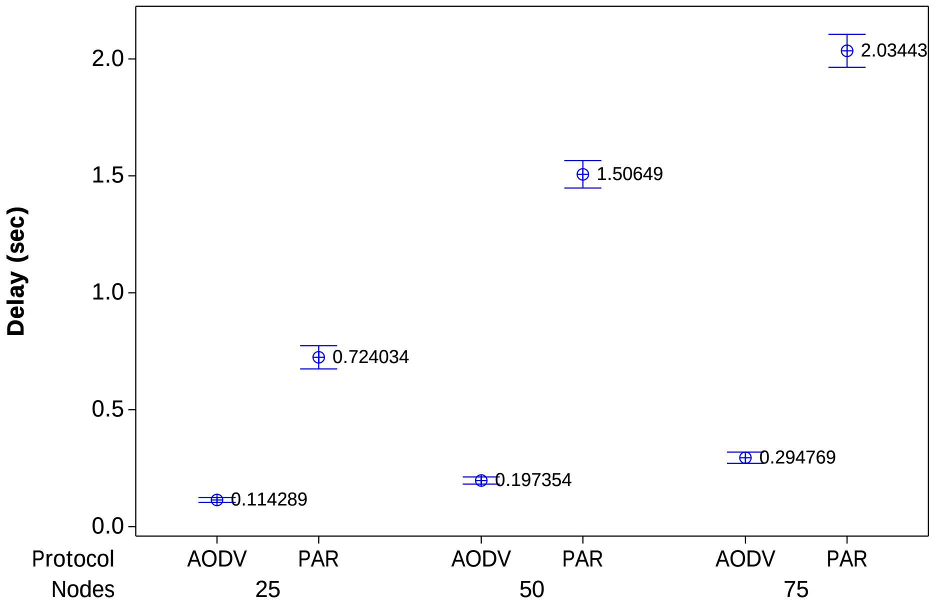


Figure 5.19: Plot of End-to-End Delay by Protocol and Node Density

evident for the PAR protocol. As described in the previous section, the initial TTL of RREQ packets is set to the network diameter. With greater numbers of nodes in the network, this value is set higher, thereby lengthening the period before a node retransmits a request and increasing delay. Additionally, PAR seeks to find reliable routes rather than short routes. In doing so, some short but unreliable routes are ignored while the protocol waits for a reliable route. In small networks, routes are short, meaning reliable routes can be found quickly. However, as the network grows it takes longer to find a reliable route, particularly to a distant neighbor. This also contributes to increase in delay as node density increases.

*5.6.3 Node Speed.* High speeds correspond to larger end-to-end delays, since packets must queue while the protocol adapts to the changes in topology and repairs failed routes. Figure 5.20 reflects these results. Furthermore, the figure demonstrates that the delay increases much more with the PAR protocol than with the AODV protocol, as node speed is increased. This is expected due to AODV's use of expanding ring searches and PAR's more selective route discovery process.

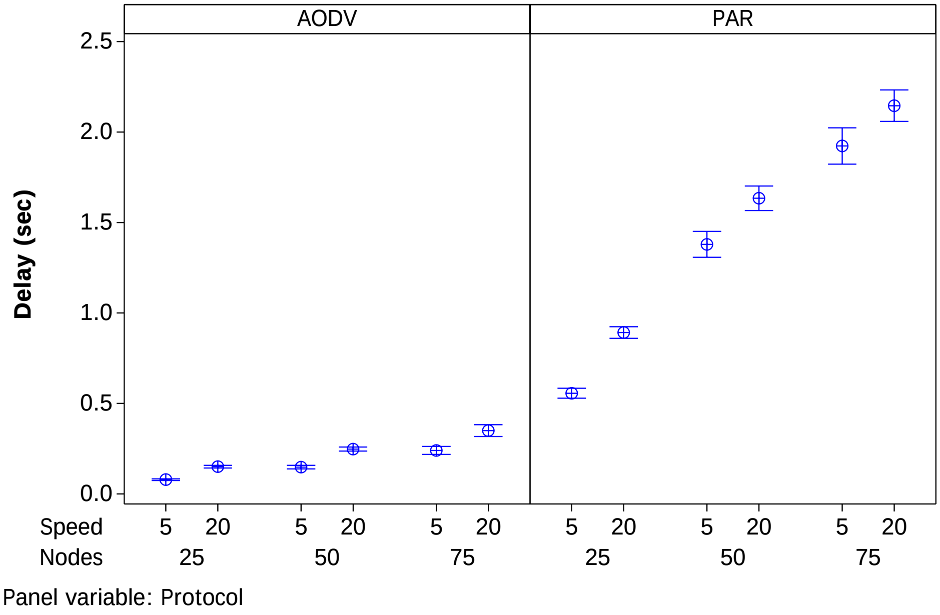


Figure 5.20: Plot of End-to-End Delay by Speed and Node Density

*5.6.4 Offered Load.* The increased load on the network adds to the congestion experienced. This congestion is compounded by the routing overhead associated with the added data. Increased congestion forces delays waiting for the wireless medium, as well as increasing the probability of collisions, thus making failed route discoveries more common. All of this leads to the expectation that end-to-end delay increases as offered load increases. Figure 5.21 confirms the expected findings. Again, the figure shows that the effects of this factor are much more substantial for PAR than for AODV. However, there are instances where the effect of offered load is minimal. For example, there is no statistically significant difference between the two packets per second and four packets per second levels for PAR with 25 nodes.

*5.6.5 Computation of Effects for End-to-End Delay.* The main effects for the factors of the experiment for AODV and PAR shown in Figures 5.22 and 5.23 respectively reveal similar behavior for both protocols. Furthermore, the main effects behave as predicted with higher node density, offered load, and node speed yielding a higher end-to-end delay. These conclusions are supported by Tables 5.7 and 5.8 which show all effects are statistically different from the mean. Additionally, for both

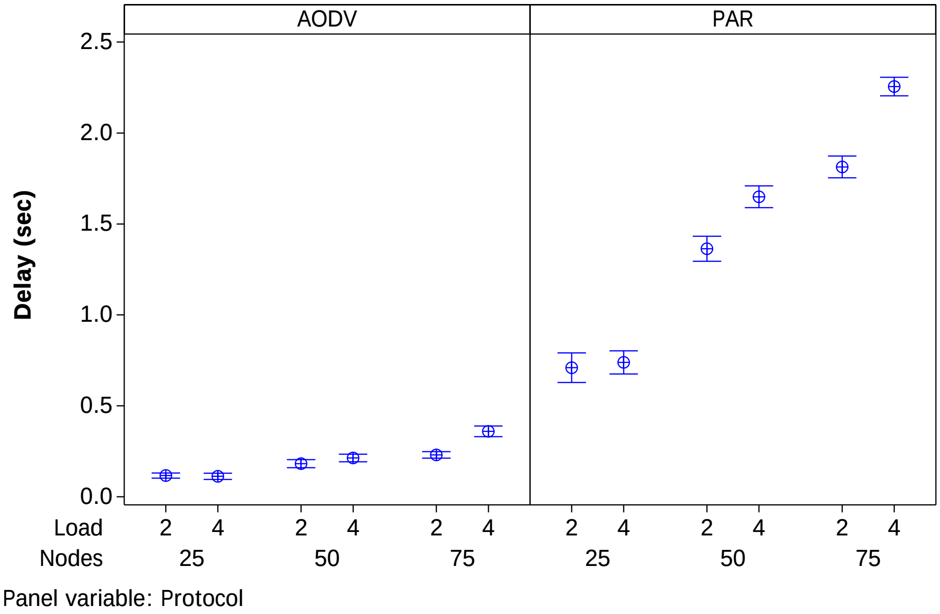


Figure 5.21: Plot of End-to-End Delay by Offered Load and Node Density

protocols, node density constitutes the largest effect. Offered load and node speed have similar effects.

Table 5.7: Table of Effects for AODV End-to-End Delay

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 0.2021 | 0.0216 | 0.1989 | 0.2054 |
| | 25 | -0.0878 | 0.0028 | -0.0925 | -0.0832 |
| Nodes | 50 | -0.0048 | 0.0028 | -0.0094 | -0.0001 |
| | 75 | 0.0926 | 0.0028 | 0.0880 | 0.0973 |
| Load | 2 pps | 0.0261 | 0.0020 | -0.0294 | -0.0228 |
| | 4 pps | -0.0261 | 0.0020 | 0.0228 | 0.0294 |
| Speed | 5 m/s | -0.0469 | 0.0020 | -0.0502 | -0.0436 |
| | 20 m/s | 0.0469 | 0.0020 | 0.0436 | 0.0502 |

## *5.7 Efficiency Analysis*

*5.7.1 Routing Protocol.* As shown in Figures 5.13 and 5.14, the PAR protocol has more overhead and successfully delivers a greater amount of application data. Thus, it is possible for the efficiency of PAR to exceed that of AODV. However, it is expected AODV has the greater efficiency due to the magnitude of the PAR routing

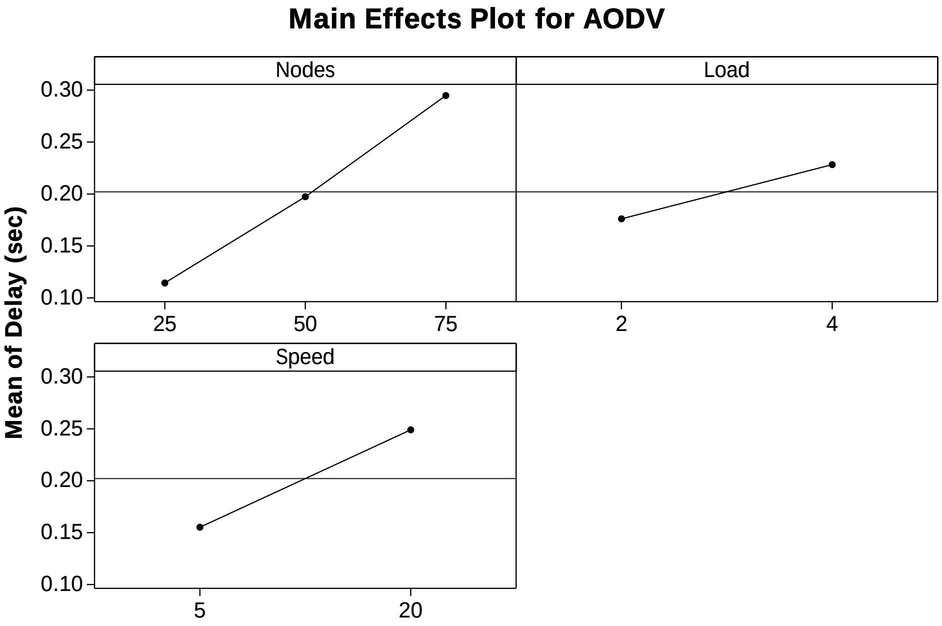


Figure 5.22: AODV Main Effects Plot of End-to-End Delay

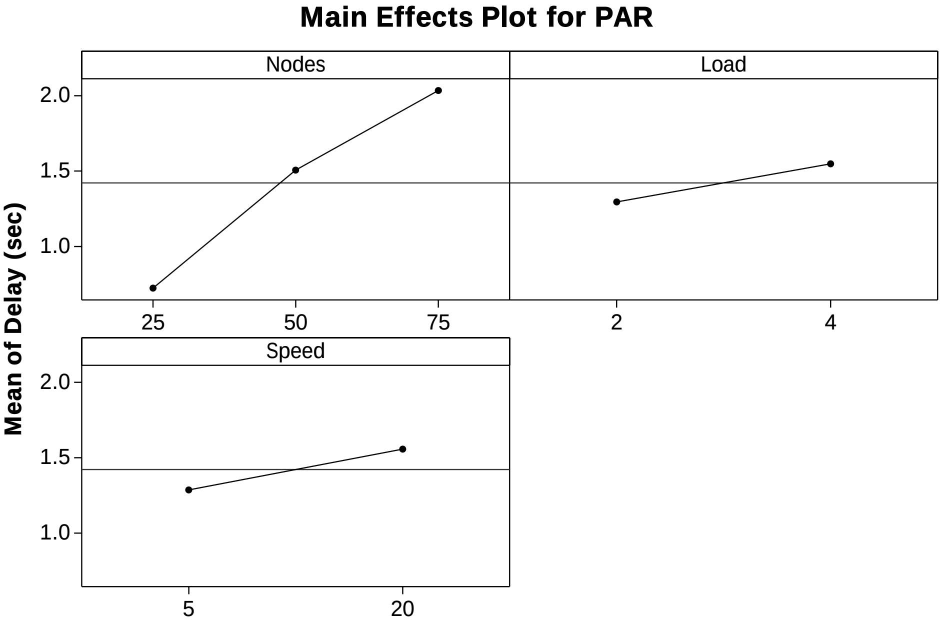


Figure 5.23: PAR Main Effects Plot of End-to-End Delay

Table 5.8: Table of Effects for PAR End-to-End Delay

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 1.4217 | 0.0914 | 1.4078 | 1.4355 |
| | 25 | -0.6976 | 0.0118 | -0.7172 | -0.6780 |
| Nodes | 50 | 0.0848 | 0.0118 | 0.0653 | 0.1044 |
| | 75 | 0.6128 | 0.0118 | 0.5932 | 0.6323 |
| Load | 2 pps | 0.1261 | 0.0083 | -0.1399 | -0.1123 |
| | 4 pps | -0.1261 | 0.0083 | 0.1123 | 0.1399 |
| Speed | 5 m/s | -0.1356 | 0.0083 | -0.1494 | -0.1218 |
| | 20 m/s | 0.1356 | 0.0083 | 0.1218 | 0.1494 |

