JAN 2 5 2007

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 26-10-2006	2. REPORT TYPE DISSERTATION	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE CREDIBLE MOBILE AD HOC NETWORK SIMULATION-BASED STUDIES.	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) MAJ KURKOWSKI STUART H	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) COLORADO SCHOOL OF MINES	8. PERFORMING ORGANIZATION REPORT NUMBER CI07-0013

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) THE DEPARTMENT OF THE AIR FORCE AFIT/ENEL, BLDG 16 2275 D STREET WPAFB OH 45433	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		180	19b. TELEPHONE NUMBER *(Include area code)*

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

CREDIBLE MOBILE AD HOC NETWORK
SIMULATION-BASED STUDIES

by

Stuart H. Kurkowski

Stuart Kurkowski's

Dissertation

Credible Mobile Ad Hoc Network Simulation-based Studies

Abstract

Simulation is the research tool of choice for a majority of the mobile ad hoc network (MANET) community. However, while the use of simulation has increased, the credibility of the simulation results has decreased. To determine the state of MANET simulation studies, we surveyed the 2000-2005 proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). We present the results of our survey and summarize common simulation study pitfalls. We develop standards and algorithms that help enable MANET researchers to move toward the goal of simulation-based research with credible scenarios. We also document a large variable analysis of the Location Aided Routing (LAR) protocol. This study discovers several variables that have a significant impact on LAR performance, but are not always considered in a MANET simulation study. Finally, we discuss tools we created that aid the development of credible simulation studies. We offer these results to the community with the hope of improving the credibility of MANET simulation-based studies.

Stuart Kurkowski's

Dissertation

Credible Mobile Ad Hoc Network Simulation-based Studies

Abstract

Simulation is the research tool of choice for a majority of the mobile ad hoc network (MANET) community. However, while the use of simulation has increased, the credibility of the simulation results has decreased. To determine the state of MANET simulation studies, we surveyed the 2000-2005 proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). We present the results of our survey and summarize common simulation study pitfalls. We develop standards and algorithms that help enable MANET researchers to move toward the goal of simulation-based research with credible scenarios. We also document a large variable analysis of the Location Aided Routing (LAR) protocol. This study discovers several variables that have a significant impact on LAR performance, but are not always considered in a MANET simulation study. Finally, we discuss tools we created that aid the development of credible simulation studies. We offer these results to the community with the hope of improving the credibility of MANET simulation-based studies.

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mathematical and Computer Sciences).

Golden, Colorado

Date _____

Signed: _____
Stuart H. Kurkowski

Approved: _____
Dr. Tracy K. Camp
Associate Professor

Golden, Colorado

Date _____

_____
Dr. Graeme Fairweather
Professor and Head,
Department of Mathematical and
Computer Sciences

ii

ABSTRACT

Simulation is the research tool of choice for a majority of the mobile ad hoc network (MANET) community. However, while the use of simulation has increased, the credibility of the simulation results has decreased. To determine the state of MANET simulation studies, we surveyed the 2000-2005 proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). We present the results of our survey and summarize common simulation study pitfalls. We develop standards and algorithms that help enable MANET researchers to move toward the goal of simulation-based research with credible scenarios. We also document a large variable analysis of the Location Aided Routing (LAR) protocol. This study discovers several variables that have a significant impact on LAR performance, but are not always considered in a MANET simulation study. Finally, we discuss tools we created that aid the development of credible simulation studies. We offer these results to the community with the hope of improving the credibility of MANET simulation-based studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview: MANET Simulation-based Studies

Mobile Ad Hoc Networks (MANET) are wireless mobile nodes that cooperatively form a network without infrastructure. Because there is no coordination or configuration prior to setup of a MANET, there are several challenges. These challenges include routing packets in an environment where the topology is changing frequently, wireless communications issues, and resource issues such as limited power and storage. The leading way to research solutions to these difficult MANET challenges is simulation.

Despite the fact that there are quality simulators already developed and in use today, the work for a simulation-study designer is far from complete. There are numerous factors involved in conducting credible simulation-based MANET research.

