2008

Routing Protocol Performance Evaluation for Mobile Ad-hoc Networks

Pedro A. Lopez-Fernandez
University of North Florida

ROUTING PROTOCOL PERFORMANCE EVALUATION FOR MOBILE AD-HOC NETWORKS

by

Pedro A. Lopez-Fernandez

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

December 2008

The thesis "Routing Protocol Performance Evaluation for Mobile Ad-Hoc Networks" submitted by Pedro A. Lopez-Fernandez in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

Signature Deleted

2/25/08

Dr. Sanjay P. Ahuja
Thesis Advisor and Committee Chairperson

Signature Deleted

7/25/08

Dr. Zornitza G. Prodanoff
Thesis Advisor

Signature Deleted

7/25/08

Dr. Susan Vasana

Accepted for the School of Computing:

Signature Deleted

12/15/08

Dr. Judith Solano
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

Signature Deleted

12/22/08

Dr. Neal S. Coulter
Dean of the College

Accepted for the University:

Signature Deleted

23 Dec 2008

Dr. David E.W. Fenner
Dean of the Graduate School

CONTENTS

FIGURES

TABLES

ABSTRACT

Currently, MANETs are a very active area of research, due to their great potential to provide networking capabilities when it is not feasible to have a fixed infrastructure in place, or to provide a complement to the existing infrastructure. Routing in this kind of network is much more challenging than in conventional networks, due to its mobile nature and limited power and hardware resources.

The most practical way to conduct routing studies of MANETs is by means of simulators such as GloMoSim. GloMoSim was utilized in this research to investigate various performance statistics and draw comparisons among different MANET routing protocols, namely AODV, LAR (augmenting DSR), FSR (also known as Fisheye), WRP, and Bellman-Ford (algorithm). The network application used was FTP, and the network traffic was generated with tcplib [Danzig91]. The performance statistics investigated were application bytes received, normalized application bytes received, routing control packets transmitted, and application byte delivery ratio.

The scenarios tested consisted of an airborne application at a high (26.8 m/s) and a low speed (2.7 m/s) on a 2000 m x 2000 m domain for nodal values of 36, 49, 64, 81, and 100 nodes, and radio transmit power levels of 7.005, 8.589, and 10.527 dBm. Nodes were paired up in fixed client-server couples involving 10% and 25% of the nodes being

clients and the same quantity being servers. AODV and LAR showed a significant margin of performance advantage over the remaining protocols in the scenarios tested.

Chapter 1

INTRODUCTION AND MOTIVATION

Mobile Ad-hoc networks (MANETs) are networks of mobile devices, typically referred to as nodes, which communicate with each other wirelessly without having to resort to any kind of pre-existing infrastructure. The nodes can communicate with existing infrastructure (such as the Internet) as well, if required.

MANETs allow the set-up of networks "on the spot" in a quick fashion, without reliance on existing facilities and infrastructure [Macker99, Murthy04]. In this manner, an organization can set up a network just about anywhere, either in a temporary (such as in response to an emergency or in mobile military operations) or in a permanent or semi-permanent manner (such as in border monitoring). The inherent ability to move the nodes about and yet maintain the connectivity is an attribute that makes MANETs the only kind of network suitable to certain situations. This kind of flexibility makes MANETs an invaluable addition to traditional networking technologies, leading to a great deal of research and development.

Wireless sensor networks (WSNs) are specialized relatives of MANETs whose main purpose is to sense or monitor a predetermined type of event, such as vibration levels, temperatures, and pressure in various environments [Ayildiz02], many of them hostile to or difficult to access by humans. WSNs do not necessarily have to be mobile,

although they can be, and often cannot rely on human intervention for operation once deployed. In contrast with regular MANETs, sensor networks often have numbers of nodes several orders of magnitude larger, are data-centric (for example, nodes in stand-by status that sense the local temperature reach a certain value become fully active), and feature data aggregation (nodes aggregate local information before relaying the information back to where it needs to go). They have nodes more prone to failure (they are simpler and cheaper nodes that rely on large numbers and further deployments rather than upkeep/maintenance), and have more limited hardware resources (such as memory, to keep cost down) [He04, Murthy04]. Economy of energy usage is even more important than in conventional MANETs due to the deployment characteristics of WSNs, leading to trade-offs between sensitivity and energy-usage [He04, Yan03]. This leads to their networking protocols at various layers being highly specialized and different from those of conventional MANETs [Murthy04].

The special characteristics that distinguish MANETs from conventional networks gives rise to certain performance issues to which they are particularly susceptible. One does not just worry about the typical issues affecting a more conventional network, but also about issues such as node mobility (which implies constant topology changes), limited bandwidth, and power conservation. Link unidirectionality is particularly important for MANETs [Chun00, Macker99], since the nodes radio equipment may be heterogeneous, some nodes may be more susceptible to interference from various sources, giving rise to different radio ranges for different nodes.

This project was directed at investigating the performance of various routing protocols in conventional, bidirectional MANETs. Various measures of performance were evaluated and compared between the different routing protocols. How each protocol was affected by different levels of mobility, nodal density, and radio range was of special interest.

Chapter 2

MOBILE AD-HOC NETWORK (MANET)

The definitions of the word ad hoc include [theFreeDictionary.com08]: (1) Formed for or concerned with one specific purpose, and (2) improvised and often impromptu. Both of these meanings, together with the idea of mobility, very adeptly characterize the nature of MANETs. MANET nodes, due to their mobile character and ad-hoc deployment capability, need to be smarter than typical hosts in conventional networks, with each node having to be able to perform routing functions in a network whose topology may change at any time in unpredictable ways and become part of a self-forming temporary network [Chun00, Macker99, Murthy04].

Efficiency and economy are also key requirements of MANET nodes; wired networks can always be made to outperform wireless ones in terms of bandwidth, and their power needs are easy to satisfy by comparison. MANETs on the other hand have to be able to provide satisfactory services to the users while making use of restrained bandwidth availability, and their nodes should ideally last as long as possible without requiring human intervention. This requires efficient use of the limited energy resources they carry with them (usually in the form of batteries, but other possibilities exist, such as the fuel that drives a generator in a vehicle).

MANETs are ideally suited for many applications, as stated previously. What follows is but a brief sampling and description of a few promising applications [Macker99, Murthy04]. There are many other possible applications, and as experience with MANETs increases and the technologies mature and improve, the number of applications and the frequency of actual deployments will undoubtedly increase.

1. Disaster relief: It often happens that after a major natural disaster the infrastructure (taken here to mean all infrastructure in general and communications infrastructure in particular) of the devastated area is destroyed or rendered inoperable. A recent example is the devastation in New Orleans and other parts of the Gulf of Mexico coast of the United States in 2005 due to hurricane Katrina. It may also happen that the affected area had a poor or non-existent infrastructure to begin with, such as many of the areas affected by the recent Indian Ocean tsunami in 2005. Rescue efforts could greatly benefit by setting up temporary MANETs whose nodes are individual rescue units, both on land and in the air.

2. Surveillance: Aerial and land vehicles/personnel tasked with keeping watch over some swath of territory, such as the southern border of the United States or parts of Afghanistan or Iraq, can greatly benefit from being able to coordinate their actions in a more integrated form than is possible with conventional forms of communication. In a MANET context, what one node sees all nodes also see, in essence allowing the force to act as a unified, coherent whole. Notice that the concept of surveillance is not restricted to detecting intruders; it also extends to

detecting or monitoring other threats or situations, such as forest fires. The surveillance nodes can be aerial drones, land robots, individuals, and simple wireless sensor nodes.

3. Military: Soldiers or vehicles operating in hostile territory cannot afford to depend on any kind of pre-existing infrastructure for communications. A MANET can provide a coherent view of the situation to all those involved, and allow those in charge to coordinate the actions of their forces in a more effective manner than would otherwise be possible. Multimedia capabilities typically provided by today's portable computers could enhance the overall level of situational awareness and make any miscommunications less likely. Notice that the surveillance application mentioned previously can also fall in the military category.

4. Exploration: Robotic or manned vehicles sent to other planets could constitute a MANET for coordinated exploration of a given area of terrain. A large interesting feature found by a node could be communicated to the other nodes, in order to have additional nodes (with potentially more suited equipment) investigate the feature. Similarly, a node could alert other nodes of some sort of dangerous condition it has encountered, such as quicksand or slippery ground, or ask for assistance if need be.

5. Air Traffic Control: The control of aircraft approaching and departing airports could be enhanced by MANETs. The nodes would be the aircraft and the ground control stations. All kinds of relevant information, such as fuel state, could be

exchanged in addition to position, velocity, altitude, and identification. Multimedia information could also be exchanged, and all the airplanes could have self-awareness of their neighbors. Aircraft further away from the airport than is common today could also become aware of the general situation by also becoming part of the network (by being within transmission range of airplanes closer to the airport). Since the aircraft are equipped with precise equipment to pinpoint their spatial location and velocity, and their general movements follow some general rules and patterns, this is an ideal application for the LAR scheme (described later) to enhance the "main" routing protocol.

6. Wireless Sensor Networks: The number of situations in which MANETs can be used for sensing activities is very large. There are many situations in which a rapid sensor deployment capability in hostile or hard to reach territory is very desirable, such as in military tracking of vehicles or personnel [He04, Galstyan04, Ayildiz02]. Civil defense and the military alike can greatly benefit from sensor networks to detect biological or chemical attack [Ayildiz02, Murthy04]. Animal studies can make use of networks of sensors that do not inhibit the behavior of the animals in the wild [Mainwaring02]. Geological and other natural activities in remote areas or too dangerous to humans, such as forest fires [Ayildiz02], monitoring of physical phenomena harmful to people, such as radiation [Brennan04, Ayildiz02] or volcanic eruptions [Werner05], are just a few more examples of applications that can effectively be carried out by sensor networks.

Chapter 3

LITERATURE SURVEY

The existing literature on MANETs is very extensive. An extremely comprehensive work is "Ad Hoc Wireless Networks – Architectures and Protocols" [Murthy04], which extensively covers most issues having to do with the subject, whereas "Tutorial on Wireless Ad Hoc Networks" [Remondo04] provides a brief introduction. MANET design issues, such as routing architecture in light of the nature of MANETS, unidirectional link support, QoS routing, and multicast support are discussed in "Routing Protocols Overview and Design Issues for Self-Organized Network" [Chun00]. In "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations" [Macker99], the authors cover some of the same design issues as in "Routing Protocols Overview and Design Issues for Self-Organized Network" [Chun00], but they augment them with some additional ones, such as limited bandwidth, energy-constrained operation, and limited physical security. Also covered are desirable properties of MANETS, such as distributed operation, loop freedom, demand-based and proactive operation modes, and security, and desirable metrics to use in quantifying MANET performance. The inclusion of important metrics to consider in "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations" [Macker99] was of great assistance to the present work.

Many routing protocols for MANETs have been developed, and the area is one of intense research. A rather exhaustive listing of existing MANET routing protocols can be found in "Ad Hoc Wireless Networks – Architectures and Protocols" [Murthy04], conveniently grouped in several categories. The sheer number of MANET routing protocols makes it impractical to list every one of them; however, a brief sampling follows, grouped by the routing information update mechanism and including one or more references.

Proactive Protocols:

- APRL (Any Path Routing without Loops) is described in "Dynamic Neighbor Discovery and Loop-Free, Multi-Hop Routing for Wireless, Mobile Networks" [Karp98].
- CGSR (Cluster-Head Gateway Switch) is described in "Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel" [Chiang97].
- DSDV (Destination Sequenced Distance-Vector) is described in "Ad-hoc On-Demand Distance Vector Routing" [Perkins94].
- FSR (Fisheye State Routing, commonly referred to as Fisheye) is described in "Scalable Routing Strategies for Ad Hoc Wireless Networks" [Iwata99].
- GSR (Global State Routing) is described in "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks" [Chen98].
- HSR (Hierarchical State Routing) is described in "Scalable Routing Strategies for Ad Hoc Wireless Networks" [Iwata99].

- OLSR (Optimized Link State Routing) is described in "The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation" [Clausen01], "Optimized Link State Routing Protocol" [Clausen03], and "Optimized Link State Routing Protocol for Ad Hoc Networks" [Jacquet01].
- STAR (Source-Tree Adaptive Routing) is described in "Transmission-Efficient Routing in Wireless Networks Using Link-State Information" [Garcia01].
- WRP (Wireless Routing Protocol) is described in "A Routing Protocol for Packet Radio Networks" [Murthy95] and "An Efficient Routing Protocol for Wireless Networks" [Murthy96].

Reactive Protocols:

- ABR (Associativity-Based Routing) is described in "Associativity Based Routing for Ad Hoc Mobile Networks" [Toh97].
- AODV (Ad Hoc On-Demand Distance-Vector Routing) is described in "Evolution and Future Directions of the Ad-hoc On-Demand Distance Vector Routing Protocol" [Belding03], "Ad-hoc On-Demand Distance Vector Routing" [Perkins99], "A Quick Guide to AODV Routing" [Klein08], and "AODV Routing Implementation for Scalable Wireless Ad-Hoc Network Simulation (SWANS)" [Lin04].
- DSR (Dynamic Source Routing) is described in "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks" [Johnson01].

- FORP (Flow-Oriented Routing Protocol) is described in "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction" [Gerla99].
- LAR (Location Aided Routing) is described in "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks" [Ko00].
- ODMRP (On-Demand Multicast Routing Protocol) is described in "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks" [Lee02].
- PAOD (Power-Aware On-Demand) is described in "Power-Aware On-Demand Routing Protocol for MANET" [Kun04].
- PLBR (Preferred Link-Based Routing) is described in "A Preferred Link-Based Routing Protocol for Ad Hoc Wireless Networks" [Sisodia02].
- RDMAR (Relative Distance Micro-discovery Ad-hoc Routing) is described in "RDMAR: A Bandwidth-efficient Routing Protocol for Mobile Ad hoc Networks" [Aggelou99].
- GRAd (Gradient Routing in Ad-hoc networks) is described in "Gradient Routing in Ad Hoc Networks" [Poor08].
- SSA (Signal-Stability Based Adaptive) is described in "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks" [Dube97].
- TORA (Temporarily Ordered Routing Algorithm) is described in "Temporarily-Ordered Routing Algorithm" [SECAN-LAB05A] and "Trusted Route Discovery with the TORA Protocol" [Pirzada04].

Hybrid Protocols:

- CEDAR (Core Extraction Distributed Ad Hoc Routing) is described in "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm" [Sinha99].
- IZR (Independent Zone Routing) is described in "Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks" [Samar04].
- STARA (System and Traffic dependent Adaptive Routing Algorithm) is described in "A System and Traffic Dependent Adaptive Routing Algorithm for Ad Hoc Networks" [Gupta97].
- ZHLS (Zone-Based Hierarchical Link State) is described in "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks" [Joa02].
- ZRP (Zone Routing Protocol) is described in "Zone Routing Protocol (ZRP)" [Beijar02], "Determining the Optimal Configuration for the Zone Routing Protocol" [Haas99], and "The Performance of Query Control Schemes for the Zone Routing Protocol" [Haas01].

Much work has been done to evaluate the performance of MANET routing protocols and compare them, often using simulation to conduct the studies. Since published studies typically choose a few of the available protocols to run experiments in the simulator of choice, it is convenient to group these studies by the simulator used. A sampling of the available papers, concentrating mainly on the GloMoSim simulator (used in this work) and the nS-2 simulator (which is the most often used simulator) follows:

GloMoSim studies:

AODV, DSR, STAR are investigated in "Performance Comparison of Three Routing Protocols for Ad Hoc Networks" [Jiang04].

- AODV, WRP, DV, DSR are investigated in "Study of MANET Routing Protocols by GloMoSim Simulator" [Pandey05].
- AODV is investigated in "Scalability Study of the Ad Hoc On-demand Distance Vector Routing Protocol," [Lee03].

nS-2 studies:

- LAR, DREAM, DSR are investigated in "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks" [Camp02].
- DSR, AODV, PAODV, CBRP, DSDV are investigated in "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks" [Boukerche04].
- DSR, AODV, CBRP are investigated in "A Performance Comparison of Routing Protocols for Ad Hoc Networks" [Boukerche01A] and "A Simulation Based Study of On-Demand Routing Protocols for Ad Hoc Wireless Networks" [Boukerche01B].
- DSDV, AODV, DSR are investigated in "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks" [Johansson99].
- DSDV, TORA, DSR, AODV are investigated in "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" [Broch98].

- DSR, AODV are investigated in "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks" [Das00].

Opnet studies:

- DSR, AODV, TORA are investigated in "A Performance Comparison of On-Demand Routing Protocols for Application Data in Mobile Ad hoc Networks" [Lee05].

Although simulation is the most expedient and least expensive way to study MANETs, some published studies that conducted actual, in-the-field experiments, are available also. "Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms" [Gray04] is one such study; it evaluates the APRL, AODV, ODMRP, and STARA routing protocols.

Many papers have been published on the subject of WSNs. An excellent survey is presented in "Wireless Sensor Networks: A Survey" [Ayildiz02], where the authors describe the main differences between WSNs and MANETs (larger order of magnitude of nodes, higher node density, and less reliability), various applications, factors influencing the design (fault tolerance, scalability, cost, and hardware constrains), the deployment environment, the general architecture, and details of the various layers. A smaller but also very complete survey can be found in "Ad Hoc Wireless Networks – Architectures and Protocols" [Murthy04]. A basic overview of the WSN routing

protocols and design issues they raise are presented in "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations" [Macker99]. The critical issue of energy conservation is discussed in "Energy-Efficient Surveillance System Using Wireless Sensor Networks" [He04], where a particular system is discussed; the work encompasses the requirements that had to be met, a system overview and description, the system operation, the actual implementation (hardware and software), and actual testing results. The work in "Differentiated Surveillance for Sensor Networks" [Yan03] presents an effective, energy efficient sensing protocol for WSNs. Particular applications of WSNs in various fields are discussed in "Radiation Detection with Distributed Sensor Networks" [Brennan04], "Wireless Sensor Networks for Habitat Monitoring" [Mainwaring02], and "Monitoring Volcanic Eruptions with a Wireless Sensor Network" [Werner05].

Chapter 4

MANET ROUTING PROTOCOLS

There are various ways to classify MANETs based on the characteristics of their routing protocols. Some typical ways to classify them are the following [Remondo04, Royer99, Murthy04]:

1. Proactive versus reactive: Proactive routing protocols, also called (routing) table-driven, seek to maintain routing tables at every node containing up to date routing information to all the other nodes in the network. Routing information updates to reflect changes in the network topology are propagated throughout the network, which entails a good deal of overhead. On the other hand, when a message needs to be sent from one node to another node, the response time (for routing at least) of the sending node is very short for actually sending the message, since the route is readily available to the node.

   Reactive protocols, which are also called demand-driven protocols, do not seek to maintain a full view of the topology of the entire network at every node. Instead, the necessary routing information is obtained when the sending node needs it, with the cooperation of other nodes. This information is kept for some time and then discarded. This process of getting routing information is known as route discovery. Since the topology of the network can change as a node sends information to

another node, a route maintenance procedure is also required when topological changes occur.

Reactive protocols require less routing overhead than proactive ones, since there is no continuous updating of the routing information at each node to all other nodes. On the other hand, they tend to suffer from a lag in sending messages when compared to proactive protocols, since the routing information, if not readily available, has to be obtained via route discovery.

Hybrid protocols attempt to combine the best features of proactive and reactive protocols. For each node, a table driven approach is applied within a given zone around the node, and a demand-driven approach is then applied outside of that zone.

2. Location-based versus non location-based: In location-based protocols, the nodes acquire information about their relative geographical location with respect to other nodes in the network. This information is then used to aid in routing decisions, which in theory results in a more efficient routing algorithm. Non-location-based routing does not make use of geographical information, relying instead on "hop" information. One disadvantage of location-based routing is the reliance on and extra complication of obtaining the geographical information. One possibility is GPS; another is some sort of INU (Inertial Navigation Unit). The additional

equipment adds to the complexity (more components that may fail) and power requirements of the nodes (shorter endurance).

3. Hierarchical-topology versus flat-topology: Hierarchical-topology protocols make use of some sort of hierarchy within the network with an associated addressing mechanism in an attempt to increase the efficiency of the routing compared to other methods; an example of hierarchical routing is the Internet itself. The hierarchy used in a MANET is not fixed since it can vary due to different factors, the most important of which is the overall topology of the network, which is a fluid entity due to node mobility.

   The flat-topology type of routing protocols is the exact opposite: The addressing scheme is flat, un-hierarchical. Each node has a unique address that does not denote any kind of membership in any subgroup of the network, that is, all nodes are equal peers.

In the next few paragraphs, one routing algorithm and five routing protocols with GloMoSim implementations tested in this study are discussed.

1. Bellman Ford [Tanenbaum03, SECAN-LAB05B]

   Bellman Ford is a proactive, non-location-based, flat-topology routing algorithm used in Distance Vector Routing (DVR) protocols. Each node i maintains a routing

table that contains for every destination node j in the network the successor node and the estimated distance (or cost) d^i_j; hops are a typical measure for distance, and an unknown distance is set to infinity. In addition, each node i is assumed to know who its k neighbors are and its distance to each of them. Node i periodically receives from each of its neighbors their routing tables. It can then refine its distance to all the nodes in the network by finding the minimum of $d^i_k + d^k_j$, for every destination j in the network and using the information supplied by each of its neighbors. The same is true of all the other nodes in the network. It takes several iterations of information exchange for each node to converge to the set of minimum distances to all the nodes in the network. Increasing the mobility of the nodes in the network will make it more and more difficult for the algorithm to obtain optimal values in the nodes' routing tables.

Distance Vector Routing gets its name from the fact that each router (or node in a MANET, since each MANET node must function as a router) keeps a one-dimensional array (i.e., a vector) of distances to all of the other routers in the network, along with "next hop" information. This brings in a problem of scalability as the network grows in size, with the routers exchanging progressively larger routing tables. A more flexible alternative is provided by Link State Routing (LSR).

In Link State Routing, each router (or node in the case of a MANET) performs a five- step process:

a) Discover its neighbors and learn their addresses (via HELLO packets).

b) Measure the delay (cost) to each of its neighbors (via ECHO packets).

c) Build a LSP (Link State Packet), which contains all the information it has learned about its neighbors. The LSP contains the address of the node, sequence and age numbers, and a table containing the address of each of its neighbors and the cost to each of them. The sequence and age numbers are included to prevent a node from using an obsolete LSP packet to update its routing information.

d) Flood the LSP packet throughout the network. Naturally, the other nodes participate in this. New and updated LSP packets are sent periodically.

e) Perform Dijkstra's algorithm once the LSP packets from all the other nodes have been received. The receipt of a full set of LSP packets means the node can construct a graph representation of the entire network.

Scalability is improved in LSR compared to DVR despite the use of flooding. The LSP packets a node sends in LSR contain information only about its neighbors (that is, its links' state), whereas in Distance Vector Routing the information

contained in the packets that contain the node's routing table pertains to all the nodes in the network.

2. FSR (Fisheye State Routing) [Iwata99, Murthy04]

FSR, commonly referred to as Fisheye, is a proactive, non location-based, hierarchical-topology routing protocol which is built on top of the GSR (Global State Routing) routing protocol. Each node i maintains four routing data structures, namely: a neighbor list A_i, a topology table TT_i, a next hop table $NEXT_i$, and a distance table D_i. The tables contain one entry per destination node j; in particular, $TT_i(j)$ contains the link state information reported by node j together with a j-assigned sequence number (to distinguish newer information from older information, akin to a time stamp). Nodes initially start out with empty data structures and send link state information (which is initially none) to their neighbors. Thus, a node learns who its neighbors are by looking at the sender field of received packages, and this information is added as appropriate to its routing data structures. Note that a node's link state information is its topology table (which includes the sequence numbers). All this link state information is now sent out to a node's neighbors, which adds to their own link state information. This process is periodically repeated, thus eventually resulting in every node having complete link state information about all the nodes in the network (i.e., a complete topology map). Newly received information is checked against existing information by comparing sequence numbers, to ensure only the most up to date

data is kept. Given a topology map (or a partial one), each node can then calculate shortest paths to the other nodes in the map (that is, computing $NEXT_j$ and D_j) using a shortest path algorithm (typically Dijkstra's). There is no flooding as such; nodes only exchange their own link state information (which eventually covers the entire network) with their neighbors. FSR improves the network efficiency by reducing the size of the update messages exchanged. For a particular node, the nodes of scope one, two, three ... are those nodes that can be reached within one, two, three ... hops, respectively. A node then sends with a high frequency update messages to its neighbors with only link information for nodes within one scope of itself (that is, its neighbors), and with a lower frequency information with the remaining link state information.

3. WRP (Wireless Routing Protocol) [Murthy95, Murthy96]

WRP is a proactive, non-location-based, flat-topology, routing protocol. For routing, each node i maintains a set of four tables: a distance table, a routing table, a link-cost table, and a message retransmission table. The distance table stores for every known destination j and neighbor k (of node i), the distance to j (D^i_{jk}) and the predecessor node (p^i_{jk}) as reported by k. The routing table has as its most important contents, for every known destination j, the destination's identifier, the distance to the destination (D^i_j), and the predecessor and successor of the chosen shortest path to j (p^i_j and s^i_j, respectively). The link-cost table lists for every node k the cost of each corresponding i to k link (l^i_k) and how much time has elapsed since

the last message from k was successfully received (to detect link breaks). The message retransmission list (MRL) contains entries comprised of the sequence number of an update message, a retransmission counter which is initially set to a small integer, an ACK-required flag (specifies whether node k has acknowledged the update message), and the list of updates contained in the update message. Note that the retransmission counter is decremented every time node i sends a new update message; any update decrements the counter, not just a retransmission of the same update. Connectivity is ascertained via HELLO messages and ACKs to successful message reception. The update messages the nodes exchange contain the identifier of the sending node, a sequence number, a list of updates or ACKs to updates (an ACK entry specifies the source node and sequence number of the update message being acknowledged), and a response list of the nodes that should ACK the update message. Note that an update specifies a destination and the distance and predecessor to the destination. A node updates its routing tables after receiving an update message or after a link status change to a neighbor. In response to an update, a node performs two basic steps:

a) It updates the distance and predecessor information in its distance table as reported in the update, also updating other entries in the distance table for other neighbors impacted by the new information (by looking at the predecessor information in the update and comparing it for the predecessor information for the other neighbors to the same destination).

b) It updates its routing table by choosing some neighbor p as its successor to destination j, if that neighbor appears to offer a lower cost path to j.

4. AODV (Ad-hoc, On-demand, Distance Vector routing) [Belding03, Perkins99, Klein08, Lin04, Remondo04, Royer99, Gray04, Jiang04, Boukerche04, Murthy04, Perkins03, Lee03]

AODV is an on-demand, non-location-based, flat-topology routing protocol. It makes use of flooding. A node knows its neighbors (nodes within radio range) via the reception of HELLO packets, which are sent out periodically by nodes to their neighbors. Nodes maintain partial (i.e., not of the entire network) routing tables that contain, among other items, a destination's IP address and last known sequence number, next hop, number of hops, active neighbors (neighbors that recently passed packets to the present node destined for that destination node), and a lifetime value. A route not used within its lifetime is discarded. When a node wants to communicate with a node for which it has no valid routing information, it begins a route discovery process by increasing its own sequence number (used to determine how current a route is) and broadcasting an RREQ (Route Request) packet (which is uniquely identified by the source's address and RREQ counter) to its neighbors. If a node receiving an RREQ (neighbor or not) does not know a route to the destination, it broadcasts the RREQ in turn to its neighbors and sets up entries in its routing table, indicating who it received the RREQ from and its original source (a node does not broadcast an RREQ it has already seen). If a node

knows a route to the destination that is not expired and has a sequence number for the destination equal to or higher than that in the RREQ, or is itself the destination, it replies with an RREP (Route Reply) packet, which is unicast to the node from which it received the RREQ. The nodes receiving the RREP do the same, each sending the RREP to the node from which they received the RREQ. This process is repeated until the source node gets the RREP and the route to the destination is then fully established. The messages sent from the source to the destination follow in reverse the path travelled by the RREP. If a node detects a neighbor is no longer active, it sends RERR (Route Error) packets to active neighbors that have recently passed it packets that were then routed through the now non-active neighbor. Those active neighbors then do the same, and so on. A source node using the now broken route will thus become notified of the break and it can re-initiate route discovery, if needed. The RERR packets are thus used for route maintenance.