overhead. Figure 5.24 is the efficiency of the two protocols as a function of their node densities and confirms AODV is more efficient than PAR.

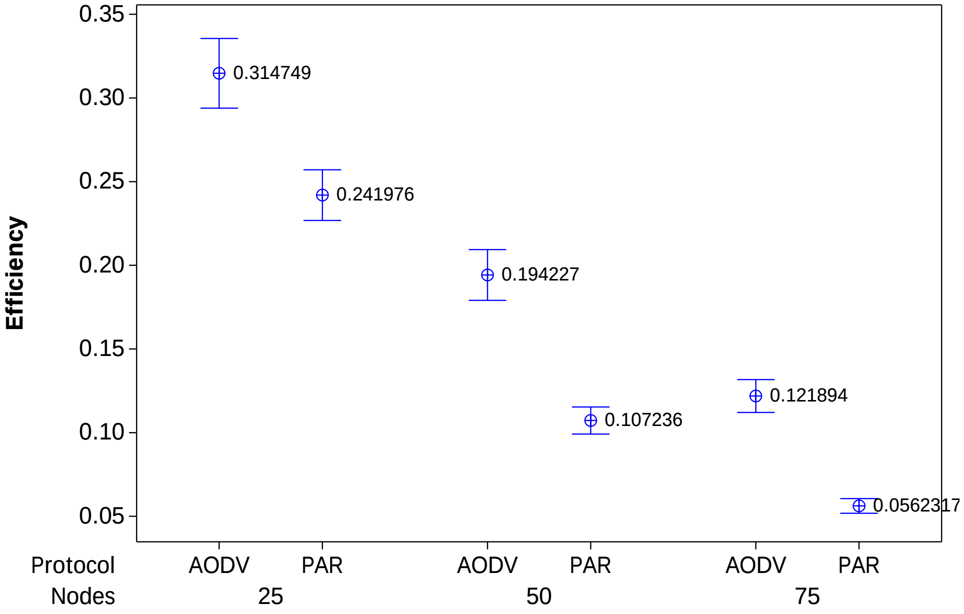


Figure 5.24: Plot of Efficiency by Protocol and Node Density

*5.7.2 Node Density.* Efficiency is expected to decrease as node density increases due to the corresponding increases in routing overhead. Figure 5.24 confirms that the efficiency of the AODV protocol declines with increasing values of node density. Similarly, the efficiency of the PAR protocol is reduced despite having successfully transmitted greater amounts of application data. The decline in efficiency

AODV experiences is smaller than that of PAR. As such, AODV remains more efficient than PAR, even at higher levels for node density.

*5.7.3 Node Speed.* It is expected that efficiency at 0-5 m/s will exceed that at 0-20 m/s. The results for efficiency in the experiments versus node speed is shown in Figure 5.25. For all AODV node densities, 5 m/s is superior to 20 m/s. The PAR results do not behave as expected. For the PAR protocol node speed has no statistically significant impact on the efficiency of the network. These unexpected results are due to the fact that PAR experiences high rates of link failures regardless of node speed. Thus, for PAR, increases in node speed are not accompanied by the loss of application layer data seen with AODV. Therefore, there is no impact to the efficiency experienced.

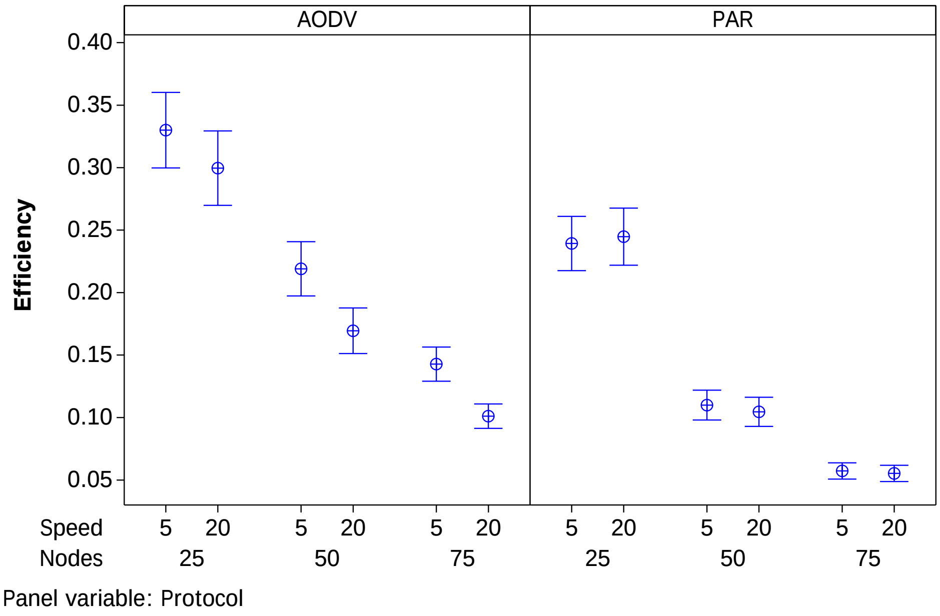


Figure 5.25: Plot of Efficiency by Speed and Node Density

*5.7.4 Offered Load.* Efficiency can be increased by transmitting more application layer data, or by reducing the overhead associated with routing. By increasing the offered load, thus providing the network more application layer data to transmit, it is expected that the efficiency of the network will improve. Thus, the efficiency for the four packets per second level should be greater than two packets per second. Figure 5.26 displays the efficiency plotted versus the offered load. Both AODV and PAR show greater efficiency at higher offered loads than at lower loads. Additionally, for every node density and offered load AODV results in higher efficiency than PAR. This is also an expected result since PAR has significantly greater routing overhead than AODV.

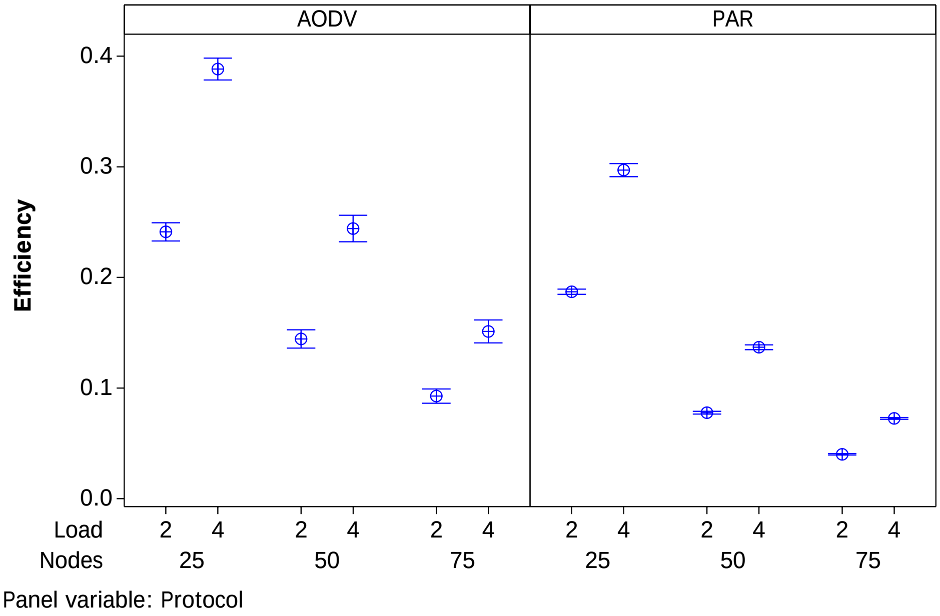


Figure 5.26: Plot of Efficiency by Offered Load and Node Density

*5.7.5 Computation of Effects for Efficiency.* Figures 5.27 and 5.28 show the main effects of the factors with respect to efficiency. These figures demonstrate that both protocols have a significant decline in efficiency between 25 and 50 nodes in density. The efficiency continues to decline between 50 and 75 nodes, but does so more slowly. The figures also confirm that the node speed has virtually no impact on the efficiency of the PAR protocol. The remaining factors have lesser effects on the efficiency of the protocols. Tables 5.9 and 5.10 quantify the effects of the factors. For both PAR and AODV the node density has the greatest effect on efficiency, with offered load having the second greatest effect. Furthermore, these tables verify that, at 90% confidence, there is no statistically significant difference between the node speed factor and the mean for PAR.

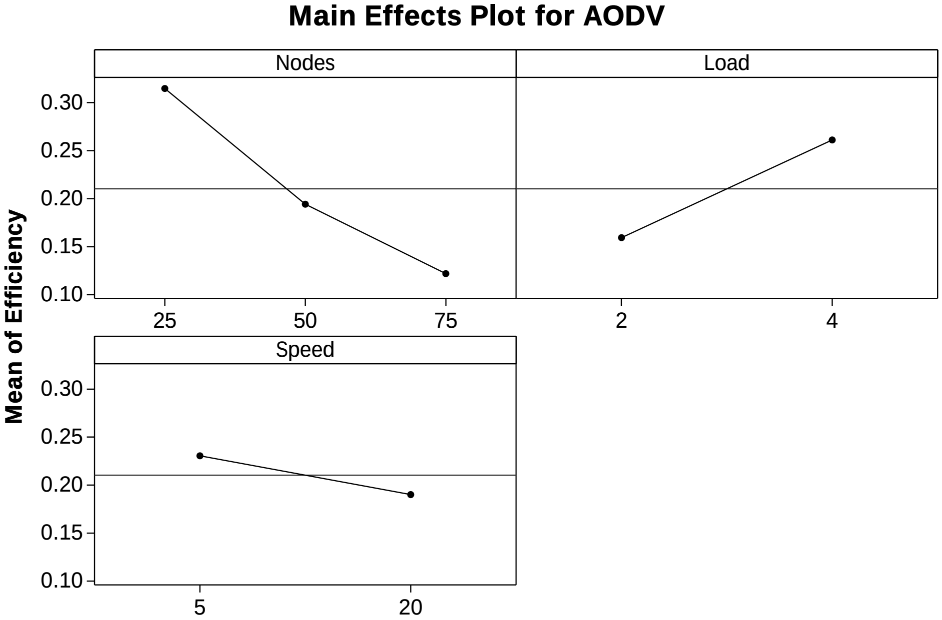


Figure 5.27: AODV Main Effects Plot of Efficiency

Table 5.9: Table of Effects for AODV Efficiency

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 0.2103 | 0.0115 | 0.2085 | 0.2120 |
| | 25 | 0.1044 | 0.0015 | 0.1020 | 0.1069 |
| Nodes | 50 | -0.0161 | 0.0015 | -0.0185 | -0.0136 |
| | 75 | -0.0884 | 0.0015 | -0.0909 | -0.0859 |
| Load | 2 pps | 0.0509 | 0.0011 | -0.0527 | -0.0492 |
| | 4 pps | -0.0509 | 0.0011 | 0.0492 | 0.0527 |
| Speed | 5 m/s | 0.0203 | 0.0011 | 0.0185 | 0.0220 |
| | 20 m/s | -0.0203 | 0.0011 | -0.0220 | -0.0185 |