First of all, a simulation-study designer must decide upon the type of simulation. Discrete-event simulations have two main types: terminating (finite-time horizon) and steady-state (non-terminating). In addition to selecting the type of simulation, the researcher must validate the simulation model. The researcher must ensure the simulator is running correctly in his or her environment. The researcher must also verify that his or her implementation of a particular protocol or set of variables is correct (or at least in line with the protocol specifications). Verification can be difficult in research where there is no truth data for the protocol or implementation.

Once the simulator and code have been validated and verified, the simulation study must be executed. Because discrete-event simulators are based on randomness, there are many potential credibility issues in the way the simulation executions are handled. The scenarios used for input must sufficiently exercise a protocol. The numerous variables of a simulator must be set appropriately. Additional issues range from seeding the random number generator, to managing parallel simulations, to monitoring steady-state.

Once a researcher completes all of the effort to generate results, there is still work to be done in analyzing and publishing results. There are several statistically sound ways to parse the output data to address covariance, capture isolated events, remove initialization bias, and construct confidence intervals. There are also many dos and don'ts for documenting and publishing results.

The steps for conducting a simulation study are many. The chance to compromise the credibility of the study is great. As the MANET community moves closer to real-world implementation of MANETs, the simulation research must be credible.

1.2 Motivation

In contrast to other fields of research that use the scientific method, the computer network communities have fallen prey to the "computer scientific method" [72]. The computer scientific method lacks rigor. In the computer scientific method, researchers change several variables at once until simulation results support the researcher's original inclination. The systematic control of experiments and use of hypotheses is not prevalent in the community. The simulators have become so popular, they are systems in and of themselves, rather than just a model. Furthermore, researchers have come to believe model output is truth [49].

We conducted a survey of all the papers published in a premiere MANET conference to evaluate the current state of MANET simulation-based research. Unfortunately, the results are extremely discouraging; in general, results published on MANET simulation-based studies lack credibility. Credibility is lacking from the methods used to execute the simulations to how the results are analyzed and published.

As a result, the MANET community needs a list of common simulation pitfalls. They also need guidance to address the pitfalls that can be avoided or accounted for in their research. Additionally, there is a lack of benchmarks or standards in the MANET community. The community needs standards for characterizing scenarios that truly test a protocol. There is also an uncertainty about the significant factors involved in a MANET routing protocol. And there is a lack of tools available to enable a researcher to conduct credible simulation-based studies. Without these items, MANET simulation-based studies have the potential to continue with misleading and questionable results.

Documenting a list of simulation pitfalls, with ways to address or avoid them, will raise the quality of simulation-based studies. Standards will enable the comparison and advancement of research results across the community. Understanding of the significant factors involved in a simulation-based study and providing tools to observe, characterize, and analyze results will improve simulation practices in the MANET community.

As the MANET community moves forward toward implementation, it is imperative that the simulation research is credible. Every MANET simulation-based study needs to satisfy at least four credibility criterion (see Chapter 2 for details). Unfortunately, at the present time, few of them do.

4

1.3 Research Overview

In this dissertation, we raise awareness of the issues and provide guidance, standards, and tools to aid MANET researchers in conducting and reporting credible simulation results. This dissertation involves documenting the pitfalls, identifying proper steps to improve the credibility of network simulation-based research, documenting standards for credibility, and providing tools to aid researchers. We note that, even though our research is based on NS-2, most of the principles apply to any simulation-based network research effort.

Our research achieved several goals:

1. A list of pitfalls common to simulation-based MANET efforts.

2. Guidance and understanding for avoiding these pitfalls and boosting credibility.

3. Standards to help researchers in improving the credibility of their simulation-based studies.

4. Algorithms to enable a researcher to create rigorous simulation scenarios for use in evaluation studies.

5. Analysis of variables that significantly impact MANET routing protocols, raising awareness of several variables that are not always discussed in the literature.

