Benchmarking Wireless Network Protocols: Threat and Challenge Analysis of the AeroRP

Dan S. Broyles

Submitted to the graduate degree program in Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas School of Engineering in partial fulfillment of the requirements for the degree of Master of Science

Thesis Committee:

____

Dr. James P.G. Sterbenz: Chairperson

____

Dr. Gary J. Minden

____

Dr. Bo Luo

____

Date Defended

The Thesis Committee for Dan S. Broyles certifies
that this is the approved version of the following thesis:

Benchmarking Wireless Network Protocols: Threat and Challenge Analysis of the AeroRP

Committee:

____

Chairperson

____

____

____

Date Approved

Abstract

To accommodate the unique conditions of mobile wireless networks, numerous protocols have been designed. Protocols are initially tested through simulation software, but often under non-realistic conditions, using simple or even ideal wireless environments not usually found in the real world. Without challenges and channel impairments, such simulations cannot accurately determine the advantages and disadvantages of the protocol nor can a reliable comparison be made between the performance of any two protocols. New protocols must be tested in a manner consistent with legacy protocols so they can be accurately compared and improved upon. The contributions of this thesis are a set of models that can create more realistic and challenging simulations, including a 3-D implementation of the Gauss-Markov mobility model, and a set of benchmarks that can be used to test the strengths and weaknesses of wireless routing protocols. These benchmarks are then applied to several MANET protocols including AODV, DSR, OLSR, DSDV, and AeroRP that is part of the Aero protocol stack developed at The University of Kansas. AeroRP outperforms the traditional MANET routing protocols in benchmarks that involve either highly-dynamic networks or disruptions in connectivity.

Acknowledgements

Thanks goes out to my wife and daughters, who have also had very busy lives and school work, for being supportive and understanding. And to my son, who joined the Marines to help preserve the blessings of Liberty we enjoy. And I would like to thank to my comittee members and the insightful Resilinets team at the University of Kansas, particularly Hemanth Narra for his help troubleshooting the AeroRP protocol in ns-3.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

Traditional mobile ad-hoc network (MANET) routing protocols such as DSR [3] and AODV [4] are designed to transmit packets only after complete end-to-end routes are known. This can be a problem in mobile ad-hoc networks, in which a complete route to the destination may not always exist, and connectivity can be sporadic. New protocols are being developed to handle these unique challenges. The Aero protocol stack [5] under development at The University of Kansas aims to improve throughput and packet data rate in highly-dynamic aeronautical telemetry environnments through the use of TCP/IP-friendly transport and network protocols and a location-aware adaptive routing algorithm. These protocol layers work together to produce a solution that is delay and disruption tolerant.

To compare the AeroRP routing algorithm [6] against the other MANET protocols, a set of simulation tests are needed. The problem with past network simulations has been the overly simplistic and unrealistic nature of the simulated channel environment, which often do not take physical layer imperfections and challenges into account. As a result, the simulation results do not accurately reflect the protocol's true performance characteristics. A more realistic channel environment must be modeled into the network simulations if the results are going to be useful in determining strengths and weaknesses. Furthermore, a representative set of such tests can be used to benchmark the performance of MANET protocols and provide a more solid basis for comparing one protocol against another. The primary contributions of this thesis are the implementation of more realistic and challenging physical layer impairment simulation models, a 3-D implementation of the Gauss-Markov mobility model, and the proposal and implementation of a set of benchmarks that utilize these models to test and compare AeroRP against a number of legacy wireless routing protocols.

This work is motivated by a desire to test AeroRP through a systematic simulation method and to compare its performance with that of traditional wireless routing protocols such as OLSR [7], DSDV [8], AODV, and DSR [9]. The idea to develop a set of benchmarks was partly inspired by the benchmark software available to the CPU and video graphic card industry. Since each protocol has its own strengths and weaknesses, it seems appropriate to run each protocol through every benchmark. Finding weaknesses where they are not expected might lead to further refinements and improvements in the algorithm. Also, finding strengths where they are not expected might allow the protocol, or certain parts of it, to be exploited for other purposes than those initially intended.

Benchmarks have been used in a variety of platforms to compare the performance of two or more test subjects. Due to a wide variety of possible configurations and system complexities, benchmarks provide a mechanism for comparing performance on equal grounds. Benchmarks are often used to test the strengths and weaknesses between test subjects in complex systems, in which complex interactions make traditional comparisons unreliable. For example, a set of benchmarks is used to compare the performance of one video card against another. The dependant variable in this case is frame rate, or frames per second (FPS). To make the test fair, the video cards are tested using the same computer setup, the same motherboard, hard drive, CPU, RAM, and even power supply. An example set of results from one set of benchmarks is shown in Figure 1.1.

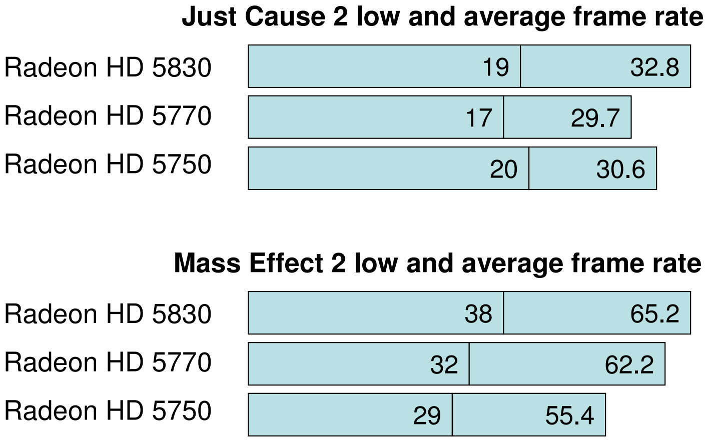


Figure 1.1. Comparison of 3 Video Cards (Adapted from [10])

From the benchmark performance results graph, it is clear to see that the Radeon 5770 outperforms the 5750 in the game Mass Effect 2, but underperforms the 5750 in the game Just Cause 2. The strengths and weaknesses of each can be attributed to the architecture and inner workings of each card. One card may have a faster GPU, but the other card might have faster internal memory, or twice as much memory, or larger cache. As a result of studying the benchmark results, users might choose the video card that most closely meets their needs.

It is expected that the same kind of comparisons can be made with wireless network protocols. Each protocol has a unique architecture, functional algorithms, and configuration options. They are used in unpredictable networks with very complex interactions. It is not always clear how an architectural decision in one protocol will affect its performance, and under what network or environmental conditions the benefit will be manifest. Based upon the results of these benchmarks, a network engineer may be able to select a routing protocol that meets a particular need or performance requirement. Additionally, researchers may use the results to identify weaknesses in a protocol algorithm and choose to optimize or improve that algorithm.

1.1 Contributions

The contributions of this thesis are:

1. Description and implementation of more realistic and challenging physical layer impairment simulation models, including a location-based impairment and sources of persistent interference such as jammers.

2. Description and implementation of a 3-D Gauss-Markov mobility model for more realistic node movement.

3. Definition of fundamental tests that utilize these more realistic and challenging models to benchmark MANET routing protocol performance.

4. Implementation of these benchmarks to compare the performance of AeroRP against legacy MANET routing protocols.

1.2 Organization

This thesis is organized as follows. Chapter 2 gives background information on the kinds of unrealistic mobility models and physical channel conditions usually found in simulation environments along with some challenges and fundamental characteristics of MANET routing algorithms. Chapter 3 discusses several enhancements to ns-3 mobility and channel models which bring more sophistication and realism to the simulations, allowing researchers to get a much better idea of how these networks will behave in the real world. A set of benchmarks is proposed in chapter 4 that leverage these more realistic and challenging models, making a more accurate assessment of strengths and weaknesses possible. Chapter 5 implements these benchmarks, comparing the performance of five MANET routing protocols including AeroRP. Chapter 6 concludes this thesis with key conclusions from the research and benchmark results and some logical next steps.

Chapter 2

Background and Related Work

Computer simulations are an important first step in testing the behavior and performance characterists of a new protocol. Network simulators such as ns-3 do a very good job of simulating the mechanics of mobile ad hoc network routing algorithms, but have tended to ignore many of the physical layer imperfections that can truly test the abilities of those protocols in dynamic wireless environments. System performance under real world conditions can be dramatically different than simulations using ideal and unrealistic conditions [11]. Unrealistic node movement and simple propagation models provide misleading performance results.

In Section 2.1 of this chapter, we will discuss problems with the memoryless mobility models that are commonly used in MANET simulations. Propagation loss models and the absence of slow fading models in ns-3 are covered in Section 2.2. Also, some challenges of designing a MANET routing protocol are discussed in Section 2.3. And several fundamental routing algorithms are described in Section 2.4.

Much of the background work has been completed by research in the Information and Telecommunication Technology Center (ITTC) department of The

University of Kansas. Section 3.1 covers recent ns-3 development work to produce a 3-dimensional Gauss-Markov mobility model which provides a more suitable model for simulating the movement of airborne network nodes [12]. A comprehensive challenge and attack simulation framework was introduced in [13] which includes some discussion of wireless challenges. Further ns-3 development associated with this work includes a more detailed discussion of how to simulate wireless threats and challenges such as jammers and signal fading, which is discussed in Section 3.2.

This more sophisticated simulation environment provides the basis for a battery of tests to compare the performance of several routing protocols used in mobile ad hoc networks. Among these protocols is one under development at The University of Kansas called AeroRP. The Aero protocol stack is presented in [5] and is discussed briefly in Section 4.6.5.

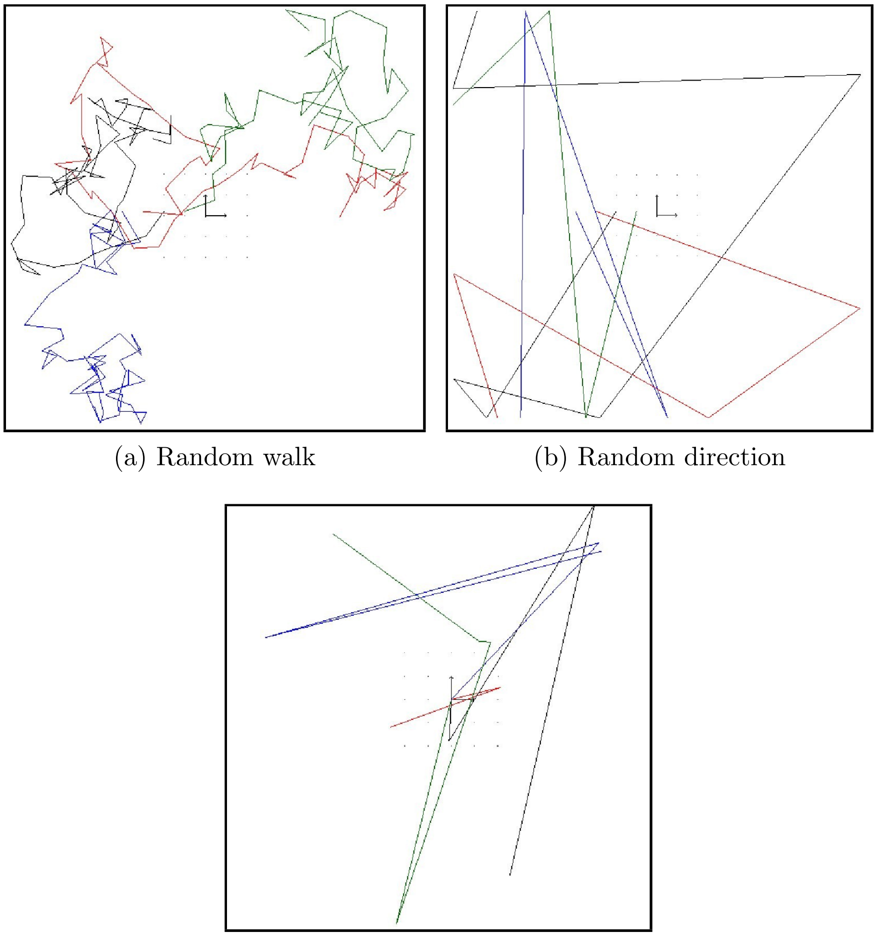
2.1 Mobility Models

Highly dynamic supersonic airborne networks provide a challenging environment for mobile ad hoc networks. High mobility, limited bandwidth and transmission range, unreliable noisy channels, intermittent connectivity, and multihop routing create a harsh environment for communications. There are several ways in which simulations of such an environment can be overly simplistic and misleading. One reason why past simulations in ns-3 and other simulator applications are unrealistic is that they use synthetic, memoryless, and random mobility models to simulate the movement of network nodes. Such models include Random Waypoint, Random Direction, and Random Walk, which are the most common mobility models used by researchers in simulating MANETs [14] [15]. Since they are memoryless, simulations using these mobility models exhibit unnatural movements with abrupt and often extreme changes in velocity and direction, uncharacteristic of mobile airborne nodes. The simulation performance of a MANET is affected by the mobility model used, and the difference between actual and simulated performance can be significant. In the case of ns-3, the existing mobility models also lacked support for 3-dimensional positioning and movement, thus making a realistic test of airborne nodes nearly impossible.

2.1.1 Memoryless Models

In Random Walk, each node is given a random trajectory and travels along that trajectory for a fixed period of time or a fixed distance as shown in Figure 2.1a. When nodes reach the limits of the 2-dimensional boundary, they bounce off in a new direction mirroring the previous direction and velocity. Figure 2.1b shows an example of the Random Direction model, in which nodes travel on a random trajectory until they reach the boundary, at which time they pause for a random period of time and head off in a new random direction and speed. In the Random Waypoint model, as illustrated in Figure 2.1c, each node travels to a random waypoint (x, y coordinate), pauses for a period of time, and then heads off to another waypoint. The waypoints, node speeds, and pause times are modeled as uniformly distributed random variables.

These simple synthetic mobility models do not mimic the motion of airborne nodes very well. The nodes undergo sudden changes in speed and direction at random, which is uncharacteristic of real objects (e.g. aircraft).



(a) Random walk

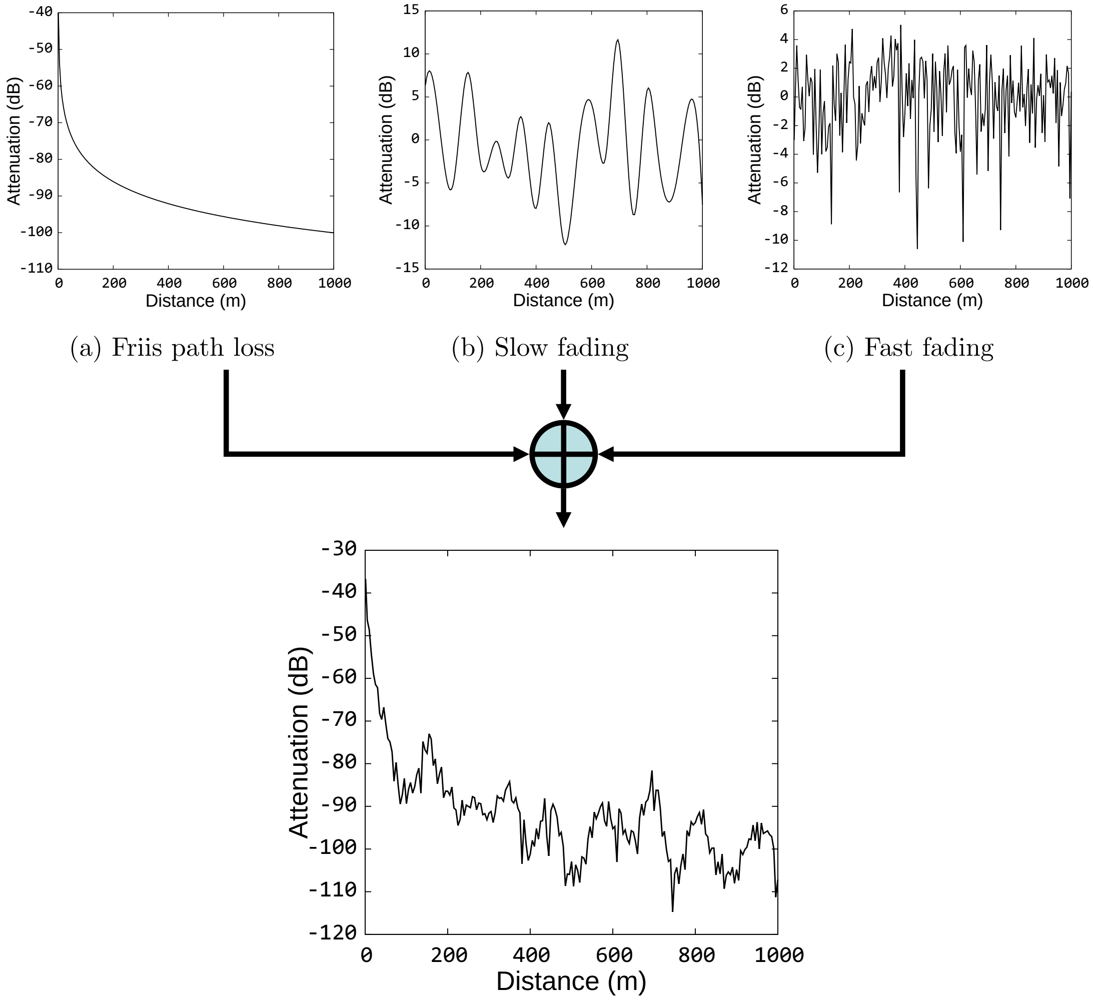
(b) Random direction

(c) Random waypoint

Figure 2.1. ns-3 memoryless mobility models

2.2 Propagation Loss Models in ns-3

Ns-3 uses a flexible approach to modeling propagation losses; one does not need hundreds of models to perfectly simulate every possible wireless environment. Instead, propagation loss models can be chained together to get the desired effect. For example, by chaining the Log-Distance and the Nakagami loss models in a wireless channel, one gets the properties of both models: a path loss model based upon distance and a fast fading model based upon statistics. This concept is illustrated in Figure 2.2. The summation of several simple propagation loss models yields a complex model that can better approximate real world channel conditions. This technique is used in most of the benchmark simulations in this research.