## *5.8 Summary*

Predicted Associativity Routing is a custom MANET routing protocol designed to improve the reliability of the routes discovered. The results of this research indicate that this protocol, despite focusing on reliable routes, does not produce more reliable routes than AODV. Specifically, AODV produces routes that are as much as

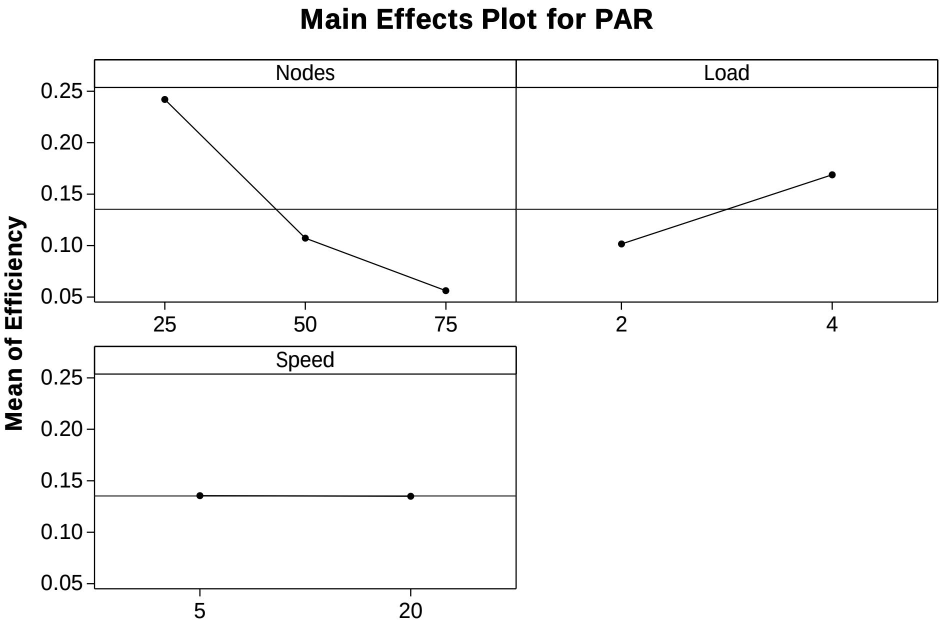


Figure 5.28: PAR Main Effects Plot of Efficiency

Table 5.10: Table of Effects for PAR Efficiency

| Parameter | Level | Mean Effect | Std Dev | Upper CL | Lower CL |
|---|---|---|---|---|---|
| Mean | | 0.1351 | 0.0070 | 0.1341 | 0.1362 |
| | 25 | 0.1068 | 0.0009 | 0.1053 | 0.1083 |
| Nodes | 50 | -0.0279 | 0.0009 | -0.0294 | -0.0264 |
| | 75 | -0.0789 | 0.0009 | -0.0804 | -0.0774 |
| Load | 2 pps | 0.0336 | 0.0006 | -0.0347 | -0.0325 |
| | 4 pps | -0.0336 | 0.0006 | 0.0325 | 0.0347 |
| Speed | 5 m/s | 0.0003 | 0.0006 | -0.0008 | 0.0014 |
| | 20 m/s | -0.0003 | 0.0006 | -0.0014 | 0.0008 |

three times longer than the routes produced by PAR. Furthermore, the results reveal that the density of nodes in the network has the greatest effect on both the reliability and performance of the network. ANOVA computations for ln(Route Lifetime) for AODV show Node Density contributes 48.62% of the total variation. Similar computations for the inverse of Route Lifetime for PAR show that Node Density is responsible for 70.66%. Furthermore, both ANOVAs show that Node Speed makes a major contribution to total variation in the respective responses. Additionally, the research demonstrated that despite its shorter lived routes, PAR delivers as much as 1.29 times more application layer data. However, to achieve this increase in data delivery, PAR requires up to 3.5 times more routing overhead than AODV. Thus, PAR displays greater throughput than AODV. However, this substantial overhead drives the efficiency of PAR down, working 0.4 times as efficiently as AODV. Additionally, PAR results in up to 7.5 times the end-to-end delay experienced by AODV. In general, AODV outperformed PAR in terms of both reliability and performance.

# VI. Conclusions

## 6.1 Introduction

This chapter summarizes the research and its results. First the objectives of the research are discussed, and their outcomes described. Next the contributions of this research are outlined. Finally, suggestions for future work in this area are proposed.

## 6.2 Objectives and Results

Three objectives are addressed by this research. The first objective is to determine the reliability of PAR. Next, the research seeks to determine the factors that affect the reliability of the protocols the most. Finally, this research seeks to compare the network performance of PAR to AODV to determine how the new protocol compares to existing protocols.

### 6.2.1 Reliability of PAR.

Despite the fact that PAR is specifically designed to address the issues of reliability in MANET routing, the protocol did not improve route reliability. In fact, the evaluation of the route lifetime metric for both PAR and AODV shows AODV produces more reliable routes under all network configurations. The difference in reliability can be as much as three times for high node densities. This unexpected result is rooted in the differences in the ways the two protocols evaluate connectivity between neighboring nodes. Links break more frequently in PAR which drives the route lifetime metric down for this protocol.

### 6.2.2 Factors Affecting Reliability.

Varying node density has the greatest effect on the reliability of the routes discovered. Node speed has the second greatest effect on the reliability of the protocol. The ANOVA for the AODV protocol show node density accounts for 48.62% of the variation in the natural logarithm of route lifetime. Similarly, node density accounts for 70.66% of the variation in the inverse lifetime of PAR. Node Speed also contributes significantly to the appropriately transformed response in each protocol.

*6.2.3 Performance of Comparison of PAR and AODV.* In addition to assessing reliability, several network performance statistics are collected and analyzed. These include throughput, end-to-end delay, and efficiency. The PAR protocol increases throughput compared to AODV by as much as two and one half times. This is due in large part to the routing overhead associated with PAR. In contrast, AODV performs significantly better with respect to end-to-end delay. This can be accounted for by the selectiveness PAR exhibits in selecting a route, as well as the expanding ring search employed by AODV. These factors contribute to the difference in delay in PAR and AODV. Finally, under most configurations, AODV has greater efficiency despite PAR delivering a greater amount of application layer traffic. This, also, is due to the routing overhead required by the PAR protocol to operate effectively.

## *6.3 Research Contributions*

The primary contribution of this research is the unique approach used to determine and employ residual link lifetimes in the generation of routes. In [BKS03], the concept of using residual lifetimes in an effort to improve route reliability is introduced. This research expands upon that by attempting provide a simple, adaptive means to determine the expected value of link lifetimes. Though unsuccessful, this type of protocol may yet result in a reliable MANET routing protocol. If the frequency of link breaks can be reduced, perhaps by improving the method by which PAR determines connectivity, the fundamental premise of PAR may prove successful at improving route lifetimes and improving routing reliability.

An additional contribution of this research is the introduction of a new routing protocol that improves the percentage of application layer packets delivered by the routing protocol. While not demonstrating improved reliability, PAR successfully delivers more data. As this is the fundamental purpose of a network, to deliver data from source to destination, this finding is significant. By improving the amount of application layer packets delivered results in more effective communication throughout the network.

## *6.4 Future Work*

There are several areas for future research. First, modify the PAR protocol to improve its reliability. As it is designed, a node's connectivity with its neighbors is determined only by consecutively-received HELLO packets. This makes the protocol particularly sensitive to losses in these packets, as evidenced by the large number of link breakages experienced. AODV, on the other hand, leverages all packets handled by the routing protocol to evaluate connectivity. Modifying the PAR protocol to operate in this manner may improve its reliability by reducing the frequency of link failures.

An additional area of research increases the sophistication of the PAR protocol. As it is implemented presently, the protocol is very basic, providing only the most fundamental functionality necessary to discover and maintain routes. Future research on PAR should focus on adding additional functionality such as local route repair, and allowing intermediate nodes to respond to route requests. In doing so, the performance of the protocol can be improved by reducing the overhead associated with the protocol.

One final area for research is to determine alternative ways to estimate link lifetimes. The PAR protocol uses the mean of experienced lifetimes and the confidence interval, or No Go Zone, to determine the expected value of link lifetimes. An alternative method may yield greater dividends in route reliability. By more accurately estimating link lifetime, residual lifetimes can better be determined. With this more precise knowledge, nodes can determine more reliable routes.

## *6.5 Summary*

Mobile Ad Hoc Networks can provide low cost, rapidly deployable, highly mobile network solutions. However, MANETs introduce challenges that make routing difficult and unreliable. Reliable routes are crucial to providing efficient and effective communications. Predicted Associativity Routing is a MANET routing protocol designed to address the issue of reliability in MANET routing. By determining an expected value for link lifetimes, PAR can make route selection decisions based on the residual lifetimes of alternative routes. In doing so, PAR seeks to produce long lasting, dependable routes.

Simulation of the protocol reveals, that despite its focus on reliability, PAR does not produce more reliable routes that AODV. Furthermore, variations in node density have the greatest impact on the route lifetimes experienced. Finally, PAR delivers as much as 1.29 times more data but requires as much as 3.5 times the routing overhead. However, PAR suffers from significantly longer delays, almost 7.5 times more, and due to its great overhead, has 0.4 times the efficiency.

# *Appendix A. Supplemental Pilot Study Results*

Figures A.1 through A.5 represent the results of pilot study experiments conducted with RREP Backoff set to 0.10 seconds. These results support the conclusions drawn in Chapter III.

Additionally, Tables A.1 through A.12 represent the confidence intervals for the corresponding figures. These intervals are reflected in the table since they are generally too tight to be seen in the figures.