5. DSR (Dynamic Source Routing) [Johnson01, Remondo04, Royer99, Murthy04, Johnson07]

In this project, DSR was not tested by itself, but as the underlying protocol for the LAR protocol. DSR is an on-demand, non-location-based, flat-topology routing protocol. Nodes maintain route caches of known routes to destinations. If a sender does not have a route to the destination it seeks, it initiates a route discovery procedure similar to that in AODV via RREQ (Route Request) packet flooding. When a node receives an RREQ and does not have a route in its route cache to the

destination, it adds its address to the ordered list contained in the RREQ of addresses the RREQ has traversed (if the node has already seen this RREQ, it discards it) and rebroadcasts it. If the node knows a route to the destination (this would include the case in which the destination receives the RREQ), it appends the route to the list of nodes in the RREQ, after checking for no repeat nodes (if it finds a repeated node, it modifies the entire route accordingly). It then sends a RREP (Route Reply) packet back to the source. The RREP will follow the same route as the RREQ but in reverse (and, of course, starting at the sending node). Route maintenance is performed by detecting broken links at the link layer level or at the routing protocol level (if ACKs at this level are requested). When a node detects a broken link, it updates its own cache to remove routes involving the failed link, and it sends a RERR (Route Error) packet to the source node via the reverse route. As the RERR traverses the route back to the source from the node that detected the link failure, the nodes that see the RERR update their caches. Notice that nodes only remove information from their caches as they learn existing links have been broken.

6. LAR (Location Aided Routing), scheme 1 [Ko00, Murthy04]

LAR is an on demand, location-based, flat-topology routing protocol. It seeks to improve the efficiency of other MANET routing algorithms that carry out route discovery via flooding (such as AODV and DSR) by restricting the flooding zone to a sub-domain of the entire network. RREQ (Route Request), RREP (Route

Reply) and RERR (Route Error) packets are used in a manner similar to that of AODV or DSR. Notice that MANETs, due to their limited resources, place emphasis in minimizing resource utilization over maximizing route optimality; thus a more efficient MANET routing algorithm in general refers to one with better resource utilization. To restrict the flooding zone, the nodes make use of location (via GPS or some other means of determining position) and average velocity information. If the source node knows the position and average velocity of a destination node at some previous time t_p, then it can determine an "expected zone" where the destination node will be at the current time, t_c. It does this by defining a circle centered on the last known position of the destination node with a radius that is the product of the average velocity of the destination node at t_0 times the difference $t_c - t_0$. Once the "expected zone" is determined, the source node determines the "request zone," which is the zone defined by the smallest rectangle that will contain the "expected zone" and its own position, and has sides parallel to predefined fixed x- and y-axes (2-D case). An RREQ broadcasted by the source to its neighbors will have in it the coordinates of the corners of the "request zone," and only nodes receiving the RREQ that are within the "request zone" will broadcast it in turn to their neighbors. When the destination node receives the RREQ, it replies with a RREP that follows the reverse path to that of the received RREQ (which includes in it the route it followed). When the source node receives the RREP, the route has been established and communication can proceed. Breaks in the route are handled via RERRs, similar to AODV and DSR. Four additional facts must be noted: (1) Circles become spheres and rectangles become boxes in

three-dimensional space. (2) The protocol defaults to a request zone that covers the entire network, if a node cannot determine an "expected zone" for a destination node (due to lack of position or velocity information). (3) LAR 1 assumes each node knows the other nodes' average velocity, but other schemes are also possible. (4) RREQ packets include the destination node's current location and a time stamp, for use in future route discoveries.

Chapter 5

RELATED WORK

The multiplicity of available MANET routing protocols and the availability of MANET simulators has led to numerous performance studies and comparisons. The studies in "Performance Comparison of Three Routing Protocols for Ad Hoc Networks" [Jiang04], "Study of MANET Routing Protocols by GloMoSim Simulator" [Pandey05], and "Scalability Study of the Ad Hoc On-demand Distance Vector Routing Protocol" [Lee03] used the GloMoSim simulator.

In "Performance Comparison of Three Routing Protocols for Ad Hoc Networks" [Jiang04], AODV, DSR, and STAR are compared in terms of data delivery, control overhead, and average latency under various scenarios of mobility, connectivity density, number of data flows, domain shape, and initial node placement. AODV turned out to be the best protocol in terms of data delivery in densely connected scenarios, whereas STAR was found to be the best performer in all the remaining cases.

The work described in "Study of MANET Routing Protocols by GloMoSim Simulator" [Pandey05] compared AODV, WRP, DV, and DSR. The metrics of interest in the experiments were end-to-end delay, packet delivery rate, and messaging overhead, with the control parameters being traffic load, node density, and node mobility. DSR was found to have the lowest messaging overhead but highest end-to-end delay of all. The

proactive protocols, DV and WRP, proved to be less vulnerable to increases in traffic load than the reactive protocols; WRP, in particular, consistently demonstrated the lowest end-to-end delay of all the protocols and had excellent packet delivery rates. All the protocols suffered from low delivery rate when the mobility was perpetual, with DV showing the largest degradation in end-to-end delay with increasing node mobility.

The work carried out in "Scalability Study of the Ad Hoc On-demand Distance Vector Routing Protocol" [Lee03] was not one to compare various protocols, but rather one to test various improvements to an existing protocol, AODV. Expanding ring search and query localization techniques were found to reduce the amount of overhead produced by the protocol, whereas the use of local route repair techniques improved the number of data packets that reached their destinations.

There are numerous MANET simulation studies that have made use of the NS-2 simulator [Boukerche01A], [Boukerche01B], [Boukerche04], [Broch98], [Camp02], [Das00], [Johansson99]. The goal in "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks" [Camp02], which compared LAR, DREAM, and DSR, was to stress the protocols at both low and high node speeds (0 to 20 m/s). The data delivery ratio stayed close to constant for DREAM, whereas for the other protocols, it started out better than DREAM at low speeds but it quickly deteriorated at higher speeds, DREAM bettering all others at the high end of the speed spectrum, while DSR fared the worst. In terms of control packet overhead, DREAM started out with the worst performance at low speeds, but it again bettered all the other

protocols at the high end of the speed spectrum. DSR in non-promiscuous mode performed the worst of all at high speeds, whereas DSR in promiscuous mode performed second best for high speeds, with LAR being just slightly worse than DREAM at high speeds.

In "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks" [Boukerche04], AODV, PAODV (Preventive AODV), CBRP, DSR, and DSDV were investigated in terms of throughput, average end-to-end delay, and overhead, using various scenarios of mobility, load, and size of the network. The findings revealed AODV to have the highest overhead of all, followed by CBRP and DSR; DSR and CBRP both showed very high throughput, whereas AODV showed a very short end-to-end delay. PAODV proved to be only slightly better than AODV.

The work in "A Performance Comparison of Routing Protocols for Ad Hoc Networks" [Boukerche01A] is the same as that in "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks" [Boukerche04], but only covering AODV, CBRP, and DSR in terms of throughput and end-to-end delay. "A Simulation Based Study of On-Demand Routing Protocols for Ad Hoc Wireless Networks" [Boukerche01B] extends "A Performance Comparison of Routing Protocols for Ad Hoc Networks" [Boukerche01A] by also considering overhead.

A new mobility metric, M, is developed in the work described in "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks"

[Johansson99], which is an average of the absolute relative speed between all the node pairs in the network. In this study, DSDV, AODV, and DSR were compared in terms of throughput, delay, and overhead for different scenarios involving mobility (M) and offered load. The reactive protocols turned out to be superior to the table-driven one (DSDV), with both AODV and DSR behaving very similarly in terms of delay and throughput, and with DSR being superior at low traffic loads and AODV at higher loads. DSR was more efficient at low traffic loads, whereas AODV was more efficient at higher packet loads.

The work in "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" [Broch98] compared DSDV, TORA, DSR, and AODV under a variety of mobility and workload scenarios with the goal of measuring the ability of the protocols to react to topology changes while continuing to deliver data to the destinations. The results in this work correlate very well with those in "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks" [Johansson99]. DSR and AODV were the superior performers in all mobility scenarios.

The work in "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks" [Das00] considered only DSR and AODV, which are emerging as possibly the two most promising protocols, under various load, mobility, node density, and domain shape scenarios. The parameters investigated were packet delivery fraction, average end-to-end delay, and normalized routing load. DSR showed a lower routing load than AODV most of the time, but when the MAC (802.11 was used)

overhead was taken into account the overhead generated using both protocols was found to be very similar. AODV was found to provide better performance in terms of packet delivery fraction and delay when the network was more stressed, whereas DSR did better with less network stress. These results agree well with those found in "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks" [Das00] and "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks" [Johansson99].

The work in "A Performance Comparison of On-Demand Routing Protocols for Application Data in Mobile Ad Hoc Networks" [Lee05] is tailored to the ROKA (Republic of Korea Army) and it used the Opnet simulator. It compared the AODV, TORA and DSR protocols in a network of 20 apparently fixed nodes for varying data rates. No protocol was better than TORA in terms of packet delivery fraction, with AODV being second best. AODV showed better delay and routing load than TORA except at the lower end of the packet generation rates investigated. The overall lower performer was DSR. The authors of the study ended up recommending the use of TORA.

The experimental (with actual networking hardware) comparison of APRL, AODV, ODMRP, and STARA in "Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms" [Gray04] compared the protocols under a random, constant speed (walking pace) conditions of mobility and low network load using 33 nodes in a 225 x 365 m^2 athletic field. The investigated parameters were message delivery ratio, communication

efficiency, hop count, and end-to-end delay. The reactive approaches were found to be better in dynamic environments than the table driven ones; ODMRP was concluded to be able to handle higher mobility than AODV due to its higher message delivery ratios measured.

Most of the comparative studies in MANETs make use of CBR traffic, as is the case with the works listed above. However, the comparative performance of protocols may be different if a different kind of traffic is used. Such is the case in "Comparative Study of CBR and TCP Performance of MANET Routing Protocols" [Clausen02], where OLSR, a proactive protocol, was found to provide better results than AODV when the traffic was TCP.

Chapter 6

METRICS AND METHODOLOGY

The work in "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations" [Macker99] provides an extensive discussion of metrics necessary to evaluate the performance and suitability of MANET routing protocols. It states the need to have metrics that are independent of any routing protocol, so comparisons can be drawn, and it groups them in two main categories: qualitative and quantitative. The qualitative metrics discussed are:

1. Distributed operation: This is inherent in MANETs.

2. Loop freedom: Although not required (as long as the packets eventually get where they are supposed to), it increases the efficiency of the routing algorithm by reducing unnecessary hops.

3. Demand-based operations: Network and energy resources in MANETs are much more limited than in conventional networks, therefore it is desirable for the routing algorithm to do its operations only when needed.

4. Proactive operation: It is desirable to reduce the latency induced by on-demand operations when resources permit it.

5. Security: The wireless nature of MANETs puts them especially at risk of attack. For this reason, security must receive a great deal of attention.

6. Sleep period operation: The routing protocol must be able to gracefully handle nodes going to sleep and waking up. Those modes of operation should be an important consideration in MANETs, due to the limited energy resources.

7. Unidirectional link support: It is desirable to be able to handle nodes that have different radio transmission and reception ranges to support heterogeneous nodes and differing conditions for each node.

The quantitative properties discussed are:

1. End to end data throughput and delay: These are measurements of the routing policy's effectiveness and performance from the "external" perspective of other policies and protocols (that make use of the routing).

2. Route acquisition time: How long it takes to obtain a route to the destination. It is especially important for applications that are time-sensitive.

3. Percentage out-of-order delivery: The transport layer protocols prefer in-order delivery; increasing it therefore reduces the amount of processing needed to rearrange the data.

4. Efficiency: Various ratios, such as average number of data bits transmitted to those delivered, average number of control bits to those delivered, and average number of control and data packets transmitted to data packets delivered. These efficiency measures quantify how effective the routing protocol is internally, that is, how many resources it must use to provide a given level of performance to the "external" users of the routing.

Those metrics have to be expressed in the context of other parameters that must be varied. Some of the most important are network size, network connectivity, topological rate of change, link capacity, fraction of unidirectional links, traffic patterns (for instance, bursty versus non-bursty traffic), mobility, frequency of sleeping modes, and physical domain shape.

To perform evaluations of routing protocols, there are two main approaches or methodologies. A researcher can measure and qualify different metrics on an actual network, or he can use simulation. Actual testing is very expensive, because enough hardware must be obtained to conduct the testing; as the size of the network is increased, the expense becomes progressively higher, eventually making it unrealistic to run actual testing. It is more feasible to make use of simulation programs, which allow

a great variation in the testing parameters, at essentially no additional cost in terms of money and resources. The simulation itself can be run in very little time, whereas actual testing would require real physical time. For these reasons, the current research was done using simulation as the tool of choice.

Chapter 7

TESTBED DESCRIPTION

The experiments and data processing for this project were executed on a personal computer with an AMD Athlon™ 2200+, 32-bit processor, and 1 Gigabyte of memory running a Fedora Core 4 Linux operating system. The software testbed consisted of a network simulator, a preprocessor (input file creator), a batch-running facility, and the post-processing components.

7.1 Network Simulator

The MANET simulator chosen for this project was GloMoSim (Global Mobile Information System Simulator) [Zeng98, Bajaj99, UCLA01], which is freely available at http://pcl.cs.ucla.edu/projects/glomosim/obtaining_ glomosim.html, and which has versions available for Windows, Linux, Solaris, and various other flavors of UNIX. The currently available release (as of 2/28/2006), is 2.03. A commercial product derived from GloMoSim, Qualnet, is available at www.qualnet.com. The commercial product is more refined and capable than GloMoSim and it has much better documentation, but it requires a licensing fee.

GloMoSim was designed using the parallel discrete-event simulation afforded by the Parsec (Parallel Simulation Environment for Complex Systems) C-based simulation environment developed at UCLA [Bragodia98]. Parsec implements a process

interaction approach to the simulation of discrete events. In this approach, the representation of objects (or entities) in the physical system under simulation is accomplished via logical processes. These logical processes interact among themselves by means of time-stamped message exchanges; the time-stamps correspond to the actual times when the corresponding physical events take place.

Extensibility is a key attribute in the design of GloMoSim. To achieve it, GloMoSim makes use of a layered approach that follows the networking layered approach and adds the additional layers needed to simulate the transmission and mobility physics. Standard APIs are provided so different models for each different layer can be added in a standardized manner with a minimum level of difficulty by various developers independently and then used interchangeably within each layer. This extends and improves the modeling capabilities of GloMoSim. A user can select from among the various models available at each layer those that best suit his purposes via a standard GloMoSim input file (a sample input file is included in the GloMoSim distribution files). If a model he needs is not available, he can make use of GloMoSim's built-in extensibility and design and write his own model, making use of the standard APIs. The principal physics (mobility and radio transmission) and network layers currently present in GloMoSim are presented in Table 1.

Network Layer/Physics	Available Types
Application	CBR (Constant Bit Rate), HTTP, Generic FTP, FTP,
Transport	UDP, TCP
Network (Routing)	Bellman-Ford, AODV, Fisheye (FSR), DSR, LAR 1, WRP
Datalink (MAC)	CSMA, TSMA, MACA, 802.11
Packet Reception	SNR-bounded, BER-based
Radio	ACC-noise, no-noise
Radio-wave Propagation	Free Space, Two-ray
Mobility	Random Drunken, Random Waypoint, Trace

Table 1: GloMoSim Network and Physics Layers

Scalability is another very important attribute of GloMoSim, since it was conceived from inception to be able to simulate very large networks of up to a million nodes [Bajaj99]. In order to be able to scale to such an extent, GloMoSim implements the concept of network gridding or partitioning [Bajaj99, UCLA01]. Applying the commonsensical approach of using one Parsec entity per network node would result in severe performance penalties as the number of nodes increased more and more. Instead, GloMoSim breaks up the network into a number of geographical partitions and uses one entity to represent all the nodes in that partition. Thus, a node's membership to a particular partition (entity) is based on the node's geographical location, which will vary with time as mobility is introduced. Within each partition entity, a data structure for each member node is used to maintain the state of that node. This way, an increase in the number of nodes does not require an increase in the number of partition entities. The only requirement in the number of partition entities is that it must at least equal the number of processors being used to run the simulation. Each partition entity

incorporates all the GloMoSim layers; communication among them is handled via function calls.

GloMoSim can simulate five applications that make use of the network: CBR (constant bit rate), generic FTP, FTP, TELNET, and HTTP. The first two make use of the UDP transport protocol, whereas the last three use TCP. GloMoSim uses the library tcplib [Danzig91] for TCP traffic generation. This library assumes an exponential distribution of message generation, and is refined by internet traffic traces performed by the authors at three different institutions.

The user of GloMoSim controls the parameters to the program via various input files. The main input configuration file, an example of which is included in Appendix 1, includes:

1. Basic parameters such as simulation time, terrain dimensions, seed (for pseudo-random number generation), number of nodes, and initial node placement.
2. Mobility parameters, such as mobility model, maximum and minimum speeds, and motion pauses.
3. Radio signal propagation parameters, such as propagation model, propagation power limit, and temperature.
4. Radio parameters, such as radio type, transmission frequency, bandwidth, power, antenna gain, sensitivity, packet reception model, and packet power reception threshold.

5. MAC protocol choice.
6. Routing protocol choice, and any parameters associated with the specific protocol.
7. Selection of layers for which statistics are desired.
8. GUI options, such as active or inactive.

The main input configuration file also references additional input files depending on some of the choices made. In regards to this project, the significant additional input files were: (1) a node placement file, which describes the initial position on the terrain of all the nodes in the network, (2) an application file, which describes the applications being used in the network, application parameters and nodes involved, and (3) a Fisheye routing protocol configuration file. Examples of all these additional input files are included in Appendix 2.

GloMoSim's standard output is written to one output file. The user chooses, via the main input configuration file to GloMoSim, the layers for which he desires statistics. The general format of each line of this file is similar for all the lines and is best described with an example:

```
Node:  10, Layer:  802.11, BCAST pkts rcvd clearly: 83
```

Four items of information are thus contained in a typical line: the identity of the node the data is for, the layer the statistic belongs to, the statistic's name, and the value of the statistic for that node. All the values are described in the line by appropriate strings.

Generating cumulative statistics for the simulation for most of the statistics available requires maintaining a summation for each statistic of interest, processing the output file line by line searching for the desired strings that identify the desired statistics and appropriately updating the summations. A few of the statistics available for certain applications are not cumulative in nature, such as nodes' average end-to-end delay. Thus, they require further processing beyond just adding values from the lines of output as-you-go. Yet, some other statistics may require modifications to GloMoSim itself, because there is currently no standard option in the main input configuration file to output them, and the data needed to compute them is simply not in the standard output file. An example of such is jitter.

As a matter of related interest, the author added a few lines of code to the stock GloMoSim for use when the application used is CBR. These lines provide additional output. This additional output is written at the end of the output file, after all the standard output has been written. It reports cumulative properties to quantify end-to-end delay and jitter characteristics for the network being evaluated. These properties were computed inside of GloMoSim. Thus, a few files in the GloMoSim simulator had code added to them for that purpose, and GloMoSim itself had to be recompiled. No existing code was changed to ensure nothing in GloMoSim was broken by the additions. It should be noted that in order to compile GloMoSim in a Linux system, one must have Parsec and gcc properly installed in one's machine.

7.2 Preprocessor

It was impractical to create the input files for GloMoSim by hand, as this project required running a large number of cases. Therefore, an automated way was created in the form of a Java class appropriately named InputCreator. InputCreator's work is controlled by two input text files, one a sample standard GloMoSim input file, for use as a template, and the other a parameter input file. The parameter input file simply consists of lines with strings separated by blank space. The first string in each line is the name of a parameter in the template file; the remaining values in that line are the values that parameter is to take, in the GloMoSim input files that InputCreator will build. InputCreator then will form all the combinations of as many elements as there are parameter names in the parameter input file, resulting from having each parameter line contribute one of its values to each of the combinations. Then the GloMoSim input files will be created, each input file containing one of the combinations created. The parameters that can be varied are NUMBER-OF-NODES, MOBILITY-WP-PAUSE, MOBILITY-WP-MIN-SPEED (which is set equal to MOBILITY-WP-MIN-SPEED), RADIO-TX-POWER, ROUTING-PROTOCOL, APP-CONFIG-FILE, CBR. This was hard-wired for the purposes of this project, but a more general class where different parameters can be varied could be written as a general-purpose GloMoSim input creation tool. Also, node input files specifying the initial location of each node for each case, and application input files specifying the application to be used (CBR or FTP) and nodes involved (clients and servers) are created. The different kinds of input files have

already pre-planned prefixes and identifiers to facilitate the batch running and post-processing needed to obtain the results from all the cases in a useful form.

7.3 Batch Processing and Post-Processing of Results

A java class, Runner, was written to run all the GloMoSim cases unattended in batch mode. It was later extended to be able to also run the gnuplot data and function plotting utility script processing facility in the same way. It collects a few items of information from the user interactively, the main ones being the name of the program to run and the prefix of the input files to use. The input files to use must be in the same directory as Runner. After the job is complete, Runner will "clean-up" after itself, putting all input and output files in directories whose names are provided to the user.

The post-processing of results was automated. Each of the GloMoSim cases run generated an output file (GloMoSim output files were discussed in the Network Simulator Section), each line of which had to be processed to collect statistics and eventually produce result plots. The massive number of lines of output data to process made an automated way to carry out the post-processing mandatory. Four Java classes were written to handle this job: Rename, OutputProcessing, FileData, and GNUPlotGraphScript.

Rename is a self-contained "utility" class whose job is to make it easier to combine run cases into cohesive sets of cases. It allows for the mass renaming and deleting of files that follow certain filename patterns, in particular having a given prefix and a given

character at a certain location in the filename. This is so because once a parameter input file is constructed and used to create a set of GloMoSim input files by use of the InputCreator class, the names of the resulting GloMoSim output files are set for that particular parameter input file. These names are of the form prefixDDDDDDD, where D represents an integer digit from 2 to 9, and prefix is what precedes the digits. The digit 1 was left as a "reserved" digit but was never used.

The first digit location is associated with the first parameter that can be varied in the parameter input file (specific to this project, the first parameter in the parameter input file was the NUMBER-OF-NODES). The second digit location is associated with the second parameter that can be varied in the parameter input file. The remaining digit locations associate following that same pattern. The digit 2 corresponds to the first value of the corresponding parameter in the parameter input file (specific to this project, the NUMBER-OF-NODES could take on the values 36, 49, 64, 81, and 100, so the digit 2 in the first digit location of the output file name corresponded to a case with 36 nodes). The digit 3 corresponds to the second value of the corresponding parameter in the parameter input file. The remaining digits associate following the same pattern. Thus, the names of the output files are tied to a particular parameter input file, and one can have identical output file names representing different cases run using different parameter input files. In order to combine case-sets that were run using different parameter input files into a single case-set, one must (by hand) create a new parameter input file and do a mass file renaming or deleting to make the "D" digits properly correspond to the new parameter input file.

OutputProcessing is the main "engine" for the output processing. It reads all the output files generated by GloMoSim for all the cases run. It creates a FileData object for each GloMoSim output file; thus, a FileData object represents a GloMoSim case run. FileData computes all the desired statistics for that case.

Once all the FileData objects needed are created, OutputProcessing will search through them to create one GNUPlotGraphScript object per graph to be created, by extracting from each FileData object the appropriate statistics for inclusion in each of the GNUPlotGraphScript objects. Each GNUPlotGraphScript object will in turn create a script file for use by the gnuplot data and function plotting utility; each will also write a corresponding text file containing all the graph information in an easy to read tabular form. After all the gnuplot scripts are created, Runner is again run to finally create each graph as a postscript (*.ps) file, which will in turn be converted to encapsulated postscript (*.eps) or jpeg format by appropriate calls (within Runner) to the ps2eps postscript to encapsulated postscript conversion program or to the ImageMagick graphics manipulation library, respectively.

Chapter 8

DESCRIPTION OF EXPERIMENTS

The goal of the experiments performed was to evaluate and compare the performance of five routing protocols implemented in GloMoSim and described in Chapter 4, namely Bellman-Ford, AODV, LAR (augmenting DSR), WRP, and Fisheye, under various scenarios. The selected application to test the networks was FTP. GloMoSim has two varieties of FTP: "Generic FTP" in which there is no acknowledgment of packet receipt on the part of the receiving node, and the standard FTP, in which the TCP layer functions as it normally does (i.e., with acknowledgments from the receiving node). The second type was the one used in this project. Thus, each server and client pair in a simulation undergo a connection establishment phase (three-way handshake), followed by a data transfer phase of segments and acknowledgements to the segments (notice that more than one segment can be sent before an acknowledgement is received, but all the acknowledgements must be timely received to preclude segment retransmissions). There was no connection teardown in the simulations run, however, because the TCP transmissions were set up to run non-stop starting at the beginning of the simulation (thus connection set-up had to occur) without a terminating time (thus no connection teardown was scheduled). The standard FTP in GloMoSim is implemented such that the client sends messages of random size at random times during the simulation to its intended server during the stipulated time in the simulation, as determined by the tcplib library [Danzig91]. The transmitting nodes in each case were free to start their

transmissions from the moment the simulation began, and the transmissions continued throughout the simulation.

The motion of the nodes was simulated using the Random Waypoint Mobility Model. In this mobility model a node moves from a current starting point to a randomly selected destination point within the physical domain at a velocity selected within a given speed range. Once the node arrives at its destination point, it pauses for a set time and then it starts on its way to the next random destination point. In this project, the speed range was set to a single velocity, so each of the nodes simply moved to its next destination at the same fixed speed.

The radio wave transmission model selected was the free-space model, which assumes the sender and receiver have an unobstructed line of sight between them. In this model, the transmission power is attenuated in proportion to the square of the distance between sender and receiver. There is another transmission model in GloMoSim, the two-ray model, which predicts the transmission power to be attenuated in proportion to the distance between sender and receiver raised to the fourth power by taking into consideration ground wave reflection effects. This last model has a hard-coded 1.5m height for the radio antenna. The radio wave transmission model selected is thus more appropriate for airborne node simulations than for ground-level node simulations.

The radio bandwidth was set to 2Mbit per second. This is the default value in the sample GloMoSim main input file distributed with the simulator, and it is the maximum

value supported in the original 802.11 standard in 1997 [Bradley08]. For comparison, the standard (no proprietary enhancements) 802.11b and 802.11g bandwidths are 10.4 Mbit per second and 54 Mbit per second, respectively. The datalink layer protocol selected was the mentioned 1997-vintage 802.11 protocol. In GloMoSim, this protocol employs CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) with virtual channel sensing [Nuevo04].

A simulation run can be reproduced since the (pseudo-) randomization is based on a seed given to GloMoSim via its main input file.

The size of the terrain was set at 2,000 m x 2,000 m (6,561.7 ft x 6,561.7 ft, which is approximately 1.54 square miles), and the simulations were set to model 10 minutes of actual time network traffic. For each routing protocol, the following parameters were varied:

1. Number of Nodes: 36, 49, 64, 81, 100. All of these values are the square of integers. The reason for this is the nodes were evenly distributed on the square-shaped terrain at the beginning of each simulation run, and square numbers facilitate this distribution easily and cleanly. The lower limit for the number of nodes was 36 because six is the lowest squared number judged to give a "reasonable" node density in the given terrain. The highest limit of 100 was set due to computational concerns with the physical time it actually took to compute

all the cases, and because it was judged to provide a reasonable upper limit for nodal density in the given terrain.

2. Node Mobility Speed: Two speeds were investigated, a low speed of 2.682 m/s (6 mph), and a "high" speed of 26.822 m/s (60 mph).

3. Node Mobility Pause: It was decided to simulate continuous ("perpetual") motion, so the pause time selected was 0 seconds.

4. Radio range: The values used were 375.0 m (1230.3 ft), 300.0 m (984.2 ft), and 250.0 m (820.2). These were obtained by keeping the default GloMoSim radio and wave propagation input values for the highest radio range, and varying just the radio transmission power to obtain the lower range values. The values for the radio transmission power respectively equivalent to the given radio ranges were 10.527 dBm (decibel milliwatts), 8.589 dBm, and 7.005 dBm. The radio range was calculated using the radio_range routine included with the GloMoSim simulator. The power units used for radio transmission are related to the more familiar milliwatts by:

$$Power_{mW} = 10^{\frac{Power_{dBm}}{10}}$$

At the higher radio range, it would theoretically take approximately 5.3 "maximum radio range" hops to move a packet from one side of the terrain to the opposite end. At the lower radio range, the corresponding number of hops is about eight.

5. Percent of communicating pairs: 10, 25. This represents the percentage of the number of nodes that were servers (receivers). The number of nodes that were servers was set equal to the number of nodes that were clients (senders). Thus, for the value of 25, 25% of the nodes send messages to 25% of the nodes. A node's role as a server or client stayed constant throughout the simulation, and no node acted as both. Pairings of servers to clients were not changed throughout the simulation. If the percentage of the total nodes was not an integer, it was rounded up to the next integer. At the lower percentage, only 20% of the nodes in the network were data-flow endpoints, whereas at the higher percentage half of the nodes were data-flow endpoints for each data flow.