(a) Friis path loss

(b) Slow fading

(c) Fast fading

(e) Composite loss model

Figure 2.2. Composition of realistic model (Adapted from Fig 5.1 of [1])

The problem, however, is that ns-3 does not have slow-fading propagation loss models to simulate localized scattering or shadowing. There are basic distance-based loss models such as Friis, Log-Distance, and Cost231 propagation models.

Furthermore, there are a couple fast-fading statistical models such as Nakagami and Jakes. But it is difficult to simulate an actual impairment event at a specific location within the simulation using the existing models. To simulate an impairment such as rain fade or any other line-of-sight obstruction that affects a specific link requires a loss model with physicality.

2.3 Routing Design Challenges

Growing from a need to handle dynamic and unpredictable environments found on the battlefield, mobile ad hoc networks find an increasing number of applications. Some common applications are discussed in [2] including mobile conferencing, emergency services, home networking, embedded computer applications, sensor networks, automotive, and personal area networks. MANETs are being considered in many situations where a temporary, resilient, and autonomous network is needed. Herein lies the challenge: to determine efficient routes for a dynamic, unpredictable network, while keeping packet overhead and delay down, and throughput high.

The question is not ”Which MANET routing algorithm is the best”, but rather ”Which MANET routing algorithm best meets the requirements of a particular application?” It depends upon the importance assigned to the various performance criteria for each application. Cumulative throughput, protocol overhead, delay, security, datarate, power consumption, and delay tolerance are a few of the metrics that may be considered when selecting or designing a MANET routing protocol. There are performance tradeoffs between the different criteria. For example, a protocol that is designed around low hop count may have lower delay, but may also tend to choose routes with weaker signal strength resulting in higher packet loss.

2.3.1 Shared Medium

Mobile wireless networks are subject to environmental challenges not found in wired networks. The shared wireless medium is unpredictable, subject to signal fading, complex propagation loss and delay characteristics, friendly and unfriendly interference, and evesdropping. Mobility further implies that nodes may not always be in range of each other, and routes may change regularly, or even disappear entirely. One MANET topology may have constant connectivity and long-lasting routes, whereas a very dynamic topology may have route segments available for only a few seconds and may not be able to wait for the routing algorithm to converge upon a complete route. MANETs must configure themselves and automatically adjust to these environmental conditions.

2.3.2 Transport Protocol

Traditional TCP assumes that all packet drops are due to congestion. This assumption works well enough for wired networks, where routes are stable and signals are generally strong. But TCP is very problematic in wireless networks, where routes change and signals fade regularly. With this single assumption, each packet loss is handled by cutting transmission rates in half, reducing throughput when it should be implementing mechanisms to counter interference or reroute the traffic. MANET routing algorithms need to utilize a transport protocol that is designed with wireless in mind, like the Aero transport protocol (AeroTP). At the time of this writing however, AeroTP is not available in ns-3, otherwise it would be integrated into these benchmarks.

2.3.3 Protocol Performance

The performance of any routing algorithm is dependent upon the interactions of the individual mechanisms within that protocol and how they interact with neighboring nodes. A basic understanding of the different mechanisms will guide us in predicting how each routing protocol may perform in each benchmark. Some of the basic functions a MANET routing protocol must perform are:

- Neighbor discovery
- Selecting neighbor for forwarding packets
- Building and managing the topology
- Establish routes

2.3.3.1 Neighbor Discovery

A MANET node must determine which other nodes are within its transmission range and use that knowledge to perform various activities related to routing. Common methods for neighbor discovery include transmitting beacon packets and promiscuous snooping. MANET nodes must keep their neighbor list updated regularly. Some protocols transmit location information periodically to their neighbors to aid in routing algorithms.

2.3.3.2 Selecting the Best Neighbor

MANET nodes must decide which neighbor to forward packets to on route to the destination. Based upon the algorithm used, the best neighbor may be chosen based upon the fewest hops, available node power, signal strength, or even its location and speed vector relative to the destination.

2.3.3.3 Managing Topology

Network nodes must use mechanisms to find out what other nodes are in the network and how they are connected to each other. This information is needed for routing calculations and must be kept current or else incorrect routing decisions can be made and performance will suffer. One challenge is knowing how often to refresh the topology. If it is done too often, the overhead affects network resources; and if it is not updated often enough, routes become stale. In either case, overall performance can suffer.

2.4 Routing Algorithms

Calculating routes is the primary purpose of a routing protocol and is accomplished through the use of various routing algorithms as described in the following sections.

2.4.1 Proactive

Often referred to as a table-driven algorithm, each node in proactive routing protocols keeps a current and complete set of routes to every other node in the network [16]. Updates are disseminated to the other nodes with waves of control packets. These algorithms tend to produce lower delay over reactive protocols at the expense of higher control traffic overhead. The more dynamic the topology, the more frequent the update control packets need to be sent. If the protocol sends updates at set intervals that are too slow, the routes will become stale and packets may be lost. If the interval is shortened, then control packet overhead will increase accordingly. Thus, proactive protocols generally perform poorly in large or highly-dynamic networks, however they can provide better quality of service over reactive protocols with the right network conditions. In proactive algorithms, packets are forwarded only after a route has been established to the target node, but that route should already exist in the source node's routing table before it is needed. The requirement to keep routing tables current at all times implies that proactive algorithms are not designed for power constrained applications.

2.4.2 Reactive

Reactive routing protocols [17] are sometimes referred to as source-initiated protocols. The fundamental premise is that nodes will only construct routes when a source node requests a route to a destination node. If the route is not already cached, the source node floods the network with route request control packets. Route maintenance packets are used to keep the routes current as long as the route is needed. The overall delay is likely to be higher than a proactive algorithm, but we should expect fewer control packets resulting in better efficiency and lower power consumption than with proactive protocols. Of course, the actual efficiency and performance will depend upon the network topology, traffic patterns, and how the algorithm controls the flooding process. Much like proactive algorithms, packets are forwarded only after the network has converged upon a route to the destination node, but reactive protocols only make the effort to find a route when needed.

2.4.3 Link State

Proactive and reactive routing algorithms are concerned with "when" a route gets calculated. However, link state, distance vector, and geographic routing algorithms tell us "how" a route is calculated. In link state [18] routing protocols, each routing node in the network creates a graph of network connectivity by broadcasting a list of its direct neighbors to each of its neighbors. Each node's neighbor list propagates through the network until all nodes know the neighbors of all other nodes. At that point, a node will use the Dijkstra or similar algorithm to calculate which neighbor is the next logical hop in the routes to every other node in the network and places this information in a forwarding table.

2.4.4 Distance Vector

Distance vector routing [19] is less complex and has less overhead than link state. Each node's neighbor list is only sent to each neighbor. Unlike link state, it does not propagate to each node in the network, hence, less overhead. This also means that nodes do not know the complete route to every other node. Instead, distance vector uses two other pieces of information: the next hop and the distance (number of hops) to the destination. The underlying Bellman-Ford algorithm suffers from routing loops and the associated "count-to-infinity" problem, and does not scale to large networks.

2.4.5 Geographic

Routing algorithms that use node location information to determine routes are considered geographic or location-aware. These protocols can use GPS technology or they may get location information through control messages from neighboring nodes [20]. One feature of geographic routing algorithms is that they do not need to know the full path to the destination, but can transfer packets to nodes that are closer to the destination or to nodes that will come within range of the destination node sooner.

2.4.6 Routing in Delay-Tolerant Networks

Recent interest in delay-tolerant networks was sparked by the desire to design an inter-planetary network (IPN). Delay and disruption-tolerant networks lack continuous end-to-end paths [21]. Traditional routing algorithms such as those classified as proactive, reactive, link state, and distance vector are unable to forward packets in such networks since a complete end-to-end path may not be available. These networks require routing algorithms in which nodes can store data until a progressive connection can be made, or until the node itself comes within transmission range of the destination. This ability is called “store-and-haul” [22], or ferrying, and is used in conjunction with a geographic routing algorithm in AeroRP.

Chapter 3

Realistic Simulation Models

Memoryless, 2-dimensional mobility models are unrealistic and inadequate for simulating airborne networks. Combat, stealth, and reconnaissance aircraft do not always travel in straight lines; yet they do not wander around randomly either. A 3-dimensional mobility model that incorporates both memory and randomness provides more realistic node movement for these applications. Furthermore, wireless networks face threats from interference and jamming, which have been overlooked in past simulation models. This chapter is organized as follows.[1] Section 3.1 details the Gauss-Markov mobility model, including fundamental formulas and use of the α tuning parameter. Section 3.2 describes a novel implementation of a slow fading loss model that has both location and mobility, and finishes with a method used to simulate jammers in the wireless channel environment.

[1] This chapter is based on [12] and [13].

3.1 Gauss-Markov Mobility Model

Aware of the problems with the existing ns-3 mobility models, a 3-dimensional version of the Gauss-Markov mobility model was developed and has been incorporated into the ns-3 release. The model itself is designed to mitigate the problems associated with random memoryless models that produce straight-line node movement interrupted by wild course alterations [23]. The Gauss-Markov mobility model is a synthetic mobility model that combines both random movement with memory to produce movement that is more natural than the memoryless mobility models previously found in ns-3. This model extends the standard 2-dimensional Gauss-Markov model to three dimensions so it can be applied to airborne nodes.

The Gauss-Markov mobility model is a relatively simple memory-based model with a single tuning parameter, alpha (α), which determines the amount of memory and variability in node movement. Other parameters have a significant impact on the dynamics and characteristics of the Gauss-Markov mobility model as well as the selection of alpha. These include the choice of Gaussian distribution standard deviation, average speed, and time step.

In the 3-dimensional implementation of the Gauss-Markov model, each mobile node is assigned an initial speed and direction, as well as an average speed, direction, and pitch. At set intervals of time, called the time step, a new speed, direction, and pitch are calculated for each node. Nodes follow their course until the next time step. This cycle repeats through the duration of the simulation.

The new speed, direction, and pitch parameters are calculated as follows [14]:

$$\begin{aligned} s_n &= \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \\ \tau_n &= \alpha \tau_{n-1} + (1-\alpha)\bar{\tau} + \sqrt{(1-\alpha^2)}\tau_{x_{n-1}} \\ p_n &= \alpha p_{n-1} + (1-\alpha)\bar{p} + \sqrt{(1-\alpha^2)}p_{x_{n-1}} \qquad (3.1) \end{aligned}$$

where α is the tuning parameter, $\bar{s}$, $\bar{\tau}$, and $\bar{p}$ are the average speed, direction, and pitch parameters, respectively. $s_{x_{n-1}}$, $\tau_{x_{n-1}}$, and $p_{x_{n-1}}$ are random variables from a Gaussian (normal) distribution that provide a degree of randomness to the new speed, direction, and pitch values.

Note that these formulæ represent basic 3-dimensional node movement. The objective of this model is to add realism to the 3-dimensional node movement while limiting the algorithm's complexity. It is not necessary to model the various flight controls, like the rudder, flaps, ailerons, angle of bank, etc. It is sufficient to model the aircraft movement itself using the Gauss-Markov algorithm, for which we assume the direction and pitch variables represent the actual angles at which the aircraft is moving.

After calculating these variables, the algorithm must determine a new velocity vector and send that information to the ns-3 constant velocity helper, in which the new node location is actually calculated. Assuming the direction and pitch variables are given in radians, the velocity vector $\bar{v}$ is calculated as:

$$\begin{aligned} v_x &= s_n \cos(\tau_n)\cos(p_n) \\ v_y &= s_n \sin(\tau_n)\cos(p_n) \\ v_z &= s_n \sin(p_n) \qquad (3.2) \end{aligned}$$

Special Case: $\alpha = 0$

When α is zero, the model becomes memoryless; the new speed and direction are based completely upon the average speed and direction variables and the Gaussian distribution. The previous values are not considered.

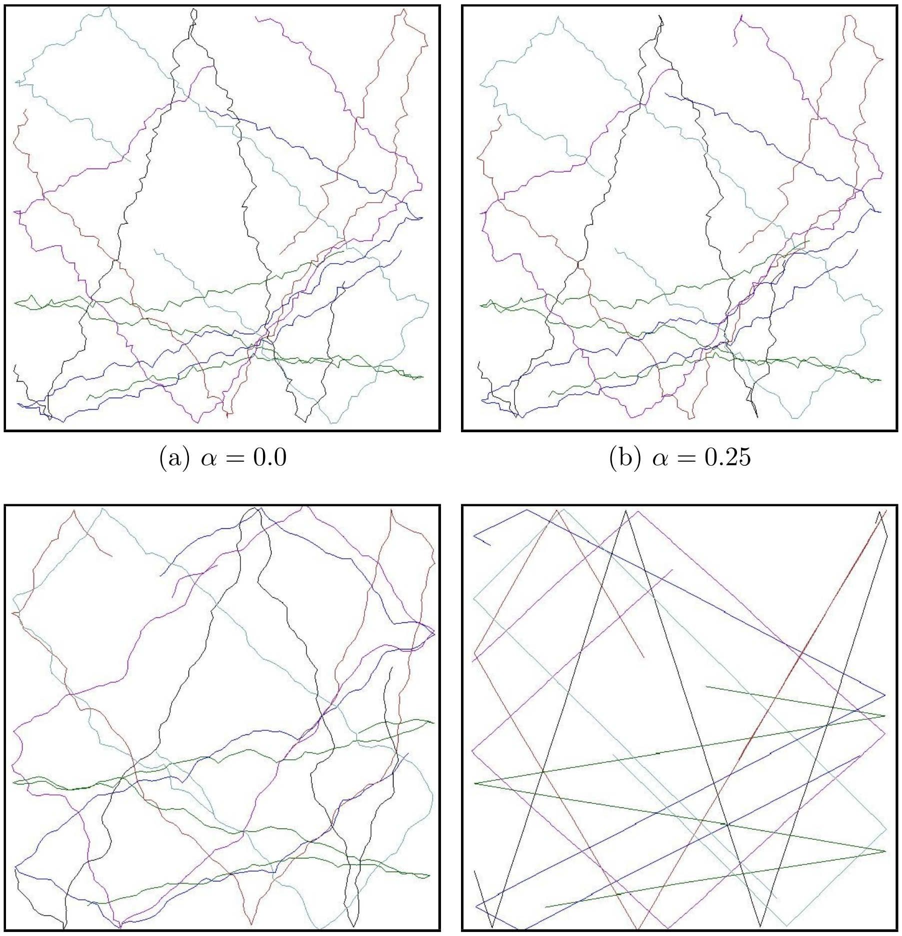
$$\begin{aligned} s_n &= \bar{s} + s_{x_{n-1}} \\ \tau_n &= \bar{\tau} + \tau_{x_{n-1}} \\ p_n &= \bar{p} + p_{x_{n-1}} \end{aligned} \tag{3.3}$$

Special Case: $\alpha = 1$

When α is 1, movement becomes predictable, losing all randomness. The new direction and speed values are identical to the previous direction and speed values. In short, the node continues in a straight line.

$$\begin{aligned} s_n &= s_{n-1} \\ \tau_n &= \tau_{n-1} \\ p_n &= p_{n-1} \end{aligned} \tag{3.4}$$

As Figure 3.1 illustrates, setting α between zero and one allows for varying degrees of randomness and memory. In addition to α, the dynamics of the Gauss-Markov mobility model are greatly influenced by other variables such as the time step, the selection of the average speed and direction, and the mean and standard deviation chosen for the Gaussian random variables. For example, choosing a standard deviation on the Gaussian distribution governing the direction that is much larger than the average direction generates a very different movement pattern than if the standard deviation and average direction values are similar.



(a) $\alpha = 0.0$

(b) $\alpha = 0.25$

(c) $\alpha = 0.85$

(d) $\alpha = 1.0$

Figure 3.1. Gauss-Markov model with $\alpha = 0.0$, 0.25, 0.85, and 1.0

3.2 Wireless Challenges

It was noticed that none of the propagation loss models in ns-3 worked well in situations with moving localized scattering, such as with rain fade, nor with stationary localized shadowing, as with buildings and other obstructions. A propagation loss model that statistically simulates scattering and shadowing can be written in ns-3, but when modeling specific challenges, it is important to control where the impairments are located and how they are moving. For example, we may want to simulate a rain-fade event moving between two network nodes, or we may want to simulate the signal shadows caused by a number of buildings between network nodes.

3.2.1 Moving Propagation Loss Model

To this end, we created the `MovingPropagationLossModel`, a new ns-3 propagation loss model that includes a mobility model parameter and range of influence. Using the mobility model and range, we can specify exactly where the loss takes place and how it moves over time. In this way, a realistic challenge can be modeled based upon a specific set of channel impairments that have locality instead of relying solely upon statistical methods.

The loss from typical propagation models is based upon the distance between the transmitter and receiver and may or may not also be influenced by random distributions. In the `MovingPropagationLossModel`, signal loss occurs when the impairment lies in the direct path between two wireless nodes as shown in Figure 3.2. The algorithm in this model calculates the distance between the center of the impairment to the closest point on the line segement between the two wireless nodes. If that distance is less than the radius of the impairment, then the loss is incurred. The distance between the wireless nodes is irrelevant; the loss is only applied if the impairment intersects the line segment between the transmitter and receiver.

Propagation loss models affect all the nodes installed on the wireless channel, so there is no need to identify and set up loss between individual nodes. The loss takes place automatically between any pair of nodes whenever the impairment intersects the path between them.