Table A.1: Confidence Interval Information for Figure 3.4

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 3.3269 | 3.5169 | 3.7123 |
| | 50% | 3.3983 | 3.4660 | 3.5337 |
| | 80% | 3.5504 | 3.6088 | 3.6672 |
| 2 | 20% | 2.0851 | 2.1570 | 2.2289 |
| | 50% | 2.0691 | 2.1690 | 2.2690 |
| | 80% | 2.1151 | 2.1801 | 2.2450 |
| 5 | 20% | 1.1493 | 1.1666 | 1.1840 |
| | 50% | 1.0453 | 1.1281 | 1.2110 |
| | 80% | 1.1308 | 1.1639 | 1.1970 |

Table A.2: Confidence Interval Information for Figure 3.5

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 3.4543 | 3.5441 | 3.6339 |
| | 50% | 3.3519 | 3.5274 | 3.7029 |
| | 80% | 3.3887 | 3.5581 | 3.7275 |
| 2 | 20% | 2.1338 | 2.2025 | 2.2711 |
| | 50% | 2.0310 | 2.1258 | 2.2206 |
| | 80% | 2.1119 | 2.1972 | 2.2825 |
| 5 | 20% | 1.1579 | 1.2145 | 1.2711 |
| | 50% | 1.2297 | 1.2525 | 1.2752 |
| | 80% | 1.1836 | 1.2393 | 1.2951 |

Table A.3: Confidence Interval Information for Figure 3.6

| Hello Periodicity | No Go Zone | Lower CL | Mean | Upper CL |
|---|---|---|---|---|
| 1 | 20% | 1.1060 | 1.2042 | 1.3023 |
| | 50% | 1.1316 | 1.2055 | 1.2794 |
| | 80% | 1.1019 | 1.2386 | 1.3752 |
| 2 | 20% | 1.4770 | 1.5670 | 1.6570 |
| | 50% | 1.5316 | 1.6255 | 1.7193 |
| | 80% | 1.4828 | 1.6360 | 1.7892 |
| 5 | 20% | 1.9105 | 2.0041 | 2.0977 |
| | 50% | 1.6609 | 1.8194 | 1.9778 |
| | 80% | 1.8124 | 1.9525 | 2.0926 |

Table A.4: Confidence Interval Information for Figure 3.7

| Hello Periodicity | No Go Zone | Lower CL | Mean | Upper CL |
|---|---|---|---|---|
| 1 | 20% | 49498.28 | 51441.44 | 53384.60 |
| | 50% | 51299.76 | 52469.76 | 53639.76 |
| | 80% | 50110.43 | 51904.28 | 53698.14 |
| 2 | 20% | 46920.43 | 48483.90 | 50047.37 |
| | 50% | 46783.51 | 48696.66 | 50609.81 |
| | 80% | 47254.82 | 48813.62 | 50372.43 |
| 5 | 20% | 38230.34 | 39682.05 | 41133.75 |
| | 50% | 38974.68 | 41373.47 | 43772.25 |
| | 80% | 38594.70 | 40114.40 | 41634.11 |

Table A.5: Confidence Interval Information for Figure 3.8

| Hello Periodicity | No Go Zone | Lower CL | Mean | Upper CL |
|---|---|---|---|---|
| 1 | 20% | 559898.17 | 606215.40 | 652532.62 |
| | 50% | 574337.31 | 616186.23 | 658035.14 |
| | 80% | 563792.39 | 588126.22 | 612460.04 |
| 2 | 20% | 837849.12 | 857057.17 | 876265.21 |
| | 50% | 806046.19 | 844199.33 | 882352.47 |
| | 80% | 772599.17 | 840054.73 | 907510.28 |
| 5 | 20% | 1268409.31 | 1331620.33 | 1394831.35 |
| | 50% | 1300642.08 | 1427944.61 | 1555247.14 |
| | 80% | 1227094.13 | 1293917.76 | 1360741.39 |

Table A.6: Confidence Interval Information for Figure 3.9

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 0.0754 | 0.0784 | 0.0814 |
| | 50% | 0.0747 | 0.0787 | 0.0826 |
| | 80% | 0.0787 | 0.0811 | 0.0836 |
| 2 | 20% | 0.0523 | 0.0535 | 0.0547 |
| | 50% | 0.0527 | 0.0546 | 0.0565 |
| | 80% | 0.0522 | 0.0551 | 0.0580 |
| 5 | 20% | 0.0284 | 0.0290 | 0.0295 |
| | 50% | 0.0272 | 0.0282 | 0.0293 |
| | 80% | 0.0294 | 0.0301 | 0.0307 |

Table A.7: Confidence Interval Information for Figure 3.10

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 3699.93 | 4009.20 | 4318.47 |
| | 50% | 3879.37 | 4046.60 | 4213.83 |
| | 80% | 3695.08 | 3736.00 | 3776.92 |
| 2 | 20% | 5513.43 | 5656.20 | 5798.97 |
| | 50% | 5322.27 | 5543.80 | 5765.33 |
| | 80% | 4985.81 | 5366.60 | 5747.39 |
| 5 | 20% | 7236.46 | 7599.80 | 7963.14 |
| | 50% | 7296.30 | 8036.20 | 8776.10 |
| | 80% | 7124.56 | 7512.60 | 7900.64 |

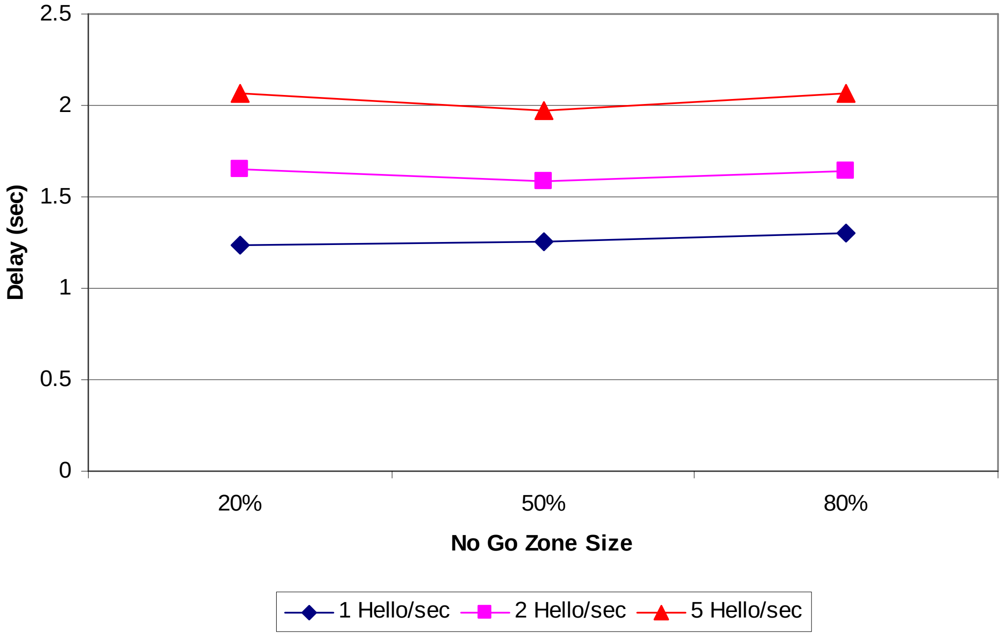


Figure A.1: End-to-end Delay versus No Go Zone Size for RREP Backoff of 0.10 s

Table A.8: Confidence Interval Information for Figure A.1

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 1.1572 | 1.2397 | 1.3223 |
|  | 50% | 1.1674 | 1.2551 | 1.3427 |
|  | 80% | 1.1213 | 1.3028 | 1.4844 |
| 2 | 20% | 1.6220 | 1.6511 | 1.6801 |
|  | 50% | 1.4887 | 1.5890 | 1.6893 |
|  | 80% | 1.5057 | 1.6393 | 1.7729 |
| 5 | 20% | 2.0072 | 2.0694 | 2.1316 |
|  | 50% | 1.8468 | 1.9675 | 2.0882 |
|  | 80% | 1.9604 | 2.0670 | 2.1736 |

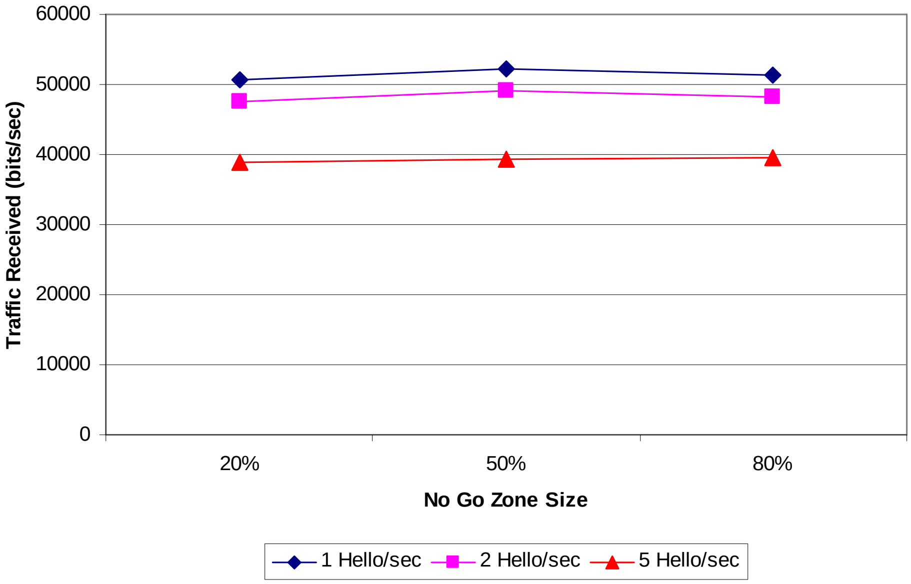


Figure A.2: Data Traffic Received versus No Go Zone Size for RREP Backoff of 0.10 s

Table A.9: Confidence Interval Information for Figure A.2

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 49583.94 | 50775.84 | 51967.73 |
| | 50% | 50750.98 | 52178.94 | 53606.90 |
| | 80% | 48259.79 | 51262.58 | 54265.36 |
| 2 | 20% | 45998.22 | 47573.67 | 49149.13 |
| | 50% | 46374.69 | 49009.10 | 51643.50 |
| | 80% | 45938.31 | 48324.15 | 50709.99 |
| 5 | 20% | 37548.07 | 38996.20 | 40444.32 |
| | 50% | 38132.15 | 39341.62 | 40551.10 |
| | 80% | 38275.38 | 39527.08 | 40778.78 |

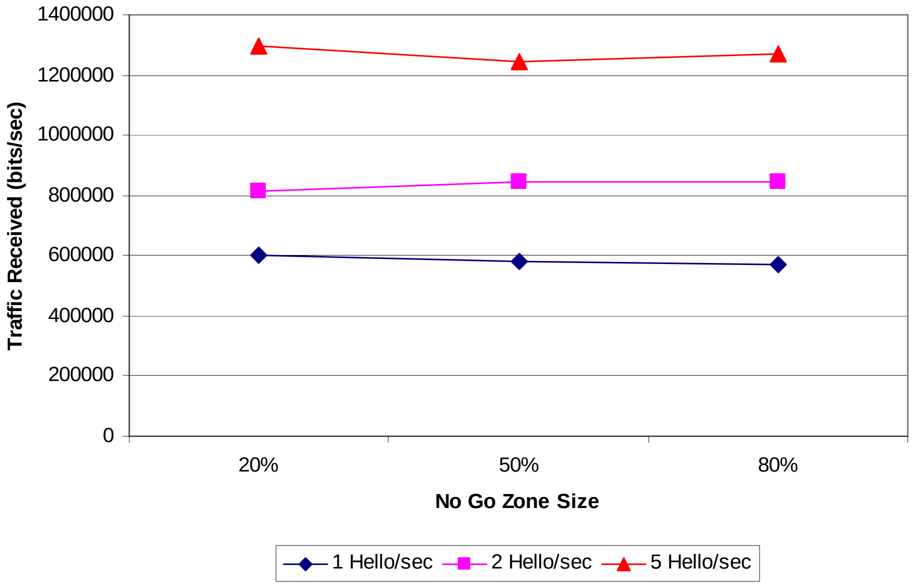


Figure A.3: Route Traffic Received versus No Go Zone Size for RREP Backoff of 0.10 s

Table A.10: Confidence Interval Information for Figure A.3

| Hello Periodicity | No Go Zone | Lower CL | Mean | Upper CL |
|---|---|---|---|---|
| 1 | 20% | 581660.54 | 601479.68 | 621298.82 |
|  | 50% | 534038.56 | 578925.61 | 623812.66 |
|  | 80% | 524597.62 | 571154.85 | 617712.09 |
| 2 | 20% | 766278.00 | 815517.23 | 864756.46 |
|  | 50% | 776487.65 | 843831.22 | 911174.78 |
|  | 80% | 792733.27 | 843565.48 | 894397.69 |
| 5 | 20% | 1226220.78 | 1294132.64 | 1362044.51 |
|  | 50% | 1196163.23 | 1246577.32 | 1296991.40 |
|  | 80% | 1176989.94 | 1269635.53 | 1362281.12 |