All the combinations obtainable, using the parameters varied as described, were run for each protocol. Thus, the number of cases run were five (protocols) x five (node counts) x two (speeds) x three (radio ranges) x two (percentages of communicating pairs) = three-hundred cases. Each case was run ten times, and the average of the results taken. Each time a case was run, it was run using a different set of seeds for pseudo-random number generation in both the InputCreator class used for GloMoSim input file generation and for GloMoSim itself. In the case of InputCreator, a given seed determines how the nodes are "paired-up" in terms of clients and servers. In regards to GloMoSim, the pseudo-randomness will control items such as the selection of a new destination point and the time and size of a new FTP transmission. The initial distribution of the nodes within each of the numbers of nodes tested was always the same, in a homogeneous grid of equally spaced nodes of dimensions (number of

nodes)$^{1/2}$ x (number of nodes)$^{1/2}$, where each node always occupied the same initial position.

Sample reference potential vehicles that could act as nodes in a MANET, matching quite well the parameters selected in this study (in particular, examine the speeds, mobility model, and radio wave propagation models selected), are helicopter-type UAVs (unmanned air vehicles). One example of such a vehicle is the French Infotron IT 180-5 coaxial-rotor helicopter drone [Infotron08], with an empty weight of 10 kg-force (22 lb), a full load weight of 15 kg-force (33 lb), a maximum speed of 90 km/h (= 25 mps = 60 mph), a ceiling of 3,000 m (9,842 ft), and an endurance of 90 minutes. Another sample vehicle is the American (vertical) ducted-fan Honeywell Micro Air Vehicle [Defense Review.com08], with a wet (with gas) weight of 5.7 kg-force (12.5 lb), a maximum speed of 92.6 km/h (25.7 mps = 57.5 mph), a ceiling of 3,200 m (10,500 ft), and an endurance of 40 minutes at 5,500 ft (1,676 m). This last vehicle in particular is inaudible at 100 m, and it has an interchangeable modular sensor package that can detect a man-sized object at 250 m during the day (electro-optical sensor option), or 125 m at night (infra-red sensor option).

Chapter 9

RESULTS

In order to quantify network performance, the following metrics were collected and computed for each case run:

1. Application Bytes Received.
2. Application Bytes Received (Normalized).
3. Application Byte Delivery Ratio.
4. Control Packets Transmitted.

Note that the term "statistic" is used in GloMoSim in lieu of "metric." The first metric provides an absolute measure of the network's performance for each case investigated. The remaining metrics provide a measure of the efficiency with which the performance was achieved, which itself is an indication of scalability. The data is presented as a series of graphs and corresponding tables. Each graph represents the value of the chosen metric (y-axis) versus the number of nodes (x-axis) at a certain combination of values for percent communicating pairs and radio transmission power (i.e., radio range), for every protocol. Each graph contains ten data curves, one data curve for every protocol (five protocols) at every speed (two speeds) and with continuous motion. Additionally, each graph lists the parameters that differentiate it from the other graphs of the same type under the main title.

Control Packets Transmitted was collected to obtain a measure of overhead for each protocol. The best way to collect this metric given the information obtained from the GloMoSim output was by setting the following: (1) Control Packets Transmitted to the sum of "(Routing) Control Packets Transmitted" for AODV. (2) Control Packets Transmitted to the sum of "Routing Control Packets Sent" for WRP. (3) Control Packets Transmitted to the sum of "Routing Table Broadcasts Transmitted" for B-F. (4) Control Packets Transmitted to the sum of "Intra-Scope Updates" for Fisheye. (5) Control Packets Transmitted to the sum of the various Routing Requests, Routing Replies, and Routing Error Packets metrics for LAR. This last measure best gives an idea of how a protocol's overhead changes as the parameters change, particularly as a scalability measure, which in turn gives an idea of the "cost" incurred by the protocols to move the application's bytes (FTP). It must be noted that these measures are not consistent among different protocols, and the way each protocol works must be kept in mind. The values for AODV, LAR, and WRP are directly comparable, since they are (routing) packets actually sent. The values for B-F and Fisheye are comparable with each other, but not with the other three, since they consist of Broadcasts Transmitted and Intra-Scope Updates (which are meant for a node's neighbors). They display, as expected, proportionality to the number of nodes. A way to make these values approximately comparable to those of AODV, LAR, and WRP would be to multiply them by the average number of neighbors a node has for each case, or by collecting Routing Broadcasts (or Scope Updates) Received, information which is missing from the GloMoSim output.

Normalization of Application Bytes Received was done by dividing the value of that metric as collected from the GloMoSim output by the number of communicating pairs (i.e., data flows). A "network building block" is defined as a relatively small collection of nodes where the ratio of clients and servers to the number of nodes in the building block is constant. Then, for tests that only differ in the number of nodes, increasing the number of nodes is akin to adding additional "network building blocks" together, to make a larger network within the same physical space. Increasing the size of the network in this manner, we preserve the basic makeup of the network, in contrast to simply increasing the number of nodes while keeping the total number of clients and servers constant. Thus, it is sometimes easier to look at scalability issues in this manner. Normalization of the metric in this manner allows it to convey how it is behaving as the network is scaled up.

Some of the metrics collected were not normalized by the number of communicating pairs as just described. Some of these metrics were ratios themselves – such as the Application Byte Delivery Ratio – and were thus implicitly normalized, just not by the number of communicating pairs. Some others simply looked at absolute performance measures, such as bytes received by the servers during the simulation time. Other types of normalization could also be done, such as using the number of nodes as the dividing quantity.

The description of the results in the graphs is presented first for the cases with 10% communicating pairs, in order of increasing transmit power. The results for Application

Bytes Received, Application Bytes Received (Normalized), Application Byte Delivery Ratio, and Routing Control Packets Transmitted in each instance are presented in that order. Then the same is done for the cases with 25 % communicating pairs. The graphs are designated with a three-character code consisting of a digit followed by a lower case letter in turn followed by an upper case letter. The digit indicates the percent of communicating pairs and it can be "1" for 10%, or "2" for 25%. The lower case letter indicates the transmit power and it can be "a" for 7.005 dBm, "b" for 8.589 dBm, or "c" for 10.527 dBm. The upper case letter can be "P" for Application Bytes Received, "D" for Application Byte Delivery Ratio or "R" for Routing Control Packets Transmitted. When necessary, the three-character code is followed by a dash and either the letter "a" (the metric is absolute), or the letter "n" (the metric is normalized). It must be noticed that at 100 nodes (the highest node value investigated), GloMoSim consistently crashed for the WRP protocol. Consequently, no results are presented for WRP at the 100-node mark.

9.1 10% Communicating Pairs, 7.005 dBm Transmit Power

Application Bytes Received is shown in Figure 1 (graph 1aP-a). At the low speed, almost none of the protocols managed to move any bytes to the servers at the lowest nodal density. The one exception was LAR, which moved 82,551 bytes. The trend continued at the next nodal number of 49 for that speed, but interestingly LAR's performance worsened. This is explained by the tenuousness of the connectivity at this low density for the low speed. The performance of the LAR protocol at these low nodal densities was better than all the other protocols, but could not be considered reliable.

There was overall improvement at 64 nodes; the best performer was LAR with 198,778 bytes moved, closely followed by B-F, AODV, WRP, and Fisheye all with similar performance at around 140,000 bytes. Past this point, both LAR and AODV improved dramatically to final values (at 100 nodes) of 1,704,449 bytes and 1,352,277 bytes, respectively. B-F showed no improvement initially, but then it increased at a moderate rate to its final value of 672,813 bytes. Fisheye made smaller gains, ending at a final value of 489,578 bytes. WRP had a mild gain from 64 to 81 nodes, increasing to 223,538 bytes.

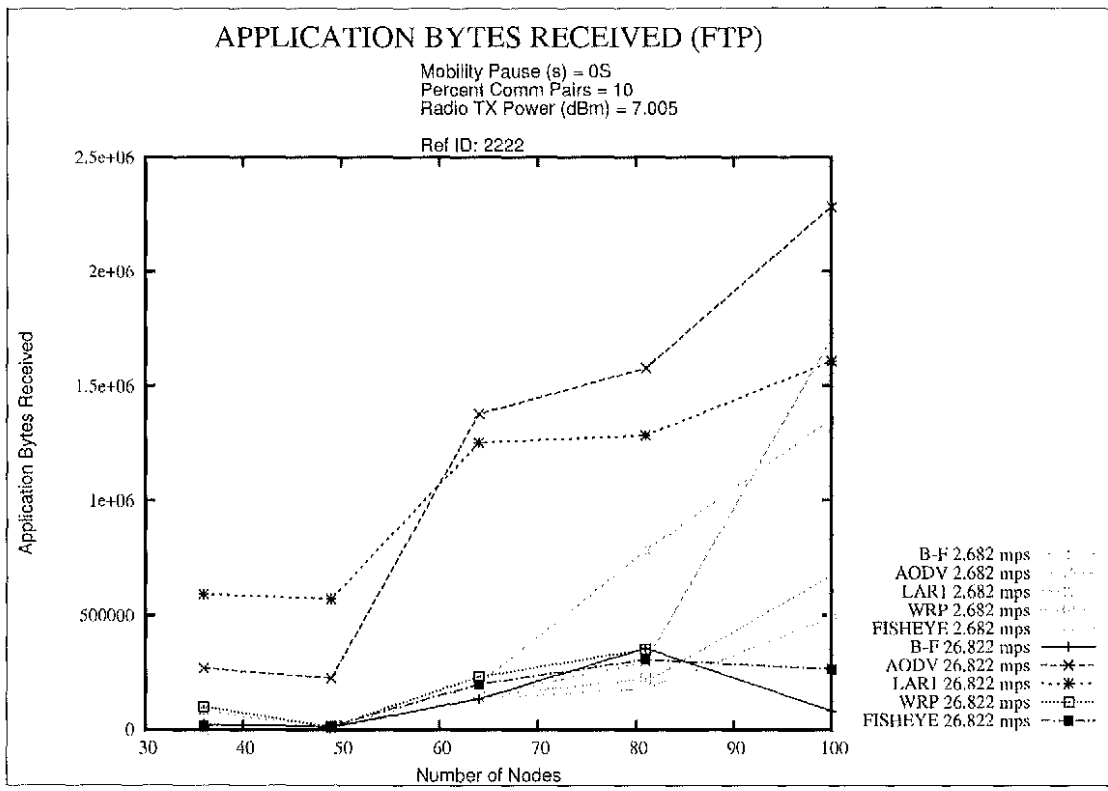

Figure 1: Application Bytes Received, 10% Communicating Pairs, 7.005 dBm Transmit Power (Graph 1aP-a).

At the high speed in Figure 1 (graph 1aP-a) all of the protocols showed large gains in performance at the three lowest node densities. The largest gains, at 64 nodes, were made by AODV to 1,377,817 bytes and LAR to 1,251,630 bytes. The other protocols in the meantime improved their performance to the range 133,260 to 230,471 bytes at 64 nodes. AODV continued to gain in performance throughout to the highest node density, with a very sharp gain from 81 to 100 nodes, ending up delivering 2,282,226 bytes at 100 nodes, whereas LAR leveled off between 64 and 81 nodes and then it showed a large gain to 1,606,955 bytes. The other protocols improved little from 64 to 81 nodes; at the higher node densities, the performance of both Fisheye and B-F deteriorated with respect to their respective low-speed performances, ending up with 263,788 byes and 80,835 bytes, respectively.

Figure 2 (graph 1aP-n) displays the results of Figure 1 (graph 1aP-a) in normalized form. At the low speed, all the protocols displayed a clear efficiency improvement at the 64-node mark. Past that point, the efficiency either leveled off and then improved or improved all the way to the 100-node mark. The highest efficiency was thus obtained at the high end of the node scale. At the high speed, a similar trend was observed all the way to 64 nodes. In this case, however, both AODV and LAR peaked at 64 nodes and then showed a local trough at 81 nodes, whereas all the remaining protocols maintained a close to constant efficiency between 64 and 81 nodes and then dropped significantly to 100 nodes, except for Fisheye, which dropped moderately to 100 nodes.

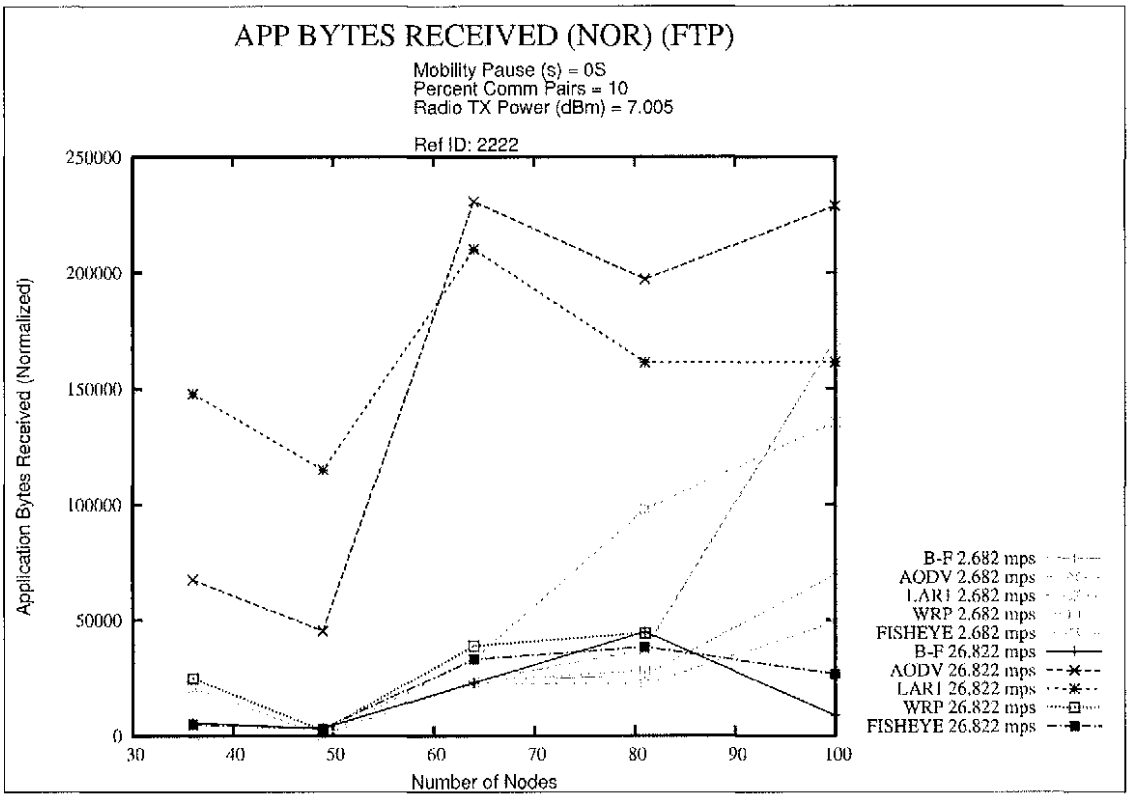

Figure 2: Application Bytes Received (Normalized), 10% Communicating Pairs, 7.005 dBm Transmit Power (Graph 1aP-n).

Application Byte Delivery Ratios (ABDRs) are shown in Figure 3 (graph 1aD). One immediately notices high speed resulted in higher byte delivery ratios. In particular, AODV and LAR achieved above 0.95 ABDR throughout the nodal range at the high speed. At the low speed, they did not achieve a similar value until they reached 81 nodes. At the low speed, all the nodes increased their ABDR from less than 0.1 at 36 nodes to higher than 0.95 at 81 nodes, in a quite steady manner. AODV and LAR were the best, closely followed by WRP, and Fisheye lagging up to 64 nodes, where it was at an ABDR of 0.3. B-F performed better than Fisheye up until 81 nodes, where it lagged all the protocols with an ABDR of 0.77. At the high speed, all the protocols other than AODV and LAR stayed closely grouped together within a 0.15 band, with the lowest

value obtained at 49 nodes by Fisheye at an ABDR of 0.4. At 81 nodes, all the protocols achieved higher than 0.95 ABDR, except Fisheye, which was at 0.9.

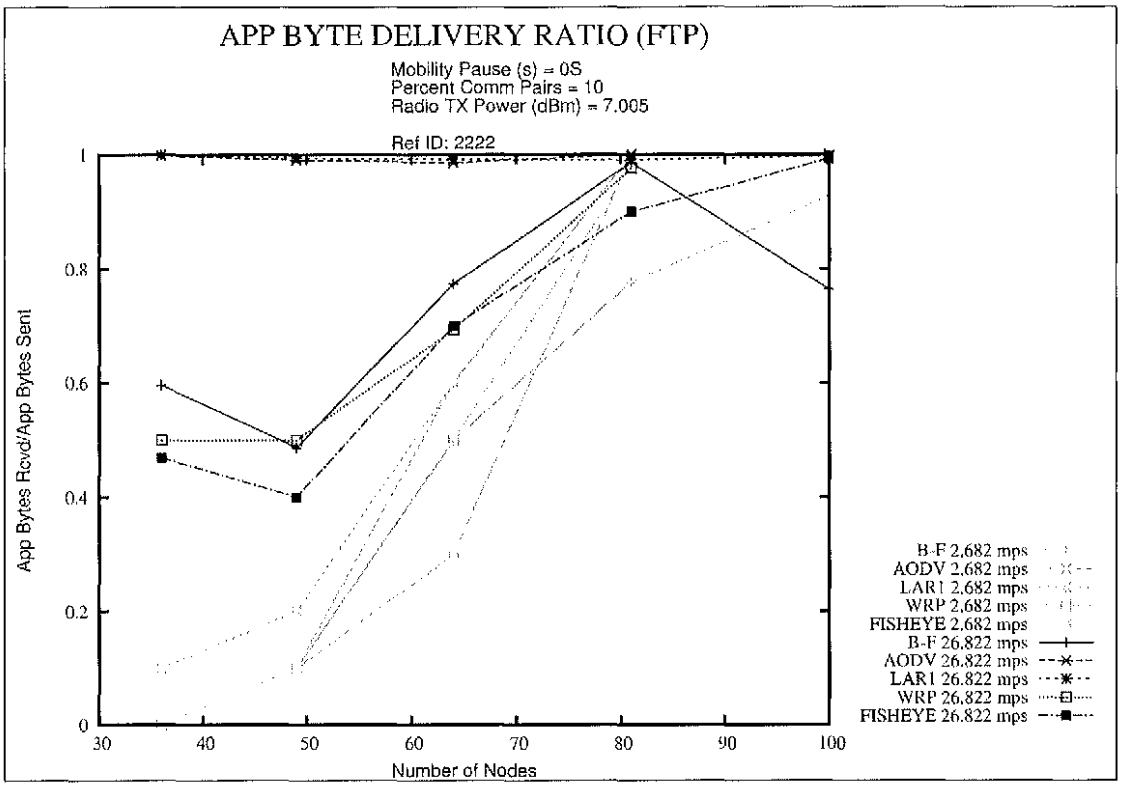

Figure 3: Application Byte Delivery Ratio, 10% Communicating Pairs, 7.005 dBm Transmit Power (Graph 1aD).

The Routing Control Packets Transmitted is contained in Figure 4 (graph 1aR). Recall that AODV, LAR, and WRP can be compared together as a group, and B-F and Fisheye can be compared together as a separate group. These last two protocols behaved linearly, increasing the number of broadcasts (B-F) and scope-updates (Fisheye) as the number of nodes increased, independent of speed, percent communicating pairs, and radio transmit power. B-F did 60 broadcasts per node and Fisheye did 161 scope updates per node for the test time. Since these results do not change in the remaining

graphs, they will not be mentioned again for each graph to avoid repetitiousness. WRP produced far higher numbers of routing control packets than AODV and LAR at both speeds, with a higher slope up to 64 nodes, where Fisheye at the low speed (only) increased faster, but still with far fewer packets. The WRP curves look almost linear, but the slopes are slightly increasing with the number of nodes (i.e., as the topography grows more complicated). The WRP number of routing control packets ranged from 18,948 (36 nodes) to 53,664 (81 nodes) at the low speed, and from 28,198 (36 nodes) to 79,846 (81 nodes) at the high speed. This works out to be 526 to 663 packets per node for the low speed and 782 to 985 packets per node at the high speed. LAR followed WRP in the Routing Control Packets Transmitted metric, smoothly increasing for the high speed from 6,192 (36 nodes) to 31,818 (100 nodes) packets, with the slope of the curve increasing slightly with increasing number of nodes. At the low speed, interestingly, LAR used just slightly more packets than at the high speed up to 64 nodes, where it had 15,794 (versus 13,528 at the high speed), and then its number of packets increased very fast to 69,240 at 100 nodes. AODV was the more economical protocol; at the low speed, it used under 200 packets up to the 64 nodes and then it increased relatively quickly to 9,393 packets at 100 nodes. At the high speed, AODV started out at 880 packets at 36 nodes and it increased smoothly to 4,431 at 64 nodes; then its routing packet usage increased much faster reaching 15,459 at 100 nodes.

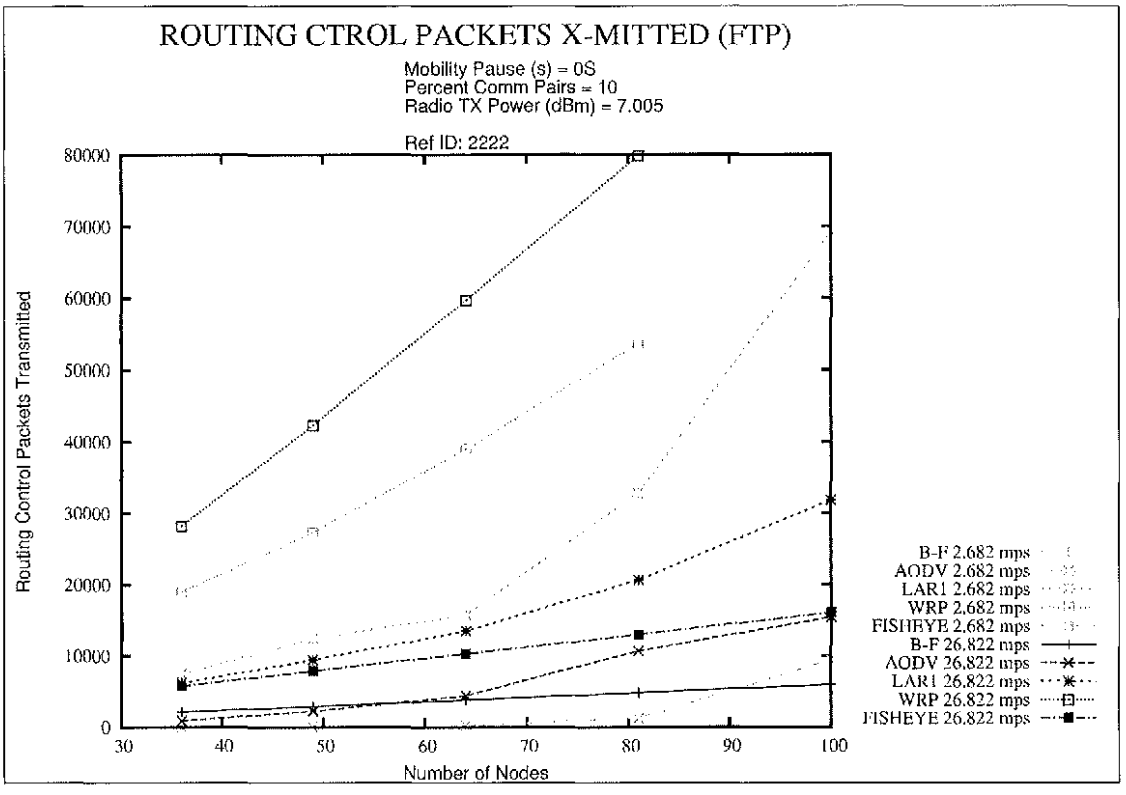

Figure 4: Routing Control Packets Transmitted, 10% Communicating Pairs, 7.005 dBm Transmit Power (Graph 1aR).

9.2 10% Communicating Pairs, 8.589 dBm Transmit Power

Application Bytes Received is shown in Figure 5 (graph 1bP-a). At the low speed, as in the previous Application Bytes Received graph and with similar performance, only LAR moved data from clients to servers at low node densities. At 49 nodes, all the protocols showed some gains. LAR improved the most, to 179,820 bytes, followed by AODV, B-F, and WRP to about 19,000 bytes; Fisheye hardly improved. All the protocols greatly improved to the next nodal density of 64 nodes, the largest gains by far being those of AODV and LAR; they were very similar in absolute terms, and approximately reached the level of 1,400,000 bytes. B-F also showed a large gain at 64

nodes, attaining almost 1,200,000 bytes. The other protocols stayed in the 440,000-640,000 byte range. LAR and B-F deteriorated to the next node density of 81 nodes, to 1,308,207 and 757,017 bytes, respectively, and then improved at 100 nodes to 1,617,377 and 1,081,649 bytes, respectively, which was similar to their performance at 64 nodes. AODV and Fisheye showed continued improvement past the 64 node mark, AODV quite sharply and Fisheye quite mildly, to end up at 2,276,474 and 790,123 bytes at 100 nodes, respectively.

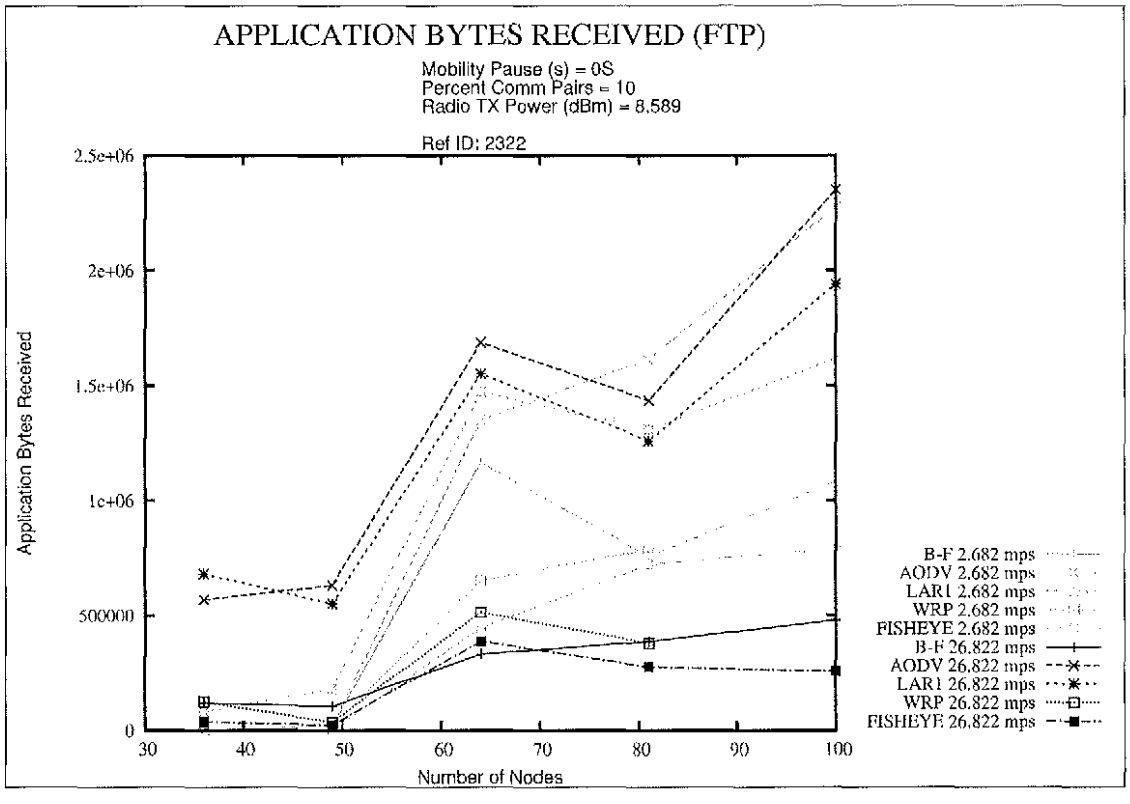

Figure 5: Application Bytes Received, 10% Communicating Pairs, 8.589 dBm Transmit Power (Graph 1bP-a).

At the high speed in Figure 5 (graph 1bP-a), the higher performers were LAR and AODV, as in the low speed but now by an even larger margin. Both displayed similar

performance throughout all node levels. Furthermore, this performance stayed in each case within a narrow range, which opened up slowly as the node density increased. Both started out at around 600,000 bytes, stayed at a similar level to 49 nodes, and then sharply increased to around 1,600,000 bytes at 64 nodes. This very large increase in performance was followed by a moderate drop in performance at the 81-node mark to 1,258,802 bytes (LAR) and 1,434,969 (AODV) bytes, and then another large increase in performance to 1,940,365 bytes and 2,349,985 bytes at 100 nodes, respectively.