6. Tools that help analyze and present MANET simulation-based output data.

First, in Chapter 2, we present the specific simulation-based study issues that exist in MANET research. In this chapter we provide detailed descriptions and results from our survey of the published papers in the 2000-2005 proceedings of the MobiHoc conference. We then document a list of pitfalls that exist in simulation-based MANET studies. The list was developed from our survey of MobiHoc papers,

from the experiences of others in the field, and from our own experiences in MANET simulations, and provides an analysis of the current state of MANET research.

In Chapter 3, we provide guidance and standards for how simulation scenarios of MANET routing protocols should be characterized and documented. The implementation of these standards will enable the improvement of MANET studies as well as the ability to credibly describe a protocol's performance and compare it to other protocols.

Having provided standards for the simulation scenarios, the next step is to provide guidance in the execution of credible simulation studies. Chapter 4 provides a large-scale variable analysis of the Location Aided Routing (LAR) [48] protocol in NS-2. In this work, we identify the variables having the greatest impact on delivery ratio. We found several significant variables that are not variables traditionally evaluated in the literature. Addressing the values used for these variables in future studies will lend further credibility to simulation studies.

To aid researchers in validation, results analysis, and results presentation, we have developed a visualization tool iNSpect (interactive NS-2 protocol and environment confirmation tool). The iNSpect program can be used in all aspects of a simulation study from mobility file generation to presentation of results. Our visualization tool enables the MANET community to improve upon the presentation and validation of simulation results. Chapter 5 discusses the tool we developed as well as its uses in this and other research efforts. Finally, Chapter 6 summarizes our conclusions for each chapter and the research as a whole.

Chapter 2

MANET SIMULATION STUDIES: THE INCREDIBLES

In this chapter, we consider the current state of MANET simulation studies published in a premiere conference for the MANET community, i.e., the Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) from 2000-2005 [28]. The results, unfortunately, are discouraging; in general, results published on MANET simulation studies lack believability. There are several factors involved in conducting trustworthy simulation-based research. For our study, we focused on the following four areas of credibility in simulation research.

1. Repeatable: A fellow researcher should be able to repeat the results for his/her own satisfaction, future reviews, or further development.

2. Unbiased: The results must not be specific to the scenario used in the experiment.

3. Rigorous: The scenarios and conditions used to test the experiment must truly exercise the aspect of MANETs being studied.

4. Statistically sound: The execution and analysis of the experiment must be based on mathematical principles.

The remainder of the chapter will focus on the current state of MANET simulations, our survey results, common pitfalls to avoid, and tools to aid the researcher in conducting simulation studies. The goal of this chapter is to raise awareness on the

lack of reliability of MANET simulation-based studies. We present our survey results and identify common issues and pitfalls as a starting point for improvement.

2.1 The Current State of MANET Simulation Studies

In this section we describe our foundation, motivation, and results for our MobiHoc conference paper survey. We use this survey to document the current state of MANET simulation studies.

2.1.1 Survey Foundation

We conducted a survey of MANET research published in MobiHoc [28]; we only included the full papers in our survey, not the poster papers. Simulation is an often used tool to analyze MANETs; 114 out of the 151 MobiHoc papers published (75.5%) used simulation to test their research.

There are many discrete-event network simulators available for the MANET community [85]. Unfortunately, 34 of the 114 published MobiHoc simulation papers (29.8%) did not identify the simulator used in the research. Figure 2.1 shows the simulator usage results of the MobiHoc authors that did identify the simulator used. Network Simulator-2 (NS-2) [82] is the most used simulator in MANET research; 35 of the 80 simulation papers that state the simulator used in the simulation study used NS-2 (43.8%).