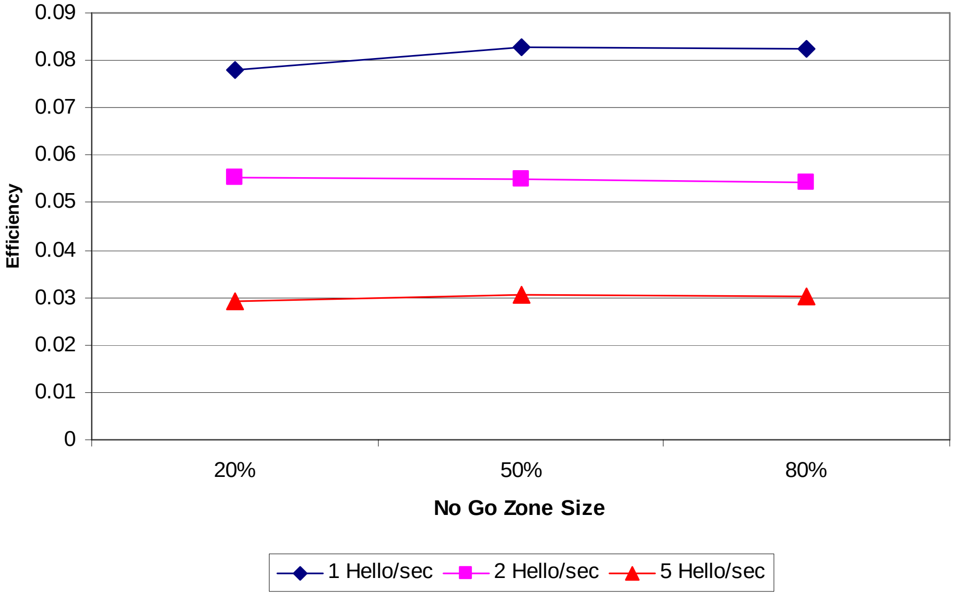


Figure A.4: Efficiency versus No Go Zone Size for RREP Backoff of 0.10 s

Table A.11: Confidence Interval Information for Figure A.4

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| | 20% | 0.0768 | 0.0779 | 0.0789 |
| 1 | 50% | 0.0790 | 0.0829 | 0.0869 |
| | 80% | 0.0802 | 0.0825 | 0.0848 |
| | 20% | 0.0534 | 0.0552 | 0.0570 |
| 2 | 50% | 0.0533 | 0.0550 | 0.0567 |
| | 80% | 0.0526 | 0.0542 | 0.0559 |
| | 20% | 0.0284 | 0.0293 | 0.0301 |
| 5 | 50% | 0.0299 | 0.0306 | 0.0314 |
| | 80% | 0.0288 | 0.0303 | 0.0318 |

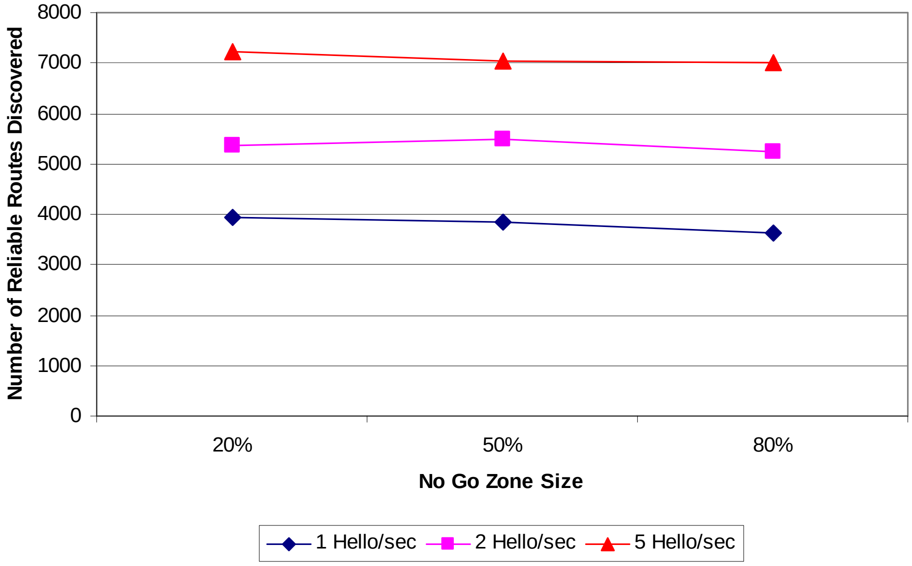


Figure A.5: Reliable Routes Discovered versus No Go Zone Size for RREP Backoff of 0.10 s

Table A.12: Confidence Interval Information for Figure A.5

| **Hello Periodicity** | **No Go Zone** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|
| 1 | 20% | 3846.21 | 3949.40 | 4052.59 |
| | 50% | 3622.27 | 3836.60 | 4050.93 |
| | 80% | 3379.74 | 3642.20 | 3904.66 |
| 2 | 20% | 5119.91 | 5351.00 | 5582.09 |
| | 50% | 5112.05 | 5485.80 | 5859.55 |
| | 80% | 4971.27 | 5252.80 | 5534.33 |
| 5 | 20% | 6896.22 | 7238.60 | 7580.98 |
| | 50% | 6747.01 | 7028.00 | 7308.99 |
| | 80% | 6559.64 | 7014.80 | 7469.96 |

# *Appendix B. Supplemental Validation Results*

Figures B.1 through B.4 represent the results of validation experiments conducted with node speed set at 0-20 m/s. These results support the conclusions drawn in Chapter V.

Additionally, Tables B.1 through B.8 represent the confidence intervals for the corresponding figures. These intervals are reflected in the tables since they are generally too tight to be seen in the figures.

Table B.1: Confidence Interval Information for Figure 5.1

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 51.87 | 60.89 | 69.90 |
| | | 4 | 38.64 | 45.44 | 52.23 |
| | PAR | 2 | 7.20 | 8.06 | 8.92 |
| | | 4 | 8.05 | 9.13 | 10.22 |
| 50 | AODV | 2 | 16.57 | 19.52 | 22.47 |
| | | 4 | 19.75 | 22.38 | 25.00 |
| | PAR | 2 | 3.36 | 3.48 | 3.61 |
| | | 4 | 3.64 | 3.73 | 3.82 |
| 75 | AODV | 2 | 14.25 | 15.44 | 16.64 |
| | | 4 | 15.56 | 17.04 | 18.51 |
| | PAR | 2 | 3.04 | 3.06 | 3.08 |
| | | 4 | 3.34 | 3.39 | 3.44 |

Table B.2: Confidence Interval Information for Figure 5.2

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 128408.93 | 133058.42 | 137707.91 |
| | | 4 | 145620.52 | 154390.19 | 163159.87 |
| | PAR | 2 | 137388.27 | 145586.95 | 153785.63 |
| | | 4 | 211463.04 | 222656.99 | 233850.94 |
| 50 | AODV | 2 | 424159.90 | 429630.97 | 435102.03 |
| | | 4 | 437585.08 | 457823.28 | 478061.48 |
| | PAR | 2 | 854135.88 | 877922.77 | 901709.66 |
| | | 4 | 987568.54 | 1006291.09 | 1025013.64 |
| 75 | AODV | 2 | 765716.09 | 779022.16 | 792328.23 |
| | | 4 | 753029.47 | 765435.07 | 777840.67 |
| | PAR | 2 | 2069318.83 | 2099526.76 | 2129734.69 |
| | | 4 | 1986226.70 | 2025976.63 | 2065726.56 |

Table B.3: Confidence Interval Information for Figure 5.3

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 0.0301 | 0.0341 | 0.0382 |
| | | 4 | 0.0311 | 0.0381 | 0.0452 |
| | PAR | 2 | 0.4270 | 0.4576 | 0.4882 |
| | | 4 | 0.4600 | 0.4822 | 0.5045 |
| 50 | AODV | 2 | 0.0405 | 0.0461 | 0.0518 |
| | | 4 | 0.0579 | 0.0680 | 0.0780 |
| | PAR | 2 | 1.0539 | 1.1009 | 1.1478 |
| | | 4 | 1.3408 | 1.4127 | 1.4847 |
| 75 | AODV | 2 | 0.0616 | 0.0732 | 0.0847 |
| | | 4 | 0.1155 | 0.1479 | 0.1803 |
| | PAR | 2 | 1.6193 | 1.6621 | 1.7048 |
| | | 4 | 2.0371 | 2.0947 | 2.1523 |

Table B.4: Confidence Interval Information for Figure 5.4

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 0.1420 | 0.1531 | 0.1642 |
| | | 4 | 0.2605 | 0.2804 | 0.3002 |
| | PAR | 2 | 0.1867 | 0.1943 | 0.2020 |
| | | 4 | 0.3055 | 0.3131 | 0.3207 |
| 50 | AODV | 2 | 0.1229 | 0.1298 | 0.1368 |
| | | 4 | 0.2022 | 0.2140 | 0.2258 |
| | PAR | 2 | 0.0797 | 0.0808 | 0.0819 |
| | | 4 | 0.1367 | 0.1391 | 0.1415 |
| 75 | AODV | 2 | 0.0892 | 0.0928 | 0.0963 |
| | | 4 | 0.1376 | 0.1430 | 0.1485 |
| | PAR | 2 | 0.0402 | 0.0408 | 0.0414 |
| | | 4 | 0.0732 | 0.0744 | 0.0756 |

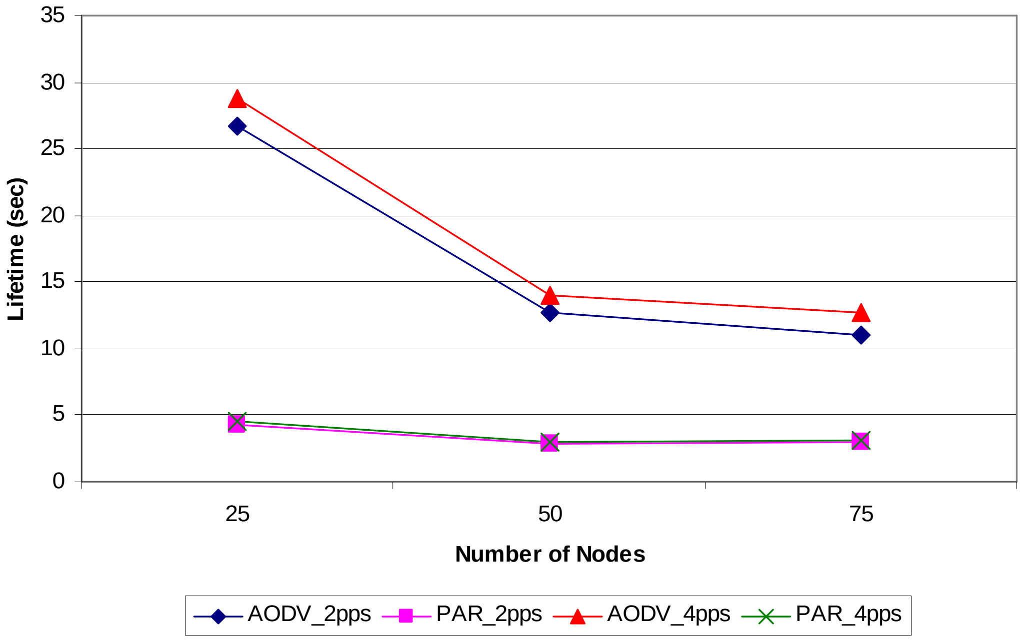


Figure B.1: Route lifetime versus Node Density at 0-20 m/s

Table B.5: Confidence Interval Information for Figure B.1

| Node Density | Protocol | Offered Load | Lower CL | Mean | Upper CL |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 24.08 | 26.74 | 29.40 |
| | | 4 | 26.73 | 28.81 | 30.89 |
| | PAR | 2 | 3.83 | 4.22 | 4.60 |
| | | 4 | 4.09 | 4.52 | 4.95 |
| 50 | AODV | 2 | 11.87 | 12.74 | 13.60 |
| | | 4 | 13.10 | 13.98 | 14.87 |
| | PAR | 2 | 2.84 | 2.88 | 2.92 |
| | | 4 | 2.95 | 3.00 | 3.05 |
| 75 | AODV | 2 | 10.31 | 11.05 | 11.79 |
| | | 4 | 11.77 | 12.71 | 13.66 |
| | PAR | 2 | 2.98 | 3.00 | 3.01 |
| | | 4 | 3.09 | 3.12 | 3.15 |