All the other protocols at the high speed attained a much lower level of performance at all node levels. B-F and WRP started out at 36 nodes at an almost identical performance level in the range 118,000 to 123,000 bytes, whereas Fisheye started out at a low performance level of 37,622 bytes. Performance worsened at 49 nodes, to slightly worse than at the previous node level for B-F and Fisheye, and to a level similar to B-F for WRP. All three protocols peaked in performance at 64 nodes as they reached 331,821 (B-F), 387,946 (Fisheye), and 513,242 (WRP) bytes. B-F continued with an increase in performance to 478,482 bytes (at 100 nodes). This increase was linear and almost imperceptible. Fisheye worsened to 81 nodes and then to 100 nodes, finishing at 257,000 bytes, whereas WRP displayed the same worsening trend to 81 nodes.

Figure 6 (graph 1bP-n) displays the results of Figure 5 (graph 1bP-a) in normalized form. It is immediately obvious that the shape of the different curves is almost the same as in the non-normalized graph. At both speeds, the largest increase in efficiency occurred at 64 nodes, which was also the highest peak for all the protocols at the high

speed and for all the protocols except Fisheye at the low speed. At the high speed, all the protocols dropped in efficiency advancing from 64 to 81 nodes, which became a local trough for AODV and LAR; the remaining protocols maintained a fairly constant efficiency between 81 and 100 nodes. At the low speed, AODV and LAR dropped in efficiency at 81 nodes as at the high speed, but LAR stayed at an approximately constant efficiency level at 100 nodes whereas AODV recovered to its prior level. B-F displayed a behavior similar to AODV's but more extreme, with a very large loss of efficiency at 81 nodes and a slight recovery (but not to its prior level) at 100 nodes. Fisheye actually had its highest efficiency at 81 nodes after a small increase from 64 nodes, and it returned to the 64-node efficiency level at 100 nodes. WRP showed a modest decrease from 64 to 81 nodes.

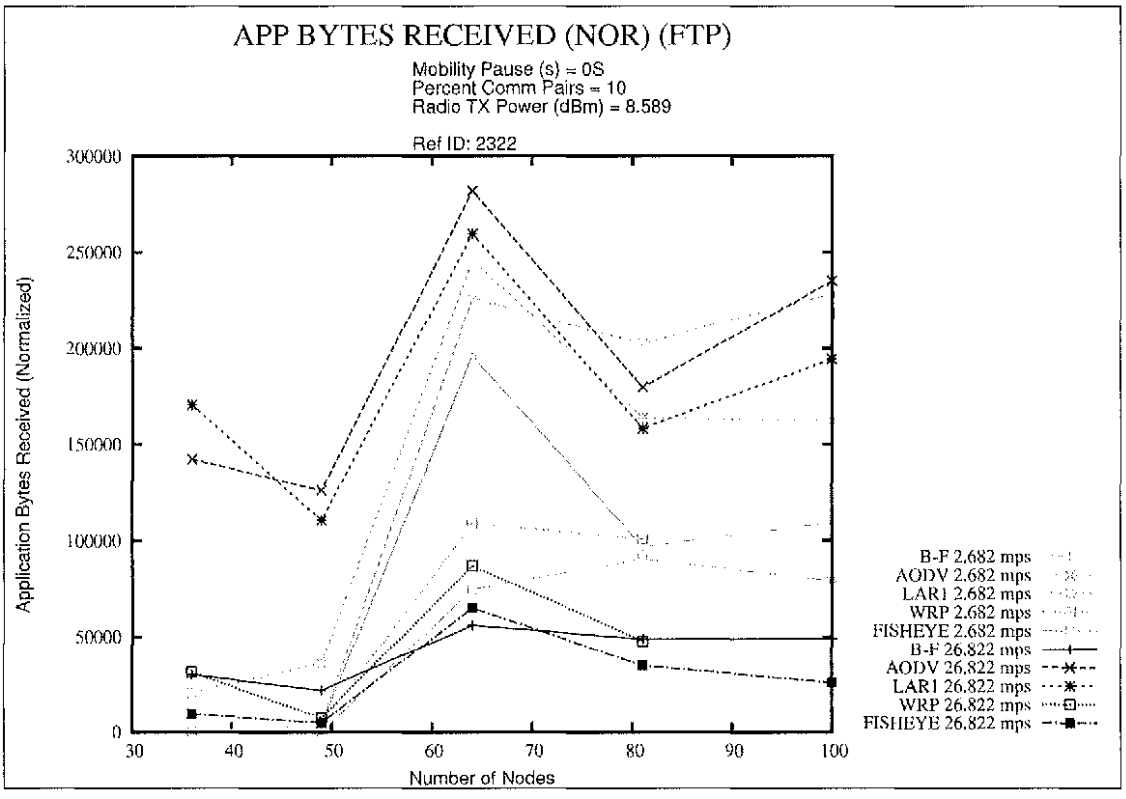

Figure 6: Application Bytes Received (Normalized), 10% Communicating Pairs, 8.589 dBm Transmit Power (Graph 1bP-n).

Byte delivery ratios are shown in Figure 7 (graph 1bD). At the high speed, AODV and LAR displayed an ABDR of 0.99 or better throughout the entire node range. Fisheye was the laggard with an ABDR of 0.7 at the lowest end of the node range, worsening at 49 nodes to around 0.6 and then showing an almost constant rate of improvement from then on to above better than 0.98 at 100 nodes. B-F remained constant at an ABDR of around 0.8 up until 64 nodes, and then it improved smoothly to better than 0.96 at 100 nodes. WRP showed an almost linear improvement from approximately 0.8 at 36 nodes to above better than 0.97 at 64 nodes and above. At the low speed, all the protocols started out at near 0 at the lower node value, except for LAR, which started out at around 0.1, and then rapidly increased to above 0.8 at 49 nodes and on to better than

0.99 at 64 nodes and above. AODV and WRP performed almost identically to 0.3 at 49 nodes and then above 0.99 for higher node counts. B-F matched them to 49 nodes but then trailed somewhat, achieving 0.90 at 64 nodes and then increasing at an almost constant rate to above 0.98 at 100 nodes. Fisheye was the laggard up to 49 nodes, with an ABDR of 0.1, but then matched the results of WRP and AODV.

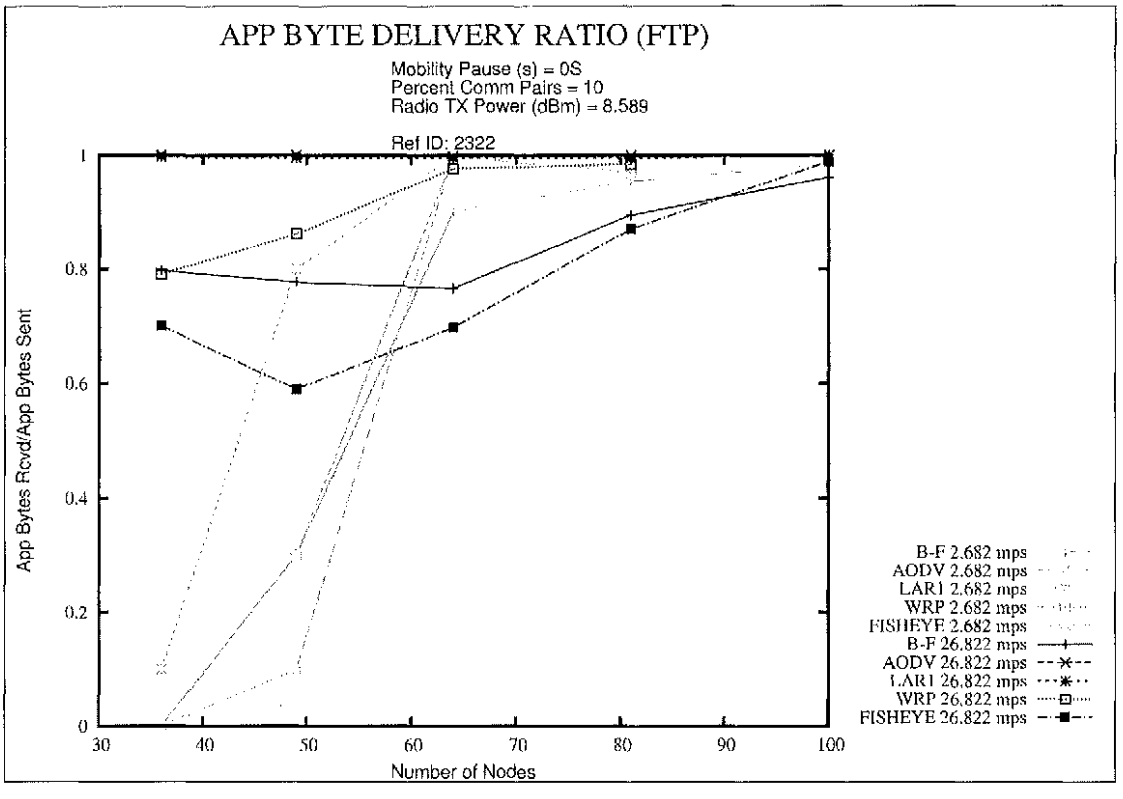

Figure 7: Application Byte Delivery Ratio, 10% Communicating Pairs, 8.589 dBm Transmit Power (Graph 1bD).

Figure 8 (graph 1bR) contains the Routing Control Packets Transmitted. WRP produced more routing packets than either AODV or LAR at both speeds. It increased close to linearly from 18,946 (36 nodes) to 53,726 (81 nodes) packets at the low speed, and from 28,166 (36 nodes) to 79,781 (100 nodes) packets at the high speed. LAR, at

the high speed, increased smoothly from 4,270 routing packets to 27,311, with the curve displaying a slightly increasing slope with increasing number of nodes. At the low speed, it started out at 5,230 packets at 36 nodes, and it increased to 17,517 packets at 64 nodes, decreasing slightly at 81 nodes and then having its greatest increase to 34,675 packets at 100 nodes. AODV was again the most economical protocol, very smoothly increasing from 2,175 packets at 36 nodes to 13,166 packets at 100 nodes at the high speed; at the low speed, it used less than 200 packets up to 49 nodes, and then it increased more rapidly to 11,310 routing packets at 100 nodes.

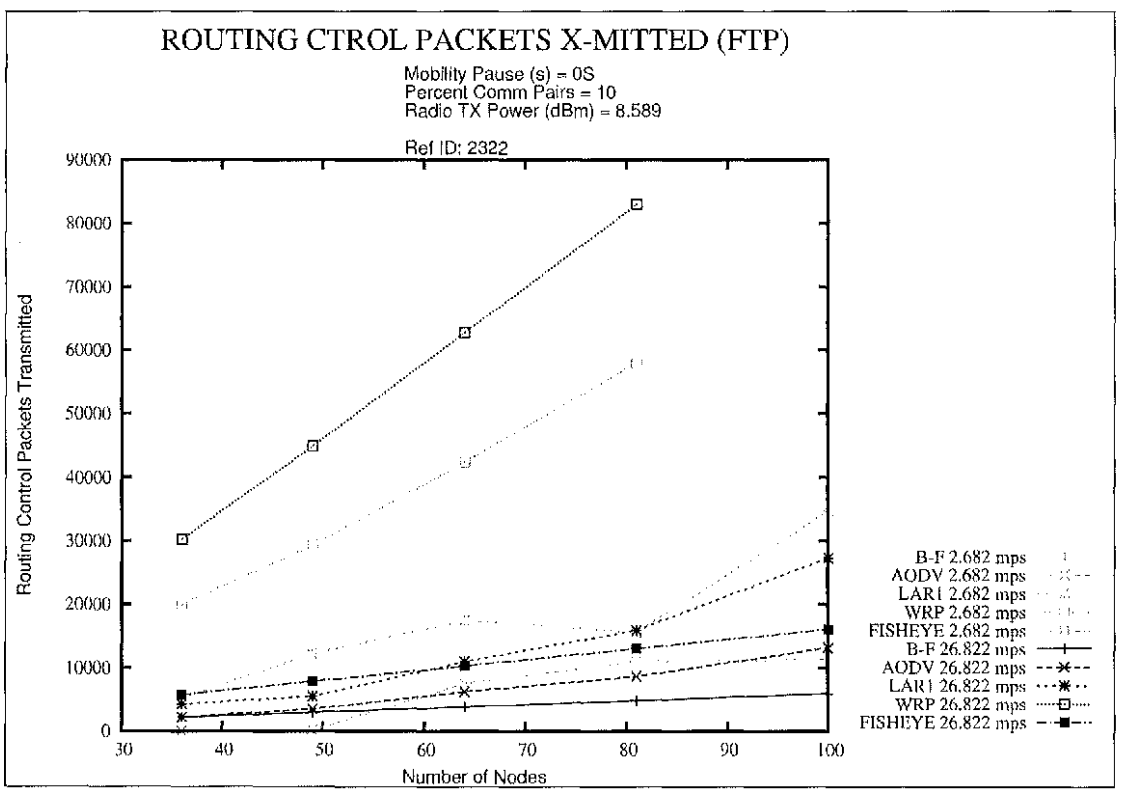

Figure 8: Routing Control Packets Transmitted, 10% Communicating Pairs, 8.589 dBm Transmit Power (Graph 1bR).

9.3 10% Communicating Pairs, 10.527 dBm Transmit Power

Application Bytes Received is shown in Figure 9 (graph 1cP-a). At the low speed, all the protocols moved a significant number of bytes even at the lowest density, unlike in the previous graphs. At 36 nodes, the bytes transferred ranged from 160,988 bytes (Fisheye) to 424,175 bytes (LAR), with AODV and WRP achieving a similar performance at approximately 337,000 bytes and B-F coming in at 264,781 bytes. From that point on, AODV became the best performer and, interestingly, its performance became indifferent to speed, being almost identical at both speeds and following the trends it displayed at the previous power level at the high speed. It had the greatest increase in performance from 49 nodes to 64 nodes, where it transferred 1,916,216 bytes. A performance trough followed at 81 nodes, with a drop to 1,666,929 bytes, and finally achieving 2,570,514 bytes at 100 nodes. LAR's performance was similar to AODV's past the lowest nodal density, displaying the same trends and lagging it slightly. B-F followed the same trends as LAR throughout the node density range, with performance peaks at 64 nodes (1,235,207 bytes) and 100 nodes (1,376,552) bytes. WRP's performance in turn closely tracked that of B-F, being just slightly better, with its greater advantage over B-F being achieved at 64 nodes with 1,361,859 bytes. Fisheye lagged the other protocols peaking at 64 nodes with 944,760 bytes and then linearly and very slightly losing performance to 100 nodes, where it ended up at 824,624 bytes.

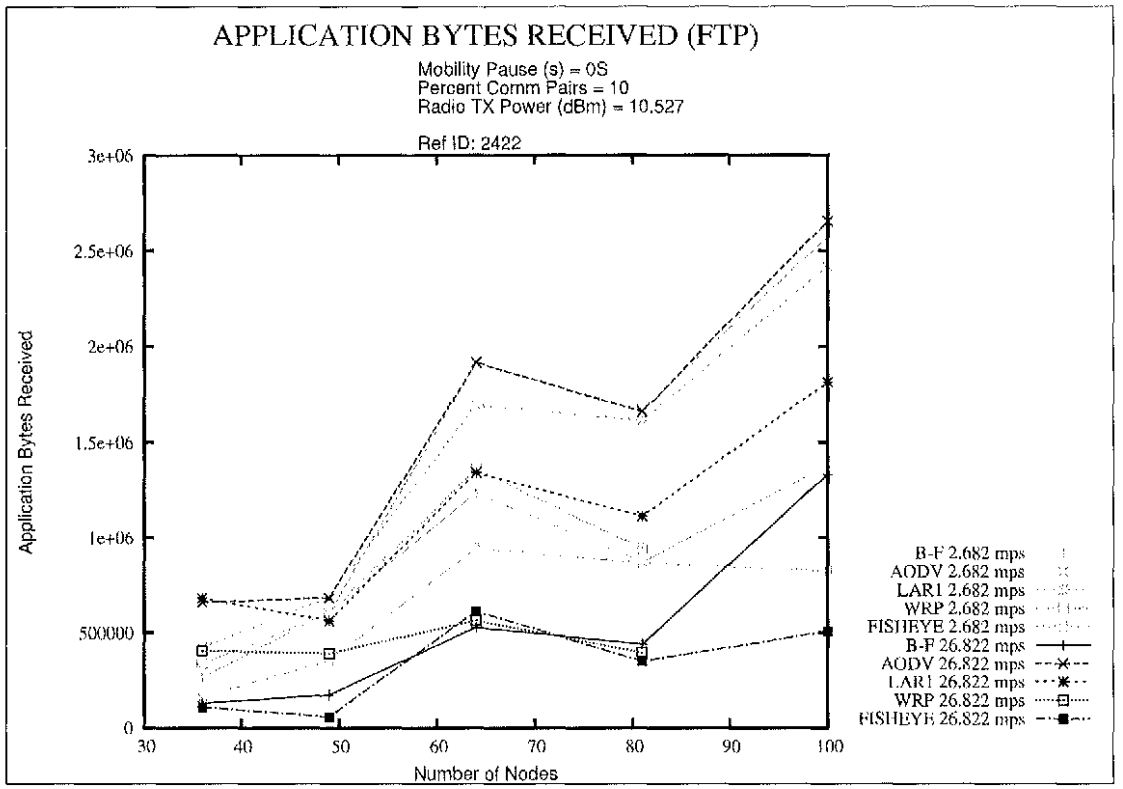

Figure 9: Application Bytes Received, 10% Communicating Pairs, 10.527 dBm Transmit Power (Graph 1cP-a).

At the high speed in Figure 9 (graph 1cP-a), the best performing protocol was AODV, followed by LAR. AODV, LAR, and WRP achieved better performance, at the lowest node density, at the high speed than at the low speed, with 661,596, 682,322, and 406,126 bytes, respectively. Fisheye and B-F suffered from the opposite effect at 36 nodes, with 107,760 and 128,051 bytes, respectively. As already mentioned, from 49 nodes on AODV showed almost complete imperviousness to speed with performance almost identical at all node levels to its performance at the low speed, peaking at 100 nodes with 2,654,324 bytes, which is just a bit higher than at the low speed. LAR's performance, on the other hand suffered with the increased speed, with peaks at 64 (1,340,895 bytes) and 100 nodes (1,813,496 bytes), and a trough at 81 nodes, with

1,112,055 bytes. WRP's performance remained within a narrow range for all its nodal values, peaking at 565,393 bytes at 64 nodes. The performances of B-F and of Fisheye did not see a real improvement until 64 nodes, where they peaked at values very similar to that of WRP, which they followed very closely also to the 81-node mark. Then B-F improved greatly when reaching 100 nodes, where it attained 1,330,291 bytes, whereas Fisheye saw a modest improvement to 506,301 bytes.

Figure 10 (graph 1cP-n) displays the results of Figure 9 (graph 1cP-a) in normalized form. It is very similar to the previous normalized graph, Figure 6 (graph 1bP-n), and thus it is easiest to describe by comparison. As in the previous normalized graph, the 64 node mark was a clear peak and 81 nodes a clear trough for all the protocols at both speeds. This time all the protocols clearly peaked at 64 nodes except for B-F at the high speed, which peaked at 100 nodes. Additionally, at the low speed, all the protocols showed a very definite increase in efficiency as the number of nodes increased from 36 to 49.

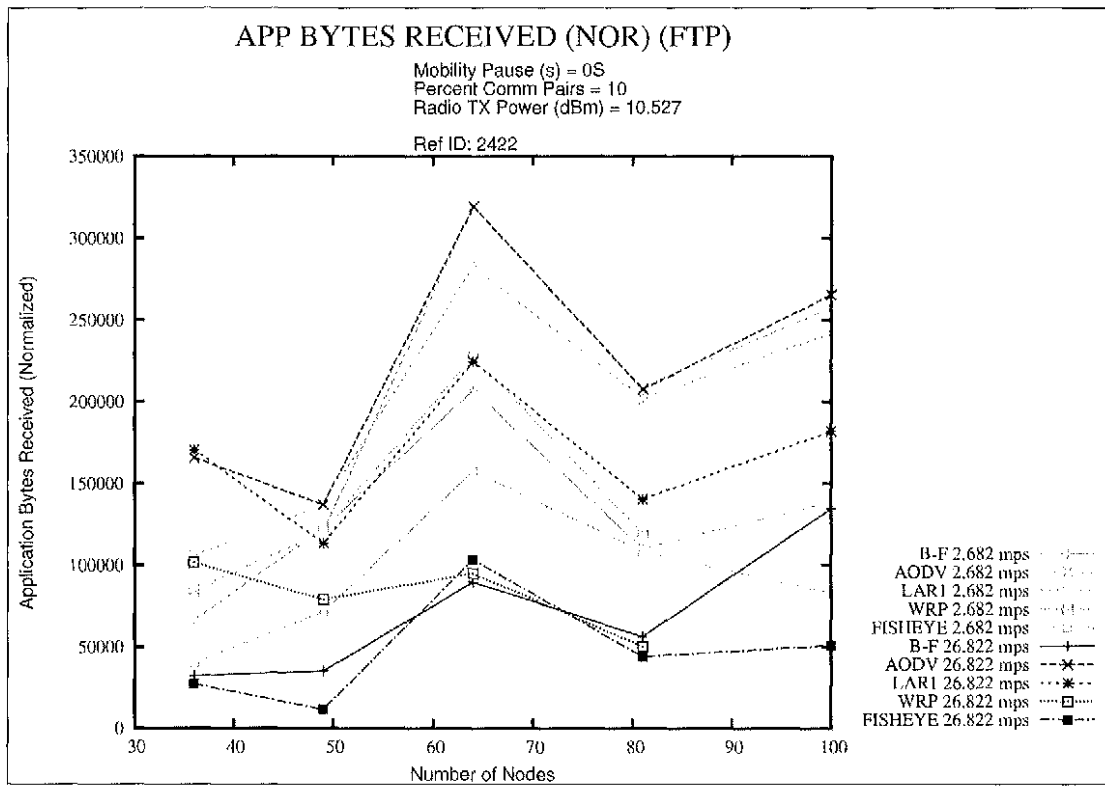

Figure 10: Application Bytes Received (Normalized), 10% Communicating Pairs, 10.527 dBm Transmit Power (Graph 1cP-n).

Application Byte Delivery Ratio appears on Figure 11 (graph 1cD). At the low speed, all the protocols have greatly improved from the results that appeared in Figure 7 (graph 1bD). At the lowest node level, the ABDR values obtained were 0.4, 0.6, 0.6, 0.7, and 0.9 for Fisheye, WRP, B-F, AODV, and LAR, respectively. All the protocols then reached values above 0.99 from the next node value on, where they stayed from then on, except for Fisheye, which reached above 0.99 only at 64 nodes and on. At the high speed, LAR, AODV, and Fisheye remained above 0.99 throughout the entire node range, B-F in a steady manner increased from 0.89 at 36 nodes to 0.95 at 100 nodes, and Fisheye stayed above 0.99 for all nodal values except 49, where it attained the overall lowest value of 0.90.

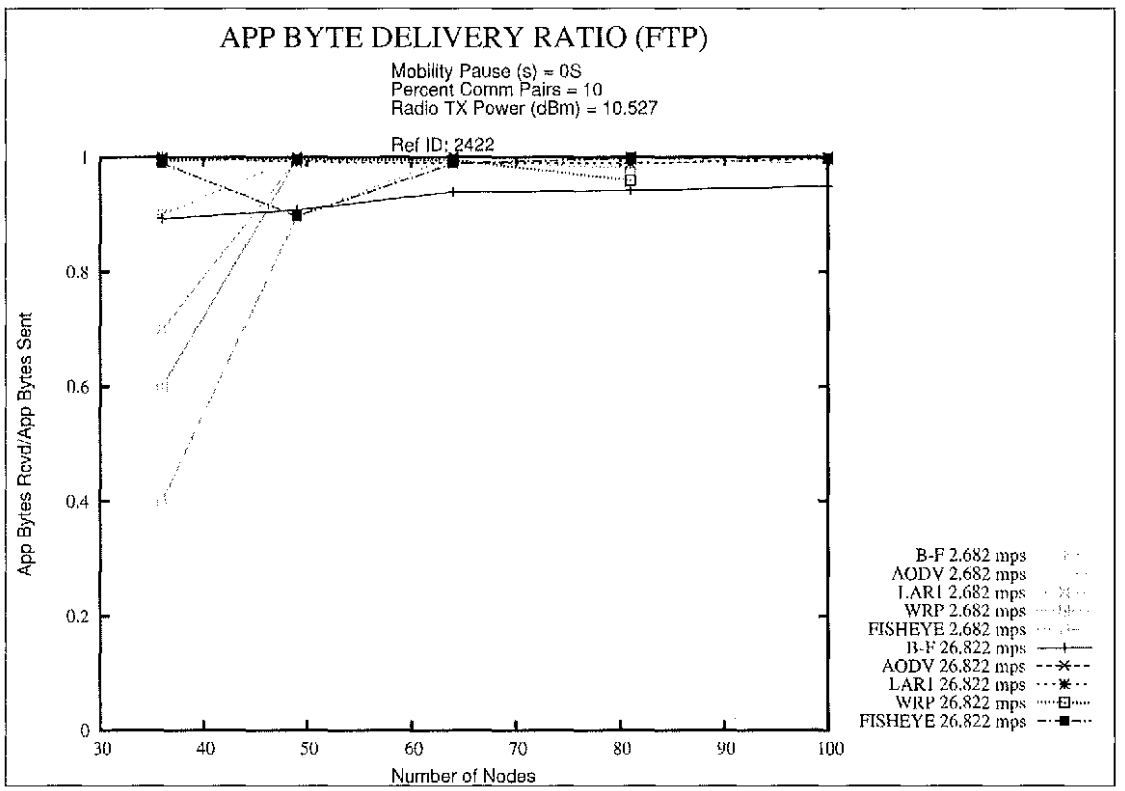

Figure 11: Application Byte Delivery Ratio, 10% Communicating Pairs, 10.527 dBm Transmit Power (Graph 1cD).

Figure 12 (graph 1cR) contains the Routing Control Packets Transmitted. WRP produced more routing control packets than either AODV or LAR at both speeds. It increased close to linearly from 21,172 (36 nodes) to 61,508 (81 nodes) packets at the low speed, and from 32,265 (36 nodes) to 85,591 (100 nodes) packets at the high speed. At the high speed, LAR increased smoothly from 3,339 routing packets to 15,959, with the curve displaying a slightly increasing overall slope as the number of nodes increases. At the low speed, LAR started out at 3,516 packets at 36 nodes, had a relatively large increase to 8,191 packets at 64 nodes, followed by a smaller increase to 9,694 packets at 81 nodes, and then had its greatest increase to 16,436 packets at 100

nodes. AODV was again the most economical protocol, very smoothly increasing from 1,627 packets at 36 nodes to 9,484 packets at 100 nodes at the high speed; at the low speed, it increased in a quite smooth manner from 381 packets at 36 nodes to 10,143 packets at 100 nodes. Interestingly it jumped by close to 2,000 packets at the low speed from each node count to the next.

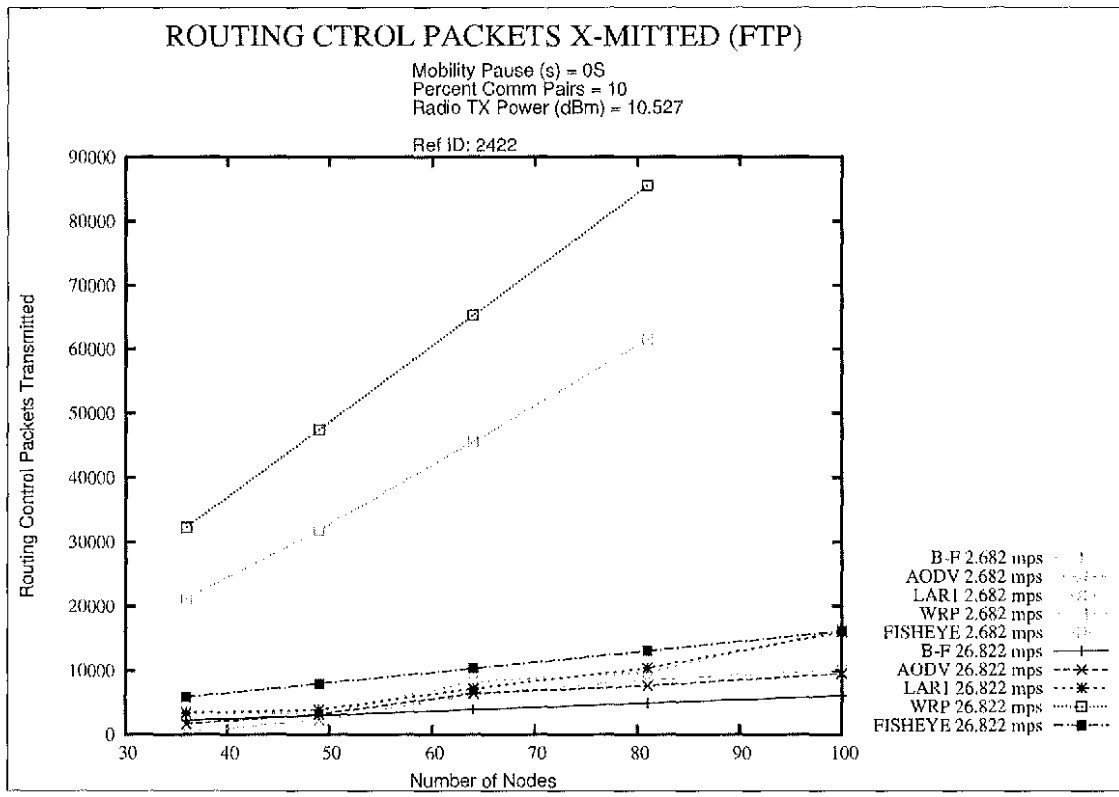

Figure 12: Routing Control Packets Transmitted, 10% Communicating Pairs, 10.527 dBm Transmit Power (Graph 1cR).