To be useful for a variety of challenge situations, the path loss incurred is based

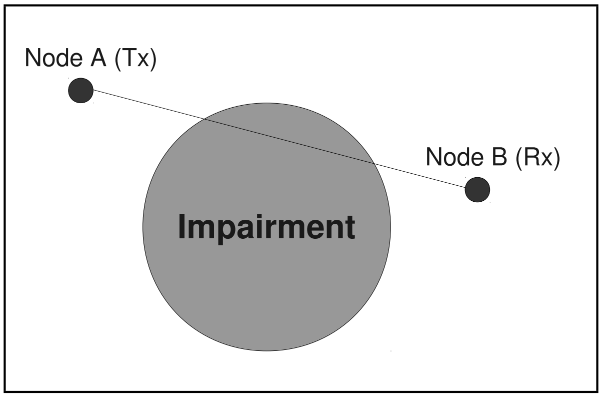


Figure 3.2. Two wireless nodes and a localized channel impairment

upon the two radii illustrated in Figure 3.3. The first is the center radius, and the second is the edge radius. These radii are configurable so as to simulate different kinds of impairments – from buildings to thunderstorms. Any path within the center radius suffers the full path loss value. Any path outside the edge radius suffers no path loss. And a path that falls between the center radius and the edge radius suffers a signal loss between zero and the full impairment loss value as the loss tapers linearly from the center radius to the edge radius.

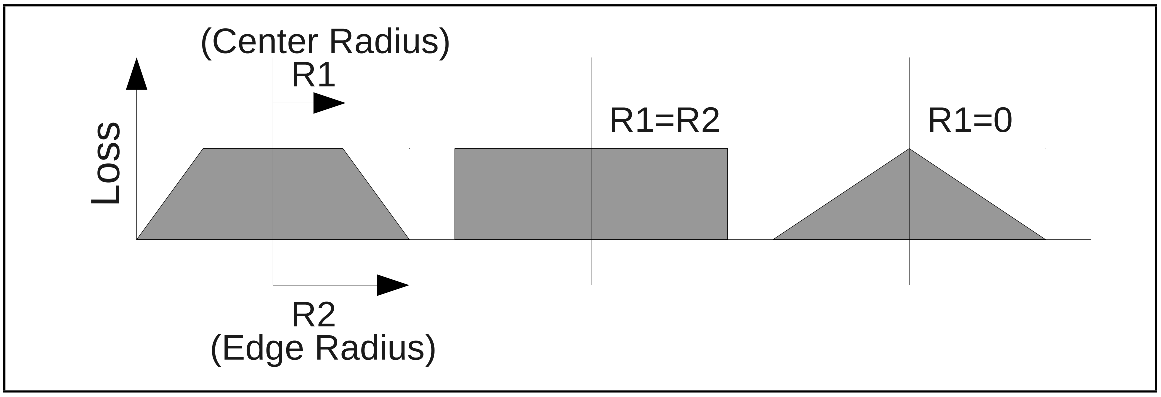


Figure 3.3. Impairment Loss based upon two radii

Note that the moving propagation loss model does not account for fast fading due to reflections caused by buildings but you can chain a Rayleigh-based loss model such as Jakes or Nakagami to the wireless channel to statistically simulate these effects.

3.2.2 Jammers and Radio Channel Interference

Unlike signal loss from scattering and line-of-sight obstacles, jammers cause radio channel interference, increase channel noise, and reduce the signal to noise ratio impacting a receiver's ability to discern data bits correctly. To understand how to simulate an effective jammer, it is important to understand how ns-3 handles the physical layer (PHY) signals.

3.2.2.1 Understanding Tx and Rx Signal Power in ns-3

The default transmission power in ns-3 $= 40$ mW $= 10 \times \log_{10}(40) = 16.02$ dBm. This and several other parameters can be adjusted in the `YansWifiPhyHelper`:

Table 3.1. YansWifiPhyHelper parameters in ns-3

Parameter	**Description**	**Default**
`RxGain`	Receiver Gain	1 dBm
`TxGain`	Transmitter Gain	1 dBm
`CCAMode1Threshold`	Rss must be higher than this to allow the `PHY` to declare `CCABusy` state	–99 dBm
`EnergyDetectionThreshold`	Rss must be higher than this to allow `PHY` to detect the signal and declare `SYNC` state	–96 dBm
`TxPowerStart, TxPowerEnd`	Min - Max Tx levels (dBm)	16 dBm
`TxPowerLevels`	Number of Tx power levels	1
`RxNoiseFigure`	Loss (dB) in S/N ratio due to non-ideal receiver	7 dB

If we use the LogDistance propagation loss model, the ns-3 receiver signal strength (R_{ss}) calculation is:

$$R_{ss} = \text{power}_{\text{Tx}} + \text{gain}_{\text{Tx}} - \text{loss}_{\text{ref}} - 10 \times n \times \log_{10}\left(\frac{\text{dist}}{\text{dist}_{\text{ref}}}\right) \tag{3.5}$$

Using 11 dBm for TxGain, 0 dBm for RxGain, distance of 100 meters, and default LogDistance model values of 46.677dBm loss at 1 meter and an exponent of 3.0:

$$R_{ss} = 16.02 + 11 - 46.677 - 30\log_{10}\left(\frac{100\text{ m}}{1\text{ m}}\right) \cong -79.7\text{ dBm} \tag{3.6}$$

Signals from all transmitter sources on the same frequency are calculated likewise and are used to determine the signal to noise ratio at the receiver. In the ns-3 physical layer model, if the PHY state is IDLE or CCABusy when a packet arrives, the receive energy of the first bit of the new signal is calculated. If the energy is higher than the EnergyDetectionThreshold, then the PHY changes state to SYNC and schedules an event for when the last bit of the packet is expected to be received. Otherwise, the PHY drops the packet and switches state to IDLE [1] (or to CCABusy if the total energy received reaches the CCA threshold).

With this in mind, the way to implement a jammer in ns-3 is to create transmitters that send high power, high datarate packets nonstop on the same channel as the network we are testing. Since transmitting Wi-Fi nodes back-off when they sense traffic, we must change the CCAMode1Threshold and EnergyDetectionThreshold parameters to prevent the jammers from backing off. In the YansWifiPhyHelper, the CCAMode1Threshold is the Rss threshold that must be reached to allow the PHY to change state from IDLE to CCABusy. The EnergyDetectionThreshold is the Rss threshold that must be reached to allow the PHY to detect a signal so it can back-off. To model the jamming devices in ns-3, these two thresholds must be set very high so that the jammers will not detect other signals on the channel and back-off. Setting both of these parameters to 0.0 dBm or higher will make sure the jammer nodes continue transmitting even in the presence of other signals.

Chapter 4

Simulations

This chapter describes many of the ns-3 parameters, models, and simulation environments chosen for use in the benchmarks. Details about how the simulations are set up and executed are explained. Fundamental characteristics about the routing protocols chosen to benchmark against are discussed. This chapter is organized as follows. Section 4.1 describes benchmark parameters, metrics, independent variables, and benchmark scoring. Section 4.2 explains the use of automation scripts which are used to setup and conduct the tests, and collect performance data for further analysis. Section 4.3 outlines the models used in the benchmarks to provide a more realistic simulation environment. The different benchmark scenarios are described in Section 4.4. Section 4.5 details ns-3 code modifications needed for simulations using distant nodes or AeroRP. Section 4.6 introduces the routing protocols used in these benchmarks. Section 4.7 lists the primary ns-3 simulation parameters and their default settings. Section 4.8 includes a few miscellaneous notes about the simulations.

4.1 Simulation Model

Benchmarking is a systematic method of testing and comparing the performance of a system, or in this case a routing protocol, against another protocol that is commonly recognized as the seminal or de-facto standard. Not only is the selection of routing protocols important to the benchmark, but also the selection of the challenges they will face. Benchmarks are not meant to test every possible permutation. The tests comprise a handful of both typical and corner cases that are representative of the kinds of environments frequently encountered in wireless ad hoc networks. Corner cases test the network at the extremities of their operational boundaries and can push an algorithm to its limits. Good performance on the corner cases can make the difference between success and failure in very challenging real world environments. If designed well, the same benchmarks can be used to test the performance of future proposed protocols with little or no modifications.

Simulations are performed in ns-3 [24], which is a growing, open-source network performance simulation environment with several ad hoc network routing protocols already built in. New protocols are emerging to address the difficult environment faced by dynamic ad hoc networks [25] and there is a need to evaluate their performance against the legacy protocols. However, when simulations ignore physical-layer conditions and challenges, such as path loss, fading, and interference, the results are not indicative of the solution's true performance.

The parameters used in this thesis to measure the resiliency of a protocol are cumulative data received and packet delay. Cumulative data received will be the predominant comparison metric and is defined as the number of bits that are successfully forwarded from source nodes to the destination node. Since the transport protocol used throughout this research is UDP, the throughput already excludes retransmitted packets. To test the protocol under various wireless network applications, the benchmarks will be divided into three categories:

1. Stationary simple networks
2. Mobile 3-D networks
3. Special applications

More detail on the individual scenarios is given in Section 4.4. The special application network benchmarks test challenging scenarios such as short window of opportunity and a dense urban environment. Some of the independent variables chosen for these benchmarks are:

1. Distance between nodes
2. Hops between transmitter and receiver
3. Transmission power
4. Number of impairments

Table 4.1. Simulation protocols to be compared

Transport	**Network**	**Routing**
UDP	IP	AODV
UDP	IP	DSDV
UDP	IP	OLSR
UDP	IP	DSR
UDP	IP	AeroRP

Table 4.1 lists the protocols that are evaluated using the proposed benchmarks. In conjunction with the AeroRP routing protocol, the AeroTP transport protocol

and the AeroNP network protocol should be used. However, since AeroTP and AeroNP are not yet available for testing in ns-3, the traditional UDP and IP protocols are implemented. UDP is chosen over TCP for these benchmarks since we are testing the performance of routing algorithms and are concerned with getting individual packets through, and not concerned with end-to-end reliability and flow control.

Each wireless network protocol is affected by mobility models as well as physical layer impairments, and it is expected that some protocols will handle these challenges better than others. Each protocol is tested against every benchmark, which should result in a better understanding of each protocol's strengths and weaknesses. For each benchmark, the theoretical maximum cumulative throughput will be calculated first, and then the protocols will be tested in ns-3 and scored based upon their performance relative to the theoretical maximum.

Impairments such as jammers and localized shadowing are used to challenge the test networks and help distinguish those protocols that perform well in challenging environments. In conducting this research, the questions we try to answer are:

- How does the AeroRP protocol perform against traditional MANET routing protocols?

- How does a challenging environment (jammers, rain fade, and high speed) change the performance?

- Can these benchmarks help to identify mechanisms in one protocol that might be beneficial if implemented in another protocol or scenario?

- How does the mobility model affect the performance?

4.2 Simulation Automation Scripts

A customized ns-3 simulation script is used for all benchmarks and tests; it is designed to read a single network configuration file and build the simulated network dynamically at run-time. The configuration file is a simple text file; the first line includes three pieces of information: the routing protocol to use, the speed of the nodes, and the random seed. This last element is used to set the random seed generator in ns-3; using the same seed generator provides a useful way to create duplicate simulation runs. The rest of the lines in the file contain information about the type, location, and signal strength of the various nodes. Four node types are defined: ground station nodes, airborne nodes, jammers, and moving propagation shadows or impairments. A separate Perl script is used to automate the generation of config files with different parameters and then execute the simulation script on each one. This script also automates the storing of results in an output data file for further processing. In tests involving random node location and movement, simulations are performed eight times using eight random seed numbers, and the average throughput values from the eight individual tests will be used in the benchmark score for that protocol.

4.3 Simulating a Realistic Environment

Overly simplistic simulation models do not test the strengths and weaknesses of MANET routing protocols. This section describes the methods used to create more realistic simulations in ns-3. This includes propagation loss models, jammers, delay, and the 3-D Gauss-Markov mobility model. All of these elements combine to create a more realistic, challenging environment with which to test the performance of the routing protocols.

4.3.1 Propagation Loss Models

One of the purposes of this research is to simulate realistic radio signal channel environments in which the different routing protocols can be tested. One way this is accomplished is to daisy-chain different propagation loss models together as described in Section 2.2. A realistic model can be composed from a handful of basic models from the three primary categories: path loss, shadowing, and fast fading. All benchmarks use the Friis propagation loss model for path loss in free space, and the Nakagami model is used for fast fading to statistically simulate the constructive and destructive interference caused by multipath signals and reflections. Nakagami implements a Rayleigh fast fading algorithm when m is set to 1, which is used when simulating dense urban environments where line-of-sight is completely obstructed and the signal suffers from many reflections.

When simulating more open environments with better line-of-sight, the fading is reduced by using a higher value for the m parameter. In these cases, a Rician fading model is preferred since it assumes that there is a dominant line-of-sight signal that is stronger than the multipath interferers [26], but the Rician model is not currently available in ns-3. To simulate slow fading caused by signal shadows or widespread scattering, the `MovingPropagationLossModel` is used in some benchmarks to generate these signal impairments at specific locations depending upon the scenario.

4.3.2 Interference and Delay

Several of the benchmarks include jammers to create a hostile channel environment. Jammers are implemented in ns-3 as indicated in section 3.2.2. To account for the propagation delay of electro-magnetic waves at the speed of light, the `ConstantSpeedPropagationDelayModel` is used in all ns-3 simulations.

4.3.3 Mobility Model

The stationary simple benchmarks consist of stationary nodes, so the mobility model chosen is the Constant Position mobility model. For the mobile benchmarks, the Gauss-Markov mobility model is selected to better mimick the motion of nodes in a MANET. The fundamental Gauss-Markov parameters used in the benchmarks are specified in Table 4.2.

Table 4.2. Gauss-Markov simulation parameters

Simulation parameter	**Value**
`Alpha`	0.85
`MeanVelocity`	[400 800 1200 2400] m/s
`MeanDirection`	[0 2π] radians
`MeanPitch`	[-0.05 0.05] radians
`NormalVelocity`	Normal Distribution w/ Std Dev of 80 m/s
`NormalDirection`	Normal Distribution w/ Std Dev of 0.8 radians
`NormalPitch`	Normal Distribution w/ Std Dev of 0.2 radians

4.4 Simulation Scenarios

The many algorithms that make up a routing protocol must interact with each other as well as the unpredictable channel environment. Benchmarks allow us to compare the performance of these complex systems. The benchmarks chosen for this research attempt to cover a wide range of variables, but are not designed to address every scenario. Table 4.3 lists the different benchmarks and their primary attributes.

4.4.1 Stationary Benchmarks

The first three benchmarks are chosen to provide a baseline of performance for each routing protocol. These three benchmarks use stationary nodes and test performance against factors such as node spacing, link outages, and number of hops in the route. These are also useful in determining if the simulations are working properly, and that routes get set up and packets arrive at the receiver.

4.4.2 3-D Mobile Benchmarks

Benchmark 4 tests routing protocol performance as nodes traverse a large 3-dimensional area using different mobility models and node density. Benchmark 5 tests cumulative throughput in the same 3-dimensional space against an increasing number of jammers.

4.4.3 Transmission Power

One common theme in a discussion of mobile ad-hoc networks is the subject of power consumption. It is often the case that nodes in ad-hoc networks are power constrained and great care must be taken to ensure that the devices have enough power to perform their assigned tasks for a given period of time. One way to conserve power is to keep transmission power as low as possible. In Section 5.6, the routing protocols are tested to see which perform better at lower transmission powers.

Table 4.3. Comparison of benchmarks

Benchmark	**Independant Variable**	**Impairments**	**Mobility**
1. Stationary nodes in freespace	Node distance	None	Stationary
2. Stationary nodes with impairments	Node distance	Nakagami, Moving impairment	Stationary
3. Stationary increasing hops	Hops	Nakagami	Stationary
4. 3D space mobility model	Mobility model, Gauss-Markov α	Nakagami	Random waypoint, Gauss-Markov, Random direction
5. 3D space with jammers	# of jammers	Nakagami, Jammers	Gauss-Markov
6. Transmission power test	Tx power	Nakagami	Gauss-Markov
7. Short window of opportunity	None	None	To waypoint and home
8. Urban canyon test	None	Nakagami, Moving impairment	Gauss-Markov

4.4.4 Short Window of Opportunity

Not all MANET challenges come in the form of physical layer channel impairments such as interference. Many challenges facing MANET routing algorithms come from conditions imposed by high mobility, which tests the efficiency and resiliency of the algorithm. There may be only brief periods of time when a complete end-to-end route exists. Consider a mobile network in which nodes are only in range of each other for a few seconds at a time, and complete end-to-end routes are sporadic. In Section 5.7, the routing protocols face a challenge in which the time window to set up a route is very small, and each new route is completely different.

4.4.5 Dense Urban Environments

The final benchmark tests the performance of each protocol at street level in a dense urban setting. This common but extremely challenging wireless environment is characterized by many deep transmission shadows and reflections caused by tall buildings in a dense grid-like arrangement.

4.5 Required Modifications in ns-3

In the initial ns-3 setup, IEEE 802.11b was used as the link. However, in ns-3, it is very difficult to get meaningful data with 802.11b – data rates must be kept extremely low, below 8 kb/s, and the distance between nodes may not exceed 4000 meters or the throughput will drop to near zero. These problems and several solutions are mentioned in [27], and the parameter changes mentioned in that work do fix the problem, but only for ns-3 version 7. Using version ns-3.11, these changes do not help. However, ns-3.11 includes support for a wider range of data rates and also supports the IEEE 802.11g standard, which fortunately allows for much longer links and higher data rates. Using 802.11g with the new `ErpOfdmRate12Mbps` rate for the WiFi manager control and data modes proves sufficient to send packets at rates of several Mb/s over hundreds of kilometers, so long as the transmission power is high enough.