When the simulator used is not specified within a published paper, the repeatability of the simulation study is directly compromised. The most direct way to make a research project repeatable is to make the code and configuration files from the simulation study available to the community; unfortunately, in our survey, no paper made a statement about code availability. In addition, the researcher must identify the
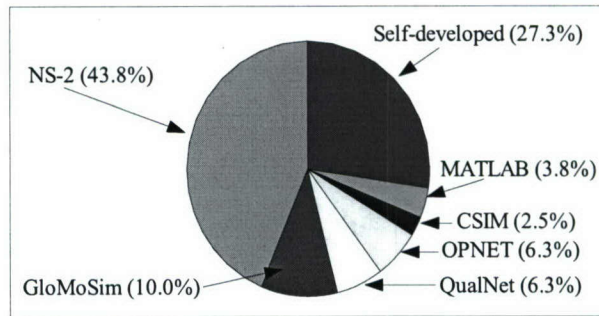
Figure 2.1. Simulator usage from our MobiHoc survey.

simulator and version, the operating system, and all variable settings. Repeatability is also based on the scenarios evaluated, the techniques used to avoid initialization bias (influence of empty queues, etc., at the start), and the techniques used to analyze the results. Thus, a published paper must discuss or reference all of these details to meet the repeatability criteria.

To be an unbiased study, a project must address initialization bias, random number issues, and use a variety of scenarios. The only time to use a single scenario is to prove a limitation or counter a generalization. To be a rigorous study, factors such as node density, node footprint, coverage, speed, and transmission range must be set to exercise the protocol under test. For example, a study that uses scenarios with average hop counts, between source and destination, below two are only testing neighbor communication and not true routing. Finally, to be a statistically sound study, a project must account for initialization bias, execute a number of simulation iterations, provide the confidence levels that exist in the results, and list any statistical assumptions made. In this chapter we use the results of our MobiHoc survey to raise awareness of the low percentage of MANET research efforts satisfying these requirements.

2.1.2 Survey Motivation

The authors of [75] completed a similar evaluation of network simulation studies in 1999. However, because the first MobiHoc conference was in 2000, this previous evaluation of simulation studies was unable to include simulations studies published in the MobiHoc conference. In addition, unlike our research, the evaluation of simulation studies from 1999 was on network simulations in general, not on MANETs in specific. Because our research is focused on the specific niche of network simulations with mobility, we completed a survey on the state of MANET simulations published in all of the previous MobiHoc proceedings (2000-2005). We found that, although it has been seven years since the previous survey study, network simulation studies (at least in the MANET community) have not improved and, in some cases, have deteriorated even further.

As an example where the reliability of simulation studies has not improved, consider the simulation type (i.e., terminating or steady-state) used in a simulation study. (See Section 2.3.1 for a discussion of simulation types.) In [74], 1690 of 2200 simulation papers (approx. 77%) did not state the type of simulation. In our MobiHoc survey, 66 of the 114 simulation papers (57.9%) did not mention the type of simulation used in the study. As an example where the credibility of simulation studies has deteriorated, consider the pseudo random number generator (PRNG) used in a simulation study. In [74], approximately 650 of the 2200 ($\approx 30\%$) papers stated which PRNG was used in the research. In our MobiHoc survey, not a single paper mentions the PRNG used.

As the MANET community moves forward toward implementation, it is imperative to have reliable simulation research and researchers addressing the design of experiments (DOE) used in their studies [11, 64]. While DOE should be used to con-

duct the overall study, we focus on issues specific to MANET research in this chapter. (See Appendix A for a summary of using DOE in a simulation study.)

The remainder of this chapter is organized as follows. In Section 2.2, we provide detailed descriptions and results from our survey of the published papers in the 2000-2005 proceedings of the MobiHoc conference. We then document a list of pitfalls that exist in simulation-based MANET studies in Section 2.3. The list was developed from our survey of MobiHoc papers, from the experiences of others in the field, and from our own experiences in MANET simulations. Section 2.4 introduces tools researchers can use to conduct credible simulation-based studies. Our goal is to raise awareness of the issues and to introduce tools to aid MANET researchers in conducting and reporting credible simulation results.