9.4 25% Communicating Pairs, 7.005 dBm Transmit Power

Application Bytes Received is shown in Figure 13 (graph 2aP-a). At the low speed, all the protocols essentially failed to transmit successfully any bytes at the lowest two or

three values of node density. The 64-node mark was when WRP, AODV, and WRP began to display some significant data transfer, 264,246 bytes for LAR and around 205,000 bytes for the other two. From that point on LAR increased quite rapidly in an almost linear fashion, peaking at 2,042,431 bytes at 100 nodes. WRP and AODV increased in a similar and moderate fashion to a performance level of 563,983 bytes (WRP) and 679,403 (AODV) bytes at 81 nodes. At that point, AODV displayed a large performance increase to 2,839,306 bytes at 100 nodes, which bested LAR. B-F and Fisheye did not see any significant data transfer until they reached 81 nodes, with 320,161 and 216,503 bytes, respectively, from which point B-F saw a large performance increase to 1,499,530 bytes at 100 nodes and Fisheye continued on an almost linearly increasing path to 509,452 bytes at 100 nodes.

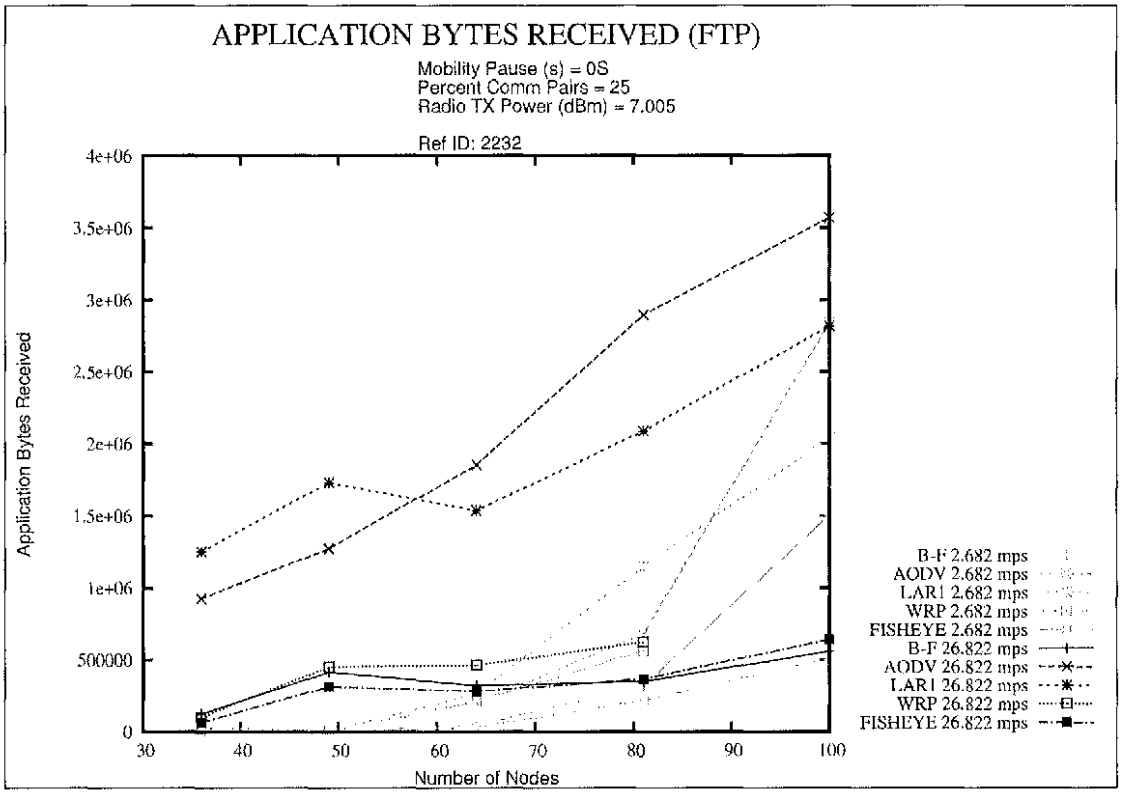

Figure 13: Application Bytes Received, 25% Communicating Pairs, 7.005 dBm Transmit Power (Graph 2aP-a).

At the high speed in Figure 13 (graph 2aP-a), the performance of the protocols clearly broke up into two groups. The first one, formed by AODV and DSR, displayed a similar performance by its members, which was very superior to that of the other protocols. Both steadily increased their performance from 926,739 (AODV) to 1,251,241 (LAR) bytes at 36 nodes to 3,569,589 (AODV) to 2,817,049 (LAR) bytes at 100 nodes. It must be noticed that LAR led AODV up to 49 nodes, at which point it suffered a drop in performance and AODV surpassed it for good.

In the second group, Fisheye, WRP, and B-F showed a repeated mild up and down performance pattern in a narrow overall range up to 81 nodes, ranging from 63,800,

102,680, and 122,515 bytes at 36 nodes, respectively, to around 360,000 bytes for B-F and WRP, and 623,595 bytes for WRP at 81 nodes. From that point on, B-F and Fisheye modestly increased to 559,073 and 639,848 bytes at 100 nodes, respectively.

The normalized results corresponding to Figure 13 (graph 2aP-a) are shown in Figure 14 (graph 2aP-n). It is immediately obvious that the graphs are extremely similar in behavior. The graph with the absolute values of performance accurately reflects the relative efficiency trends of all the data points except at the high speed for AODV and LAR, which traded efficiency leadership as progress was made to 64 nodes, with LAR dropping and AODV gaining at that point. However, the efficiencies at the high speed of both of these protocols taken together at the lower node densities stayed within the same bounds as at the higher node densities.

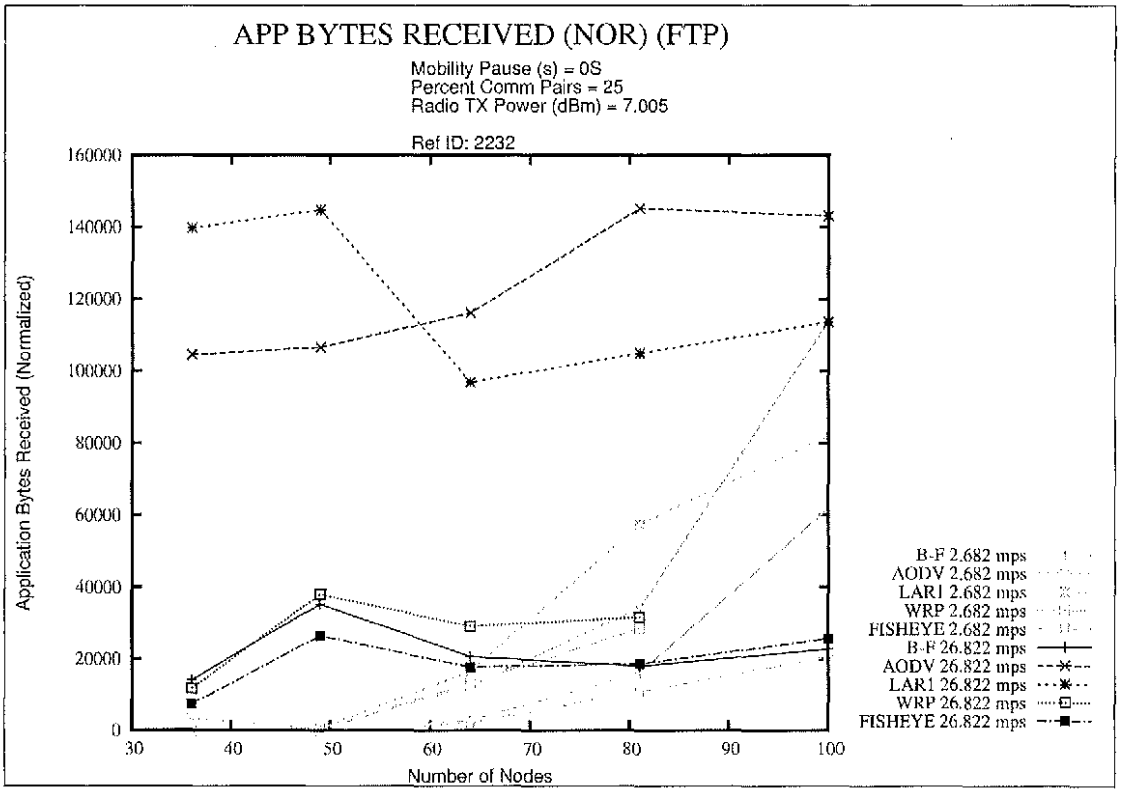

Figure 14: Application Bytes Received (Normalized), 25% Communicating Pairs, 7.005 dBm Transmit Power (Graph 2aP-n).

The Application Byte Delivery Ratio is shown in Figure 15 (graph 2aD). At the high speed, AODV and LAR maintained always an ABDR above 0.97; B-F started out at 0.97 at 36 nodes, degraded to around 0.85 at 64 nodes, and then smoothly improved to above 0.96 at 100 nodes. Continuing with the high speed, WRP and Fisheye obtained similar values throughout beginning at around 0.78 at 36 nodes, and then remained above 0.98 for all the remaining nodal values. At the low speed, AODV, WRP, and Fisheye obtained practically the same ABDR values throughout the node range, starting out at 0.1 at 36 and 49 nodes, increasing to 0.6 at 64 nodes, and then remaining above 0.99 from 81 nodes on; B-F obtained the same values except at 100 nodes, where it obtained 0.965. The best overall values at the low speed were LAR's, which started out

at 0.3 at 36 nodes, decreased to 0.2 at 49 nodes, and then remained above 0.99 the rest of the nodal range.

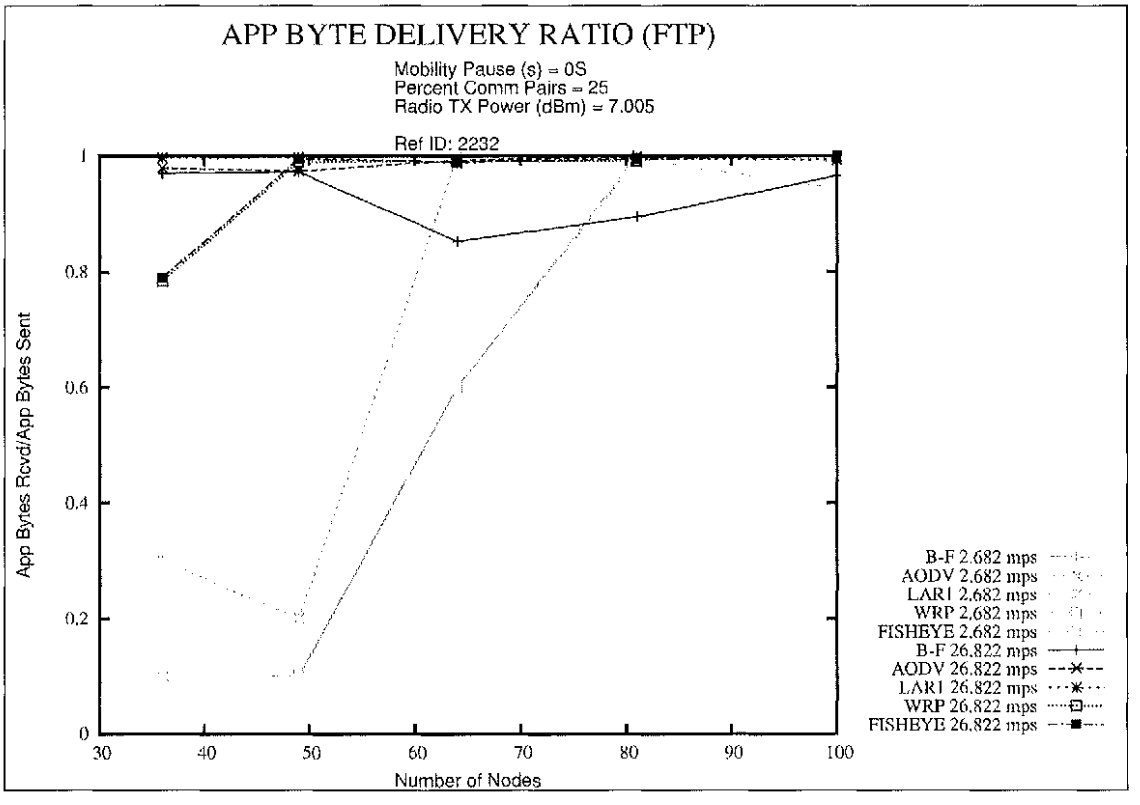

Figure 15: Application Byte Delivery Ratio, 25% Communicating Pairs, 7.005 dBm Transmit Power (Graph 2aD).

Figure 16 (graph 2aR) contains the Routing Control Packets Transmitted. WRP produced more routing control packets than either AODV or LAR at high speed, but less than LAR at the low speed at 81 nodes. It increased close to linearly from 18,946 (36 nodes) to 53,726 (81 nodes) packets at the low speed, and from 28,166 (36 nodes) to 79,781 (100 nodes) packets at the high speed. At the high speed, LAR increased smoothly from 7,545 packets at 36 nodes to 24,723 packets at 64 nodes, then it increased its curve slope significantly all the way to 100 nodes, where it produced

73,563 packets, with a slope that approximately matched that of the WRP high-speed curve. At the low speed, LAR produced more routing control packets than at the high speed. It started out at 17,073 packets at 36 nodes, increasing slowly to 35,176 packets at 64 nodes; the slope of the curve increases considerably to 81 nodes where it reaches 62,396 packets, and then it undergoes a very large increase in slope to 100 nodes where the 154,528 packet-level is reached. AODV again was very economical by comparison to all the other protocols. At the high speed it smoothly increased from 3,753 packets at 36 nodes to 31,045 packets at 100 nodes; the curve displays a definite increase in slope from 49 nodes on, the slope increase becoming larger as the number of nodes increases. At the low speed, AODV remained under 500 routing control packets transmitted up to 64 nodes, and then its number of routing control packets transmitted underwent two significant increases, the largest occurring as the nodes increased from 81 to 100, where it reached a value of 29,919 packets.

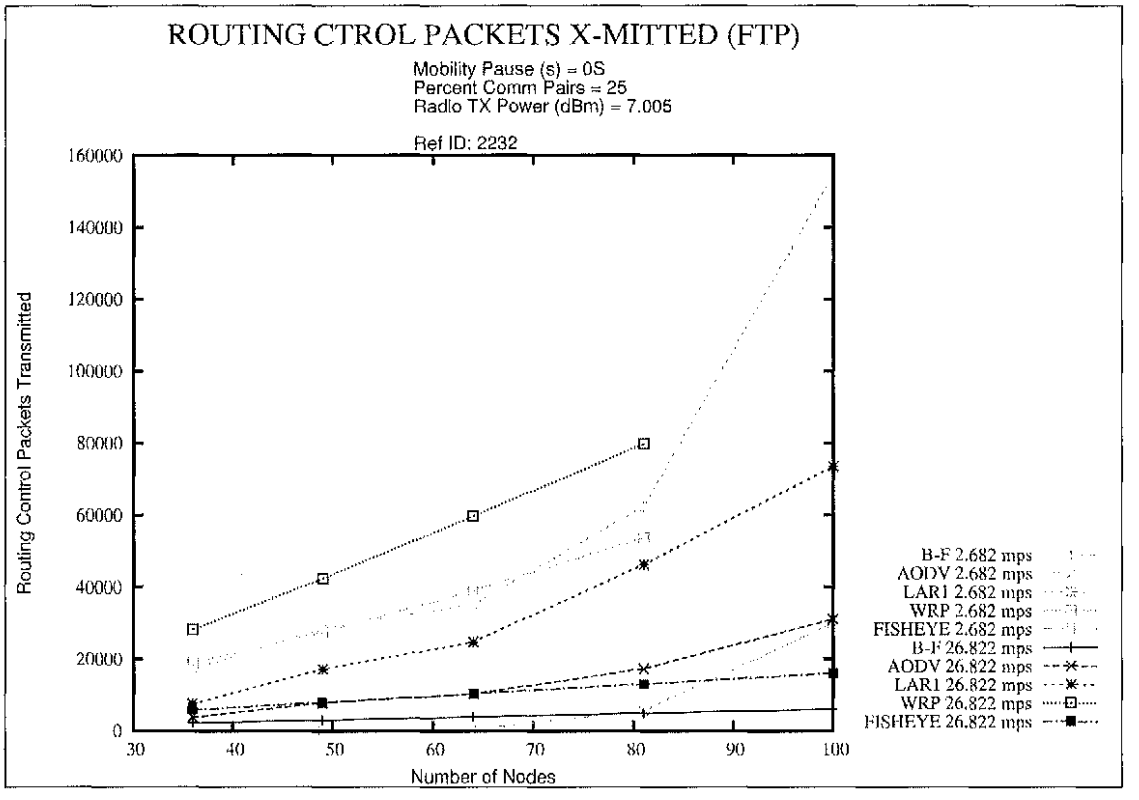

Figure 16: Routing Control Packets Transmitted, 25% Communicating Pairs, 7.005 dBm Transmit Power (Graph 2aR).

9.5 25% Communicating Pairs, 8.589 dBm Transmit Power

Application Bytes Received is shown in Figure 17 (graph 2bP-a). At the low speed, all the protocols displayed a small amount of data transfer at the lowest nodal density, with LAR transferring the most with 28,913 bytes. The same occurred at 49 nodes, except for LAR, which improved significantly to 285,200 bytes. LAR's greatest increase occurred at the next node value, to 2,227,655 bytes, from where there was a more moderate and almost linear in nature increase to the final value of 3,122,790 bytes at 100 nodes. All the other protocols had their first significant increase in performance at 64 nodes, with AODV increasing in an almost linear fashion to 3,074,343 bytes at 81

nodes, at which point it surpassed LAR's performance, WRP emulating AODV's behavior attaining 2,770,452 bytes at 81 nodes, and Fisheye increasing more slowly to 1,509,870 at 81 nodes. AODV continued its increase, albeit at a slower pace, to 3,648,441 bytes at 100 nodes, whereas B-F and Fishcye suffered from a performance deterioration at 100 nodes, attaining 2,175,131 and 1,143,681 bytes, respectively.

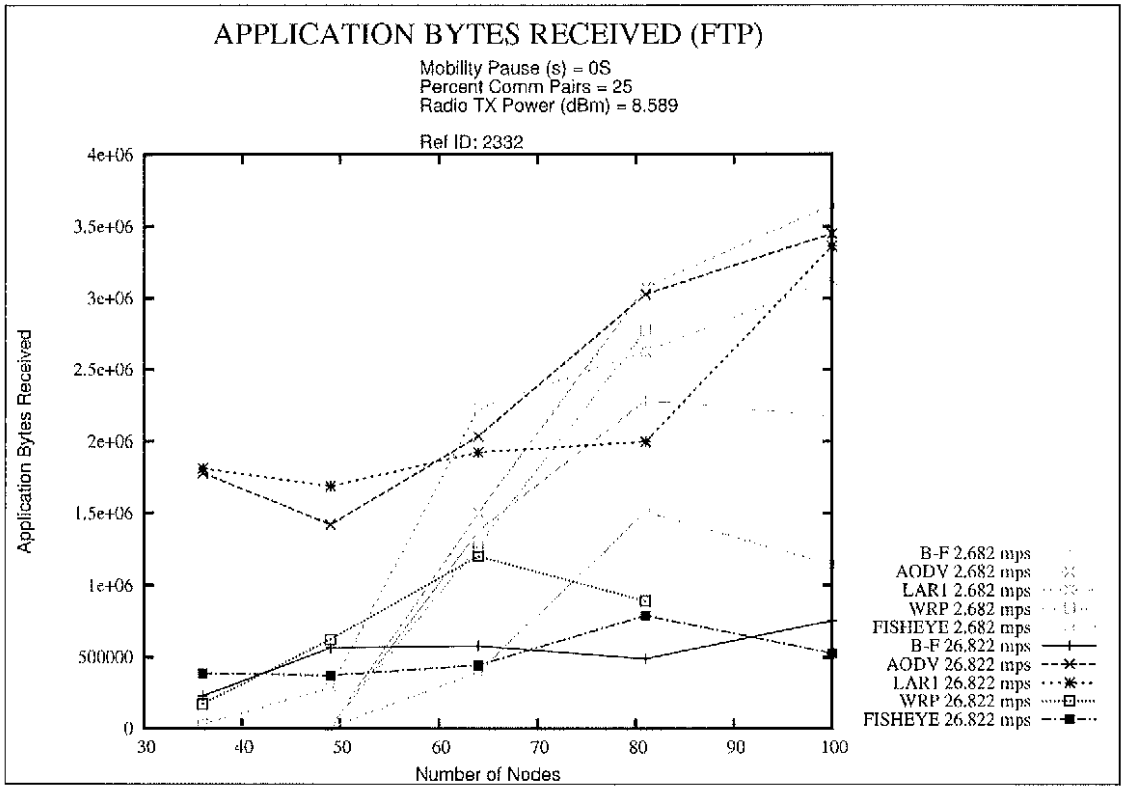

Figure 17: Application Bytes Received, 25% Communicating Pairs, 8.589 dBm Transmit Power (Graph 2bP-a).

At the high speed in Figure 17 (graph 2bP-a), AODV and LAR were clearly the better performing protocols at all node densities. AODV significantly benefited from better performance at all node densities but the highest (where it had about equal performance), as compared to its performance at the lower speed. Both LAR and

AODV started out at around 1,800,000 bytes at 36 nodes, followed by a small performance drop at 49 nodes. At that point, AODV continuously increased its performance at an almost constant pace up to 3,449,783 bytes at 100 nodes, whereas LAR increased more slowly to 81 nodes where it attained 1,996,284 bytes, followed by a sharp increase to 3,361,351 at 100 nodes. B-F and Fisheye oscillated around each other throughout the node density range, moving within a narrow and overall slowly increasing performance band. They started out in the range 226,771 (B-F) to 386,458 bytes (Fisheye) at 36 nodes and finished up in the range 753,676 (B-F) to 525,759 (Fisheye) at 100 nodes. WRP started out at a performance value slightly below that of B-F at 36 nodes, 171,101 bytes, but it initially increased quickly in a close-to-linear fashion to 1,201,421 bytes at 64 nodes, to then suffer a performance slow-down at 81 nodes with 889,993 bytes.

Figure 18 (graph 2bP-n) shows the normalized data corresponding to Figure 17 (graph 2bP-a). Again, the graphs are so similar that the absolute performance graph describes very well the relative efficiency trends of the protocols and data-points within each protocol, with a few exceptions, as follows. At the high speed, AODV's and LAR's efficiencies were highest at the lowest node count. Past the low node point, AODV's efficiency attained a local peak at 81 nodes and then dropped slightly, whereas LAR's dropped all the way to 81 nodes, where it was at its lowest value, and then it increased to end up almost even with the efficiency of AODV at 100 nodes. At the low speed, AODV showed a slight drop in efficiency from 81 nodes to 100 nodes, whereas LAR displayed the same behavior starting at 64 nodes.

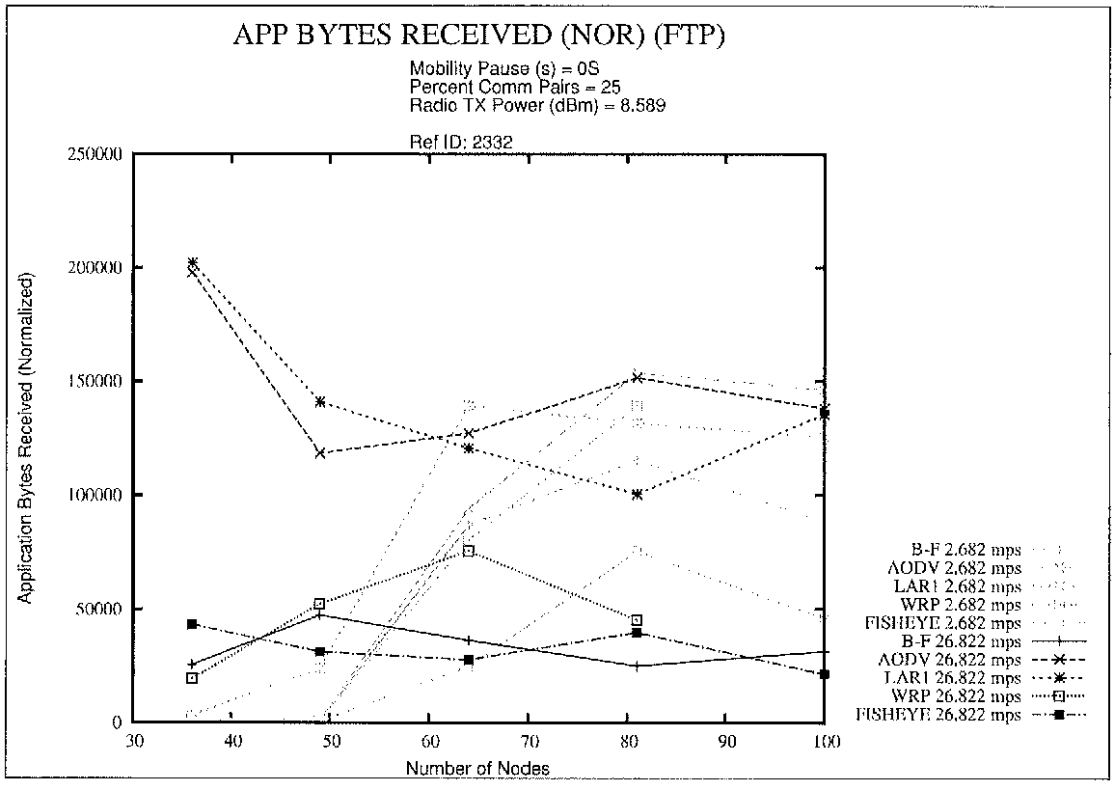

Figure 18: Application Bytes Received (Normalized), 25% Communicating Pairs, 8.589 dBm Transmit Power (Graph 2bP-n).

Application Byte Delivery Ratio is shown in Figure 19 (graph 2bD). At the high speed, AODV, LAR, and WRP attained better than 0.97 ABDR throughout the entire nodal range. Fisheye started out at 0.87 at 36 nodes and then stayed at a value better than 0.98 or better throughout the rest of the nodal range. B-F stayed at better than 0.99 or better up to 64 nodes, then dipped to 0.93 at 81 nodes and moved up to 0.96 at 100 nodes. At the low speed, LAR and AODV started out at the same level at 36 nodes, with an ABDR of 0.40, LAR then led AODV to 49 nodes, achieving 0.80 versus 0.50 for AODV, and then both remained above 0.99 starting at 64 nodes. Fisheye started at an ABDR of 0.30 at 36 nodes, remained at the same level at 49 nodes, and then achieved

an ABDR better than 0.99 for the remaining nodal values. WRP and B-F began at the same value as Fisheye, then remained within 0.1 of Fisheye at 49 (WRP above, B-F below), and then achieved 0.97 or better for the rest of the nodal range.