A new ns-3 code error emerged when using AeroRP in ns-3.11 that required a few code modifications. The primary problem was that the IPv4 protocol in ns-3 was not checking the UDP packet payload size properly after it was altered by AeroRP, leading to misreading IP header information and failing assert statements. To correct the problem, the code indicated with plus signs is added around line 628 of the `ipv4-l3-protocol.cc` file:

```
if (newRoute)
  {
  + // Routing protocol might have added some more stuff to the packet
  + // incase of AeroRP. So recalculate the packet size and modify the
  + // ip header field if necessary.
  + ipHeader.SetPayloadSize(packet->GetSize());
    int32_t interface=GetInterfaceForDevice(newRoute->GetOutputDevice());
    m_sendOutgoingTrace (ipHeader, packet, interface);
    SendRealOut (newRoute, packet->Copy (), ipHeader);
  }
```

Furthermore, the zero in line 299 of `udp-l4-protocol.cc` must be changed to `route` so that it reads:

```
m_downTarget (packet, saddr, daddr, PROT_NUMBER, route);
```

Lastly, a line of code is added to the `else` statement at line 880 of the `mac-low.cc` ns-3 file so that it reads:

```
else
  {
    goto rxPacket;
    //NS_LOG_DEBUG_VERBOSE ("rx not-for-me from %d", GetSource (packet));
  }
```

Implementing the `MovingPropagationLossModel` requires placing the code file and the header file in the `ns-3.11/src/propagation/model` directory and adding references to these files in the `wscript` file located in the `propagation` folder.

4.6 Competing Routing Protocols

It would be impossible to simulate all the different proposed MANET routing protocols and algorithms for two reasons; first, there are more than 1000 such protocols and variations [2], and second, we can only simulate and benchmark those protocols that are openly available in ns-3 and those under development by members of the ResiliNets team we are permitted to test. It would be beneficial to test AeroRP against other location-based protocols, but there are currently no other location-aware routing protocols in ns-3. However there is no harm in benchmarking AeroRP against the traditional routing protocols such as OLSR, DSDV, AODV, and DSR since one purpose of these benchmarks is to identify applications for which these protocols were not initially designed but in which they perform well.

Table 4.4. Comparison of routing protocols
(Adapted from [2] Table 5.1)

Protocol	**Category**	**Metrics**	**Route Recovery**	**Route Repository**	**Overhead**	**Feature**
OLSR	Proactive	Shortest Path	Periodic Updates	Table	High	Uses MPRs as routers
DSDV	Proactive	Shortest Path	Periodic Broadcast	Table	High	Distributed Algorithm
AODV	Reactive	Newest Route, Shortest Path	Same as DSR, local repair	Table	High	Only keeps track of next hop in route
DSR	Reactive	Shortest Path, Next Available	New Route, Notify Source	Cache	High	Completely On-demand
AeroRP	Geographical	Shortest Time-to-Intercept	Periodic Neighbor Update	Cache	Low	Per-Hop Routing, Store-and-Haul

4.6.1 OLSR

The Optimized Link State Routing (OLSR) protocol employs a proactive link state algorithm as defined in Sections 2.4.1 and 2.4.3. However, in the case of OLSR, the algorithms are optimized for use in mobile ad-hoc networks. By reducing message size and limiting the amount of flooding in the network, OLSR significantly reduces control overhead and improves efficiency. With these characteristics, we expect OLSR to perform well in small, fully connected, relatively stable networks.

4.6.2 DSDV

Destination-Sequenced Distance-Vector (DSDV) is also proactive, yet uses the Bellman-Ford distance vector algorithm described in Section 2.4.4. Each node creates a routing table containing all possible destinations, the number of hops to reach them, and the next node on the route. These two pieces of information, the number of hops and the next node respectively, make up the distance and vector parts of DSDV's name. DSDV uses sequence numbers to guarantee loop-free operation. It is also expected to do well in benchmarks with fairly small, fully connected networks that are not highly-dynamic.

4.6.3 AODV

Ad-hoc On-demand Distance-Vector (AODV) is a reactive (on-demand) distance vector routing protocol. Routes are only created as they are needed, and nodes only store the distance and vector information to all destinations in their routing tables. Like its proactive counterpart, DSDV, AODV prevents loops and the "count-to-infinity" problem by sequentially numbering the route updates. To reduce unnecessary overhead, route request messages are sequentially numbered to prevent nodes from repeating duplicate requests. We expect AODV to perform better than DSDV due to reduced message overhead, and also to perform better than OLSR and DSDV in benchmarks with highly-dynamic, fully connected networks.

4.6.4 DSR

Dynamic Source Routing (DSR) is also a reactive protocol. But unlike AODV, DSR uses source routing, a mechanism by which the route request data packet carries the addresses of all nodes in the route as it traverses the network to the destination node. When a route is found, a reply message containing all the addresses of the intermediate nodes is sent back along the same path to the source node who requested it. As this message makes its way back to the source, the intermediate nodes store the route in their own route cache in an effort to conserve network resources should a route to the same destination be requested again. The route cache can become stale quickly in highly-dynamic networks, reducing its performance. We expect DSR to perform well in relatively stable, fully connected networks with few hops between the source and destination. Performance is expected to drop significantly as the network environment becomes more dynamic.

4.6.5 AeroRP

The Aero protocol stack is designed to be delay and disruption tolerant to accommodate the requirements of highly mobile ad-hoc networks. AeroTP is a TCP-friendly transport protocol that uses opportunistic connection estabilshment and open-loop rate-based transmission control. AeroNP is an IP-compatible network protocol with efficient address mapping and header and support for cross-layer optimizations. AeroRP is a location-aware routing protocol supporting partial-path, store and haul, and dynamic routing. Each of these work together to improve performance of highly-dynamic airborne communication.

Initial tests on AeroRP were presented in [5] comparing AeroRP to DSDV and AODV, however these simulations did not include wireless challenges or a memory-based mobility model. The benchmarks presented in this thesis will simulate AeroRP using the 3-D Gauss-Markov mobility model and challenges such as jammers and localized signal fading. Whereas DSR, DSDV, and AODV favor routes with the fewest hops, AeroRP includes a `TransmissionRange` parameter which prevents it from transmitting data beyond a specified distance thereby helping to ensure that routes are chosen that have good signal strength. It is anticipated that this will also improve AeroRP's overall performance. We expect AeroRP to perform well in highly-dynamic networks and those that suffer from intermittent connectivity.

4.7 General Scenario Setup

Table 4.5 defines several simulation models and parameters used in these benchmarks. Note that the Nakagami fast fading model is only used because the Rician model is not available in ns-3. The Rician model assumes there is a dominant line-of-sight signal with weaker multipath signals causing fast stochastic fading. The Nakagami model, however, assumes no line-of-sight, but its effects can be mitigated somewhat by setting m=10. As m approaches ∞, Nakagami fast fading goes to zero.

4.8 Simulation Notes

Simulation tests that involve random variables such as Gauss-Markov mobility are run 8 times with different random seed generator numbers and the median value was used in the calculation of the score. The ns-3 implementation of the DSR protocol is still under beta test so its performance may not be up to par with the other protocols. Its inclusion in these benchmarks is still important since any performance missteps may point to problematic algorithms in the code and speed up their resolution, and the current benchmark scores can be compared to future

Table 4.5. Simulation scenario parameters

Parameter	**Application**	**Setting**
WiFi Mode		802.11g
Tx Power		40 dBm
Channel DataRate		`ErpOfdmRate12Mbps`
Data Packet Size		1000 bytes
Application Data Rate	Only one transmitter Multiple transmitters Jammers/Interferers	800 kb/s, 50% duty cycle 80 kb/s, 50% duty cycle 4 Mb/s, 100% duty cycle
Mobility Model	Stationary Benchmarks Mobile Benchmarks	Constant Position Gauss-Markov
Propagation Delay		Constant Speed Propagation Delay Model
Propagation Loss Model	Path Loss Fast Fading Localized Shadowing	Friis (using $\lambda = 0.125$) Nakagami (using $m = 10.0$) Moving Propagation Loss Model
Node Speed		0, 400, 800, 1200, 2400 m/s

scores to judge the impact of the corrections made.

Chapter 5

Analysis

This chapter presents analysis of the five routing protocols: OLSR, DSDV, AODV, DSR, and AeroRP. In some of the tests, AeroRP is tested twice: once with ferrying turned on, and a second time with ferrying turned off. The results of such tests is to show the effect that ferrying has on the overall performance, and to see how AeroRP without ferrying compares to legacy routing protocols, which do not have this feature. The following eight sections comprise the benchmarks that make up this research. Each benchmark is divided into four sections. The first is a description of the benchmark. The second section details the setup and network topology used. The third gives a brief statement on the expected results based upon what we know about each protocol's algorithms. The last section of each benchmark gives the simulation results and comments on noteworthy results or anomalies.

5.1 Benchmark 1: 6 Stationary Nodes in Free-Space

5.1.1 Description

The 6-node benchmark is a simple baseline test with stationary equidistant nodes. There is one transmitter and one receiver positioned at the far ends of the network. There are also four intermediate nodes between the transmitter and receiver. The network layout is depicted in Figure 5.1. The nodes are placed some distance from each other in a straight line and the simulation is run. Then the nodes are placed further away from each other and the test is run again. As the distance between the nodes grows from 0 to 33 km with each simulation, the cumulative data received is measured and plotted. This simple benchmark provides a baseline for the performance of the routing protocols in a stable, motionless environment.

5.1.2 Setup

In this benchmark, only the Friis path loss propagation model is used. Nodes are assumed to have unobstructed line-of-sight without reflections, scattering, or channel interference. This is the kind of unrealistic simulation refered to in Section 2, devoid of physical layer imperfections, challenges, and threats. As indicated in Table 4.5, the nodes transmit at 40 dBm, which gives them a free-space range of about 32 km. The sending node transmits at 800 kb/s with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the sender transmits data for 30 seconds. The theoretical maximum cumulative data received is 800 kb/s $\times$ 50% $\times$ 30 sec = 12 Mb.

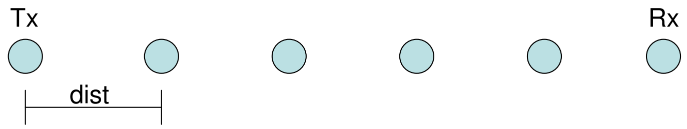


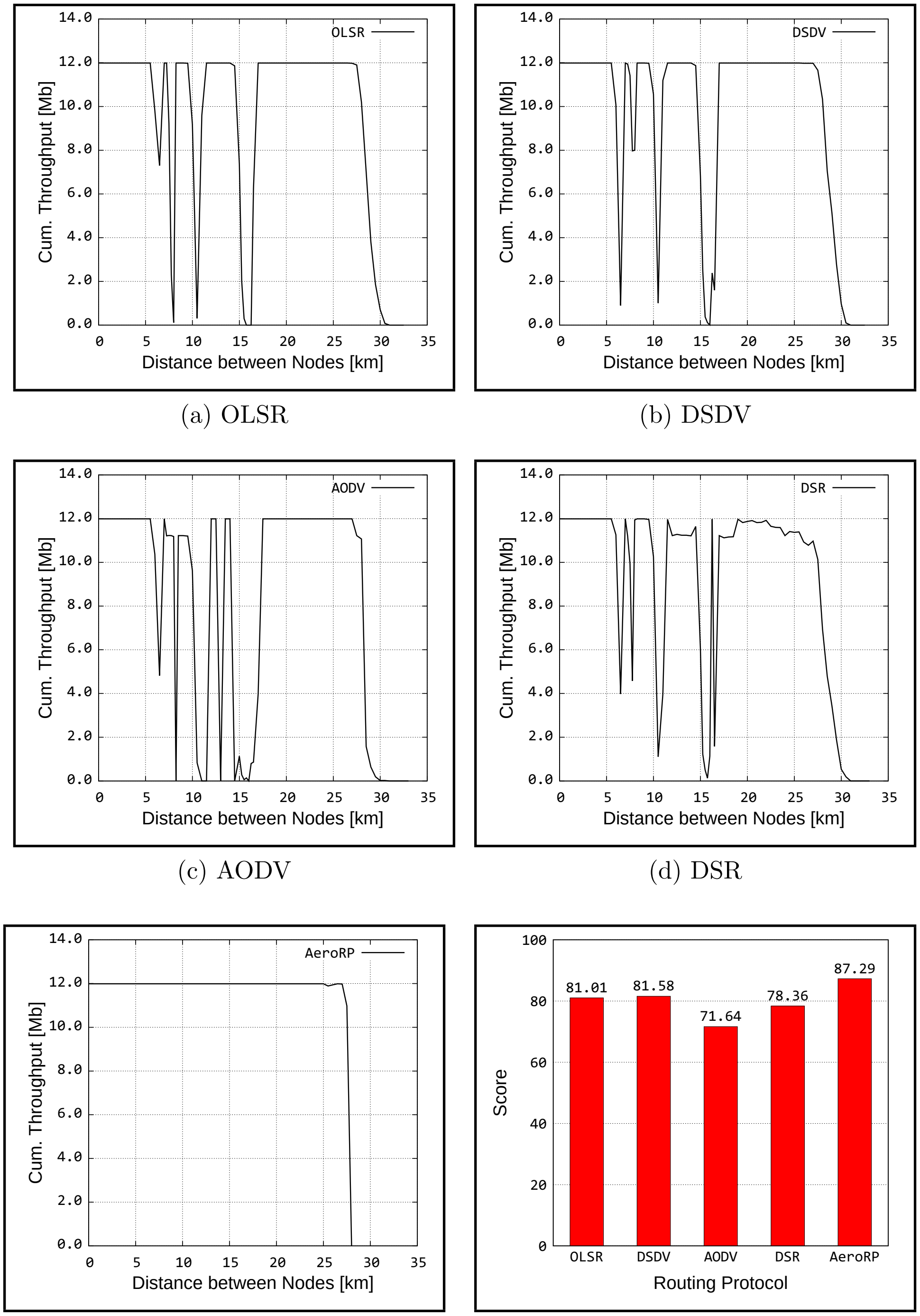
Figure 5.1. Network setup for the 6-node benchmark

5.1.3 Expected Results

In this baseline benchmark, all the routing protocols are expected to perform nearly identically. Once the routing algorithm sets up the first route, it does not need to change. All the tested routing protocols are capable of setting up routes with many hops in a connected stationary environment, so any problems creating routes in this test would be an indication that either the algorithm cannot handle a network of that size or the algorithm is not properly implemented in ns-3.

5.1.4 Results

With reference to Figure 5.2, there are points along the distance-axis where the cumulative data received briefly and unexpectedly drops to zero. In retrospect, these zeros should have been expected from the beginning. The legacy routing protocols do not route through every node if they don't have to; they will attempt to route through those nodes that result in the fewest hops. In the process of doing so, there are times when the route includes hops that are at the edge of the node's transmission range, resulting in very low throughput. Essentially, these zeros should occur in the 6-node benchmark at inverse integer multiples of the maximum transmission distance, where the hop distance is at or near the maximum transmission range, signal strength is weak, and the routing algorithm has not yet added or altered hops on the route. This is illustrated in Figure 5.3.



(a) OLSR

(b) DSDV

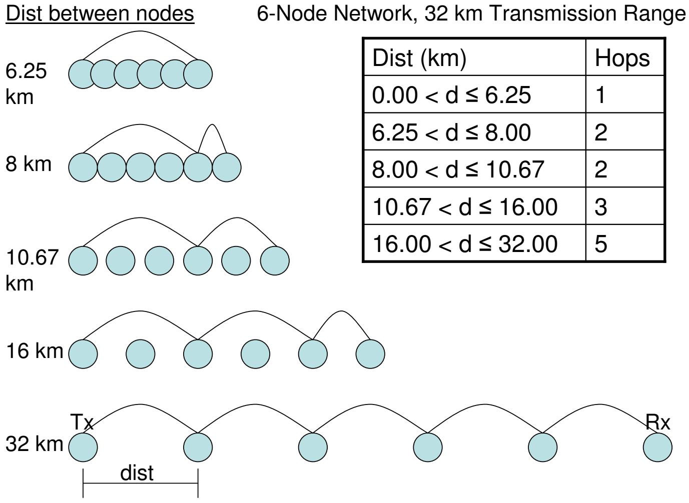
(c) AODV

(d) DSR

(e) AeroRP

(f) Scores

Figure 5.2. 6-Node benchmark with Friis propagation loss



Dist (km)	Hops
0.00 < d ≤ 6.25	1
6.25 < d ≤ 8.00	2
8.00 < d ≤ 10.67	2
10.67 < d ≤ 16.00	3
16.00 < d ≤ 32.00	5

Figure 5.3. Zeros expected when route adds/alters hops

The AeroRP graph in Figure 5.2e does not exhibit these zeros; this is one advantage to using a routing algorithm that isn't based strictly upon the fewest number of hops. In AeroRP, a maximum transmission range variable is used to prevent the algorithm from using hops that might have weak signal strength. The resulting routes may occasionally have more hops, but the signal strength will be better.

Scores shown in Figure 5.2f were calculated using a weighted average of all cumulative data values between 0 and 32 km. The sum total average score is then divided by 12 Mb and multiplied by 100%. Thus, a score of 100 means that every data point between 0 and 32 km was 12 Mb. Even though AeroRP rejects hops beyond 27.8 km and receives zero score points above that distance, it still receives the highest score due to its unrivaled 5-hop performance below 27.8 km. Although not as high, the OLSR and DSDV routing protocols performed similarly, with zeros at the very distances expected from algorithms whose metric is fewest hops.

5.2 Benchmark 2: 6 Stationary Nodes with Impairments

5.2.1 Description

As an extension of the baseline benchmark, the second benchmark performs the same test as the first benchmark with 6 stationary nodes, but includes more realistic physical layer impairments such as fast fading and localized moving shadows. As before, there is one transmitter and one receiver with four intermediate nodes between them. The setup for this benchmark, depicted in Figure 5.4, incorporates a moving impairment that blocks the transmission path of the route for a period of time. As before, the distance between the nodes grows from 0 to 33 km with each simulation, and the cumulative data received is measured and plotted.