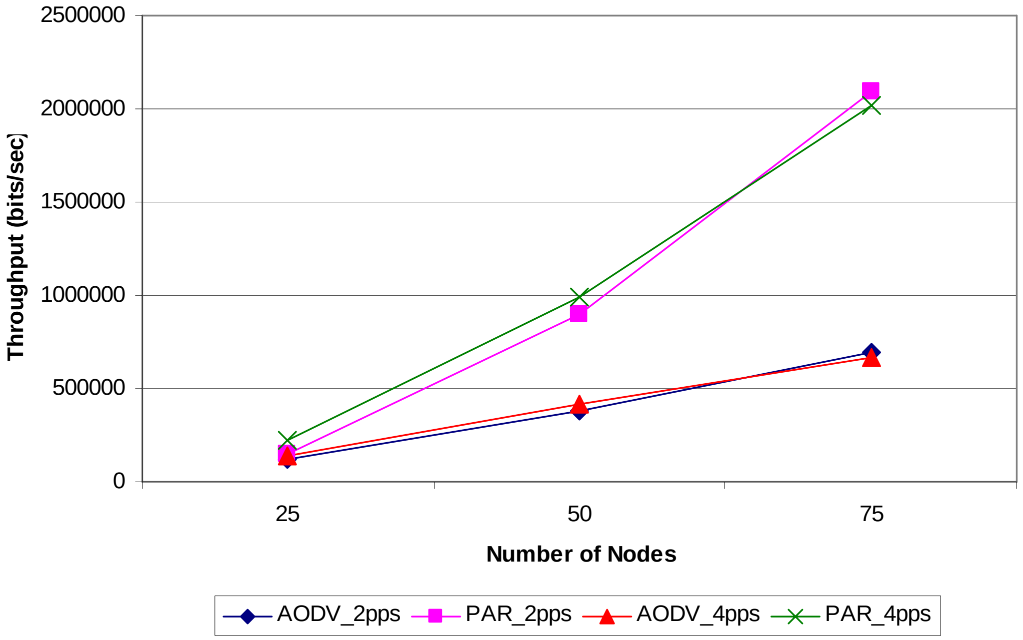


Figure B.2: Throughput versus Node Density at 0-20 m/s

Table B.6: Confidence Interval Information for Figure B.2

| Node Density | Protocol | Offered Load | Lower CL | Mean | Upper CL |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 116271.84 | 121265.44 | 126259.05 |
| | | 4 | 136956.50 | 140713.62 | 144470.75 |
| | PAR | 2 | 140857.12 | 146533.85 | 152210.58 |
| | | 4 | 218530.89 | 226555.02 | 234579.15 |
| 50 | AODV | 2 | 372561.30 | 377829.66 | 383098.03 |
| | | 4 | 405271.62 | 413109.98 | 420948.34 |
| | PAR | 2 | 881429.96 | 895272.20 | 909114.43 |
| | | 4 | 974491.41 | 993156.62 | 1011821.83 |
| 75 | AODV | 2 | 680213.67 | 690666.25 | 701118.83 |
| | | 4 | 650304.77 | 664835.69 | 679366.61 |
| | PAR | 2 | 2062807.36 | 2090982.93 | 2119158.50 |
| | | 4 | 1988979.31 | 2016854.80 | 2044730.30 |

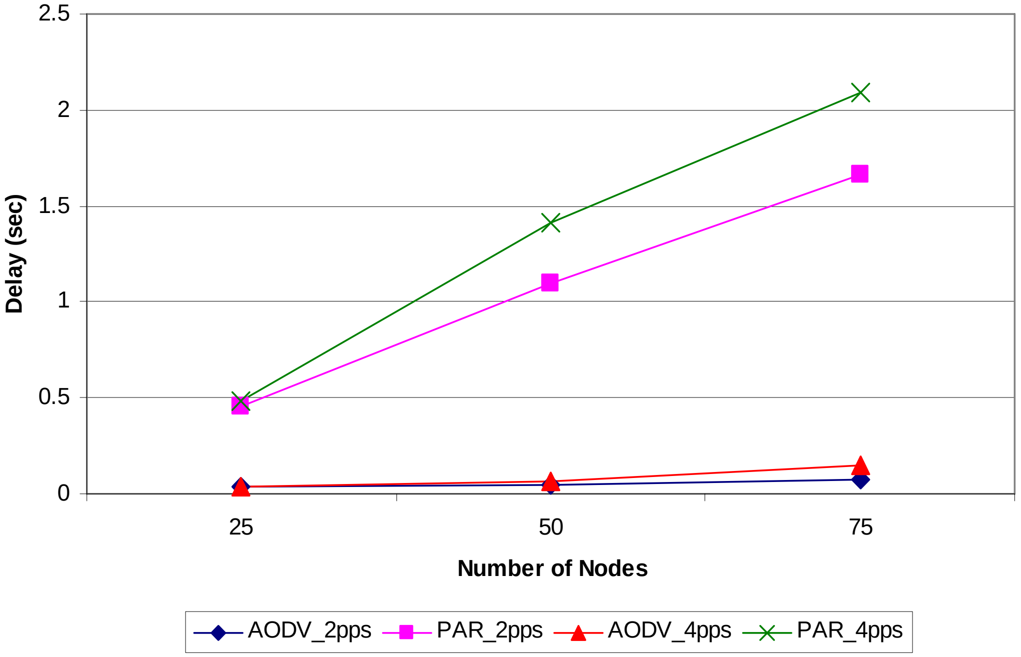


Figure B.3: Delay versus Node Density at 0-20 m/s

Table B.7: Confidence Interval Information for Figure B.3

| Node Density | Protocol | Offered Load | Lower CL | Mean | Upper CL |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 0.0381 | 0.0422 | 0.0462 |
| | | 4 | 0.0331 | 0.0390 | 0.0449 |
| | PAR | 2 | 0.8047 | 0.8358 | 0.8670 |
| | | 4 | 0.7436 | 0.7745 | 0.8055 |
| 50 | AODV | 2 | 0.0438 | 0.0530 | 0.0622 |
| | | 4 | 0.0598 | 0.0692 | 0.0786 |
| | PAR | 2 | 1.3821 | 1.4250 | 1.4680 |
| | | 4 | 1.6590 | 1.6907 | 1.7323 |
| 75 | AODV | 2 | 0.0679 | 0.0808 | 0.0938 |
| | | 4 | 0.0887 | 0.1154 | 0.1422 |
| | PAR | 2 | 1.8617 | 1.9169 | 1.9721 |
| | | 4 | 2.2792 | 2.3465 | 2.4200 |

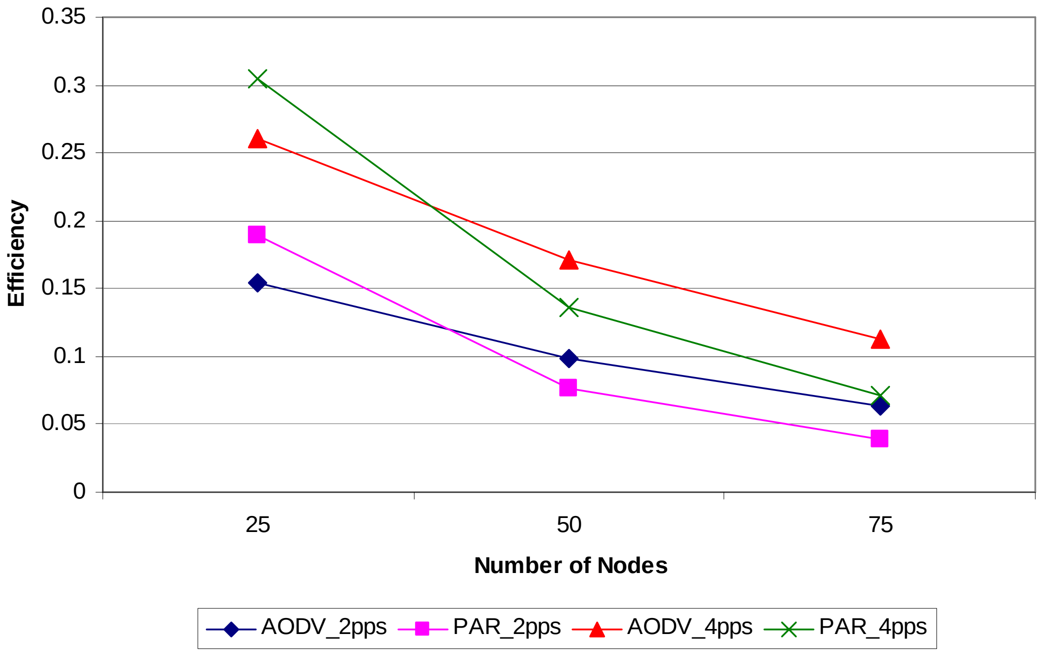


Figure B.4: Efficiency versus Node Density at 0-20 m/s

Table B.8: Confidence Interval Information for Figure B.4

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 0.1452 | 0.1538 | 0.16424 |
| | | 4 | 0.2447 | 0.2602 | 0.2756 |
| | PAR | 2 | 0.1859 | 0.1899 | 0.1939 |
| | | 4 | 0.2964 | 0.3042 | 0.3120 |
| 50 | AODV | 2 | 0.0943 | 0.0981 | 0.1018 |
| | | 4 | 0.1651 | 0.1706 | 0.1762 |
| | PAR | 2 | 0.0758 | 0.0767 | 0.0776 |
| | | 4 | 0.1336 | 0.1357 | 0.1378 |
| 75 | AODV | 2 | 0.0620 | 0.0637 | 0.0654 |
| | | 4 | 0.1093 | 0.1124 | 0.1155 |
| | PAR | 2 | 0.0389 | 0.0393 | 0.0397 |
| | | 4 | 0.0700 | 0.0715 | 0.0729 |

# *Appendix C. Validation of ANOVA Assumptions*

This appendix provides figures validating the assumptions made in the route lifetime ANOVAs in Chapter V. Additionally, discussion is provided on the methods used to transform the data to meet these ANOVA assumptions.

## *C.1 AODV*

The residual plots for AODV route lifetimes, Figure C.1, reveal there exist several high outliers. These outliers make the distribution of residuals non-normal. Further inspection of these points reveal that most of these points correspond to network configurations with node density of 25 nodes. These outliers are expected since, as discussed in Section 5.4, networks with a density of 25 nodes produce significantly longer lasting routes than either 50 or 75 nodes, which produce similar route lifetimes. Given the non-normality of the residuals a transformation is necessary.

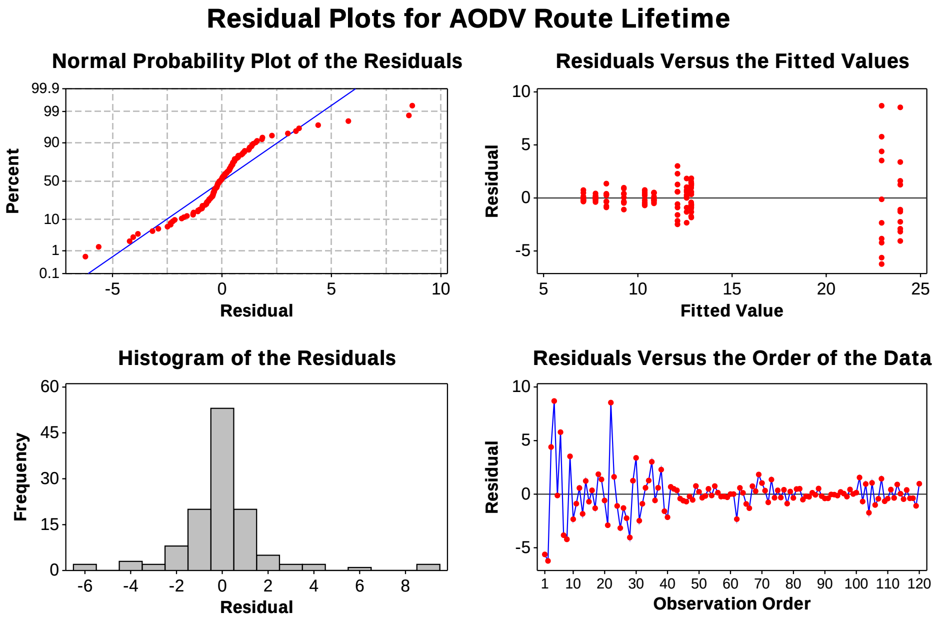


Figure C.1: Residual Plots for AODV Route Lifetime

A series of transformations is performed on the data, including inverse, square root, power, and natural logarithmic transformations. The latter of these transformations provides the best results. The initial transformation demonstrated similar outliers to the initial residual plots. The extreme high outliers are removed from the data set, as they represent anomalous performance. The residual plots representing this transformed and modified data set are depicted in Figure C.2.