2.2 Survey Results

As mentioned, to evaluate the current state of MANET simulation research, we surveyed the published papers of MobiHoc, a premiere MANET conference. For each paper in the proceedings, we distilled the answers to several simulation study questions. Only the appropriate questions were asked of each paper, e.g., if a paper did not use plots, the detailed plot questions were not surveyed for that paper. Additionally, we reviewed each paper individually avoiding word searches or other means of automatically gathering results; in other words, papers that described the study without using explicit descriptors were counted. For consistency, the same person reviewed all of the papers; to validate the results, we had a second person review all of the papers with a subset of the questions and a third person to correct the few inconsistencies found.

Table 2.1. Survey results for 151 published papers in ACM's MobiHoc Conference, 2000-2005: simulator and environment, and plots/charts/graphs.

Simulator and Environment		
Totals	**Percent**	**Description**
114 of 151	75.5%	Used simulation in the research.
0 of 114	0.0%	Stated the code was available to others.
80 of 114	70.2%	Stated which simulator was used.
35 of 80	43.8%	Used the NS-2 simulator.
8 of 80	10.0%	Used the GloMoSim simulator.
5 of 80	6.3%	Used the QualNet simulator.
5 of 80	6.3%	Used the OPNET simulator.
3 of 80	3.8%	Used MATLAB/Mathematica.
2 of 80	2.5%	Used the CSIM simulator.
22 of 80	27.3%	Used self-developed or custom simulators.
7 of 58	12.1%	Stated which version of the public simulator was used.
3 of 114	2.6%	Stated which operating system was used.
8 of 114	7.0%	Addressed initialization bias.
48 of 114	42.1%	Addressed the type of simulation.
0 of 114	0%	Addressed the PRNG used.
Plots/Charts/Graphs		
Totals	**Percent**	**Description**
112 of 114	98.2%	Used plots to illustrate the simulation results.
14 of 112	12.5%	Used confidence intervals on the plots.
100 of 112	89.3%	Had legends on the plots.
84 of 112	75.0%	Had units on the data or labels.

We used the database of survey data to compile the results shown in Tables 2.1 and 2.2, and we discuss some of these results in Section 2.3. Overall, the results in Tables 2.1 and 2.2 indicate trends in the lack of believability in MANET simu-

Table 2.2. Survey results for 151 published papers in ACM's MobiHoc Conference, 2000-2005: simulation input parameters.

Simulation Input Parameters		
Totals	Percent	Description
109 of 114	95.6%	Conducted MANET protocol simulation studies.
62 of 109	56.9%	Stated the number of nodes used in the study.
58 of 109	53.2%	Stated the size of the simulation area.
62 of 109	56.9%	Stated the transmission range.
49 of 109	45.0%	Stated the simulation duration.
41 of 109	37.5%	Stated the traffic send rate.
31 of 109	28.4%	Stated the traffic type (e.g., CBR, etc.)
39 of 109	35.8%	Stated the number of simulation runs (iterations).
42 of 109	38.5%	Used mobility in the study.
34 of 42	81.0%	Stated the mean speed of the nodes.
26 of 42	61.9%	Stated the speed variance about the mean.
21 of 42	50.0%	Stated the mean pause time of the nodes.
16 of 42	38.1%	Stated the pause time variance about the mean.
38 of 42	90.5%	Stated which mobility model was used.
25 of 38	65.8%	Used the random waypoint mobility model [43].
2 of 25	8.0%	Used the steady-state version of the random waypoint mobility model [67].
2 of 38	5.3%	Used a group mobility model [38, 78].
4 of 38	10.5%	Used a grid/road mobility model (e.g., [17]).
5 of 38	13.2%	Used the random direction mobility model (e.g., [89]).

lation research, even though using MANET simulation research to test performance is prominent; that is, 114 out of the 151 (75.5%) published MobiHoc papers used simulation as the basis for the study. Simulation is a large portion of the research in the MANET community making its lack of believability a concern.

2.3 Common Simulation Pitfalls

We have developed a list of simulation pitfalls that impact the reliability of a simulation-based study. We have accumulated the list from our own experiences with simulations as well as the experience of others in the field. Pitfalls identified from our survey of MobiHoc papers are also included in the list. Because the pitfalls impact different phases of a simulation-based study, we have grouped the pitfalls into the following categories: simulation setup, simulation execution, output analysis, and publishing.