5.2.2 Setup

In this benchmark, the Nakagami fast fading propagation loss model is added to the Friis path loss model in the wireless channel. Nodes are assumed to have somewhat obstructed line-of-sight and a fair amount of reflections and scattering. As before, the sending node transmits at 800 kb/s with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the sender transmits data for 30 seconds. During this test, a moving impairment crosses the route between the third and fourth nodes and disrupts data transmission for 10 seconds. The impairment causes 80 dBm signal loss which effectively isolates the transmitter from the receiver. As in the first benchmark, the theoretical maximum cumulative data received is 800 kb/s × 50% × 30 sec = 12 Mb.

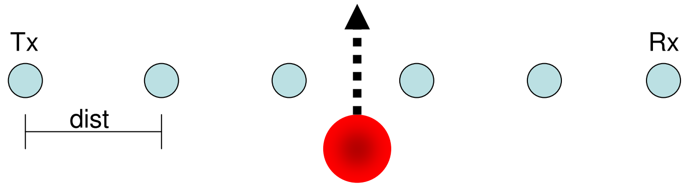


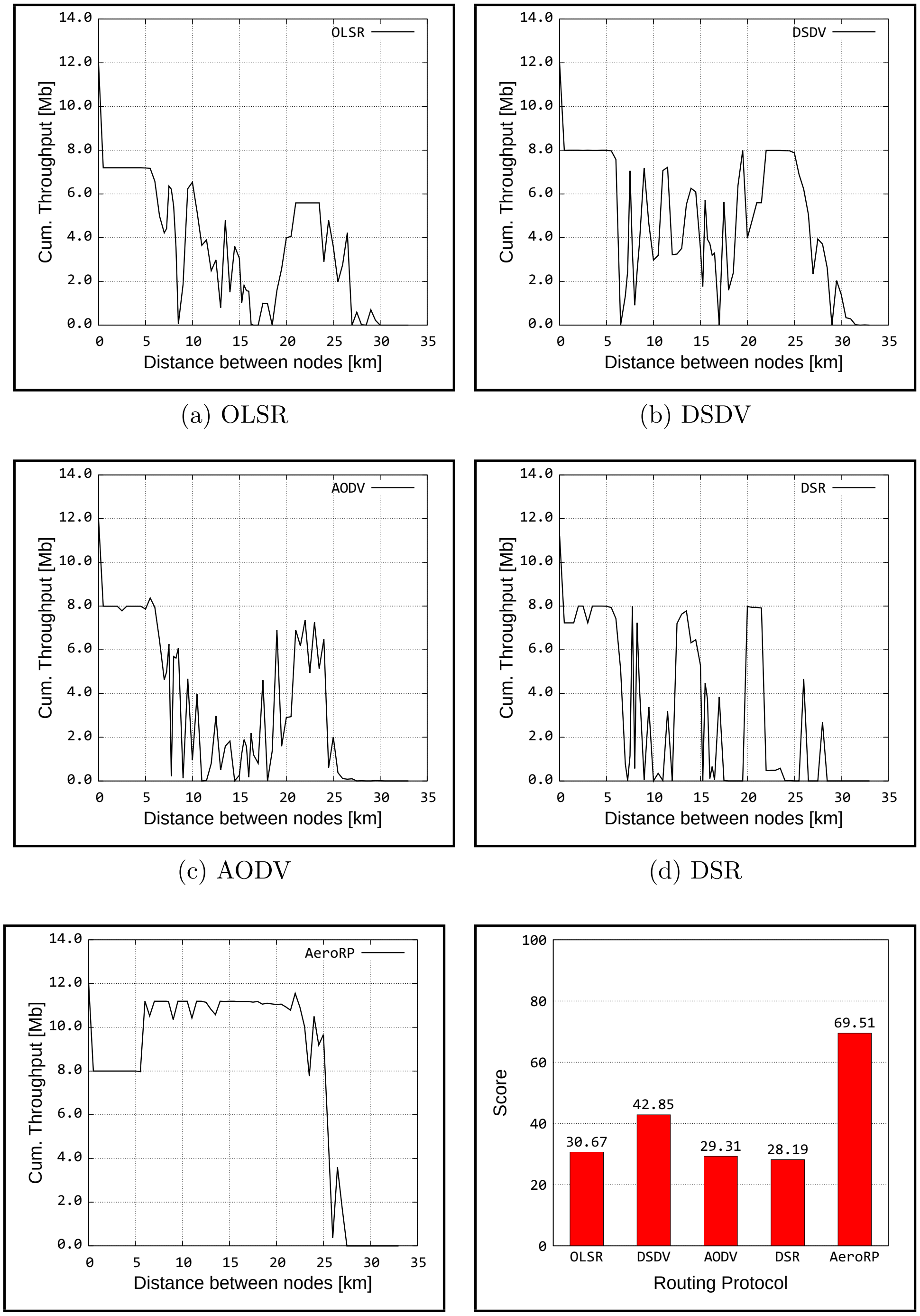
Figure 5.4. Setup for 6-node benchmark with moving impairment

5.2.3 Expected Results

In this benchmark, OLSR and DSDV are expected to perform similar to the non-impaired benchmark, except for a 33% drop in the data received due to the 10 second network disruption from the moving impairment. The Nakagami fast fading is going to play a role in final performance, but it is uncertain how much it will affect the final numbers.

5.2.4 Results

AeroRP uses store-and-forward techniques which allow data to continue moving toward the destination node using any available opportunity. The transmitter continues sending packets during the outage, unaware of whether the previous packets have made it to their destination, if they are in transit, or if they are in store-and-haul mode. The opportunistic routing algorithm makes a difference in this benchmark, as seen in Figure 5.5. It even appears to make up for many of the performance drops caused by the Nakagami fast fading. Figure 5.6 shows the transmitter sending 800 kb/s with a duty cycle of 50% (`on` for one second and `off` for one second). From 35 to 45 seconds, the impairment prevents packets from reaching the receiver. After the impairment has passed, packets are received once again. By looking at throughput vs. time over the duration of the simulation, it is easy to see how AeroRP's store-and-haul mechanism makes up for the physical



(a) OLSR

(b) DSDV

(c) AODV

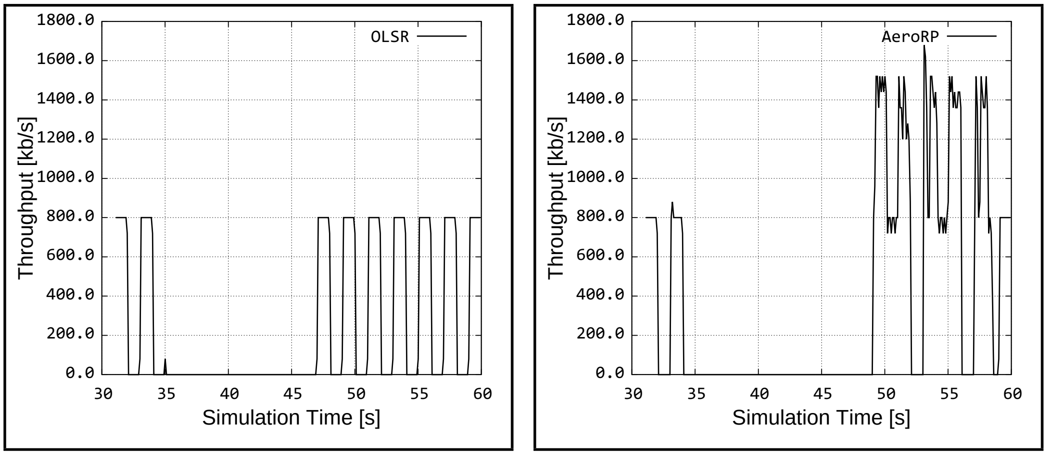
(d) DSR

(e) AeroRP

(f) Scores

Figure 5.5. Results of 6-node benchmark with challenges

channel impairments by delivering the packets that were stored during the outage and sending them along with later data packets.



(a) OLSR

(b) AeroRP

Figure 5.6. 6-node benchmark: throughput over time

Scores shown in Figure 5.5f are calculated like in the first benchmark, using a weighted average of each cumulative data value between 0 and 32 km. The sum total average score is divided by 12 Mb and multiplied by 100%. Thus, a score of 100 means that every data point between 0 and 32 km was 12 Mb. In this benchmark, AeroRP with its store-and-haul and opportunistic routing mechanisms allow packets that are sent during an outage to be delivered to their destination. The effect of the fast fading takes a real toll on the performance of the OLSR, AODV, and DSR protocols.

5.3 Benchmark 3: Stationary Increasing Hop Test

5.3.1 Description

In this test, we look at the cumulative data received and packet delay characteristics of the different routing protocols as the number of hops grows. As with the previous benchmarks, this is also a stationary test; but in this case, the nodes are always 20 km apart and the number of hops from transmitter to receiver grows from 1 to 19. The 20 km distance between each node ensures that every node will be used in the route, and that the signal strength is sufficient.

5.3.2 Setup

There are no moving impairments or jammers in this test, but both the Friis and Nakagami propagation models are installed on the wireless channel. The network is shown in Figure 5.7. Nodes are assumed to have somewhat obstructed line-of-sight and a fair amount of reflections and scattering. Due to the random nature of the Nakagami fast fading algorithm, each test is run 8 times and the average value is used. The sending node transmits at 800 kb/s with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the sender transmits data for 30 seconds. The theoretical maximum cumulative data received is 800 kb/s × 50% × 30 sec = 12 Mb. In addition to testing AeroRP with ferrying turned on, AeroRP will also be tested with ferrying turned off to see how this affects its cumulative data and packet delay characteristics.

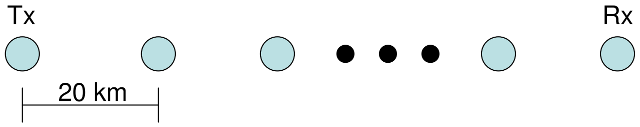


Figure 5.7. Stationary multi-hop network setup

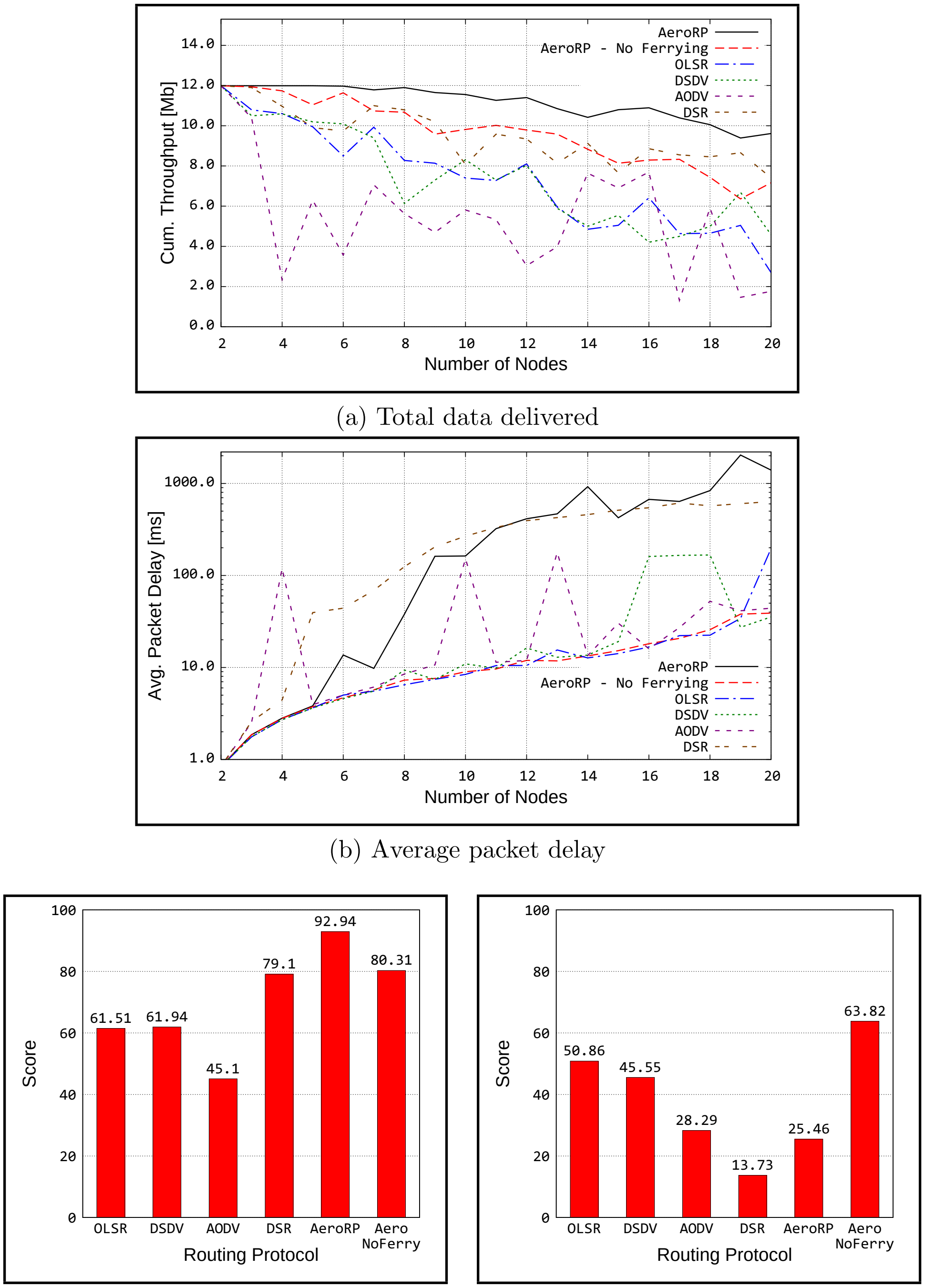
5.3.3 Expected Results

Ideally, the average delay for each simulation should grow linearly with the number of hops in the route. However, the store-and-haul mechanism in AeroRP makes it possible to have some packets arrive several seconds after they were initially transmitted, greatly increasing the average packet delay metric. OLSR, DSDV, DSR, AODV, and AeroRP without ferrying are expected to exhibit packet delay that grows linearly with the number of hops, whereas AeroRP with ferrying is expected to have much higher average delay. However, the cumulative data received using AeroRP should be much higher than the other protocols due to this same mechanism.

5.3.4 Results

As shown in Figure 5.8, AeroRP outperforms the other protocols due to its ability to continue forwarding packets when a complete route is not available. Packets that are sent during fully-connected times have the same delay as with other protocols; and packets sent during route disruptions are stored on the last connected node until the outage is over, and are then forwarded to their destination. These packets are the cause of the higher delay in AeroRP as can be seen in Figure 5.9. It is surprising to see the performance of AeroRP without ferrying. Not only does it outperform the traditional routing protocols in terms of total data received, but it does so without sacrificing packet delay, thus giving it the highest packet delay score.

Scores shown in Figure 5.8c were calculated using the average of all cumulative data delivery values divided by 12 Mb and multiplied by 100%. A score of 100 means that every data point was 12 Mb. For the delay scores (Figure 5.8d), a combination of cumulative data received and packet delay was used. Scores were based against the best possible cumulative data of 12 Mb and against the best possible delay, which is calculated using the best 1-hop delay multiplied by the



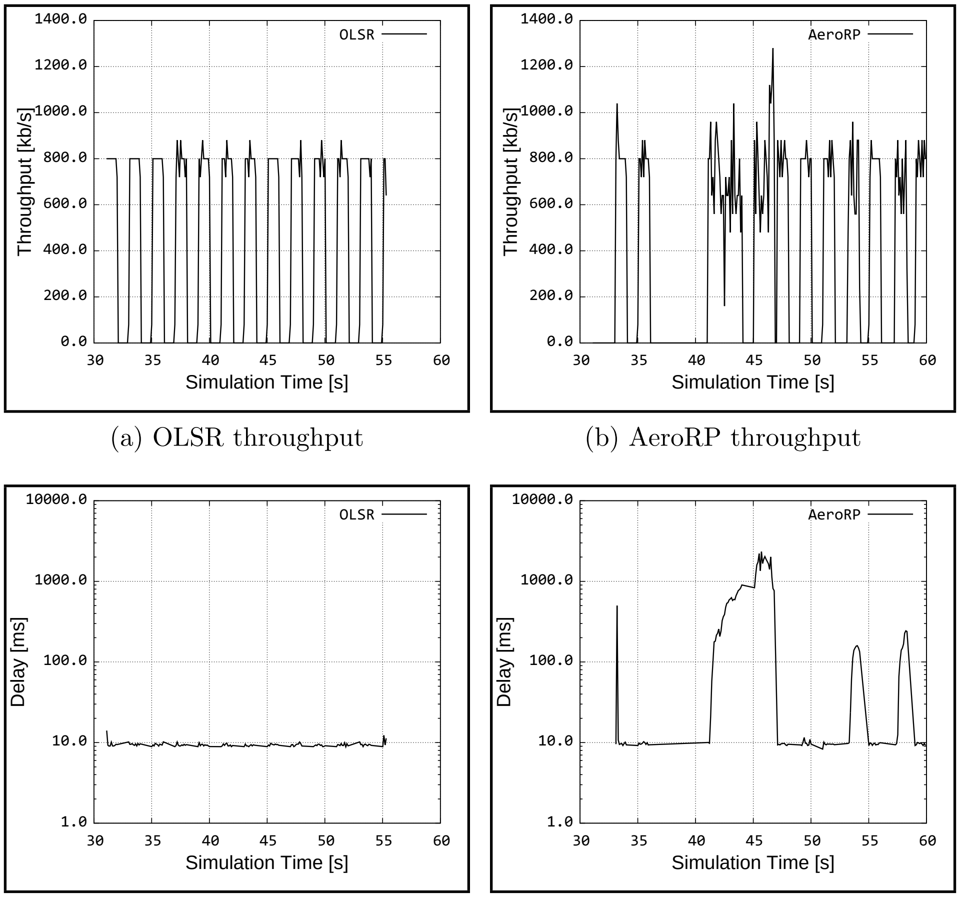
(a) Total data delivered

(b) Average packet delay

(c) Cumulative data scores

(d) Packet delay scores

Figure 5.8. Stationary multi-hop benchmark performance



(a) OLSR throughput

(b) AeroRP throughput

(c) OLSR delay

(d) AeroRP delay

Figure 5.9. Multi-hop benchmark throughput and delay vs. time

number of hops. These calculations were done for each data point. For example, if the cumulative data received for 10 hops was 7.5 Mb and the delay was 10 ms, and the best one-hop delay was 0.83 ms, the score for 10 hops would be calculated as follows:

$$\begin{aligned} s_{10} &= \frac{7.5 \text{ Mb}}{10 \text{ ms}} \times \frac{0.83 \text{ ms} \times 10 \text{ hops}}{12 \text{ Mb}} \\ s_{10} &= 0.52875 \end{aligned} \tag{5.1}$$

The final score for each routing protocol is determined by calculating the average of the individual hop scores and multiplying by 100. In this way, a low delay is only considered good if packets were actually being sent, and high delay is not as bad if it is accompanied by high throughput. A score of 100 means that all 12 Mb worth of packet data were received and the average packet delay was no greater than 0.83 ms per hop.