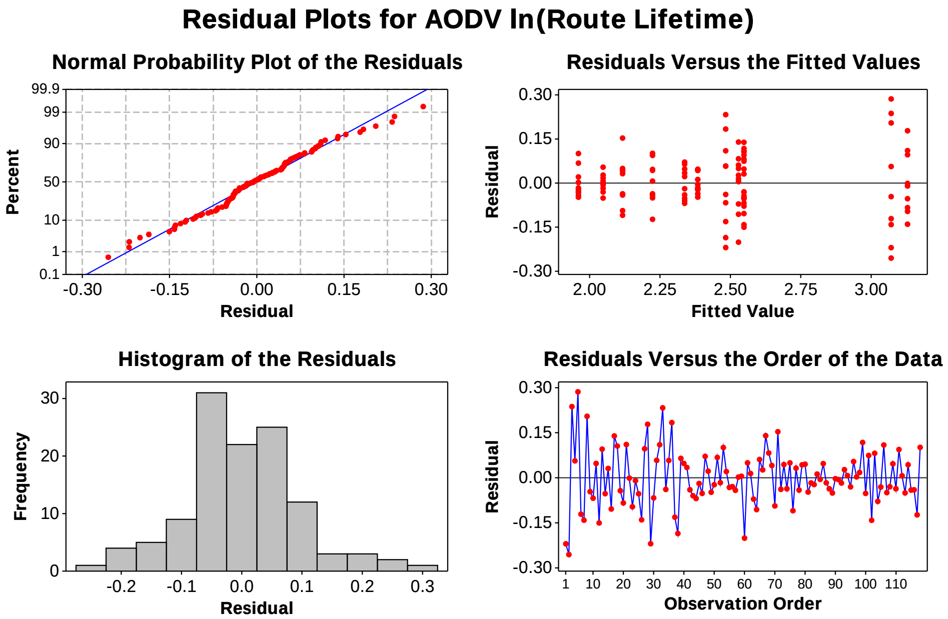


Figure C.2: Residual Plots for AODV ln(Route Lifetime)

Figure C.2 validates the assumptions of the ANOVA. The Normal Probability Plot of Residuals is linear, indicating that the residuals are normally distributed. Furthermore, the Residuals versus Fitted Value Plot demonstrates that the height of the plot is generally consistent and trend free, with only minor growth as the fitted value increases. This indicates constant variance and independent errors. Finally, the plot of Residuals versus Order of Data shows no significant trend, indicating that the residuals are independent of the order of the data.

## *C.2 PAR*

The initial residual plots for PAR route lifetimes, shown in Figure C.3, demonstrate a similar trend to that seen in AODV. In this case, there are both significant high and low outliers in the residuals plots. Like AODV, these points belong to the data for node density of 25 nodes, since this node density produces greater route lifetimes than the other densities included in this research. The result is residuals which are not distributed normally.

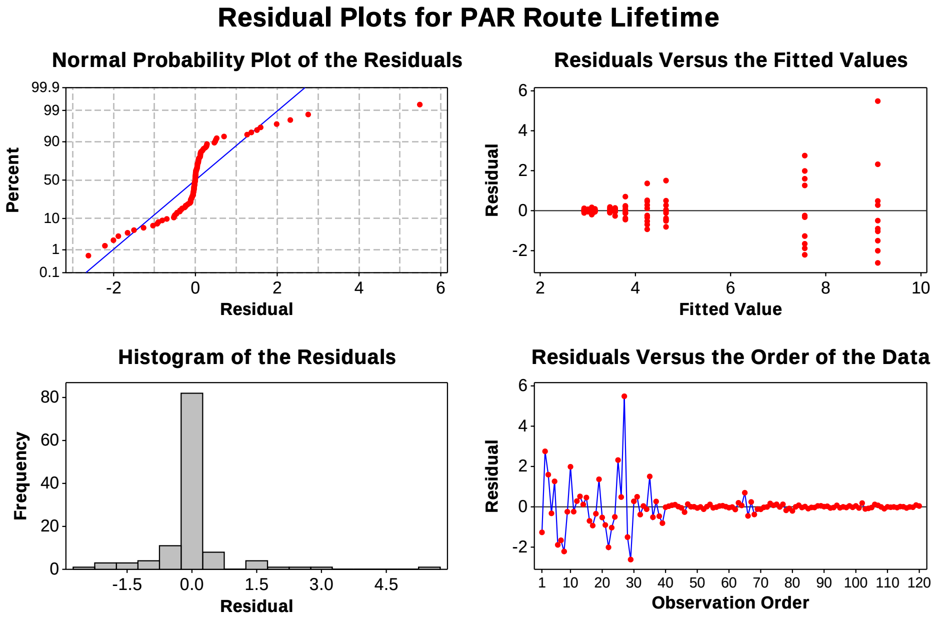


Figure C.3: Residual Plots for PAR Route Lifetime

Given the non-normality of the data, the same set of transformations are applied to the PAR data, as were applied to the data for AODV. In this case the natural logarithmic transform did not achieve data with normally distributed residuals. Rather, the inverse transformation produced data that most closely approximates the normal distribution required. As before, even in the transformed data, the outliers appear. To nullify the effects of this atypical behavior, these outliers are removed from the data set, and the residual plots are reconstructed. Figure C.4 represents the residual plots resulting from this manipulation of the data.

The ANOVA assumptions can be validated by visually inspecting the plots in Figure C.4. Inspection of the Normal Probability Plot of the Residuals reveals that, though not totally linear, the residuals of the transformed data are generally linear in nature. This indicates that the residuals are approximately normal in distribution. The plot Residuals versus Fitted Values demonstrates constant and independent variance, since there is no fanning of the plot, and no discernable trends exist. Finally,

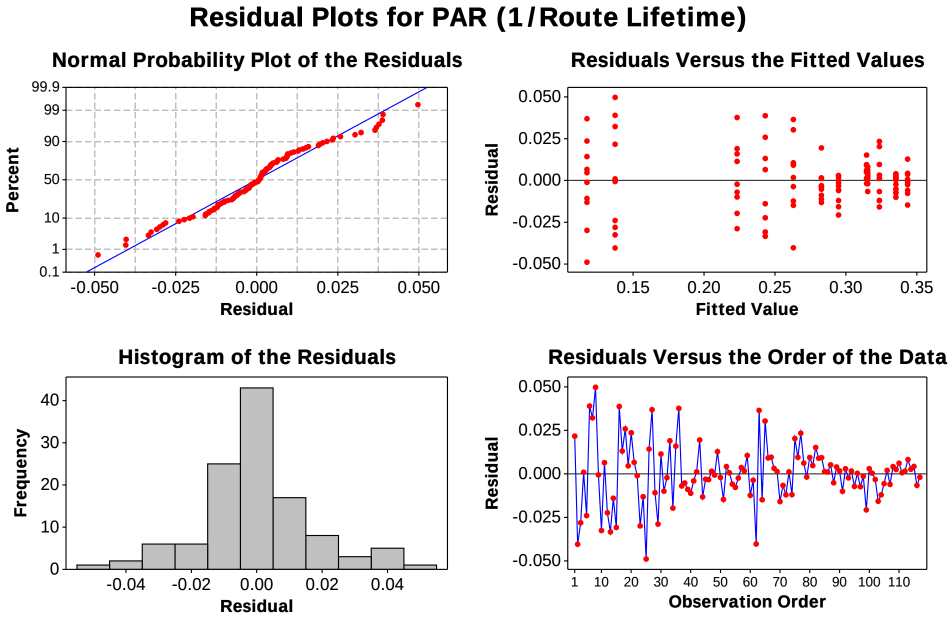


Figure C.4: Residual Plots for PAR (1/Route Lifetime)

examination of the Residuals versus Order of Data plot demonstrates there is no correlation between the residual and the order of the data.

Table D.1: Confidence Interval Information for Figure 5.6

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 1880.98 | 2003.50 | 2126.02 |
| | | 4 | 2084.10 | 2173.70 | 2263.30 |
| | PAR | 2 | 4313.80 | 4473.20 | 4632.60 |
| | | 4 | 4548.27 | 4953.20 | 5358.13 |
| 50 | AODV | 2 | 6875.86 | 7164.20 | 7452.54 |
| | | 4 | 7626.98 | 7848.00 | 8069.03 |
| | PAR | 2 | 26741.95 | 27284.70 | 27827.45 |
| | | 4 | 30131.68 | 31168.70 | 32205.72 |
| 75 | AODV | 2 | 12841.48 | 13185.80 | 13530.12 |
| | | 4 | 12925.60 | 13302.50 | 13679.40 |
| | PAR | 2 | 75031.53 | 77863.00 | 80694.47 |
| | | 4 | 79135.51 | 81346.00 | 83556.49 |

Table D.2: Confidence Interval Information for Figure 5.7

| **Node Density** | **Protocol** | **Offered Load** | **Lower CL** | **Mean** | **Upper CL** |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 1880.98 | 2003.50 | 2126.02 |
| | | 4 | 2084.10 | 2173.70 | 2263.30 |
| | PAR | 2 | 4313.80 | 4473.20 | 4632.60 |
| | | 4 | 4548.27 | 4953.20 | 5358.13 |
| 50 | AODV | 2 | 6875.86 | 7164.20 | 7452.54 |
| | | 4 | 7626.98 | 7848.00 | 8069.03 |
| | PAR | 2 | 26741.95 | 27284.70 | 27827.45 |
| | | 4 | 30131.68 | 31168.70 | 32205.72 |
| 75 | AODV | 2 | 12841.48 | 13185.80 | 13530.12 |
| | | 4 | 12925.60 | 13302.50 | 13679.40 |
| | PAR | 2 | 75031.53 | 77863.00 | 80694.47 |
| | | 4 | 79135.51 | 81346.00 | 83556.49 |

Table D.3: Confidence Interval Information for Figure 5.13

| Node Density | Protocol | Offered Load | Lower CL | Mean | Upper CL |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 51172.78 | 52833.33 | 54493.89 |
| | | 4 | 48112.76 | 50361.26 | 52609.75 |
| | PAR | 2 | 81636.63 | 86121.93 | 90607.24 |
| | | 4 | 78671.04 | 85606.72 | 92542.39 |
| 50 | AODV | 2 | 217721.87 | 224770.77 | 231819.67 |
| | | 4 | 195143.03 | 199553.25 | 203963.46 |
| | PAR | 2 | 583315.98 | 594889.81 | 606463.64 |
| | | 4 | 514245.27 | 539593.64 | 564942.02 |
| 75 | AODV | 2 | 474946.40 | 484109.01 | 493271.61 |
| | | 4 | 384281.32 | 390620.98 | 396960.65 |
| | PAR | 2 | 1621490.09 | 1670960.66 | 1720431.23 |
| | | 4 | 1338230.41 | 1366840.84 | 1395451.27 |