2.3.1 Simulation Setup

Simulation setup is the phase of a MANET research effort that is most often skipped or overlooked; and if the setup phase is done improperly, the credibility of the simulation study is flawed from the start. An NS-2 simulation study must start with proper input and configuration.

NS-2 with its support for all areas of network simulation has a dynamic control mechanism in the form of Tcl (Tool Command Language) driver files. These driver files provide the input of the scenario data and the setting of variable values. Constructing these driver files properly is a key step in executing a credible simulation effort. Setup begins with determining the simulation type, validating the model, verifying user code, validating the PRNG, defining variables, and developing scenarios.

Simulation Type Although selecting the type of simulation appears to be a trivial step, not identifying the type of simulation (terminating vs. steady-state) is a commonly overlooked step for researchers. As mentioned, 66 out of the 114 simulation

papers (57.9%) in our MobiHoc survey did not state whether the simulation was terminating or steady-state.

In terminating simulations there is a natural event which determines the length of the simulation [49]. In other words, terminating simulations model a system from a specific starting time through a specific stopping time. For example, suppose you were trying to simulate the MANET activity in a university's academic building from 9:00 a.m. to 12:00 p.m. The simulation would start at 9:00 a.m. with students in the classrooms. At transition times, students would move from one classroom to another for the next hour's class. The key with terminating simulations is understanding scenarios and the impact of time on the nodes and activity. Although the setup can be more cumbersome, the execution of terminating simulations is more straightforward than steady-state.

Steady-state simulations do not use a time window. Steady-state simulations measure performance during the median running environment of the system. As a result, there is no natural event that dictates when to terminate the simulation [49]. There is also the difficulty of trying to define what is steady-state and determine when a given network reaches it. Until a network reaches steady-state there is an initialization bias that influences output, because queues are emptier, less neighbors are known, and more packets may be dropped. Therefore, the output produced during a system's startup can differ significantly from the steady-state behavior of the system. As a result, steady-state simulations are easier to setup, but they are more difficult to execute.

Not determining the simulation type can lead to poorly designed simulations with statistically unsound results. The most common error made by researchers is to execute one type of simulation and report results on the other type of simulation.

For example, executing a terminating simulation for a set number of seconds and claiming the results represent the steady-state behavior [74]. This can produce results much different from the steady-state if the simulation terminated well before the statistics converged. The researcher should always determine the type of simulation and measure convergence if it is a steady-state simulation (see Section 2.3.2 for more detail). See [60] for an example of a MobiHoc paper identifying the simulation type used in the study.

Validation and Verification   As stated in [4, 61, 74] the simulation model must be validated as a baseline for any experimentation. Validation means the process of determining the degree to which the simulation model accurately represents the intended use [71]. According to [37], validation is assurance that the model can provide detailed enough answers to the questions being investigated–"it is the right model" [53]. Validation does not ensure that it is the right set of questions.

Verification, on the other hand, means the process of determining that the model accurately represents the conceptual design and description. Verification tells the researcher whether his or her protocol was "built correctly" [71]. Additionally, [37] states that verification shows how fully the implementation matches the developer's intent. Notice that verification does not answer whether the "right thing was built"; the research results should answer this question. Verification ensures that the model and the researcher's code does what he or she intended it to do [4].

Model Validation and Protocol Verification   After the type of simulation is determined, the simulation model itself must be prepared. As stated in [61] the model must be validated as a baseline to start any experimentation. Many researchers download the NS-2 simulator, compile it, and begin to execute simulations

with a model that has not been validated in his or her environment. Additionally, many researchers make changes to NS-2 during the study and these modifications or enhancements need to be validated.