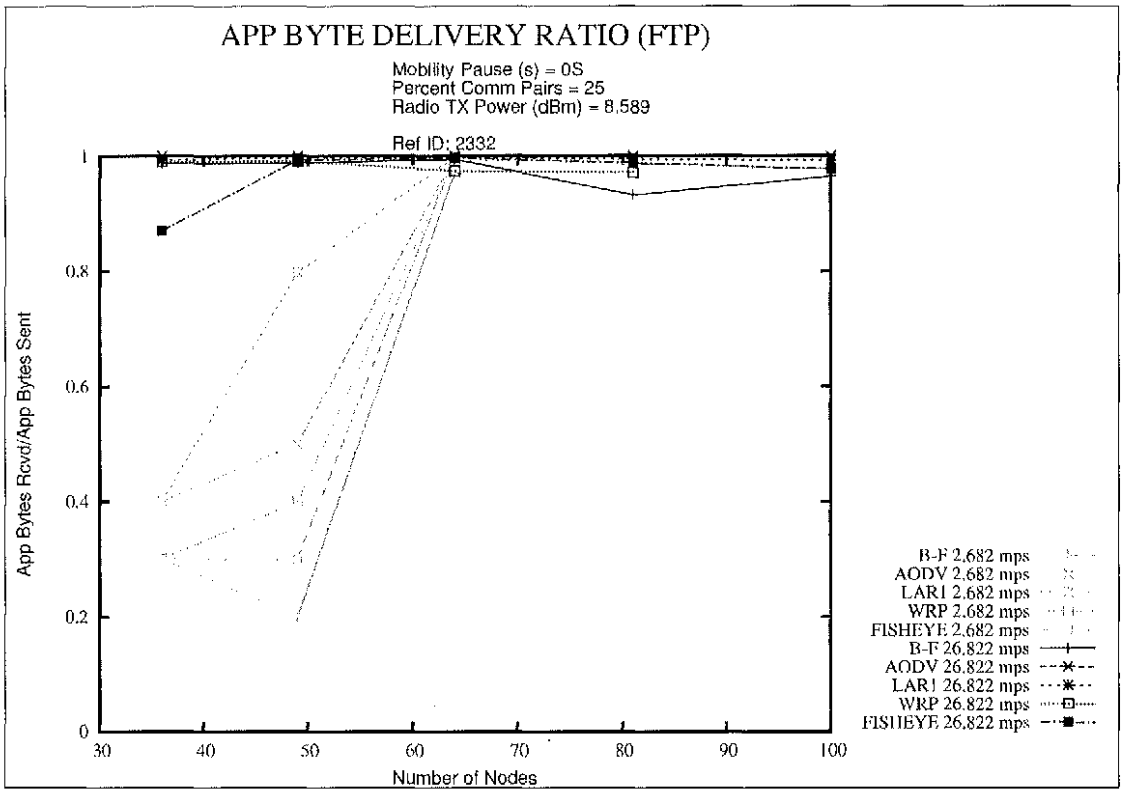

Figure 19: Application Byte Delivery Ratio, 25% Communicating Pairs, 8.589 dBm Transmit Power (Graph 2bD).

Figure 20 (graph 2bR) contains the Routing Control Packets Transmitted. WRP produced more routing control packets than either AODV or LAR at both speeds all the way up to 81 nodes. It increased close to linearly from 19,864 (36 nodes) to 58,283 (81 nodes) packets at the low speed, and from 30,114 (36 nodes) to 82,940 (100 nodes) packets at the high speed. At the high speed, LAR increased smoothly from 8,504 packets at 36 nodes to 29,547 packets at 81 nodes, the slope of the curve then increasing

significantly to 100 nodes, where 61,552 routing control packets were produced. At the low speed, it started out at 14,692 packets, increasing smoothly to 37,761 packets at 64 nodes. At this number of nodes a plateau was reached, with the packet count staying almost constant to 81 nodes and then finally undergoing a very large increase to 129,260 packets at 100 nodes. AODV once again was the thriftiest of all the protocols at both speeds. At the high speed, it increased from 2,550 packets at 36 nodes to 26,183 packets at 100 nodes, with the largest increase occurring from 64 nodes to 81 nodes, where it increased from 8,746 packets to 22,684 packets.

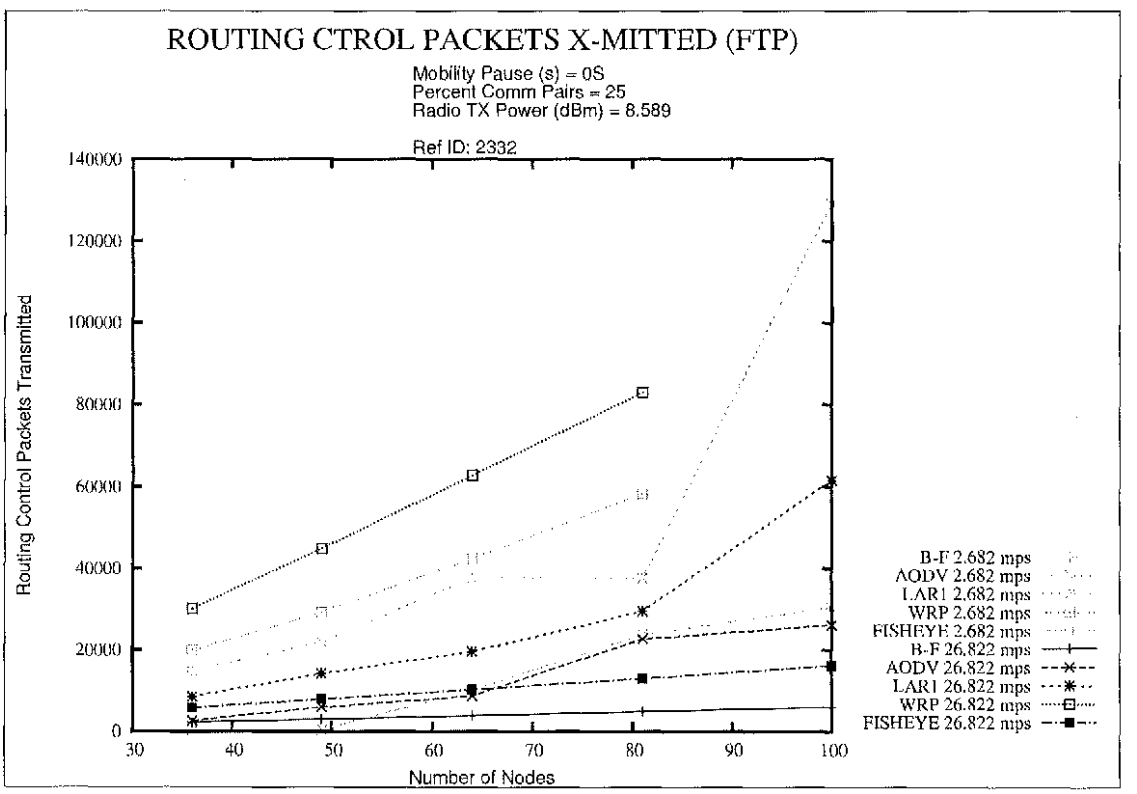

Figure 20: Routing Control Packets Transmitted, 25% Communicating Pairs, 8.589 dBm Transmit Power (Graph 2bR).

9.6 25% Communicating Pairs, 10.527 dBm Transmit Power

Application Bytes Received is shown in Figure 21 (graph 2cP-a). At the low speed, LAR and AODV were the best performers and behaved in a similar fashion throughout the nodal density range, with LAR slightly outperforming AODV. LAR displayed an almost linear performance increase, from 1,073,664 bytes at 36 nodes to 4,122,256 bytes at 100 nodes. AODV steadily increased as well, oscillating a bit, but overall close to linearly, from 667,026 bytes at 36 nodes to 4,004,738 bytes at 100 nodes. WRP displayed a behavior close to that of AODV from 36 nodes to 81 nodes, beginning at a slightly higher performance level at 36 nodes of 779,886 bytes and progressing increasingly losing performance compared to AODV to end up at 2,550,459 bytes at 81 nodes, where AODV had attained 3,224,019 bytes. B-F closely tracked WRP all the way to 64 nodes where it attained 1,974,575 bytes, past which point it improved more slowly in a close to linear fashion ending up at 2,770,773 bytes at 100 nodes. Fisheye began at 524,770 bytes at 36 nodes and increased linearly to 1,830,809 bytes at 64 nodes to decrease then very slowly to 1,814,052 bytes at 100 nodes.

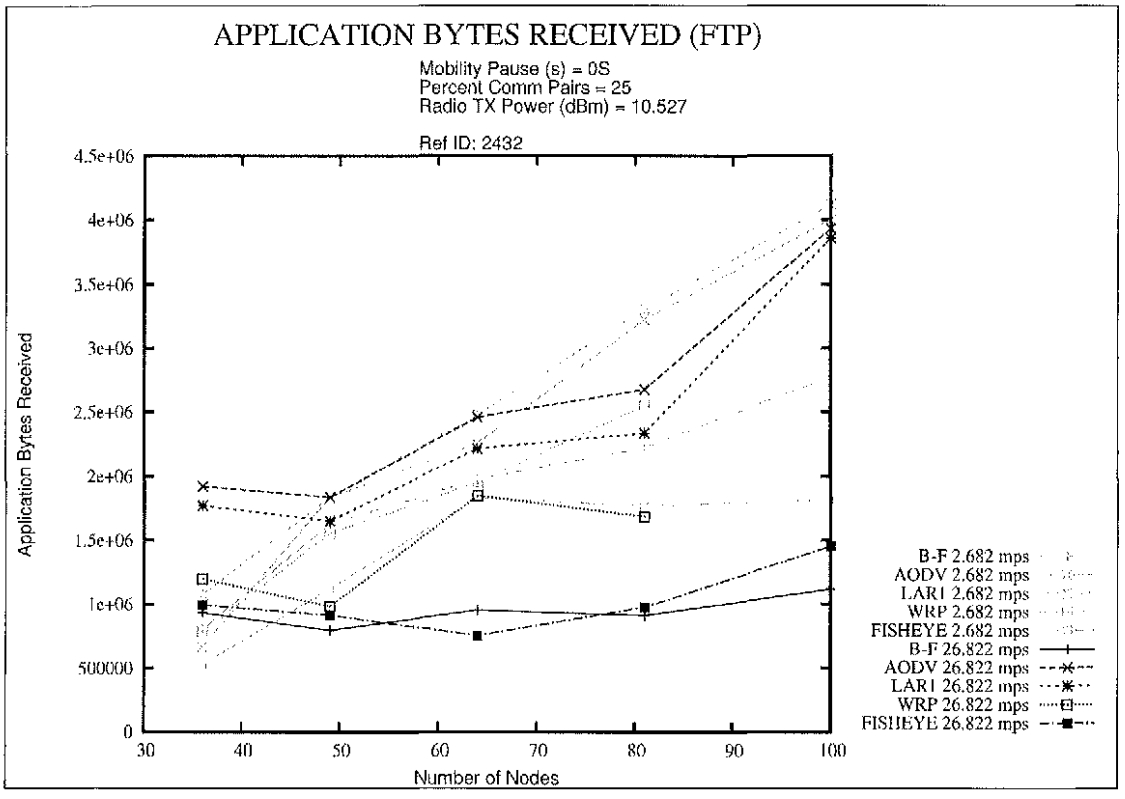

Figure 21: Application Bytes Received, 25% Communicating Pairs, 10.527 dBm Transmit Power (Graph 2cP-a).

As can be seen at the high speed in Figure 21 (graph 2cP-a), AODV and LAR were clearly the best performers by a large margin. Their performance paths behaved in almost exactly the same way, with AODV being somewhat better. They started at 1,919,809 (AODV) and 1,770,976 (LAR) bytes at 36 nodes, after which both slightly decreased in performance, to then go on a continuous increase to 3,934,18 (AODV) and 3,859,002 (LAR) bytes at 100 nodes. Interestingly each one just about intersected the other's performance curve from the low speed at 64 nodes, after which point both showed modest gains to 81 nodes, leading to large performance jumps to 100 nodes. Fisheye displayed slight and linear performance deterioration from 994,926 bytes at 36 nodes to 751,933 bytes at 64 nodes, and then a linear and slight performance increase to

1,449,005 bytes at 100 nodes. B-F oscillated up and down slightly starting out at 932,961 bytes at 36 nodes and ending up at 1,117,478 bytes at 100 nodes. WRP followed a behavior similar to that of LAR from 36 to 81 nodes but at lower performance levels, starting out at 1,197,093 bytes at the lowest node density and achieving 1,683,992 bytes at 81 nodes.

Figure 22 (graph 2cP-n) is the normalization of Figure 21 (graph 2cP-a). At the low speed, all the protocols increased in efficiency as they moved to the 49-node mark. Past that point, AODV and LAR maintained an efficiency that stayed within a narrow band, with an overall slightly increasing tendency from 61 nodes on, attaining the highest value at 100 nodes. B-F peaked at 49 nodes and from then on displayed a slight decrease in efficiency all the way to 100 nodes. Fisheye peaked at 64 nodes, and then decreased significantly from that point on, whereas WRP peaked at 49 nodes and then oscillated within a narrow band to 81 nodes. The high-speed performance started out from 36 to 49 nodes with the reverse behavior compared to the low-speed performance. All the protocols showed their peak at the lowest nodal value at this speed, and showed the largest efficiency drop when moving to 49 nodes. LAR and AODV behaved in a generally similar manner, and maintained their efficiency within a narrow band all the way from 49 to 100 nodes. Fisheye and B-F displayed a mildly decreasing efficiency from 49 to 81 nodes, and then remained constant (B-F) or slightly improved (Fisheye). WRP displayed a significant up and down oscillation all the way up to 81 nodes.

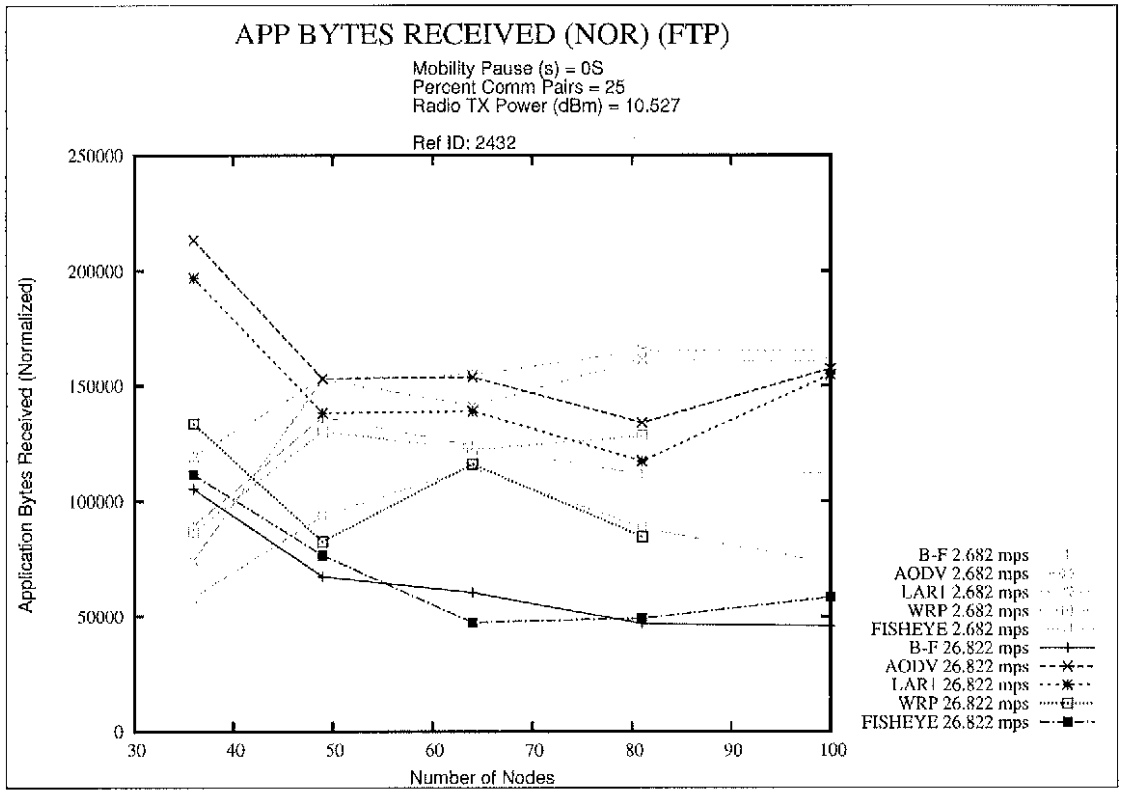

Figure 22: Application Bytes Received (Normalized), 25% Communicating Pairs, 10.527 dBm Transmit Power (Graph 2cP-n).

Figure 23 (graph 2cD) displays the Application Byte Delivery Ratio. At the high speed, all the protocols attained 0.97 ABDR or above through the entire nodal range, except for Fisheye at 36 nodes with an ABDR of 0.80. At the low speed, all the protocols attained 0.98 or better throughout the entire nodal range except for Fisheye, B-F, and WRP at 36 nodes, where they attained values of 0.80, 0.90, and 0.90, respectively.

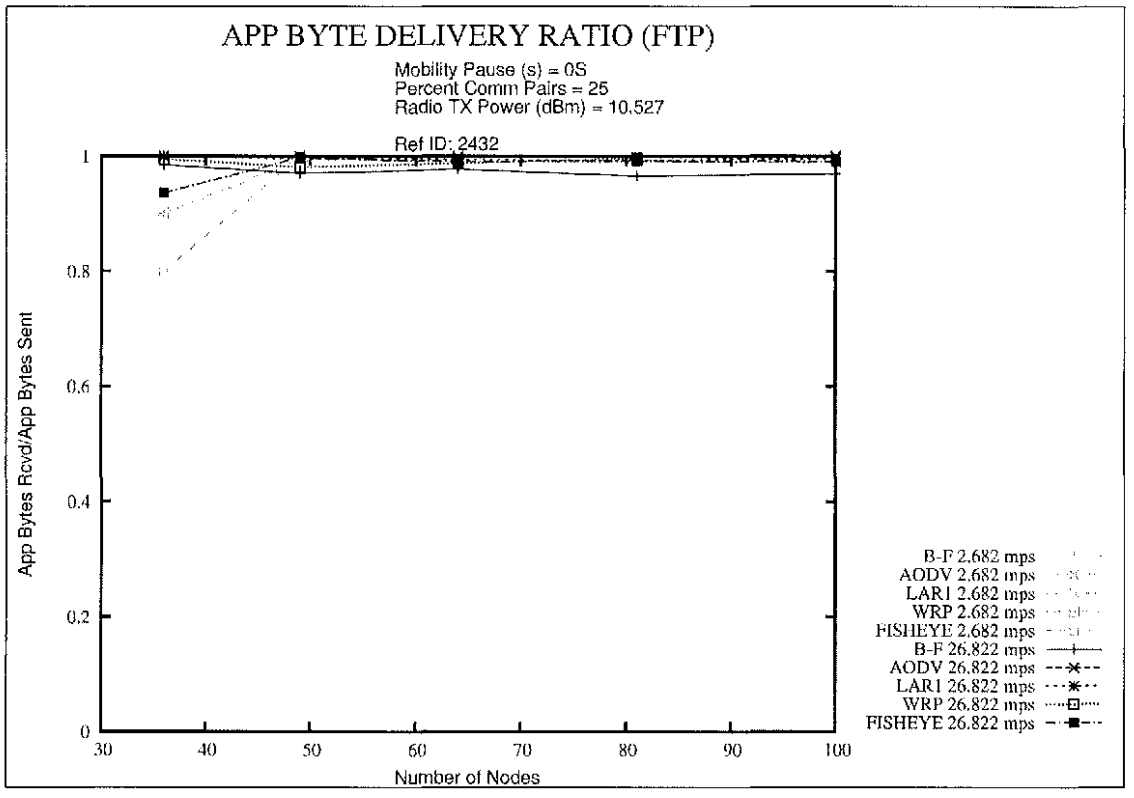

Figure 23: Application Byte Delivery Ratio, 25% Communicating Pairs, 10.527 dBm Transmit Power (Graph 2cD).

Figure 24 (graph 2cR) contains the Routing Control Packets Transmitted. WRP produced many more routing control packets than either AODV or LAR at both speeds up to 81 nodes. It increased close to linearly from 21,186 (36 nodes) to 61,923 (81 nodes) packets at the low speed, and from 32,233 (36 nodes) to 85,659 (100 nodes) packets at the high speed. LAR, at the high speed, increased its number of routing control packets transmitted all the way from 5,865 packets at 36 nodes to 53,451 packets at 100 nodes, with the largest increase occurring from 81 to 100 nodes; the slope of the curve from the 49-node mark on becomes steeper as the number of nodes increases. At the low speed, LAR produced slightly more routing control packets than at the high speed at 36 and 100 nodes only, starting out at 9,804 packets at 36 nodes,

and increasing to 53,451 packets at 100 nodes. The slope of the curve is slightly negative from 36 to 49 nodes, at which point it becomes increasingly positive, with the greatest increase in slope occurring from 81 to 100 nodes. AODV performed better than the other protocols at both speeds. At the high speed, it increased quite smoothly from 2,883 packets at 36 nodes to 22,388 packets at 100 nodes. At the low speed, AODV increased from 2,686 nodes to 29,533 nodes, with the slope becoming noticeably higher past 64 nodes, where the packet value was 11,603.

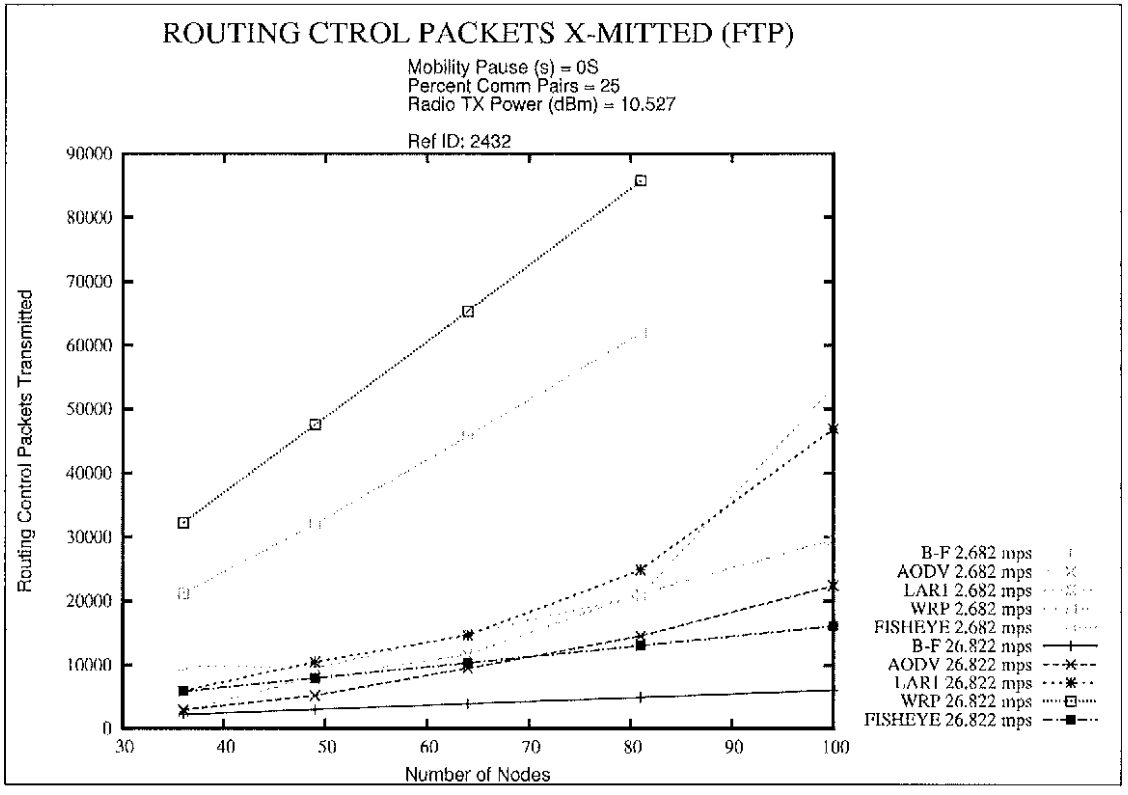

Figure 24: Routing Control Packets Transmitted, 25% Communicating Pairs, 10.527 dBm Transmit Power (Graph 2cR).

Chapter 10

OBSERVATIONS AND CONCLUSIONS

10.1 Observations

In both the low-speed and high-speed scenarios depicted in Figure 1 (graph 1aP-a), AODV and LAR provided the highest values of Application Bytes Received. Their performance was much better throughout at the high speed, where their performance advantage over the other protocols was also greatest, than at the low speed. At the high speed, LAR performed better at the lower node densities, whereas AODV outperformed LAR at the higher densities. At the low speed, LAR was a better performer than AODV except at the very highest nodal density.

At 64 nodes or less, the high speed actually made performance better for all the protocols. Only LAR and AODV improved at both the low and high speed as they progressed to the highest node density. The remaining protocols improved when reaching 100 nodes only at the low speed, while displaying the opposite behavior at the high speed. Figure 2 (normalized graph 1aP-n) clearly shows the largest efficiency gains occurring at 64 nodes at the high speed, and at 64 and 100 nodes at the low speed. AODV and LAR were much more efficient at the high speed (except for AODV at 100 nodes, where there is not much difference), whereas the other protocols were just slightly so, except at 100 nodes.

Figure 5 (graph 1bP-a) differs parametrically from Figure 1 (graph 1aP-a) only by an increase in the transmitter power (and thus radio range) to the mid-power level. AODV and LAR were again the best overall performers, and the 64-node mark again signaled a point where the performance of all the protocols greatly improved. The high-speed scenario at this power level was similar to the one in Figure 1 (graph 1aP-a), generally displaying better performance due to the higher power as expected, except for the following facts:

- The 64-node mark is now a local performance peak. The performance of all the protocols, except B-F, actually became worse as they progressed to 81 nodes (B-F showed hardly any improvement). At 81 nodes, AODV and LAR dipped below their performance level at this node density at the lower power level Figure 1 (graph 1aP-a).
- The performance spread between AODV and LAR was tighter in the cases depicted in Figure 5 (graph 1bP-a) than in those depicted in Figure 1 (graph 1aP-a).

The low-speed scenario in Figure 5 (graph 1bP-a) was quite different from that in Figure 1 (graph 1aP-a). The following observations can be made regarding Figure 5 (graph 1bP-a):

- The performance of AODV and the performance of LAR at the low speed were close to their high-speed performance except at the two lowest nodal densities.

- The performance of the remaining protocols was better at the low speed than at the high speed, except again at the lowest two nodal densities.
- B-F and LAR exhibited the performance deterioration effect at the 81-node mark at the low speed also.

Figure 6 (normalized graph 1bP-n) shows the largest efficiencies in bytes received. All occurred at 64 nodes at both the low and the high speeds. AODV and LAR were more efficient at the high speed than at the low speed, except at 81 nodes, where there was not much difference. The other protocols behaved the opposite way.

Figure 9 (graph 1cP-a) differs parametrically from Figure 5 (graph 1bP-a) by a further increase in transmitter power. It is very similar to Figure 5 (graph 1bP-a), with some moderate performance gains due to the larger transmitter power. The following observations can be made from Figure 9 (graph 1cP-a):

- There were hardly any differences in performance for AODV between the two speed levels (the high-speed one gave barely better performance).
- LAR exhibited a large performance differential between its low-speed and its high-speed curves, with its high-speed performance being worse (indeed, it was about the same as at the prior power level curve).
- The 81-node mark was clearly a local performance trough for all the protocols.

Figure 10 (normalized graph 1cP-n) shows the most efficient nodal value for Application Bytes Received was 64 nodes for all the protocols at both speeds, except for B-F at the high speed, for which the most efficient nodal value was 100 nodes, followed by 64 nodes. Figure 10 (normalized graph 1cP-n) also shows all the protocols were generally more efficient at the low speed, except for AODV, which was (slightly) better at the high speed.

LAR and AODV were again the best performers in Figure 13 (graph 2aP-a), their performance being much higher at the high speed, where they also enjoyed the largest performance differential over the other protocols. This graph is quite close in character to graph Figure 1 (1aP-a), from which it differs parametrically only by having more servers and corresponding clients. A higher performance is expected simply because of that parametric difference. Indeed such was the case at the higher speed, where all the protocols displayed gains. The most noticeable differences at the high speed with respect to Figure 1 (graph 1aP-a) are the following:

- The smoother slope distribution of the AODV and LAR curves.
- LAR actually lost performance from 49 to 64 nodes.
- The remaining protocols showed a performance improvement earlier on, at 49 nodes. Beginning at 49 nodes, their curves display a slightly convex overall curvature.

At the low speed, again comparing to Figure 1 (graph 1aP-a), the following is noted:

- LAR and AODV behaved the same way except that their performance was higher (except at the very lowest node density).
- Bellman-Ford and Fisheye, which tracked each other very closely in Figure 1 (graph 1aP-a), only improved at the higher node densities, even though the overall shape of their curves is very similar to those in Figure 1 (graph 1aP-a), and the B-F performance became much better at the highest node density.
- The performance of WRP improved throughout.

Figure 14 (normalized graph 2aP-n) shows AODV and LAR were most efficient at the high speed, with AODV leading at higher node densities; in all other cases, higher speeds resulted in higher efficiency at lower node values (decreasing with more nodes), and lower speeds at node values 81 and above (increasing with increasing nodes).