Table D.4: Confidence Interval Information for Figure 5.14

| Node Density | Protocol | Offered Load | Lower CL | Mean | Upper CL |
|---|---|---|---|---|---|
| 25 | AODV | 2 | 17101.53 | 18225.72 | 19349.91 |
| | | 4 | 32272.51 | 34110.12 | 35947.73 |
| | PAR | 2 | 18697.57 | 19677.75 | 20657.93 |
| | | 4 | 32597.79 | 35294.21 | 37990.62 |
| 50 | AODV | 2 | 43137.48 | 44350.35 | 45563.22 |
| | | 4 | 71892.56 | 75074.79 | 78257.02 |
| | PAR | 2 | 50814.52 | 51694.59 | 52574.67 |
| | | 4 | 84381.94 | 87550.75 | 90719.55 |
| 75 | AODV | 2 | 58061.67 | 58966.81 | 59871.96 |
| | | 4 | 81967.11 | 83849.67 | 85732.23 |
| | PAR | 2 | 70026.20 | 71464.05 | 72901.90 |
| | | 4 | 106706.16 | 108145.66 | 109585.16 |

Table D.5: Computation of Effects for AODV Route Lifetime

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 22.9409 | 12.8398 | 23.9233 | 12.1062 | 718.1024 | 17.9526 | 5.3768 |
| **50 Node** | 10.3681 | 7.1081 | 12.5924 | 8.3315 | 384.0014 | 9.6000 | -2.9758 |
| **75 Node** | 10.8560 | 7.7534 | 12.8299 | 9.2599 | 406.9918 | 10.1748 | -2.401 |
| **Col Sum** | 441.6505 | 277.0136 | 493.4556 | 296.9758 | 1509.0956 | | |
| **Col Mean** | 14.7217 | 9.2338 | 16.4485 | 9.8992 | | 12.5758 | |
| **Col Effect** | 2.1459 | -3.3420 | 3.8727 | -2.6766 | | | |

Table D.6: Computation of Effects for AODV Throughput

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 1.245E5 | 1.151E5 | 1.797E5 | 1.642E5 | 5.834E6 | 1.459E5 | -3.482E5 |
| **50 Node** | 4.695E5 | 4.083E5 | 6.214E5 | 4.903E5 | 1.990E7 | 4.974E5 | 3.331E3 |
| **75 Node** | 8.606E5 | 7.345E5 | 9.661E5 | 7.945E5 | 3.356E7 | 8.389E5 | 3.449E5 |
| **Col Sum** | 1.455E7 | 1.258E7 | 1.767E7 | 1.449E7 | 5.929E7 | | |
| **Col Mean** | 4.849E5 | 4.193E5 | 5.890E5 | 4.830E5 | | 4.941E5 | |
| **Col Effect** | -9.187E3 | -7.476E4 | 9.499E4 | -1.105E4 | | | |

Table D.7: Computation of Effects for AODV End-to-End Delay

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 0.0842 | 0.1486 | 0.0730 | 0.1513 | 4.5716 | 0.1143 | -0.0878 |
| **50 Node** | 0.1310 | 0.2327 | 0.1639 | 0.2619 | 7.8942 | 0.1974 | -0.0048 |
| **75 Node** | 0.1879 | 0.2718 | 0.2915 | 0.4279 | 11.7908 | 0.2948 | 0.0926 |
| **Col Sum** | 4.0304 | 6.5309 | 5.2837 | 8.4115 | 24.2565 | | |
| **Col Mean** | 0.1343 | 0.2177 | 0.1761 | 0.2804 | | 0.2021 | |
| **Col Effect** | -0.0678 | 0.0156 | -0.0260 | 0.0782 | | | |

Table D.8: Computation of Effects for AODV Efficiency

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 0.2562 | 0.2261 | 0.4037 | 0.3730 | 12.5900 | 0.3147 | 0.1045 |
| **50 Node** | 0.1649 | 0.1238 | 0.2731 | 0.2151 | 7.7691 | 0.1942 | -0.0161 |
| **75 Node** | 0.1086 | 0.0767 | 0.1768 | 0.1255 | 4.8758 | 0.1219 | -0.0884 |
| **Col Sum** | 5.2973 | 4.2651 | 8.5360 | 7.1364 | 25.2348 | | |
| **Col Mean** | 0.1766 | 0.1422 | 0.2845 | 0.2379 | | 0.2103 | |
| **Col Effect** | -0.0337 | -0.0681 | 0.0742 | 0.0276 | | | |

Table D.9: Computation of Effects for PAR Route Lifetime

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 7.5587 | 4.2463 | 9.0935 | 4.6426 | 255.1440 | 6.3853 | 2.0814 |
| **50 Node** | 3.5732 | 2.9231 | 3.7896 | 3.0827 | 133.6860 | 3.3422 | -0.9618 |
| **75 Node** | 3.1202 | 3.0030 | 3.4646 | 3.1495 | 127.3724 | 3.1843 | -1.1196 |
| **Col Sum** | 142.5209 | 101.7234 | 163.4766 | 108.7485 | 516.4695 | | |
| **Col Mean** | 4.7507 | 3.3908 | 5.4492 | 3.6250 | | 4.3039 | |
| **Col Effect** | 0.4468 | -0.9131 | 1.1453 | -0.6790 | | | |

Table D.10: Computation of Effects for PAR Throughput

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 1.605E5 | 1.499E5 | 2.217E5 | 2.125E5 | 7.446E6 | 1.862E5 | -8.816E5 |
| **50 Node** | 8.645E5 | 9.060E5 | 9.922E5 | 1.001E6 | 3.763E7 | 9.408E5 | -1.269E5 |
| **75 Node** | 2.089E6 | 2.151E6 | 2.031E6 | 2.034E6 | 8.305E7 | 2.076E6 | 1.009E6 |
| **Col Sum** | 3.114E7 | 3.207E7 | 3.245E7 | 3.247E7 | 1.281E8 | | |
| **Col Mean** | 1.038E6 | 1.069E6 | 1.082E6 | 1.082E6 | | 1.068E6 | |
| **Col Effect** | -2.967E4 | 1.137E3 | 1.391E4 | 1.462E4 | | | |

Table D.11: Computation of Effects for PAR End-to-End Delay

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 0.5248 | 0.8942 | 0.5874 | 0.8896 | 28.9614 | 0.7240 | -0.6976 |
| **50 Node** | 1.2353 | 1.4921 | 1.5227 | 1.7759 | 60.2598 | 1.5065 | 0.0848 |
| **75 Node** | 1.6879 | 1.9389 | 2.1582 | 2.3527 | 81.3771 | 2.0344 | 0.6128 |
| **Col Sum** | 34.4801 | 43.2529 | 42.6833 | 50.1818 | 170.5982 | | |
| **Col Mean** | 1.1493 | 1.4418 | 1.4228 | 1.6727 | | 1.4217 | |
| **Col Effect** | -0.2723 | 0.0201 | 0.0011 | 0.2511 | | | |

Table D.12: Computation of Effects for PAR Efficiency

| | **2 pps** | | **4 pps** | | **Row** | **Row** | **Row** |
|---|---|---|---|---|---|---|---|
| | **5 m/s** | **20 m/s** | **5 m/s** | **20 m/s** | **Sum** | **Mean** | **Effect** |
| **25 Node** | 0.1861 | 0.1879 | 0.2924 | 0.3015 | 9.6790 | 0.2420 | 0.1068 |
| **50 Node** | 0.0800 | 0.0753 | 0.1398 | 0.1338 | 4.2894 | 0.1072 | -0.02791 |
| **75 Node** | 0.0411 | 0.0390 | 0.0734 | 0.0715 | 2.2493 | 0.0562 | -0.0789 |
| **Col Sum** | 3.0714 | 3.0218 | 5.0560 | 5.0686 | 16.2177 | | |
| **Col Mean** | 0.1024 | 0.1007 | 0.1685 | 0.1690 | | 0.1351 | |
| **Col Effect** | -0.0328 | -0.0344 | 0.0334 | 0.0338 | | | |

# *Bibliography*

BKS03. Vartika Bhandari, Nupur Kothari, and Dheeraj Sanghi. A reliable on-demand routing paradigm for mobile ad hoc networks". Technical report, Indian Institute of Technology, Kanpur, India, 2003.

CBD02. Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5), 2002.

ChG98. Tsu-Wei Chen and Mario Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. In *IEEE International Conference on Communications*, June 1998.

HaP01. Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for zone routing protocol. *IEEE/ACM Transactions on Networking*, 9(4), August 2001

Ily03. Mohammad Ilyas. *The Handbook of Ad Hoc Wireless Networks*, pages 7-1 - 7-19. CRC Press, 2003.

JoM96. David B. Johnson and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, volume 353. Kluwer Academic Publishers, 1996.

Joh94. David B. Johnson. Routing in ad hoc networks of mobile hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.

KaO02. Tom Karygiannis and Les Owens. Wireless network security: 802.11, bluetooth, and handheld devices. Technical report, National Institute of Standards and Technology, Gaithersburg, Maryland, November 2002.

MiA03. J. Susan Milton and Jesse C. Arnold. *Introduction to Probability and Statistics.* McGraw Hill, fourth edition, 2003.

MaD01. Mahesh K. Marina and Samir R. Das. On-demand multipath distance vector routing in ad hoc networks. In *ICNP*, pages 14-23. IEEE Computer Society, 2001.

Mis99. Padmini Misra. Routing protocols for ad hoc mobile wireless networks. Technical report, The Ohio State University, Columbus, Ohio, November 1999.

MuM04. C. Siva Ram Murthy and B. S. Manoj. *Ad Hoc Wireless Networks Architecture and Protocols*, pages 299-359. Prentice Hall, 2004.

PeB94. Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computing. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols, and Applications*, pages 234-244, 1994.

PGC00. Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *IEEE International Conference on Communications (1)*, pages 70-74, 2000.

PeR97. Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *MILCOM 97*, November 1997.

SSB99. Rahhupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. CEDAR: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas In Communications*, 17(8), August 1999.

Toh97. Chai-Keong Toh. Associativity-based routing for ad-hoc mobile networks. *Wireless Personal Communications*, 4(2), March 1997.

YKT03. Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In *INFOCOM*, San Francisco, California, 2003.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704–0188*

**1. REPORT DATE** *(DD–MM–YYYY)*
13–06–2006

**2. REPORT TYPE**
Master's Thesis

**3. DATES COVERED** *(From — To)*
Aug 2004 — Jun 2006

**4. TITLE AND SUBTITLE**

Evaluation of the Effects of Predicted Associativity
On the Reliability and Performance
Of Mobile Ad Hoc Networks

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Esteban Francisco Sanchez, Capt, USAF

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way Bldg 640
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GCE/ENG/06-05

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Mr. Scott Gardner (Scott.Gardner.2@scott.af.mil)
AFCA/ENAN (HAF)
203 West Losey St
Scott Air Force Base, IL 62225
DSN779-6794

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approval for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT** Routing in Mobile Ad Hoc Networks (MANETs) presents unique challenges not encountered in conventional networks. Predicted Associativity Routing (PAR) is a protocol designed to address reliability in MANETs. Using associativity information, PAR calculates the expected lifetime of neighboring links. Nodes use this expected lifetime, and their neighbor's connectivity to determine a residual lifetime. The routes are selected from those with the longest residual lifetimes. In this way, PAR attempts to improve the reliability of discovered routes. PAR is compared to AODV using a variety of reliability and performance metrics. Despite its focus on reliability, PAR does not provide more reliable routes. Rather, AODV produces routes which last as much as three times longer than PAR. However, PAR delivers more data and has greater throughput. Both protocols are affected most by the node density of the networks. Node density accounts for 48.62% of the variation in route lifetime in AODV, and 70.66% of the variation in PAR. As node density increases from 25 to 75 nodes route lifetimes are halved, while throughput increases drastically with the increased routing overhead. Furthermore, PAR increases end-to-end delay, while AODV displays better efficiency.

**15. SUBJECT TERMS**

reliable routing, MANET routing, Predicted Associativity Routing, residual route lifetime

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Dr. Barry Mullins |
| | | | | | **19b. TELEPHONE NUMBER** *(include area code)* |
| U | U | U | UU | 147 | (937) 255–6565, ext 7979 Barry.Mullins@afit.edu |