The NS-2 validation suite consists of automated validation scripts that exercise the various parts of NS-2 and compare the results with known values from the developer [70, 97]. The validation scripts ensure that the researcher's environment operates as the developer intended [35]. The scripts do not validate that NS-2 is the right model [11], which is a different area of research; see [4] for details. There will never be a way to completely test the model. For most researchers, he or she is using NS-2 as designed. The validation is to ensure he or she has a properly executing version of NS-2 [35].

In addition to the core model of NS-2, the researcher's own protocol code must be verified to ensure it has been coded correctly and operates in accordance with the protocol specifications [7]. Verification of his or her code can be done in several different ways. A taxonomy of 45 verification and testing techniques are presented in [7]. The 45 techniques in [93], cover all areas of verification and testing, but the list of techniques in Table 2.3 are specific to simulation model and protocol verification. More detail is provided for each of these techniques in [92, 93].

One of the most common methods for validation is fixed value. Fixed value exercises the model with input data for which the outcomes are known a priori [61]. The data for the fixed value method are divided into two categories, input data and model parameter data/settings [8]. It is important to note that errors found during validation tests may be due to the data [93].

Table 2.3. List of simulation model verification techniques [93]

Technique	Description
Animation	Operation is shown graphically, allowing visual analysis.
Model Comparison	Comparing results generated from another model.
Degenerate Tests	Making sure logic behavior is present (e.g., do the queues increase as input is increased).
Event Validity	"Events" in the simulator are compared to events occurring in the real system.
Extreme Condition Tests	Output should be in line with extreme input (e.g., zero packet sends should produce zero receives).
Face Validity	Asking experts if the behavior is as he or she would expect.
Fixed Values	Using certain input that produce a known output.
Historical Data Validation	Comparing to historical data from the same model.
Internal Validity	Several iterations of a stochastic model are made to determine the variability of the model and ultimately the stability of the model.
Operational Graphics	Displaying various characteristics of the model (e.g., queue length) as it is executed, for visual and logical validation.
Sensitivity Analysis	Changing specific variables to see the effect on the overall model execution and output.
Traces	Specific aspects of the model are tracked throughout a simulation to determine logical progression.
Turing Tests	People who are knowledgeable about the model are asked if he or she can discriminate between real and model output.

The burden of verification for new protocols is that there may not be "ground truth" to compare to the results [11]. In the case of no "truth" data, verification may be limited to reduced scenarios where deterministic results can be calculated and used to compare to generated results. One recommendation by [37] is to identify appropriate metrics to use for comparison of "truth" and generated results. You

want to verify metrics that help answer the questions at hand. Metrics collection may require code instrumentation [7], where test code is added to the protocol code to track these metrics.

Not validating the model or verifying code is a common pitfall [4]. For example, when we upgraded to a new compiler we found that it implemented a broadcast function in one of our protocols differently than before. This difference had an impact on protocol performance. See [106] as an example of MobiHoc authors discussing code validation prior to evaluation.

PRNG Validation & Verification With the computing power available to researchers today and the complexity of the NS-2 model, MANET researchers need to ensure the pseudo random number generator (PRNG) is sufficient for his or her study. For example, the NS-2 PRNG does not allow a separate request stream for each dimension (i.e., a unique request stream) that exists in a simulation study. A 3-dimensional example is when a simulation has three different random pieces, such as jitter, noise, and delay. A researcher wants all three of these series (dimensions) to be uniformly distributed with each other and within each stream (e.g., the jitter stream needs to be uniformly distributed). The authors of [50, 74, 75, 76] show that a 2-dimensional request on a PRNG is valid for approximately $8\sqrt[3]{L}$, where L is the cycle length. In NS-2, the cycle length is $2^{31} - 1$, which means that only (approximately) 10,000 numbers are available in a 2-dimensional simulation study. Thus, [76] estimates that the NS-2 PRNG is only valid for several thousand numbers before the potential non-uniformity of numbers or the cycling of numbers. This cycling time occurrence is obviously dependent on the number of PRNG calls made during a simulation, but the study in [76] found most network simulations spent as much as 50% of the CPU cycles generating random numbers. Our testing of PRNG cycling shows cycling impact is

Scenario Development   NS-2 simulation studies all start with a given scenario that describes the size of the simulation area, number of nodes, node positions, node mobility (speed and pause time), traffic patterns (packet source and destinations), transmission range, and duration. These scenarios can be built directly into the Tcl files or in separate external files that are passed to the Tcl files. The advantage of external network topology and connectivity files is reuse and automation [6]. The files are separate from the Tcl files; thus, they can be reused with other protocols, especially for comparison studies.