Figure 17 (graph 2bP-a) differs parametrically from Figure 13 (graph 2aP-a) only by an increase in the transmitter power level, leading to generally higher performance, as expected. At the high speed, AODV and LAR were clearly superior to all the proactive protocols, as before. The high-speed curves are similar to those in Figure 13 (graph 2aP-a). The following observations can be made comparing Figure 17 (graph 2bP-a) with Figure 13 (graph 2aP-a):

- Both LAR and AODV lost performance as they progressed to 49 nodes. LAR was still better at low node densities and AODV at higher ones, with the LAR performance improvement being very small between 49 and 81 nodes.

- B-F and Fisheye stayed within a narrow performance range, but instead of tracking each other, they traded their performance advantage repeatedly.
- WRP had an initial large performance increase, peaking at 64 nodes and then subsiding.

The low-speed curves display the greatest differences with respect to the previous graph. The performance of the proactive protocols became quite similar to that of the reactive protocols. The first large increase in performance began for all the protocols in the transition from 49 to 64 nodes, with AODV, WRP, and LAR attaining close levels of performance (in that order, from higher to lower) at 81 nodes, and B-F at a somewhat lower level. B-F peaked at 81 nodes and then declined, with AODV and LAR continuing their improvement. Fisheye followed the behavior of B-F, but at lower levels of performance. Figure 18 (normalized graph 2bP-n) shows the greatest efficiency in the application bytes received was that of AODV and LAR, both of them being more efficient at the high speed. Additionally, at the low speed there was a clear peaking in efficiency at 64 and 81 nodes, whereas at the high speed, except for LAR and AODV at 36 nodes, the efficiencies remained within a relatively narrow range for each protocol.

A further increase in transmitter power from the cases in Figure 17 (graph 2bP-a) to those in Figure 21 (graph 2cP-a) resulted in further generalized performance gains, as expected. At the high speed, the curves for AODV and LAR have a shape very similar to that of LAR in Figure 17 (graph 2bP-a). B-F, Fisheye, and WRP behaved in a very

similar fashion to how they did in Figure 17 (graph 2bP-a). At the low speed, the behavior of all the protocols was similar to how they behaved in Figure 17 (graph 2bP-a), except for the fact that the large performance increase began right away from the lowest node density. Figure 22 (normalized graph 2cP-n) shows the bytes received efficiencies were similar (AODV and LAR) or superior (B-F, Fisheye, WRP) at the low speed compared to the high speed. There were no great changes past 49 nodes for both speeds, but there was a relatively large drop from 36 to 49 nodes at the high speed and the opposite at the low speed.

Protocol overhead in the Routing Control Packets Transmitted graphs, Figure 4, Figure 8, and Figure 12 (graphs 1aR through 1cR, respectively), and Figure 16, Figure 20, and Figure 24 (graphs 2aR through 2cR, respectively), provides an insight into protocol overhead changes as the cases change. B-F results were linear and the same in all the run cases, coming out at 60 broadcasts per node for the test time. This is consistent with the fact that in this protocol each node periodically broadcasts routing table information to its neighbors. The number of broadcasts thus grew proportionally to the number of nodes, and the size of the broadcasts grew too.

The routing behavior of Fisheye in terms of protocol overhead was similar to that of B-F; indeed its results were also the same for all cases, linearly increasing with the number of nodes and coming out at 161 broadcasts per node for the test time. The same comments apply as for B-F, except the size of the routing messages was kept smaller by

varying the information in them based on the scope of the nodes involved in the information, and the frequency of messages sent was higher.

The performance and general behavior of both B-F and Fisheye (refer to Application Bytes Received graphs) was quite close for the cases run. B-F was always slightly superior at the low speed, but at the high speed, one was superior at times and the other one at the other times without one clearly leading. Both protocols suffered greatly with increasing speed (except at the lower power, where performance was lowest but similar at both speeds), suggesting these protocols have a great deal of trouble keeping up with increasing connectivity change. A possible improvement for this situation would be to make the nodes "speed aware" and increase the routing table broadcasting (to the neighbors) as a function of speed. The price would be, of course, increased overhead.

The curves for WRP in all the Routing Control Packets Transmitted graphs show an increase in the number of control routing packets transmitted with increased node density, as expected, in a fashion close to being linear (but not quite). They are impervious to the number of communicating pairs, being only dependant on speed (higher speed, higher topology change rate) and transmit power (lower transmit power, lower connectivity). Since it is a proactive protocol, WRP goes about maintaining up-to-date routes from each node to all the other nodes, regardless of actual messages sent. Higher speed and higher transmit power bring increased routing packet traffic. Thus, the ranges of control packets per node transmitted by WRP were as follows for the different figures:

- 526 to 663 at the low speed and 782 to 985 at the high speed for Figure 4 (graph 1aR) and Figure 16 (graph 2aR).
- 553 to 715 at the low speed and 839 to 1,024 at the high speed for Figure 8 (graph 1bR) and Figure 20 (graph 2bR).
- 588 to 759 at the low speed and 896 to 1057 at the high speed for Figure 12 (graph 1cR) and Figure 24 (graph 2cR).

It is obvious WRP had much higher routing overhead than AODV in all cases, and higher than LAR except at the low speed for 81 nodes in Figure 16 (graph 2aR). Past 81 nodes, the trend (curve slope) at low speed favored WRP at the two lower transmit powers. AODV always used much fewer routing control packets than WRP and fewer than LAR. Its routing control packet usage increased quite smoothly with increasing number of nodes in each Routing Control Packets Transmitted graph. As expected, it was larger when the number of communicating pairs was higher. As the speed was increased, holding everything else constant (i.e., comparing within each Routing Control Packets Transmitted graph the low-speed and high-speed curves), the following was observed regarding the Number of Routing Control Packets metric:

- It increased at the low-power setting.
- It became almost the same at the mid-power setting and low-power setting with lower data flow density.
- It became lower at the high-power setting at the higher data flow density.

However, the difference in values was not large within the same number of data flows. LAR was much less efficient than AODV in the number of routing control packets it generated, with a trend towards much higher overhead as the number of nodes increased. LAR showed a tendency to "take off" after 81 nodes, particularly at the low speed. It was only competitive with AODV at the higher power setting, and better than WRP except at the take-off points. Its overhead tended to be higher at the low speed than at the high speed except at the high-power settings.

The Application Byte Delivery Ratio metric, displayed in Figure 3, Figure 7, and Figure 11 (graphs 1aD through 1cD, respectively) and in Figure 15, Figure 19, and Figure 23 (graphs 2aD through 2cD, respectively), improved within each figure as the number of nodes increased, indicating that constructing delivery routes got easier within the cases represented in the figure. Also within each figure, the delivery ratios were initially better at the high speed, with the low-speed ratios usually catching up after a certain node density was reached. The nodal value at which this catching up occurred in the simulations was lower and lower as the radio transmit power level was increased, being 81 nodes at the lowest power level, 64 nodes at the middle power level, and 49 nodes at the highest power level. This effect was due to the larger radius a node's transmissions could reach, which increased the number of neighbors a node had, making the routing easier. At the lower nodal densities, it seems counterintuitive that the Application Byte Delivery Ratio was better at the high speed. This seems to indicate that the (much) lower speed kept nodes that could not communicate with each other (due to not having

a route) unable to do so longer, as opposed to the higher speed, where that state was bound to change soon (back and forth from getting a route to not getting a route).

Moving from figure to figure by only increasing the radio transmit power resulted in better ratios, as has been discussed. Increasing just the number of communicating pairs provided somewhat higher delivery ratios, especially at the lower densities and higher speeds. This means that of the packets sent, more were delivered successfully, but does not mean the success in delivering gross numbers of packets increases with some proportionality to the number of communicating pairs at the lower density. This effect was more marked at the lower transmit powers. Consider, for example, WRP at the high speed, comparing its Application Bytes Delivered and its Application Byte Delivery Ratios for the 10% and the 25% communicating pairs at the lowest power level. The values were, respectively, 98,782 bytes with a ratio of 0.5 and 102,680 bytes with a ratio of 0.784. Therefore, even though there were two and a half times the number of communicating pairs, the number of received bytes was about the same. FTP transfers were set up less often but were more successful when they occurred. It is not clear why this effect was so, except for the speculation that taken together, the extra communicating pairs present in the 25% communicating pairs cases were more successful overall pairs than the first 10%.

10.2 Conclusions

Some overall conclusions can be drawn from the above observations. Regarding the Application Bytes Received metric applied to FTP data flows in the scenarios investigated, the following general statements can be made:

1. Trivially, higher transmission power and higher client-server density led to higher performance. Higher nodal density generally tended to do the same in an overall sense but displayed some exceptions.

2. The reactive protocols (AODV, LAR) performed much better than the proactive protocols handling FTP. The relative advantage increased with (increasing) speed, which points to the trouble the proactive paradigm has in keeping up with increasing mobility.

3. LAR and AODV had similar performance, but AODV performed somewhat better past the lower nodal densities.

4. B-F, Fisheye, and WRP had much lower performance than AODV and LAR. All three suffered greatly at the high speed at the mid and high transmission power levels. WRP was generally the best of the three at the high speed. B-F and WRP were comparable at the low speed. Fisheye was the lowest overall performer.

5. At the low client-server density, the greatest gain in performance was usually at 64 nodes, particularly at mid and high transmission power levels. That node value was clearly the most efficient one for all protocols at mid and high power (and for AODV and LAR at low power and high speed as well). The 81-node mark often displayed a performance stall or loss.

6. At the high client-server density and high speed, while AODV and LAR displayed an overall performance increase trend, the remaining protocols did not increase their performance much past 49 nodes.

7. At the high client-server density and low speed, most protocols at low and mid transmission power showed a definite nodal value where the performance increase was very large compared with all their other performance increases. At high power, the rate of performance increase was more constant, and, thus, so was the efficiency.

8. Higher speed did not always mean lower performance. At the lower nodal densities, performance at lower transmission powers tended to be improved by the higher speed. This suggests the balance between connection establishment/re-establishment and connection fleetingness worked out favorably for those conditions.

Regarding the Routing Control Packets Transmitted metric, it must be recalled that the nature of the figures obtained was comparable for AODV, LAR, and WRP as a group, and for B-R and Fisheye as a separate group. It represents the number of packets actually transmitted for the first group and the number of periodic "broadcasts" or "interscope updates" transmitted for the second group. The following conclusions can be drawn:

1. Generally, WRP transmitted more routing control packets than LAR, and LAR transmitted more control packets than AODV. However, LAR showed very high increases to 100 nodes at the low and mid transmission power levels.

2. WRP increased almost linearly with the number of nodes, but it was clearly dependent on the speed and transmission power. However, it was not dependent on the number of communicating pairs.

3. At the two lower transmission power levels, LAR transmitted significantly more routing control packets at the low speed than at the high speed, at both the low and high client-server densities.

4. The number of routing control packets transmitted by both B-F and Fisheye was proportional to the number of nodes and displayed no speed or transmission power dependency.

The Byte Delivery Ratio metric showed an increase with increasing nodal density and increased transmission power, since it was easier to establish routes from sources to destinations for those conditions. At the lower nodal densities, the high speed improved the ratio compared to the low speed. This is thought to be due to more frequent, although more fleeting, connectivity re-establishment between nodes. The reactive protocols displayed a significantly better overall ratio than the proactive ones.

This study demonstrates in general terms the usefulness of simulation as a research tool, allowing one to quickly, efficiently, and cheaply run a very large number of (numerical) experiments. If done with actual hardware, running as many cases as have been run would have been a very large undertaking in terms of time, money, and effort. This ability to run so many cases quickly and cheaply makes it possible to steer researchers towards well-refined and optimized candidate solutions to real-world situations, which could then be developed and tested with real hardware.

In the current work, five protocols were tested in a series of scenarios. Based on the results obtained, the two protocols of those evaluated that showed the most promise, by a large margin, were LAR (with DSR as the underlying protocol) and AODV. Thus, further simulation of more scenarios could now concentrate on those two candidates, or perhaps just one, and lead to some testing with actual hardware in real-world cases covered already by simulation. Modifications to those two protocols could also be developed and tested in the simulator in an attempt to improve and optimize them further. Both testing of more scenarios concentrating on the two best protocols

evaluated and improving and optimizing them could be subjects for further research. Some general discoveries were also made, such as the fact that a higher speed, within the limits of the scenarios tested, may lead to higher network performance in some cases.

REFERENCES

Print Publications:

[Aggelou99]
Aggelou G. and R. Tafazolli, "RDMAR: A Bandwidth-Efficient Routing Protocol for Mobile Ad Hoc Networks," Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia (August 1999), pp. 26-33.

[Ayildiz02]
Ayildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," Computer Networks 38, 4 (March 2002), pp. 393-422.

[Bajaj99]
Bajaj, L., M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," Technical Report #990027, UCLA Computer Science Department, 1999.

[Belding03]
Belding-Royer, E. M. and C. E. Perkins, "Evolution and Future Directions of the Ad-hoc On-Demand Distance Vector Routing Protocol," Ad Hoc Networks 1, 1 (July 2003), pp. 125-150.

[Boukerche01A]
Boukerche, A., "A Performance Comparison of Routing Protocols for Ad Hoc Networks," Proceedings of the 15th International Parallel and Distributed Processing Symposium (2001), pp. 185-191.

[Boukerche01B]
Boukerche, A., "A Simulation Based Study of On-Demand Routing Protocols for Ad Hoc Wireless Networks," Proceedings of the 34th Annual Simulation Symposium (2001), pp. 1-8.

[Boukerche04]
Boukerche, A., "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks," Mobile Networks and Applications 9, 4 (August 2004), pp. 333-342.

[Bragodia98]
Bragodia, R., R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "Parsec: A Parallel Simulation Environment for Complex Systems," IEEE Computer 31, 10 (October 1998), pp. 77-85.

[Brennan04]
Brennan, S. M., A. M. Mielke, D. C. Torney, and A. B. Maccabe, "Radiation Detection with Distributed Sensor Networks," IEEE Computer 37, 8 (August 2004), pp. 57-59.

[Broch98]
Broch, J., D. A. Maltz, D. B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (August 1998), pp. 85-97.

[Camp02]
Camp, T., J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," Proceedings of the 20th Annual Joint Conference IEEE Computer and Communications Society (June 2002), pp. 1678-1687.

[Chen98]
Chen, T.W. and M. Gerla, "Global State Routing: A New Routing Scheme for Ad-Hoc Wireless Networks," Proceedings of IEEE International Conference on Communications (June 1998), pp. 171-175.

[Chiang97]
Chiang, C. C., H. S. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel," Proceedings of IEEE SICON 1997 (April 1997), pp. 197-211.

[Chun00]
Chun, Y., L. Qin, and S. M. Lin, "Routing Protocols Overview and Design Issues for Self-Organized Network [sic]," International Conference on Communication Technology Proceedings 2 (2000), pp. 1298-1303.

[Clausen01]
Clausen, T. H., G. Hansen, L. Christensen, and G. Behrmann, "The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation," Proceedings of IEEE Symposium on Wireless Personal Mobile Communications 2001 (September 2001).

[Clausen02]
Clausen, T. H., P. Jacquet, and L. Viennot, "Comparative Study of CBR and TCP Performance of MANET Routing Protocols," First Annual Mediterranean Ad Hoc Networking Workshop (September 2002).

[Danzig91]
Danzig, P.B. and S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics," USC Tech. Report, USC-CS-91-495, 1991.

[Das00]
Das, S. R., C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," Proceedings of the IEEE Infocom 2000 (March 2000), pp. 3-12.

[Dube97]
Dube, R., C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," IEEE Personal Communications 4, 1 (February 1997), pp. 36-45.

[Galstyan04]
Galstyan, A., B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed Online Localization in Sensor Networks Using a Moving Target," Proceedings of the Third International Symposium on Information Processing in Sensor Networks (2004), pp. 61-70.

[Garcia01]
Garcia-Luna-Aceves, J.J. and M. Spohn, "Transmission-Efficient Routing in Wireless Networks Using Link-State Information," Mobile Networks and Applications 6, 3 (June 2001), pp. 223-238.

[Gerla99]
Su, W. and M. Gerla, "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction," Proceedings of IEEE GLOBECOM 1999 (December 1999), pp. 271-275.

[Gray04]
Gray, R. S., N. Dubrovsky, C. Masone, D. Kotz, A. Fiske, S. McGrath, C. Newport, J. Liu, and Y. Yuan, "Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms," Proceedings of the 7th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (October 2004), pp. 220-229.

[Gupta97]
Gupta, P. and P. R. Kumar, "A System and Traffic Dependent Adaptive Routing Algorithm for Ad Hoc Networks," Proceeding of the 36th IEEE Conference on Decision and Control (December 1997), pp. 2375-2380.

[Haas99]
Haas, Z. and M. Pearlman, "Determining the Optimal Configuration for the Zone Routing Protocol," IEEE Journal on Selected Areas in Communications (Ad-Hoc Networks Special Issue) 17, 8 (August 1999), pp. 1395-1414.

[Haas01]
Haas, Z. J. and M. R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," IEEE/ACM Transactions on Networking 9, 4 (August 2001), pp. 427-438.

[He04]
He, T., S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-Efficient Surveillance System Using Wireless Sensor Networks," Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (2004), pp. 270–283.

[Iwata99]
Iwata, A., C. C. Chiang, G. Pei, M. Gerla, and T. W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," IEEE Journal on Selected Areas in Communications 17, 8 (August 1999), pp. 1369-1379.

[Jacquet01]
Jacquet, P., P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," Proceedings of the IEEE International Multi-topic Conference (December 2001), pp. 62-68.

[Joa02]
Joa-Ng, M. and I. T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks," Journal of Communications and Networks 4, 1 (March 2002), pp. 14-21.

[Johansson99]
Johansson, P., T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," ACM/IEEE International Conference on Mobile Computing and Networking (August 1999), pp. 195-206.

[Johnson01]
Johnson, D. B., D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," Ad Hoc Networking, Addison-Wesley, Reading, MA, 2001, pp. 139-172.

[Ko00]
Ko, Y. and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," Wireless Networks 6 (2000), pp. 307-321.

[Kun04]
Kun, W., X. Yin-long, C. Guo-liang, and W. Ya-feng, "Power-Aware On-Demand Routing Protocol for MANET," Proceedings of the 24th International Conference on Distributed Computing Systems Workshops 7 (2004), pp. 723-728.

[Lee02]
Lee, S., W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," Mobile Networks and Applications 7, 6 (December 2002), pp. 441- 453.

[Lee03]
Lee, S., E. M. Belding-Royer, and C. E. Perkins, "Scalability Study of the Ad Hoc On-demand Distance Vector Routing Protocol," International Journal of Network Management 13, 2 (March/April 2003), pp. 97-114.

[Lee05]
Lee, K. S., S. J. Lee, and Y. K. Chung, "A Performance Comparison of On-Demand Routing Protocols for Application Data in Mobile Ad Hoc Networks," Proceedings of the 2005 Third ACIS International Conference on Software Engineering Research, Management and Applications (2005), pp. 331-339.

[Macker99]
Macker, J. P. and M. S. Corson, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," IETF Request for Comments 2501, January 1999.

[Mainwaring02]
Mainwaring, A., J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (September 2002), pp. 88-97.

[Murthy95]
Murthy, S. and J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks," Proceedings of 1st Annual ACM International Conference on Mobile Computing and Networking (1995), pp. 86-95.

[Murthy96]
Murthy, S. and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks 1, 2 (October 1996), pp. 183-197.

[Murthy04]
Murthy, C. S. R. and B. S. Manoj, Ad Hoc Wireless Networks – Architectures and Protocols, Prentice Hall, Upper Saddle River, NJ, 2004.

[Pandey05]
Pandey, A. K. and H. Fujinoki, "Study of MANET Routing Protocols by GloMoSim Simulator," International Journal of Network Management 15 (2005), pp. 393-410.

[Perkins94]
Perkins, C. E. and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Computer Communications Review 24, 4 (October 1994), pp. 234-244.

[Perkins99]
Perkins, C. E. and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (February 1999), pp. 90-100.

[Pirzada04]
Pirzada, A. A. and C. McDonald, "Trusted Route Discovery with the TORA Protocol," Proceedings of the Second Annual Conference on Communication Networks and Services Research (2004), pp. 121-130.

[Royer99]
Royer, E. M. and C. K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications 6, 2 (April 1999), pp. 46-55.

[Samar04]
Samar, P., M. R. Pearlman, and Z. J. Haas, "Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks," IEEE/ACM Transactions on Networking 12, 4 (August 2004), pp. 595-608.

[Sinha99]
Sinha, P., R. Sivakumar, and V. Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm," IEEE Journal on Selected Areas in Communications 17, 8 (August 1999), pp. 1454-1466.

[Sisodia02]
Sisodia, R. S., B. S. Manoj, and C. S. R. Murthy, "A Preferred Link-Based Routing Protocol for Ad Hoc Wireless Networks," Journal of Communications and Networks 4, 1 (March 2002), pp. 14-21.

[Tanenbaum03]
Tanenbaum, A. S., Computer Networks, Fourth Edition, Prentice Hall, Upper Saddle River, NJ, 2003.

[Toh97]
Toh, C. K., "Associativity Based Routing for Ad Hoc Mobile Networks," Wireless Personal Communications 4, 2 (March 1997), pp. 1-36.

[Werner05]
Werner-Allen, G., J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring Volcanic Eruptions with a Wireless Sensor Network," Proceedings of the Second European Workshop on Wireless Sensor Networks (January 2005), pp. 108-120.

[Yan03]
Yan, T., Y. He, and J. A. Stankovic, "Differentiated Surveillance for Sensor Networks," Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (November 2003), pp. 51-62.

[Zeng98]
Zeng, X., R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks," Proceedings of the 12th Workshop on Parallel and Distributed Simulation (May 1998), pp. 154-161.

Electronic Sources:

[Beijar02]
Beijar, N., "Zone Routing Protocol (ZRP)," http://keskus.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf, last revision 2002, last accessed May 4, 2008.

[Bradley08]
Bradley, M., "Wireless Standards, 802.11b, 802.11a, and 802.11n," http://compnetworking.about.com/cs/wireless80211/a/aa80211standard.htm, last accessed May 4, 2008.

[Clausen03]
Clausen, T., and P. Jacquet, "Optimized Link State Routing Protocol," MANET Working Group RFQ 3626, http://www.ietf.org/rfc/rfc3626.txt, last revision October 2003, last accessed May 4, 2008.

[Defense Review.com08]
DefenseReview.com, "Micro Air Vehicle: Backpackable UAV for Tactical Surveillance," http://www.defensereview.com/modules.php?name=News&file=article&sid=811, last accessed May 4, 2008.

[Infotron08]
Infotron, "Drone IT 180-5," http://www.infotron.fr/drone.php?lang=US, last accessed May 4, 2008.

[Jiang04]
Jiang, H. and J. J. Garcia-Luna-Aceves, "Performance Comparison of Three Routing Protocols for Ad Hoc Networks," www.cse.ucsc.edu/ccrg/publications/hjiang.ic3n01.pdf, last revision July 29, 2004, last accessed May 04, 2008.

[Johnson07]
Johnson, D. B., D. A. Maltz, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IIPv4," Network Working Group RFC 4728, http://www.ietf.org/rfc/rfc4728.txt, last revision February 2007, last accessed May 4, 2008.

[Karp98]
Karp, B. and H.T. Kung, "Dynamic Neighbor Discovery and Loop-Free, Multi-Hop Routing for Wireless, Mobile Networks," http://actcomm.dartmouth.edu/papers/./karp:aprl.ps.gz, last revision 1998, last accessed May 4, 2008.

[Klein08]
Klein-Berndt, L., "A Quick Guide to AODV Routing," http://w3.antd.nist.gov/wctg/aodv_kernel/aodv_guide.pdf, last accessed May 4, 2008.

[Lin04]
Lin, C., "AODV Routing Implementation for Scalable Wireless Ad-Hoc Network Simulation (SWANS)," http://jist.ece.cornell.edu/docs/040421-swans-aodv.pdf, last revision April 21, 2004, last accessed May 4, 2008.

[Nuevo04]
Nuevo, J., "A Comprehensive GloMoSim Tutorial," http://www.icis.ntu.edu.sg/wagio/campus/research/glomosim/glomoman.pdf, last revision March 2004, last accessed June 22, 2008.

[Perkins03]
Perkins, C. E., E. M. Belding-Royer, and S.R. Das, "Ad-Hoc On-Demand Distance Vector Routing," Mobile Ad Hoc Networking Working Group Internet Draft, http://moment.cs.ucsb.edu/AODV/ID/draft-ietf-manet-aodv-13.txt, last revision 17 February 2003, last accessed May 4, 2008.

[Poor08]
Poor, R. D., "Gradient Routing in Ad Hoc Networks," http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf, last accessed May 4, 2008.

[Remondo04]
Remondo, D., "Tutorial on Wireless Ad Hoc Networks," www.comp.brad.ac.uk/het-net/HET-NETs04/CameraPapers/T2.pdf, last revision July 12, 2004, last accessed May 04, 2008.

[SECAN-LAB05A]
SECAN-LAB, "Temporarily-Ordered Routing Algorithm," http://wiki/uni/lu/secan-lab/Temporally-Ordered+Routing+Algorithm.html, last revision January 30, 2005, last accessed May 4, 2008.

[SECAN-LAB05B]
SECAN-LAB, "Distributed Bellman-Ford," http://wiki.uni.lu/secan-lab/Distributed+Bellman-Ford.html, last revision March 16, 2005, last accessed May 4, 2008.

[theFreeDictionary.com08]
"Ad hoc", http://www.thefreedictionary.com/ad+hoc, last accessed November 09 2008.

[UCLA01]
UCLA Parallel Computing Laboratory, GloMoSim Manual, Version 1.2, http://pcl.cs.ucla.edu/projects/glomosim/glomomimManual.html, last revision 2001, last accessed May 04, 2008.