5.4 Benchmark 4: 3D Space Mobility Model

5.4.1 Description

In this test, we look at the cumulative data performance of the different routing protocols as the number of mobile transmitters grows in 3D space; we also compare the results using two different mobility models. This is the first benchmark testing each routing protocol's ability to handle dynamic routes caused by mobile nodes.

5.4.2 Setup

There are no moving impairments or jammers in this test, but both the Friis and Nakagami propagation models are installed on the wireless channel. The network is shown in Figure 5.10. The 3D field is 150 km × 150 km × 2 km. There is a single stationary receiver located on the ground in the middle of the simulation field. The number of mobile transmitters increases from 1 to 40, filling up the space in a random fashion and traveling at roughly 1200 m/s. Each test is run 8 times with different random seed numbers and the average cumulative data received value is used. The sending nodes transmit at 80 kb/s with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the senders transmit data for 160 seconds. The theoretical maximum cumulative data received is

$$80 \text{ kb/s} \times n_{Tx} \times 50\% \times 160 \text{ s} = 6.4 \text{ Mb} \times n_{Tx} \tag{5.2}$$

where n_{Tx} is the number of transmitters in the simulation space. The first set of test runs uses the Random Waypoint mobility model, and then the benchmark is run again using the Gauss-Markov mobility model, which should yield results that better approximate real world movement and performance.

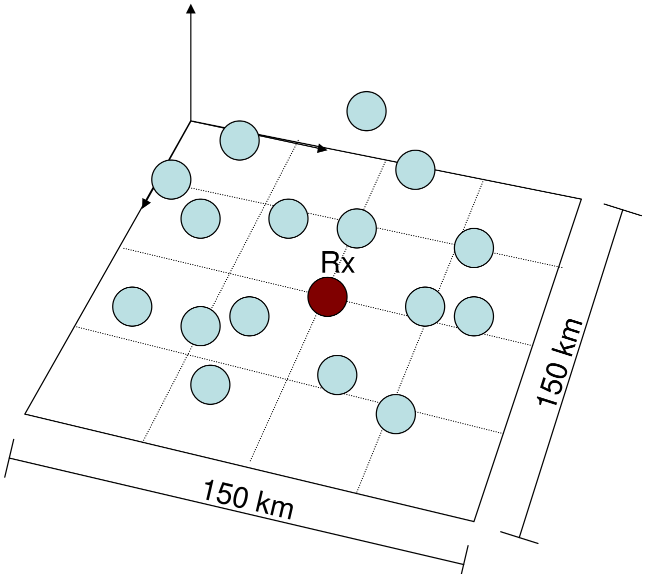


Figure 5.10. Setup for the 3D mobility model benchmark

The Random Waypoint parameters used in this test are identified in Table 5.1. The Gauss-Markov parameters used in this test are outlined in Table 4.2 and the mean velocity is a uniform random variable between 1100 m/s and 1300 m/s.

5.4.3 Expected Results

Nodes that are outside the range of the receiver must rely upon intermediate nodes through which a route can be created and packets can be sent. Since this

Table 5.1. Random waypoint mobility model parameters

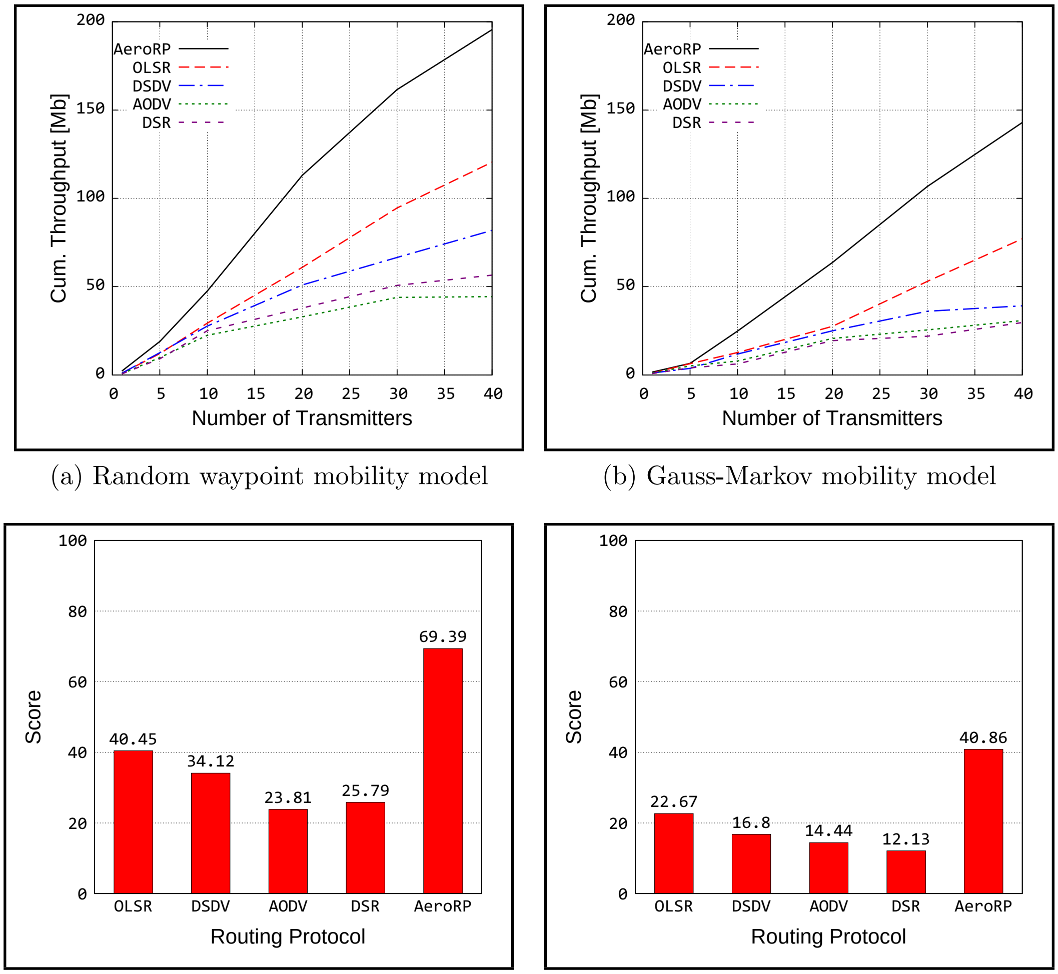
Parameter	**Setting**
Speed	1200 m/s
Pause	0 sec

will not always be the case, this test shows the real benefit of the store and forward algorithm; there are times when nodes cannot find a complete path to the receiver, and it pays to be able to carry those packets and take advantage of progressive hop opportunities. Therefore, AeroRP is expected to outperform the legacy routing protocols.

5.4.4 Results

With reference to Figure 5.11, AeroRP is able to move packets towards the destination node even when a complete path is not available, leading to more packets received. Even with 40 transmitters, the network is considered rather small, and OLSR appears to have no difficulty in proactively setting up and using routes to the receiver, so long as a complete route exists. It appears that DSDV loses a little time in setting up routes reactively and suffered a small penalty as a result.

The mobility model does make a difference. Performance suffered using the Gauss-Markov model just as if it introduced an impairment of some kind into the simulation. The reduction in total data delivered for each routing protocol is approximately 40%. This is attributed to the belief that the nodes will tend to be closer to each other when using the Random Waypoint mobility model, which does not produce paths that bounce off the walls of the simulation field. By contrast, the Gauss-Markov and Random Direction mobility models generate paths that



(a) Random waypoint mobility model

(b) Gauss-Markov mobility model

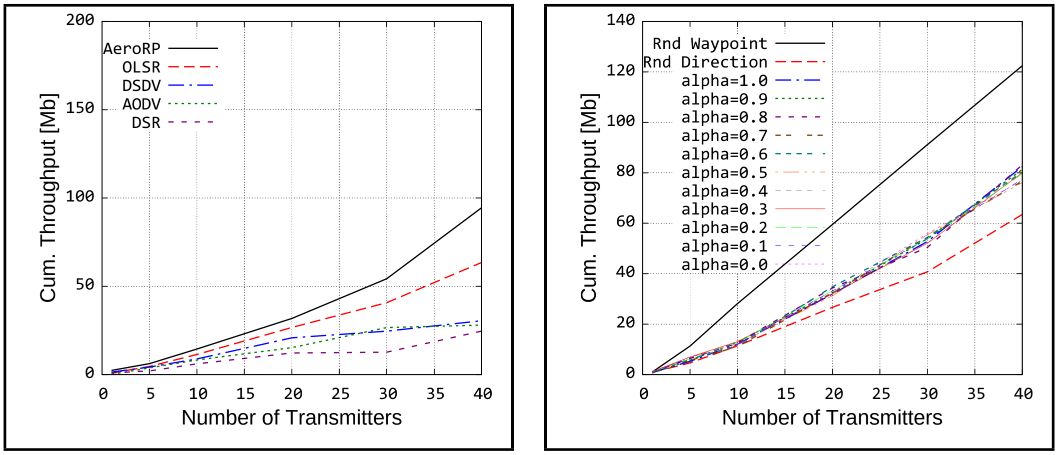
(c) Random waypoint scores

(d) Gauss-Markov scores

Figure 5.11. 3D space random mobility performance and scores

bounce off the walls, so their nodes will tend to be spaced further apart.

To test this hypothesis, benchmark 4 is repeated using the Random Direction mobility model. Furthermore, simulations are generated using the Gauss-Markov mobility model testing sensitivity to changes in the α parameter. Figure 5.12a shows the results of this benchmark using the Random Direction mobility model. AeroRP still outperforms the other routing protocols, but the results are even lower than when using the Gauss-Markov model. Figure 5.12b shows the results of the sensitivity testing. For homogeneity, only the OLSR routing protocol is tested. The Gauss-Markov mobility model is used and the independent variable is the α parameter. For comparison purposes, results from the Random Waypoint and Random Direction models are also plotted. The results indicate that while the routing protocol performance is highly dependent upon the mobility model chosen, within the Gauss-Markov model itself, performance is only slightly dependent upon the selection of α.



(a) Random direction mobility model

(b) OLSR Gauss-Markov α sensitivity

Figure 5.12. Mobility model performance and sensitivity

5.5 Benchmark 5: 3D Space with Jammers

5.5.1 Description

In this benchmark, we analyze the performance of a mobile network in 3D space as the number of jammers increases from 0 to 7. The presence of challenges helps us identify protocols that can deliver packets even in the face of intense interference.

5.5.2 Setup

As in the previous benchmark, both the Friis and Nakagami propagation models are installed on the wireless channel. But in this benchmark, there are always 10 mobile transmitter nodes and one stationary receiver node. The network is shown in Figure 5.13. The 3D field is 150 km × 150 km × 2 km. The number of jammers grows from 0 to 7, making communications in the space increasingly difficult. Both the transmitters and the jammers will use the Gauss-Markov mobility model traveling at roughly 1200 m/s. Each test is run 8 times with different random seed numbers and the average cumulative data value is used. The sending nodes transmit at 80 kb/s each with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the senders transmit data for 100 seconds. The theoretical maximum cumulative data received is

$$80 \text{ kb/s} \times 10 \times 50\% \times 100 \text{ s} = 40 \text{ Mb} \tag{5.3}$$

5.5.3 Expected Results

The cumulative data of each routing protocol is expected to be inversely proportion to the number of jammers present, as depicted in Figure 5.14. The store-and-haul mechanism combined with the limited strong transmission range should give AeroRP better performance than the other protocols.

5.5.4 Results

AeroRP's performance is notably better than the other protocols, as illustrated in Figure 5.15. Jammers cause the network to become more partitioned making it

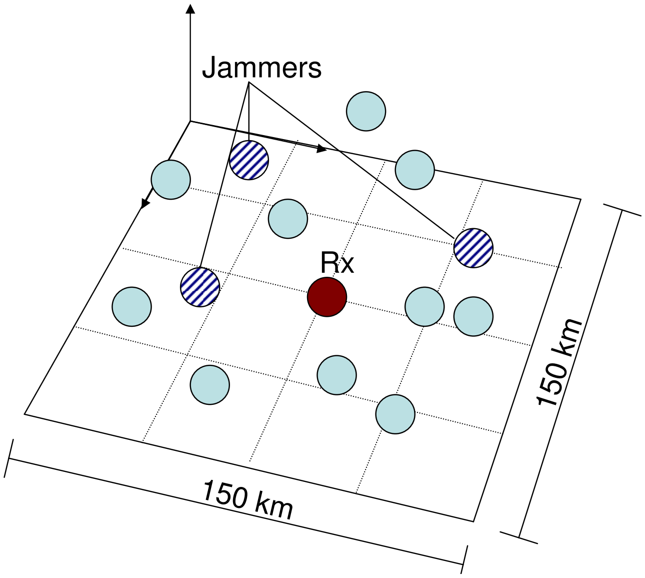


Figure 5.13. Network setup: 3D space benchmark with jammers

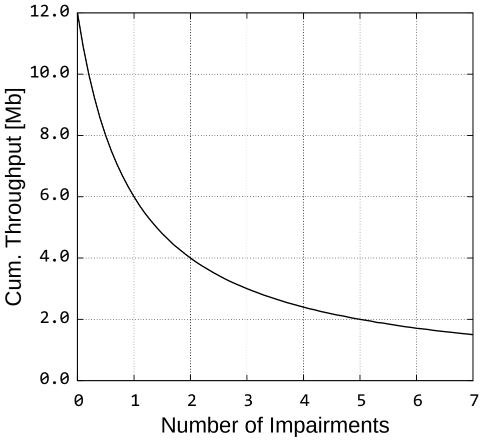
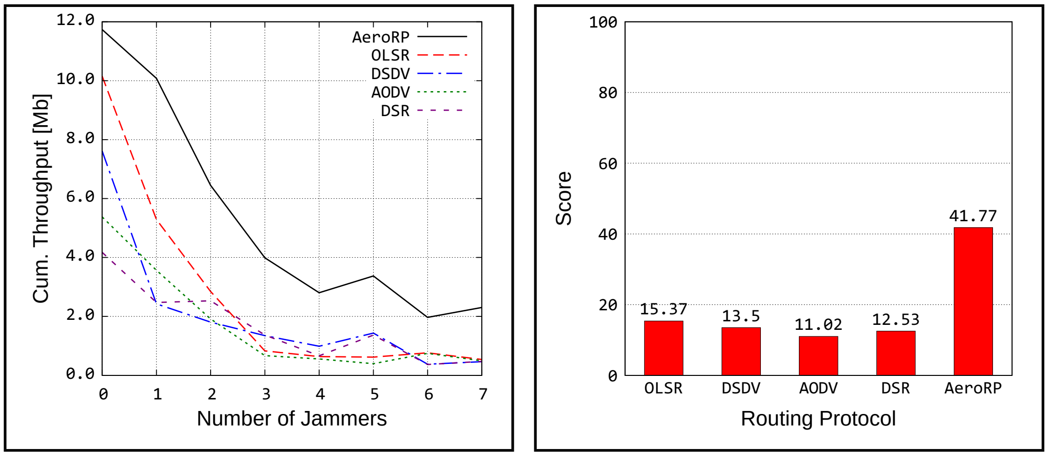


Figure 5.14. Expected performance as jammers increase

harder for complete routes to be set up to the receiver. Opportunistic forwarding and the store-and-haul mechanisms are able to incrementally move packets to the destination over time and improve the performance. As expected, the general shape of the performance curve follows an inverse relationship. Of the traditional MANET routing protocols, OLSR performed better when the number of jammers was small, and DSDV performed better as the number of jammers increased.



(a) Cumulative data received

(b) Scores

Figure 5.15. Results of 3D space benchmark with jammers

5.6 Benchmark 6: Transmission Power Test in 3D Space

5.6.1 Description

It is very common for MANET and telemetry networks to be concerned with node power consumption. In this benchmark, we analyze the performance of 10 randomly moving network nodes in 3D space as their transmission power grows from 20 to 48 dBm. This will test each protocol for its ability to perform well in power constrained applications. As usual, performance will be judged by analyzing the cumulative data characteristics for each protocol.