APPENDIX 1

GloMoSim Main Configuration Input File Sample

```
# Sample GloMoSim main configuration input file.  This file is
# the input for case 3222332.
#
# ***** GloMoSim Configuration File *****
# Glomosim is COPYRIGHTED software.  It is freely available
# without fee for education, or research, or to non-profit
# agencies. No cost evaluation licenses are available for
# commercial users. By obtaining copies of this and other
# files that comprise GloMoSim, you, the Licensee, agree to
    abide
# by the following conditions and understandings with respect to
# the copyrighted software:
#
# 1.Permission to use, copy, and modify this software and its
#   documentation for education, research, and non-profit
#   purposes is hereby granted to Licensee, provided that the
#   copyright notice, the original author's names and unit
#   identification, and this permission notice appear on all
    such
#
#   copies, and that no charge be made for such copies. Any
#   entity desiring permission to incorporate this software
#   into commercial products or to use it for commercial
    purposes
#   should contact:
#
#   Professor Rajive Bagrodia
#   University of California, Los Angeles
#   Department of Computer Science
#   Box 951596
#   3532 Boelter Hall
#   Los Angeles, CA 90095-1596
#   rajive@cs.ucla.edu
#
# 2.NO REPRESENTATIONS ARE MADE ABOUT THE SUITABILITY OF THE
#   SOFTWARE FOR ANY PURPOSE. IT IS PROVIDED "AS IS" WITHOUT
#   EXPRESS OR IMPLIED WARRANTY.
#
# 3.Neither the software developers, the Parallel Computing Lab,
#   UCLA, or any affiliate of the UC system shall be liable
#   for any damages suffered by Licensee from the use of this
#   software.
#
# $Id: config.in,v 1.32 2001/04/12 18:35:00 jmartin Exp $
```

```
#
# Anything following a "#" is treated as a comment.
#
#################################################################
    #
# The folowing parameter represents the maximum simulation time.
# The numberd portion can be followed by optional letters to
# modify the simulation time.
# For example:
#        100NS    - 100 nano-seconds
#        100MS    - 100 milli-seconds
#        100S     - 100 seconds
#        100      - 100 seconds (default case)
#        100M     - 100 minutes
#        100H     - 100 hours
#        100D     - 100 days
#

SIMULATION-TIME     600S

#
# The following is a random number seed used to initialize part
# of the seed of various randomly generated numbers in the
# simulation. This can be used to vary the seed of the
    simulation
# to see the consistency of the results of the simulation.
#

SEED                1

#
# The following two parameters stand for the physical terrain in
# which the nodes being simulated. For example, the following
# are represents an area of size 100 meters by 100 meters. All
# rang e parameters are in terms of meters.
#
# Terrain Area we are simulating.
#

TERRAIN-DIMENSIONS  (2000, 2000)

#
# The following parameter represents the number of nodes being
# simulated.
#

NUMBER-OF-NODES 49

#
#
#The following parameter represents the node placement strategy.
```

```
#- RANDOM: Nodes are placed randomly within the physical
    terrain.
#- UNIFORM: Based on the number of nodes in the simulation, the
#  physical
#  terrain is divided into a number of cells. Within each cell,
    a
#  node is placed randomly.
#- GRID: Node placement starts at (0, 0) and are placed in grid
#  format with each node GRID-UNIT away from its neighbors. The
#  number of nodes has to be square of an integer.
#- FILE: Position of nodes is read from NODE-PLACEMENT-FILE. On
#  each line of the file, the x and y position of a single node
#  is separated by a space.
#

NODE-PLACEMENT      FILE
NODE-PLACEMENT-FILE ./nodes49.input
# NODE-PLACEMENT      GRID
# GRID-UNIT           30
# NODE-PLACEMENT      RANDOM
# NODE-PLACEMENT      UNIFORM

#
# The following represent parameters for mobility. If MOBILITY
    is
# set to NO, than there is no movement of nodes in the model.
    For
# the RANDOM-DRUNKEN model, if a node is currently at position
# (x, y), it can possibly move to (x-1, y), (x+1, y), (x, y-1),
# and (x, y+1); as long as the new position is within the
# physical terrain. For random waypoint, a node randomly selects
# a destination from the physical terrain. It moves in the
# direction of the destination in a speed uniformly chosen
# between MOBILITY-WP-MIN-SPEED and MOBILITY-WP-MAX-SPEED
# (meter/sec). After it reaches its destination, the node stays
# there for MOBILITY-WP-PAUSE time period.
# The MOBILITY-INTERVAL is used in some models that a node
# updates its position every MOBILITY-INTERVAL time period. The
# MOBILITY-D-UPDATE is used that a node updates its position
# based on the distance (in meters).
#

#MOBILITY  NONE

# Random Waypoint and its required parameters.

MOBILITY RANDOM-WAYPOINT
MOBILITY-WP-PAUSE 0S
MOBILITY-WP-MIN-SPEED 2.682
MOBILITY-WP-MAX-SPEED 2.682
```

```
#MOBILITY TRACE
#MOBILITY-TRACE-FILE ./mobility.in

#MOBILITY PATHLOSS-MATRIX

# The following parameters are necessary for all the mobility
# models

MOBILITY-POSITION-GRANULARITY 0.5

##################################################################
    #
#
# PROPAGATION-LIMIT:
#   Signals with powers below PROPAGATION-LIMIT (in dBm)
#   are not delivered. This value must be smaller than
#   RADIO-RX-SENSITIVITY + RADIO-ANTENNA-GAIN of any node
#   in the model. Otherwise, simulation results may be
#   incorrect. Lower value should make the simulation more
#   precise, but it also make the execution time longer.
#
PROPAGATION-LIMIT      -111.0

#
# PROPAGATION-PATHLOSS: pathloss model
#   FREE-SPACE:
#     Friss free space model.
#     (path loss exponent, sigma) = (2.0, 0.0)
#   TWO-RAY:
#     Two ray model. It uses free space path loss
#     (2.0, 0.0) for near sight and plane earth
#     path loss (4.0, 0.0) for far sight. The antenna
#     height is hard-coded in the model (1.5m).
#   PATHLOSS-MATRIX:
#
PROPAGATION-PATHLOSS   FREE-SPACE
#PROPAGATION-PATHLOSS   TWO-RAY
#PROPAGATION-PATHLOSS   PATHLOSS-MATRIX

#
# NOISE-FIGURE: noise figure
#
NOISE-FIGURE    10.0

#
# TEMPARATURE: temparature of the environment (in K)
#
TEMPARATURE     290.0


########################################
```

```
#
# RADIO-TYPE: radio model to transmit and receive packets
#   RADIO-ACCNOISE: standard radio model
#   RADIO-NONOISE: abstract radio model
#   (RADIO-NONOISE is compatible with the current version
    (2.1b5)
#    of ns-2 radio model)
#
RADIO-TYPE              RADIO-ACCNOISE
#RADIO-TYPE              RADIO-NONOISE

#
# RADIO-FREQUENCY: frequency (in heltz)  (Identifying variable
# for multiple radios)
#
RADIO-FREQUENCY     2.4e9

#
# RADIO-BANDWIDTH: bandwidth (in bits per second)
#
RADIO-BANDWIDTH     2000000

#
# RADIO-RX-TYPE: packet reception model
#   SNR-BOUNDED:
#     If the Signal to Noise Ratio (SNR) is more than
#     RADIO-RX-SNR-THRESHOLD (in dB), it receives the signal
#     without error. Otherwise the packet is dropped.
#     RADIO-RX-SNR-THRESHOLD needs to be specified.
#   BER-BASED:
#     It looks up Bit Error Rate (BER) in the SNR - BER table
#     specified by BER-TABLE-FILE.
#
RADIO-RX-TYPE           SNR-BOUNDED
RADIO-RX-SNR-THRESHOLD  10.0
#RADIO-RX-SNR-THRESHOLD  8.49583

#RADIO-RX-TYPE           BER-BASED
#BER-TABLE-FILE          ./ber_bpsk.in

#
# RADIO-TX-POWER: radio transmition power (in dBm)
#
RADIO-TX-POWER 7.005

#
# RADIO-ANTENNA-GAIN: antenna gain (in dB)
#
RADIO-ANTENNA-GAIN  0.0

#
```

```
# RADIO-RX-SENSITIVITY: sensitivity of the radio (in dBm)
#
RADIO-RX-SENSITIVITY -91.0

#
# RADIO-RX-THRESHOLD: Minimum power for received packet (in dBm)
#
RADIO-RX-THRESHOLD -81.0

#
##############################
#

MAC-PROTOCOL         802.11
#MAC-PROTOCOL         CSMA
#MAC-PROTOCOL         MACA

#MAC-PROTOCOL         TSMA
#TSMA-MAX-NODE-DEGREE      8

#MAC-PROPAGATION-DELAY 1000NS

#
# PROMISCUOUS-MODE defaults to YES and is necessary if nodes
    want
# to overhear packets destined to the neighboring node.
# Currently this option needs to be set to YES only for DSR is
# selected as routing protocol.  Setting it to "NO" may save a
# trivial amount of time for other protocols.
#

PROMISCUOUS-MODE     YES

##############################
#
# Currently the only choice.

NETWORK-PROTOCOL    IP
NETWORK-OUTPUT-QUEUE-SIZE-PER-PRIORITY 100

#RED-MIN-QUEUE-THRESHOLD 150
#RED-MAX-QUEUE-THRESHOLD 200
#RED-MAX-MARKING-PROBABILITY 0.1
#RED-QUEUE-WEIGHT .0001
#RED-TYPICAL-PACKET-TRANSMISSION-TIME 64000NS


##############################
#
```

```
ROUTING-PROTOCOL AODV
#ROUTING-PROTOCOL    AODV
#ROUTING-PROTOCOL    DSR
#ROUTING-PROTOCOL    LAR1
#ROUTING-PROTOCOL    WRP
#ROUTING-PROTOCOL    FISHEYE

#ROUTING-PROTOCOL    ZRP
#####ZONE-RADIUS          2

#ROUTING-PROTOCOL    STATIC
#STATIC-ROUTE-FILE   ROUTES.IN


#
# The following is used to setup applications such as FTP and
# Telnet.
# The file will need to contain parameters that will be use to
# determine connections and other characteristics of the
# particular application.
#


APP-CONFIG-FILE ./myApp25-49-1S.conf


#
# The following parameters determine if you are interested in
    the
# statistics of a single or multiple layer. By specifying the
# following parameters as YES, the simulation will provide you
# with statistics for that particular layer. All the statistics
# are compiled together into a file called "GLOMO.STAT" that is
# produced at the end of the simulation. If you need the
# statistics for a particular node or particular protocol, it is
# easy to do the filtering. Every single line in the file is of
# the following format:
# Node:      9, Layer:  RadioNoCapture, Total number of
# collisions is 0
#

APPLICATION-STATISTICS          YES
TCP-STATISTICS                  YES
UDP-STATISTICS                  YES
ROUTING-STATISTICS              YES
NETWORK-LAYER-STATISTICS        YES
MAC-LAYER-STATISTICS            YES
RADIO-LAYER-STATISTICS          NO
CHANNEL-LAYER-STATISTICS        NO
MOBILITY-STATISTICS             YES
```

```
#
# GUI-OPTION: YES allows GloMoSim to communicate with the Java
# Gui Vis Tool. NO does not
#

GUI-OPTION    NO
GUI-RADIO     NO
GUI-ROUTING   NO
```

APPENDIX 2

GloMoSim Additional Input Files

```
# Sample nodes.in file (36 nodes)
# This file specifies the initial position of the network's
# nodes at the beginning of the simulation
# Format: nodeAddress  0  (x, y, z)
# Note: The zero in the second field in the format is for
# compatibility with the mobility trace format
# Note: Free Space Propagation Model used for radio waves, which
# assumes an unobstructed line of sight between node, so z value
# has no effect. That would not be the case if using the Two-Ray
# (Ground Reflection) Propagation Model.
#
0 0 (0.0, 0.0, 0.0)
1 0 (400.0, 0.0, 0.0)
2 0 (800.0, 0.0, 0.0)
3 0 (1200.0, 0.0, 0.0)
4 0 (1600.0, 0.0, 0.0)
5 0 (2000.0, 0.0, 0.0)
6 0 (0.0, 400.0, 0.0)
7 0 (400.0, 400.0, 0.0)
8 0 (800.0, 400.0, 0.0)
9 0 (1200.0, 400.0, 0.0)
10 0 (1600.0, 400.0, 0.0)
11 0 (2000.0, 400.0, 0.0)
12 0 (0.0, 800.0, 0.0)
13 0 (400.0, 800.0, 0.0)
14 0 (800.0, 800.0, 0.0)
15 0 (1200.0, 800.0, 0.0)
16 0 (1600.0, 800.0, 0.0)
17 0 (2000.0, 800.0, 0.0)
18 0 (0.0, 1200.0, 0.0)
19 0 (400.0, 1200.0, 0.0)
20 0 (800.0, 1200.0, 0.0)
21 0 (1200.0, 1200.0, 0.0)
22 0 (1600.0, 1200.0, 0.0)
23 0 (2000.0, 1200.0, 0.0)
24 0 (0.0, 1600.0, 0.0)
25 0 (400.0, 1600.0, 0.0)
26 0 (800.0, 1600.0, 0.0)
27 0 (1200.0, 1600.0, 0.0)
28 0 (1600.0, 1600.0, 0.0)
29 0 (2000.0, 1600.0, 0.0)
30 0 (0.0, 2000.0, 0.0)
31 0 (400.0, 2000.0, 0.0)
32 0 (800.0, 2000.0, 0.0)
```

```
33 0 (1200.0, 2000.0, 0.0)
34 0 (1600.0, 2000.0, 0.0)
35 0 (2000.0, 2000.0, 0.0)



# Sample Application File for 64 total nodes, 25% communicating
# pairs.
# Format: Application, addressFrom, addressTo, itemsToSend,
# startTime.  Setting itemsToSend to zero causes tcplib to
# decide the number of items to send. The size of the items is
# chosen by tcplib.  Setting startTime to zero means
# transmissions may begin as soon as the simulation starts.
#
FTP 15 46 0 0S
FTP 42 53 0 0S
FTP 0 34 0 0S
FTP 32 52 0 0S
FTP 28 23 0 0S
FTP 60 54 0 0S
FTP 6 55 0 0S
FTP 1 26 0 0S
FTP 4 8 0 0S
FTP 61 9 0 0S
FTP 48 36 0 0S
FTP 40 50 0 0S
FTP 58 10 0 0S
FTP 11 18 0 0S
FTP 37 31 0 0S
FTP 35 45 0 0S



# Sample Fisheye configuration file.  No modifications made.
# This is the standard GloMoSim file for FSR parameters as
# described below
# Format: <size of the scope>, <time out for the neighboring
# nodes>, <intra scope update interval> and <inter scope update
# interval>.
# The description of these parameters are listed below.
# <size of the scope>: this parameter specifies the scope radius
# of a node in number of hops.
# <time out for the neighboring nodes>: If a node does not hear
# from a neighbor specified by this value, the neighbor node
# will be deleted from the neighbor list.
# <intra scope update interval>: The update interval of sending
# the updates of the nodes within the scope radius.
# <inter scope update interval>: The update interval of sending
# the updates of the nodes outside the scope radius.
#
2 15S 5S 15S
```

APPENDIX 3

Results Numerical Values

GRAPH 1aP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 7.005

Ref ID: 2222

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0	0	82,551	0	0
49	461	461	564	461	461
64	140,857	141,164	198,778	140,857	136,394
81	207,463	297,863	781,725	223,538	177,068
100	672,813	1,704,449	1,352,277	0	489,578

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	21,314	268,241	591,626	98,782	17,820
49	11,202	222,932	571,327	10,871	14,680
64	133,260	1,377,817	1,251,630	230,471	197,900
81	356,065	1,576,388	1,283,221	350,898	306,432
100	80,835	2,282,226	1,606,955	0	263,788

****

GRAPH 2aP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 7.005

Ref ID: 2232

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	53	53	28,769	53	53
49	104	104	3,906	104	104
64	42,338	200,608	264,246	211,147	31,337
81	320,161	679,403	1,147,463	563,983	216,503
100	1,499,530	2,839,306	2,042,431	0	509,452

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	122,515	926,739	1,251,241	102,680	63,800
49	415,667	1,274,386	1,732,530	451,093	314,109
64	321,373	1,852,524	1,535,122	463,316	280,295
81	351,075	2,895,617	2,088,122	623,595	368,080

100	559,073	3,569,589	2,817,049	0	639,848

****

GRAPH 1bP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 8.589

Ref ID: 2322

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0	0	82,551	0	0
49	18,913	18,913	179,820	18,913	461
64	1,166,698	1,351,552	1,472,532	651,732	446,136
81	757,017	1,613,872	1,308,207	784,159	721,409
100	1,081,649	2,276,474	1,617,377	0	790,123

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	118,834	568,386	680,098	123,734	37,622
49	104,041	630,725	550,504	35,776	23,069
64	331,821	1,687,257	1,550,852	513,242	387,946
81	386,598	1,434,969	1,258,802	378,788	276,345
100	478,482	2,349,985	1,940,365	0	257,083

****

GRAPH 2bP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 8.589

Ref ID: 2332

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	15,713	17,484	28,913	15,713	12,541
49	3,906	10,801	285,200	9,118	5,316
64	1,376,765	1,500,201	2,227,655	1,277,695	394,624
81	2,284,704	3,074,343	2,626,156	2,770,452	1,509,870
100	2,175,131	3,648,441	3,122,790	0	1,143,681

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	226,771	1,780,023	1,811,235	171,101	386,458
49	563,508	1,419,550	1,687,324	622,685	371,732
64	577,680	2,034,841	1,923,382	1,201,421	442,904
81	487,989	3,026,482	1,996,284	889,993	789,265
100	753,676	3,449,783	3,361,351	0	525,759

****

GRAPH 1cP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 10.527

Ref ID: 2422

Speed: 2.682 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	264,781	338,230	424,175	336,137	160,988
49	610,216	601,423	684,734	604,389	354,238
64	1,235,207	1,916,216	1,695,526	1,361,859	944,760
81	870,863	1,666,929	1,612,903	939,305	865,149
100	1,376,552	2,570,514	2,417,083	0	824,624

Speed: 26.822 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	128,051	661,596	682,322	406,126	107,760
49	173,268	684,734	563,023	392,936	57,951
64	529,191	1,916,216	1,340,895	565,393	614,730
81	440,024	1,658,870	1,112,055	395,944	351,065
100	1,330,291	2,654,324	1,813,496	0	506,301

****

GRAPH 2cP-a
Application Bytes Received vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 10.527

Ref ID: 2432

Speed: 2.682 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	804,656	667,026	1,073,664	779,886	524,770
49	1,626,513	1,836,268	1,836,314	1,553,916	1,122,107
64	1,974,575	2,255,804	2,475,725	1,948,696	1,830,809
81	2,223,988	3,224,019	3,304,173	2,550,459	1,768,658
100	2,770,773	4,004,738	4,122,256	0	1,814,052

Speed: 26.822 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	932,961	1,919,809	1,770,976	1,197,093	994,926
49	794,976	1,836,108	1,650,293	983,256	913,862
64	952,001	2,458,534	2,215,331	1,847,044	751,933
81	911,037	2,676,772	2,334,219	1,683,992	976,788
100	1,117,478	3,934,181	3,859,002	0	1,449,005

****

GRAPH 1aP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 7.005

Ref ID: 2222

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0	0	20,638	0	0
49	92	92	113	92	92
64	23,476	23,527	33,130	23,476	22,732
81	26,118	37,233	97,873	28,096	22,134
100	69,678	171,082	135,663	0	48,963

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	5,367	67,432	147,906	24,696	4,826
49	3,060	45,199	115,003	2,174	2,936
64	22,824	230,626	210,060	38,864	32,984
81	44,847	197,062	161,347	44,387	38,304
100	8,642	228,696	161,411	0	26,698

****

GRAPH 2aP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 7.005

Ref ID: 2232

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	6	6	3,197	6	6
49	9	9	325	9	9
64	2,646	12,538	16,515	13,197	1,959
81	16,070	33,970	57,373	28,410	10,825
100	61,702	113,837	82,075	0	20,499

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	14,125	104,473	139,661	11,699	7,237
49	35,228	106,592	144,686	37,951	26,358
64	20,585	116,084	96,830	29,071	17,694
81	17,955	145,097	104,941	31,600	18,595
100	22,714	143,031	113,493	0	25,594

****

GRAPH 1bP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 8.589

Ref ID: 2322

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0	0	20,638	0	0
49	3,783	3,783	35,964	3,783	92
64	196,060	225,788	245,659	108,824	74,490
81	96,688	202,489	163,712	100,691	90,406
100	109,390	228,180	161,999	0	79,187

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	30,118	142,097	170,398	31,279	9,405
49	21,793	126,145	110,506	7,442	4,798
64	55,898	281,719	259,529	86,820	64,914
81	48,721	179,570	158,218	47,676	34,935
100	48,902	234,998	194,365	0	25,908

****

GRAPH 2bP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 8.589

Ref ID: 2332

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	1,746	1,943	3,213	1,746	1,393
49	325	900	23,873	760	443
64	87,015	94,045	139,373	80,423	24,760
81	115,356	153,797	131,607	139,268	75,494
100	88,486	146,188	125,464	0	45,855

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	25,538	197,919	202,181	19,361	43,241
49	47,356	118,408	140,995	52,323	31,289
64	36,341	127,178	120,654	75,535	27,771
81	24,999	151,614	100,429	45,215	39,780
100	31,306	138,099	135,451	0	21,383

****

GRAPH 1cP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 10.527

Ref ID: 2422

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	66,195	84,557	106,044	84,034	40,247
49	122,557	120,800	136,947	121,175	71,278
64	207,190	319,369	282,827	228,103	157,460
81	110,068	208,366	201,980	119,184	108,414
100	138,647	257,475	241,847	0	83,005

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	32,397	165,802	170,580	101,728	27,552
49	35,177	136,947	113,330	78,939	11,754
64	89,620	319,369	224,767	94,997	103,453
81	56,247	207,560	140,292	50,266	43,980
100	134,497	265,432	182,043	0	51,045

****

GRAPH 2cP-n
Application Bytes Received (Normalized) vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 10.527

Ref ID: 2432

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	89,406	74,114	119,296	86,654	58,308
49	136,413	153,022	153,026	130,156	93,509
64	124,067	141,178	154,733	122,280	114,543
81	111,644	161,414	165,347	128,241	88,508
100	111,878	160,312	165,274	0	72,886

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	105,206	213,316	196,946	133,512	111,573
49	67,239	153,009	138,152	82,504	76,530
64	60,376	153,761	138,996	116,145	47,327
81	46,821	133,975	117,357	84,591	49,212
100	45,781	157,367	154,867	0	58,408

****

GRAPH 1aD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 7.005

Ref ID: 2222

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.000	0.000	0.100	0.000	0.000
49	0.100	0.100	0.200	0.100	0.100
64	0.500	0.600	0.600	0.500	0.300
81	0.776	1.000	0.999	0.978	1.000
100	0.927	0.996	0.995	0.000	1.000

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.596	0.999	1.000	0.500	0.470
49	0.486	0.990	0.994	0.500	0.400
64	0.774	0.986	0.991	0.693	0.700
81	0.986	1.000	0.990	0.976	0.900
100	0.764	0.999	0.997	0.000	0.992

****

GRAPH 2aD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 7.005

Ref ID: 2232

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.100	0.100	0.300	0.100	0.100
49	0.100	0.100	0.200	0.100	0.100
64	0.600	0.600	1.000	0.600	0.600
81	0.994	1.000	1.000	0.992	1.000
100	0.943	0.997	0.995	0.000	0.991

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.970	0.979	0.996	0.784	0.789
49	0.972	0.974	0.997	0.990	0.994
64	0.853	0.992	0.988	0.991	0.989
81	0.894	0.998	0.995	0.991	0.994
100	0.965	0.999	0.993	0.000	1.000

****

GRAPH 1bD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 8.589

Ref ID: 2322

Speed: 2.682 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	0.000	0.000	0.100	0.000	0.000
49	0.300	0.300	0.800	0.300	0.100
64	0.901	0.996	0.999	0.997	1.000
81	0.954	0.998	0.999	0.969	0.996
100	0.985	0.996	0.998	0.000	0.996

Speed: 26.822 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	0.798	1.000	0.997	0.791	0.700
49	0.777	1.000	0.995	0.861	0.590
64	0.766	0.997	0.995	0.976	0.698
81	0.895	0.999	0.995	0.984	0.870
100	0.960	1.000	0.999	0.000	0.988

****

GRAPH 2bD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 8.589

Ref ID: 2332

Speed: 2.682 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	0.300	0.400	0.400	0.300	0.300
49	0.200	0.500	0.798	0.400	0.300
64	0.970	0.998	0.999	0.992	0.999
81	0.986	0.999	0.998	0.993	1.000
100	0.977	0.998	0.995	0.000	0.998

Speed: 26.822 m/s

Nodes	B-F	AODV	LAR1	WRP	FISH
36	0.988	1.000	0.993	0.990	0.870
49	0.987	0.999	0.997	0.991	0.993
64	0.994	1.000	0.997	0.974	0.996
81	0.932	0.998	0.995	0.971	0.988
100	0.964	0.999	0.992	0.000	0.977

****

GRAPH 1cD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 10.527

Ref ID: 2422

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.600	0.700	0.900	0.600	0.400
49	0.996	0.995	1.000	0.994	0.897
64	0.992	1.000	0.999	0.991	1.000
81	0.990	1.000	0.997	0.981	0.998
100	0.991	0.998	1.000	0.000	0.988

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.893	0.999	1.000	0.995	0.990
49	0.908	1.000	0.992	0.996	0.897
64	0.940	1.000	0.989	0.995	0.989
81	0.942	0.999	0.989	0.959	0.998
100	0.949	1.000	0.997	0.000	0.998

****

GRAPH 2cD
App Bytes Rcvd/App Bytes Sent vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 10.527

Ref ID: 2432

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.900	1.000	1.000	0.900	0.800
49	0.991	1.000	1.000	0.995	1.000
64	0.994	0.999	1.000	0.995	1.000
81	0.993	0.999	0.998	0.994	1.000
100	0.987	0.998	0.998	0.000	0.991

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	0.985	1.000	0.999	0.995	0.937
49	0.971	1.000	0.996	0.981	0.998
64	0.978	0.999	0.994	0.988	0.990
81	0.966	0.998	0.993	0.996	0.991
100	0.970	1.000	0.996	0.000	0.991

****

GRAPH 1aR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 7.005

Ref ID: 2222

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	74	7,436	18,948	5,793
49	2,940	105	12,437	27,339	7,888
64	3,840	175	15,794	38,983	10,300
81	4,860	1,310	32,847	53,664	13,040
100	6,000	9,393	69,240	0	16,095

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	880	6,192	28,198	5,795
49	2,940	2,284	9,438	42,300	7,885
64	3,840	4,431	13,528	59,673	10,302
81	4,860	10,719	20,664	79,846	13,036
100	6,000	15,459	31,818	0	16,094

****

GRAPH 2aR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 7.005

Ref ID: 2232

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	169	17,073	18,946	5,795
49	2,940	244	28,508	27,346	7,885
64	3,840	435	35,176	38,953	10,304
81	4,860	5,547	62,396	53,726	13,037
100	6,000	29,919	154,528	0	16,097

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	3,753	7,545	28,166	5,794
49	2,940	7,762	17,223	42,344	7,890
64	3,840	10,248	24,723	59,680	10,301
81	4,860	17,350	46,217	79,781	13,040
100	6,000	31,045	73,563	0	16,101

****

GRAPH 1bR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 8.589

Ref ID: 2322

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	76	5,230	19,910	5,793
49	2,940	159	12,179	29,266	7,885
64	3,840	7,490	17,517	42,363	10,301
81	4,860	11,033	15,589	57,958	13,037
100	6,000	11,310	34,675	0	16,092

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	2,175	4,270	30,220	5,794
49	2,940	3,475	5,541	44,841	7,885
64	3,840	6,273	11,001	62,737	10,301
81	4,860	8,686	15,944	82,994	13,038
100	6,000	13,166	27,311	0	16,095

****

GRAPH 2bR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 8.589

Ref ID: 2332

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	179	14,692	19,864	5,796
49	2,940	341	21,892	29,239	7,886
64	3,840	10,198	37,761	42,488	10,301
81	4,860	23,761	37,579	58,283	13,040
100	6,000	30,567	129,260	0	16,093

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	2,550	8,504	30,114	5,796
49	2,940	5,918	14,132	44,808	7,890
64	3,840	8,746	19,625	62,726	10,300
81	4,860	22,684	29,547	82,940	13,040
100	6,000	26,183	61,552	0	16,099

****

GRAPH 1cR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 10
Radio TX Power (dBm) = 10.527

Ref ID: 2422

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	381	3,516	21,172	5,796
49	2,940	2,235	3,121	31,874	7,886
64	3,840	6,136	8,191	45,560	10,297
81	4,860	8,489	9,694	61,508	13,037
100	6,000	10,143	16,436	0	16,097

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	1,627	3,339	32,265	5,792
49	2,940	3,129	3,815	47,409	7,886
64	3,840	6,391	7,131	65,305	10,298
81	4,860	7,570	10,380	85,591	13,038
100	6,000	9,484	15,959	0	16,096

****

GRAPH 2cR
Routing Control Packets Transmitted vs. Number of Nodes

Mobility Pause (s) = 0S
Percent Comm Pairs = 25
Radio TX Power (dBm) = 10.527

Ref ID: 2432

Speed: 2.682 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	2,686	9,804	21,186	5,795
49	2,940	8,001	9,473	31,968	7,886
64	3,840	11,603	14,907	45,717	10,300
81	4,860	21,179	20,810	61,923	13,037
100	6,000	29,533	53,451	0	16,098

Speed: 26.822 m/s

Nodes	_B-F	AODV	LAR1	_WRP	FISH
36	2,160	2,883	5,865	32,233	5,794
49	2,940	5,138	10,448	47,496	7,887
64	3,840	9,466	14,627	65,307	10,303
81	4,860	14,535	24,835	85,659	13,037
100	6,000	22,388	46,841	0	16,098

APPENDIX 4

Acronyms

ABDR	Application Byte Delivery Ratio
AODV	Ad Hoc On-Demand Distance Vector
B-F	Bellman Ford
CBR	Constant Bit Rate
CSMA/CA	Carrier-Sense Multiple Access/Collision Avoidance
dBm	decibel milliwatt
DSR	Dynamic Source Routing
DVR	Distance Vector Routing
FSR	Fisheye State Routing
FTP	File Transfer Protocol
GNU	GNU is Not Unix
GPS	Global Positioning System
INU	Inertial Navigation Unit
LAR	Location-Aided Routing
LSR	Link State Routing
MANET	Mobile Ad Hoc Network
mph	Miles Per Hour
mps	Meters Per Second
UAV	Unmanned Air Vehicle
WRP	Wireless Routing Protocol
WSN	Wireless Sensor Network

VITA

Pedro Lopez-Fernandez has Bachelor of Science and Master of Science degrees in Aerospace Engineering from the University of Florida, awarded in 1989 and 1991, respectively. He expects to receive a Master of Science degree in Computer and Information Sciences from the University of North Florida in December 2008. Dr. Sanjay P. Ahuja and Dr. Zornitza G. Prodanoff are serving as Pedro's thesis advisors. Pedro is currently employed as an Aerospace Engineer by the United States Air Force, and has been with the Air Force for 17 months. Previously, Pedro worked as an Aerospace Engineer for AIC (Aerospace Integration Corporation) for 6 months and the United States Navy for approximately 13 years.

Pedro has on-going interests in computational fluid dynamics, structural analysis, parallel programming, and networking. Pedro's academic work has included use of C, Java, FORTRAN, and SQL. Pedro is fluent in Spanish and has a working knowledge of French.