5.6.2 Setup

As in the previous benchmarks, both the Friis and Nakagami propagation models are installed on the wireless channel. There are 10 mobile transmitter nodes, one stationary receiver node, and zero jammers or moving impairments in this test. The network is shown in Figure 5.16. The 3D field is 150 km × 150 km × 2 km and the independent variable is transmission power, which is

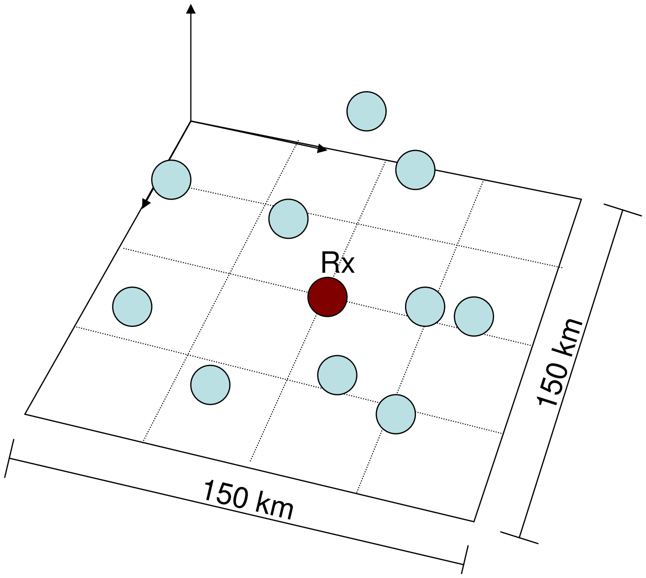


Figure 5.16. Network setup for transmission power benchmark

tested in a range between 20 and 48 dBm. The ten transmitters use a Gauss-Markov mobility model traveling at roughly 2400 m/s. Each test is run 8 times with different random seed numbers and the average cumulative data value is used. The sending nodes transmit at 80 kb/s each with a 50% duty cycle. The benchmark allows 30 seconds for the nodes to set up and then the senders transmit data for 160 seconds. The theoretical maximum cumulative data received is

$$80 \text{ kb/s} \times 10 \times 50\% \times 160 \text{ s} = 64 \text{ Mb} \tag{5.4}$$

Since transmission power affects the transmission range, tests were run to see what the ranges are at the different transmission power levels. Those values are then used in the AeroRP `TransmissionRange` parameter. The tested range results were very similar to the theoretically expected transmission ranges from the Friis transmission equation.

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2 \tag{5.5}$$

Equation 5.5 shows the basic Friis transmission equation. G_t and G_r are the transmitter and receiver antenna gain values. In these benchmarks, these are set to 1 dBi, which translates to $10^{\frac{1}{10}} = 1.258$ for the Friis equation. Lambda is 0.125 meters, which corresponds to a frequency of 2.4 GHz. Transmission power, P_t, is converted from dBm units into Watts (for example, 40 dBm = 10 Watts). The receiver power is set at –86.93 dBm ($2.0262e^{-12}$ Watts), which is the signal strength calculated using the Friis equation with 40 dBm transmission power and a transmission range of 27.8 km. Here is an example of how to calculate the transmission distance using 20 dBm (0.1 Watts) transmission power:

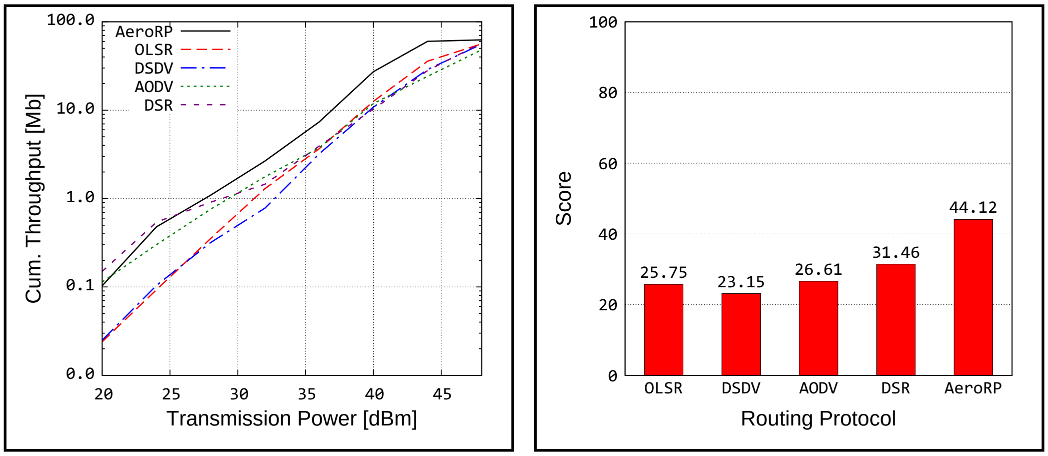
$$R = \frac{\lambda}{4\pi} \sqrt{\frac{P_t G_t G_r}{P_r}} = \frac{0.125}{4\pi} \sqrt{\frac{0.1 \times 1.258 \times 1.258}{2.0262e^{-12}}} = 2.78 \text{ km} \tag{5.6}$$

5.6.3 Expected Results

It is expected that total data received will grow exponentially with transmission power and then taper off as it approaches the theoretical maximum value of 64 Mb. Due to AeroRP's ability to handle more challenging environments better, it is expected to perform better than the other routing protocols. However, it is uncertain how the other protocols will fare against one another.

5.6.4 Results

From the results shown in Figure 5.17, it is surprising to see how well AODV and DSR performed with low transmission power compared to the other protocols. However, as power increased, their delivered data performance waned with respect



(a) Cumulative data received

(b) Scores

Figure 5.17. Transmission power benchmark results

to AeroRP. This test suggests that, among the traditional MANET routing protocols, AODV and DSR might make good candidates for low-power networks, especially considering the fact that they are both reactive protocols which is also important to power constrained networks. AeroRP performed very well again, transmitting nearly every packet at 44 dBm. The results suggest that AeroRP with ferrying yields a 3 to 5 dBm advantage over the other routing protocols. If power consumption is more important than delay, AeroRP is a very worthy candidate for low-power applications. Scores were based upon the total data delivered divided by the maximum possible data delivered, then divided by the node density. Node density is a ratio of the total transmission area covered by nodes to the total area of the simulation field. Table 5.2 illustrates how node density affects the score in the calculation of OLSR's performance.

Table 5.2. OLSR benchmark results and score calculation

Tx Power	**Tx Range**	**Node Density (with 10 nodes)**	**Data Received**	**Score**
20 dBm	2.78 km	$\frac{10\pi \times 2.78^2}{150^2} = 0.01$	0.024 Mb	$\frac{0.024}{64 \times 0.011} = 0.03$
24 dBm	4.4 km	$\frac{10\pi \times 4.4^2}{150^2} = 0.027$	0.093 Mb	$\frac{0.093}{64 \times 0.027} = 0.05$
28 dBm	7 km	$\frac{10\pi \times 7^2}{150^2} = 0.068$	0.353 Mb	$\frac{0.353}{64 \times 0.068} = 0.08$
32 dBm	11 km	$\frac{10\pi \times 11^2}{150^2} = 0.169$	1.294 Mb	$\frac{1.294}{64 \times 0.169} = 0.12$
36 dBm	17.5 km	$\frac{10\pi \times 17.5^2}{150^2} = 0.43$	3.666 Mb	$\frac{3.666}{64 \times 0.428} = 0.13$
40 dBm	27.8 km	$\frac{10\pi \times 27.8^2}{150^2} \geq 1.0$	12.58 Mb	$\frac{12.582}{64 \times 1.0} = 0.20$
44 dBm	44 km	$\frac{10\pi \times 44^2}{150^2} \geq 1.0$	35.82 Mb	$\frac{35.819}{64 \times 1.0} = 0.56$
48 dBm	70 km	$\frac{10\pi \times 70^2}{150^2} \geq 1.0$	56.40 Mb	$\frac{56.403}{64 \times 1.0} = 0.88$
Score				Avg × 100% = 25.75%

5.7 Benchmark 7: Short Window of Opportunity

5.7.1 Description

The purpose of this benchmark is to investigate a plausible scenario which presents MANET challenges that aren't caused by physical layer imperfections. In this benchmark, we analyze the performance of each routing protocol in a scenario in which the time to set up a route and transmit data is short. A series of nodes are spaced 50 km apart in follow-the-leader fashion, traveling at 500 m/s towards a target 250 km away from home base. When the nodes are 15 km (30 seconds) from the target distance, they transmit data back to the base for ten seconds at 800 kb/s. Upon reaching the target distance, the node heads back to base. Since nodes traveling in the same direction are 50 km apart, they are too far apart to transmit data through directly; they must use nodes passing in the opposite direction to bridge the transmission gap to get the data back to base as shown in Figure 5.18. Due to the speed and distance between these nodes, there is only a 10-second window of opportunity to set up a route and transmit the data to the base.

5.7.2 Setup

Nodes travel at 500 m/s and are placed 50 km apart in a line heading towards the target. Node transmission power is 40 dBm which provides a transmission range of roughly 30 km. Nodes transmit a single 10-second data message at 800 kb/s when they are between 235 and 240 km from base (10 to 15 km from the target distance) and then begin their return once they reach the target distance of 250 km. Figure 5.19 illustrates that a transmission window of about 10 seconds exists followed by 40 seconds in which a complete route does not exist. Starting with time t=0, nodes A and D are within transmission range, as are nodes B and C; this allows a route to be set up from node C to B, B to D, and from D to A. After time t=10 seconds, nodes B and D are out of transmission range and a route no longer exists between them. Finally, at time t=50 seconds, nodes A and C are within range and a new route can be established and used for up to 10 seconds again. To save time and simplify the simulation, the setup includes a full chain of returning nodes so that data can be sent all the way back to base starting in the first transmission window.

The benchmark test runs for a total of 600 seconds. The first 20 seconds are used to perform initial setup and bootstrapping routines, and then once every 100 seconds the node that is 15 km from the target begins its 10 second transmission. The theoretical maximum cumulative data received is

$$800 \text{ kb/s} \times 10 \text{ s} \times 6 = 48 \text{ Mb} \tag{5.7}$$

In ns-3, the ONOFF application is used to transmit constant bitrate traffic from the moving nodes to the stationary base. Since this benchmark requires each node to transmit for 10 seconds only one time, the ON time was initially set to 10 seconds and the OFF time to 2000 seconds. However, the result was that no data was ever transmitted. As it turns out, the ONOFF application runs the OFF time first, so no data would be sent until after 2000 seconds, at which time the test had already finished. Therefore, the OFF time is set to 0 seconds and the MaxBytes parameter is set to 1000 kilobytes, which equates to 10 seconds of data at 800 kb/s. An OFF time of 0 seconds implies that the transmission does not stop until the end of the simulation, but using the MaxBytes parameter tells the application to stop sending packets after a certain amount of data has been sent; in this case, data is sent for only 10 seconds.

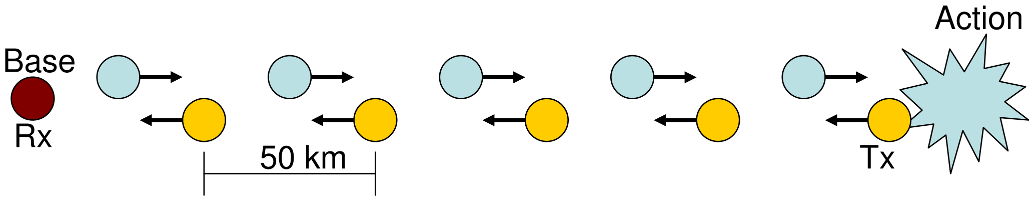


Figure 5.18. Setup for short window of opportunity benchmark

5.7.3 Expected Results

Since nodes only transmit for 10 seconds one time (when they are 15 km from the target), are spaced 50 km apart, and travel at 500 m/s, transmissions will only occur every 100 seconds. However, 10-second windows of opportunity occur every 50 seconds as shown in Figure 5.19. This means that AeroRP with the store-and-haul feature, called ferrying, will have twice the number of opportunities as the other protocols to transport data back to the base node since it can transmit

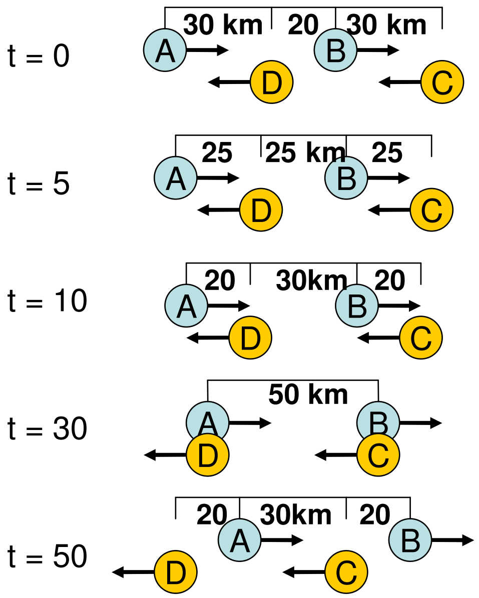
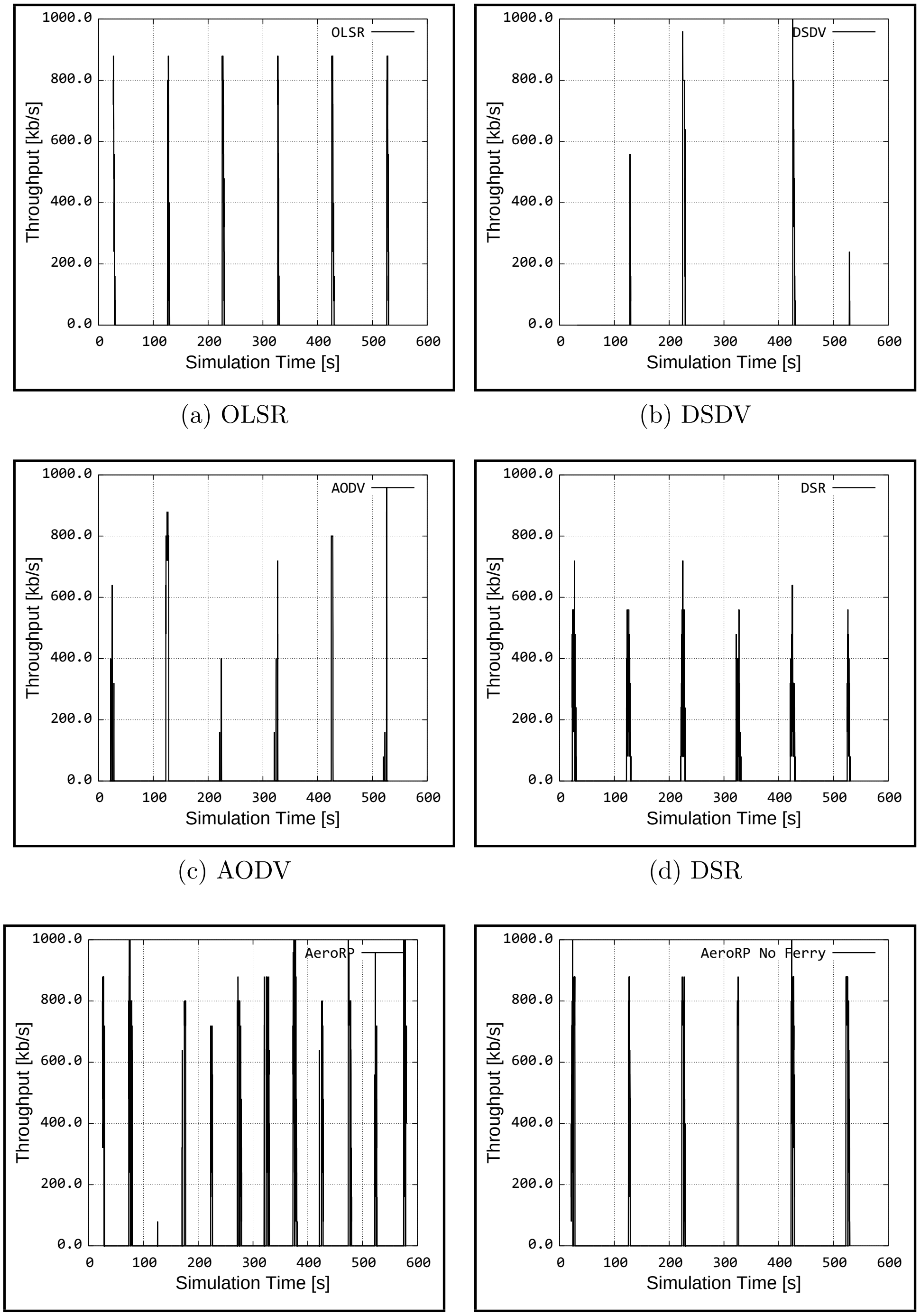


Figure 5.19. 10-second transmit window every 50 seconds

during the otherwise unused windows. To test the difference that ferrying makes, this benchmark will test AeroRP twice, once with ferrying and once without.

5.7.4 Results

Scores in Figure 5.21 are based upon the cumulative data received divided by the maximum possible data received of 48 Mb. As expected, AeroRP outperforms the other protocols. By ferrying and taking advantage of all the windows of opportunity, AeroRP gets roughly three times as many packets through compared to the legacy MANET protocols. This is also supported by Figure 5.20e, which shows that AeroRP with ferrying was able to take advantage of each 50-second window of opportunity to transmit data. AeroRP without ferrying also outperformed the other protocols. This is attributed to the fact that AeroRP does not need to wait for a complete route to begin transmitting data. At the moment when the opportunity presents itself, packets are routed towards the base node. According to the



(a) OLSR

(b) DSDV

(c) AODV

(d) DSR

(e) AeroRP

(f) AeroRP no Ferrying

Figure 5.20. Window of opportunity benchmark results

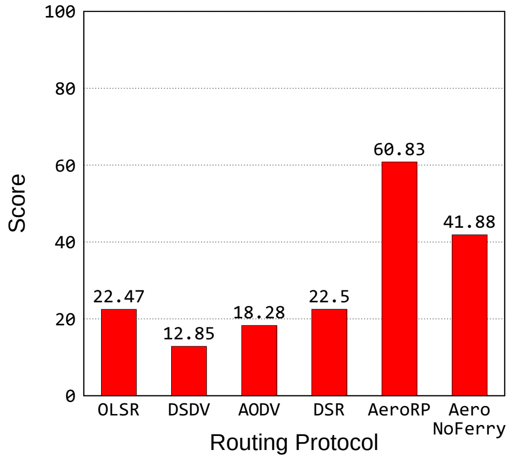


Figure 5.21. Window of opportunity benchmark scores

results, about 31% of AeroRP's cumulative data performance can be attributed to ferrying. The extremely dynamic nature of this test effectively eliminates any benefit that may come from using a proactive routing protocol. We should expect the performance of OLSR, DSDV, AODV, and DSR to be quite similar. Yet it is interesting that their performances are so different. OLSR appears to send some packets through during each window every 100 seconds, while DSDV does not. DSR has lower throughput than OLSR, yet utilizes the 10-second window better, allowing it to perform overall as well as OLSR.

5.8 Benchmark 8: Urban Canyon Test

5.8.1 Description

This benchmark simulates an urban canyon environment to test each routing protocol's ability to handle dense urban settings in which buildings cast very large shadows and strong reflections cause significant fast fading. As shown in Figure 5.22, the dense urban test involves many stationary buildings in a grid-like pattern, and many mobile nodes that move between and within the buildings.

5.8.2 Setup

To simulate this environment statistically, others have used Log-Distance propagation loss models [28]. However, since we can actually add impairments at specific locations with the `MovingPropagationLossModel`, and these will provide the slow shadow fading needed, we can continue to use the Friis propagation loss model. For fast fading channel characteristics which would be expected in this environment, the Nakagami model is used with $m = 10$.

The test uses an 11×11 grid of stationary buildings, simulated using the MovingPropagationLossModel with a constant position mobility model. There are 30 transmitting nodes and 1 receiver node (denoted with "Rx"), all moving at approximately 2 m/s. The buildings are spaced 60 meters apart on center, have a radius of 22 meters, and incur a propagation loss of 15 dB each. As a result, the simulated buildings are 44 meters wide and the simulated streets are 16 meters wide. The simulation allows 30 seconds for initialization and setup, followed by 270 seconds of transmissions. The 30 sending nodes transmit at 80 kb/s with a duty cycle of 50% so the maximum traffic that could be successfully transmitted is

$$80 \text{ kb/s} \times 30 \times 50\% \times 270 \text{ s} = 324 \text{ Mb} \tag{5.8}$$

where the transmission power is 20 mW, or 13 dBm, and node movement is simulated using the Gauss-Markov mobility model with $\alpha = 0.85$.

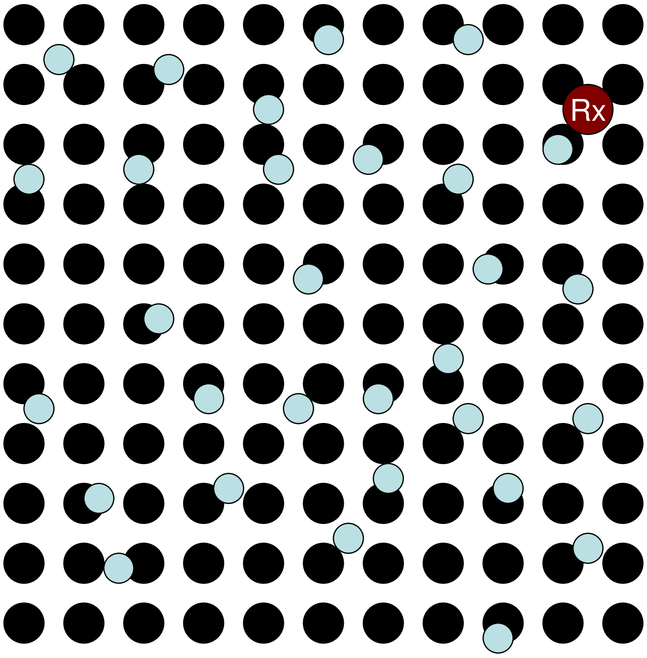


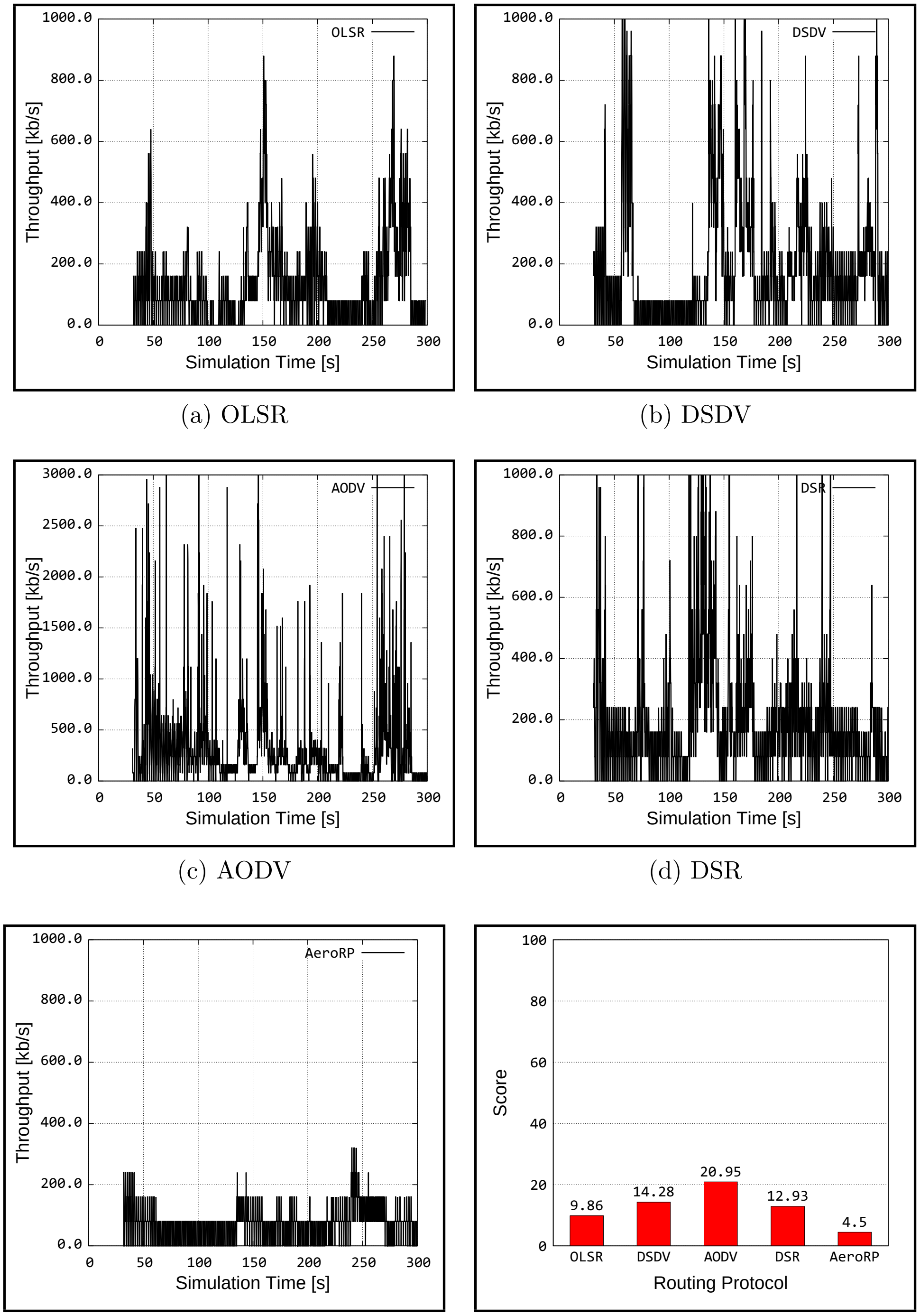
Figure 5.22. Network setup for urban canyon test

5.8.3 Expected Results

Due to the many buildings present, the topology is fragmented and disconnected much of the time. It may be very difficult to get nodes to successfully transmit data to the receiver node if they are are not in direct line-of-sight of each other. AeroRP should handle the intermittent connectivity better and use opportunities to forward packets towards the receiver, but it should be difficult to find those opportunities.

5.8.4 Results

The results in Figure 5.23 indicate that this is a very challenging test. There are very few opportunities for complete end-to-end paths. Scores are based upon the protocol's total data delivered divided by the maximum possible data delivered



(a) OLSR

(b) DSDV

(c) AODV

(d) DSR

(e) AeroRP

(f) Scores

Figure 5.23. Dense urban canyon benchmark results

of 324 Mb. In this benchmark, AODV is able to take advantage of small windows of opportunity and send more packets during those windows than OLSR. Finally we see a challenge that AeroRP does not handle well. The reason for AeroRP's poor performance is likely due to its optimistic forward routing algorithm. AeroRP assumes that nodes that are closer to the target, or have a shorter time to intercept with the target, are better handoff candidates. This is normally true in airborne environments, but in this tough urban environment, there are many large shadows cast by our simulated buildings, and picking the closer targets may lead to dead end situations. The dead end node may be close to the target, but buildings incur too much signal loss to transmit to it, and no closer candidates can be found, resulting in lost packets.

Chapter 6

Conclusions and Future Work

This chapter summarizes this thesis, with conclusions in Section 6.1 and suggestions for future work presented in Section 6.2.

6.1 Conclusions

The problem with simplistic simulation models was discussed, prompting the requirement for more complex and realistic simulation environments. Furthermore, we covered the need to benchmark wireless protocols so their performance could be measured and compared upon equal grounds. As the British scientist Lord Kelvin [29] observed, "If you cannot measure it, you cannot improve it." New models were described in detail that add the needed complexity and realism to these simulations. These include the memory-based 3-D Gauss-Markov mobility model to better simulate highly-dynamic airborne network nodes, and a more realistic physical layer channel environment that includes a new localized shadow fading model and jammers. We highlighted some challenges in designing a MANET routing protocol and some basic characteristics of the different routing algorithms to gain an understanding on how they may behave in real world networks.

Details were given about the simulation models and parameter settings that were used to make the benchmarks more challenging and realistic. Then we discussed the importance of the tests and simulation scenarios that would make up the different benchmarks. The routing protocols that were selected to be benchmarked were examined along with their expected strengths. Finally, each routing protocol was run through each of the eight benchmarks and given performance scores.

The primary contributions of this thesis are:

- Describing and implementing a 3-D version of the Gauss-Markov mobility model in ns-3
- Describing and implementing wireless impairment models that contribute to a more realistic and challenging simulation channel environment
- Defining fundamental tests that can be used to benchmark and compare the performance of different MANET routing protocols
- Analyzing these benchmarks against five MANET routing protocols, including AeroRP, and presenting the results and performance scores of these protocols

AeroRP outperforms the traditional MANET protocols in nearly every benchmark. The two mechanisms that give AeroRP an advantage in challenging real world environments are the ability to forward packets without waiting for a complete end-to-end route to be determined, and the ability to store-and-haul traffic during interruptions in connectivity. The use of these features must be carefully considered with respect to the environment that the network will operate in. As we saw in the dense urban benchmark, AeroRP can suffer from dead end situations if there are too many obstructions between the nodes.

Challenging environmental conditions that cause interruptions in connectivity are devastating to the traditional MANET protocols we tested. Without knowledge of a complete route, they must wait until the conditions improve before sending packets, even if the problem is localized to a small area of the network. AeroRP is able to mitigate these disruptions to some degree by opportunistically forwarding packets whenever possible. High speed and mobility create short windows of opportunity; those routing protocols that can become aware of the window quickly and start sending packets immediately will perform better than those that cannot. These are situations in which AeroRP has the advantage over the other protocols we tested.

It was discovered that DSR and AeroRP might be good choices in power constrained MANET applications. DSR shows good performance at low transmission power and it is a reactive, on-demand, protocol, which also helps reduce power consumption. AeroRP also shows good performance at low power, and is capable of storing data for long periods of time, which would be a real asset in power constrained sensor networks that use sleeping nodes.

It was shown that the selection of a mobility model affects the outcome of the simulation. It appears to affect all routing protocols, but some more than others. For example, using the results from benchmark 4, the performance difference between the Random Waypoint and the Gauss-Markov model was 41% for AeroRP, but was over 50% for DSDV. By using a more realistic mobility model, we introduce a more realistic, and hence more challenging, environment. And as we have discovered, protocols that have mechanisms to handle those challenges better will not be affected as much as those that don't.

6.2 Future Work

A number of items stand out which would further this work.

1. The Rician fast fading algorithm should be implemented in ns-3. This would allow future simulations of MANET protocols to use a fast fading model more suited to the kinds of environments in which they are used. This is especially true for airborne networks; the use of the Friis path loss model alone is not challenging enough, and the addition of the Nakagami fast fading model is too harsh.

2. The benchmarks in this thesis should be run again when the ns-3 implementation of the DSR routing algorithm is complete. Likewise, these tests should be run again when there is another disruption-tolerant routing protocol available in ns-3. Naturally, we should test the performance when AeroTP and AeroNP become available in ns-3.

3. These benchmarks were designed with total data delivered as the primary performance metric. This is reasonable if cumulative received data is the overall design objective. However, the benchmarks could be expanded to include scores for packet delay and channel overhead as well.

4. Benchmark 4 could be expanded to include more mobility models. This also implies that more mobility models will need to be added to ns-3.

References

[1] Timo Bingmann. Accuracy Enhancements of the 802.11 Model and EDCA QoS Extensions in ns-3. In *Institute of Telematics, Decentralized Systems and Network Services Research Group*, Universität Karlsruhe, April 2009.

[2] Azzedine Boukerche. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks*, chapter 5. Wiley-IEEE Press, 2008.

[3] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), February 2007.

[4] C. Perkins, E. Belding-Royer, and S. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[5] Justin P. Rohrer, Abdul Jabbar, Egemen K. Çetinkaya, Erik Perrins, and James P.G. Sterbenz. Highly-Dynamic Cross-Layered Aeronautical Network Architecture. *IEEE Transactions on Aerospace and Electronic Systems (TAES)*, 2010.

[6] Abdul Jabbar and James P.G. Sterbenz. Aerorp: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments. In

International Telemetering Conference (ITC) 2009, Las Vegas, NV, October 2009.

[7] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[8] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *SIGCOMM '94: Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 234–244, New York, NY, USA, 1994. ACM.

[9] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[10] Alexey Stepin, Yaroslav Lyssenko, and Anton Shilov. Fermi Light: Nvidia GeForce GTS 450 Review. *X-bit Labs.*

[11] Mineo Takai, Rajive Bagrodia, Ken Tang, and Mario Gerla. Efficient wireless network simulations with detailed propagation models. *Wirel. Netw.*, 7:297–305, May 2001.

[12] Dan Broyles, Abdul Jabbar, and James P. G. Sterbenz. Design and analysis of a 3–D Gauss-Markov Mobility Model for Highly-Dynamic Airborne Networks. In *Proceedings of the International Telemetering Conference (ITC)*, San Diego, CA, October 2010. (to appear).

[13] Egemen K. Çetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and James P.G. Sterbenz. A Comprehensive Framework to Simulate Network

Attacks and Challenges. In *RNDM'10 - Second International Workshop on Reliable Networks Design and Modeling*, Moscow, Russia, October 2010. to appear.

[14] Tracy Camp, Jeff Boleng, and Vanessa Davies. A Survey of Models for Ad Hoc Network Research. *Wirelss Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, 2(5):483–502, 2002.

[15] Vanessa Ann Davies. Evaluating Mobility Models within an Ad-hoc Network. Master's thesis, Colorado School of Mines, 2000.

[16] C. Mbarushimana and A. Shahrabi. Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks. *Advanced Information Networking and Applications Workshops, International Conference on*, 2:679–684, 2007.

[17] J. Jetcheva D. A. Maltz, J. Broch and D. B. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks. *IEEE Journal of Selected Areas of Communications*, 17:1439–1453, August 1999.

[18] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*, chapter 4, pages 371–375. Addison Wesley, 4th edition, 2008.

[19] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*, chapter 4, pages 375–382. Addison Wesley, 4th edition, 2008.

[20] A. Widmer M. Mauve and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. *IEEE Network*, 15(6):30–39, 2001.

[21] A. Hooke L. Torgerson R. Durst K. Scott K. Fall V. Cerf, S. Burleigh and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838, 2007.

[22] James P. G. Sterbenz, Rajesh Krishnan, Regina Rosales Hain, Alden W. Jackson, David Levin, Ram Ramanathan, and John Zao. Survivable mobile wireless networks: issues, challenges, and research directions. In *WiSE '02: Proceedings of the 1st ACM workshop on Wireless security*, pages 31–40, New York, NY, USA, 2002. ACM.

[23] Ben Liang and Zygmunt J. Haas. Predictive distance-based mobility management for multidimensional PCS networks. *IEEE/ACM Trans. Netw.*, 11:718–732, October 2003.

[24] NS3. The ns-3 network simulator. url http://www.nsnam.org/.

[25] Erik Perrins Justin P. Rohrer, Abdul Jabbar and James P.G. Sterbenz. Cross-Layer Architectural Framework for Highly-Mobile Multihop Airborne Telemetry Networks. In *IEEE MILCOM*, November 2008.

[26] Ricky Dye Michael Rice and Kenneth Welling. Narrowband Channel Model for Aeronautical Telemetry. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1371–1376, October 2000.

[27] Kevin Peters. Design and Performance Analysis of a Geographic Routing Protocol for Highly Dynamic MANETs. Master's thesis, The University of Kansas, 2010.

[28] Daniel B. Faria. Modeling Signal Attenuation in IEEE 802.11 Wireless LANs. Technical report, Stanford University, 2005.

[29] Lord Kelvin. Electrical Units of Measurement. In *Popular Lectures and Addresses*, volume 1. 1883.