Also, external files can be generated automatically with mobility generator programs. Automatic generation allows the researcher to cover much larger scenarios than with manual development. For a given mobility model, mobility generators construct node positions and movements based on speed, pause time, simulation area, and scenario duration. There are several mobility models ranging from random movement, to group movement, to vehicular traffic flow. See [18] for a survey of mobility models and generators used with NS-2 simulation.

Tables 2.4 and 2.5 list the parameters used by the authors who provided the number of nodes, the size of the simulation area, *and* the transmission range of nodes used in the simulations. Only 48 of the 109 MANET protocol simulation papers in our survey of published MobiHoc papers provided all three of these input parameters, detailing 61 simulation scenarios. Tables 2.4 and 2.5 show the wide range of values in these 61 scenarios. We note that scenario #36 and scenario #37 are the only two scenarios that match; the other scenarios are all unique. The number of nodes in a scenario ranged from 10 nodes to 30,000 nodes. The simulation area ranged from 25 m x 25 m to 5000 m x 5000 m. The transmission ranges varied from 3 m to 1061 m. Tables 2.4 and 2.5 also show the variety of width and height values,

Table 2.4. Input parameters for scenarios 1-30 from the 61 published scenarios in the proceedings of the MobiHoc conference, 2000-2005, sorted by number of nodes.

No.	# Nodes	Area (m x m)	Range (m)
1	10	1000 x 1000	100
2	20	350 x 350	100
3	20	1000 x 750	250
4	24	800 x 1200	250
5	25	200 x 200	100
6	25	900 x 900	250
7	30	350 x 350	100
8	36	3000 x 3000	1061
9	40	350 x 350	100
10	40	900 x 900	250
11	40	5000 x 5000	250
12	50	40 x 40	10
13	50	350 x 350	100
14	50	500 x 500	100
15	50	1500 x 300	250
16	50	1500 x 300	275
17	50	1000 x 1000	250
18	50	1000 x 1000	100
19	60	350 x 350	100
20	70	25 x 25	10
21	70	350 x 350	100
22	80	350 x 350	100
23	90	350 x 350	100
24	100	100 x 100	20
25	100	350 x 350	100
26	100	300 x 1500	250
27	100	400 x 400	100
28	100	1200 x 1200	250
29	100	500 x 500	100
30	100	575 x 575	250

Table 2.5. Input parameters for scenarios 31-61 from the 61 published scenarios in the proceedings of the MobiHoc conference, 2000-2005, sorted by number of nodes.

No.	# Nodes	Area (m x m)	Range (m)
31	100	575 x 575	125
32	100	650 x 650	67
33	100	1000 x 1000	250
34	100	1000 x 1000	150
35	100	1000 x 1000	50
36	100	1000 x 1000	100
37	100	1000 x 1000	100
38	100	2200 x 600	275
39	100	2000 x 600	250
40	100	150 x 1500	250
41	100	3000 x 900	250
42	100	1000 x 1000	100
43	110	350 x 350	100
44	120	2500 x 1000	250
45	200	100 x 100	40
46	200	500 x 500	70
47	200	1700 x 1700	250
48	200	1981.7 x 1981.7	250
49	225	100 x 100	20
50	225	600 x 600	100
51	400	100 x 100	20
52	400	800 x 800	100
53	500	3000 x 3000	67
54	600	3000 x 3000	250
55	625	1000 x 1000	100
56	1000	40 x 40	3
57	1000	81.6 x 81.6	300
58	1000	100 x 100	10
59	1000	500 x 500	20
60	10000	600 x 600	35
61	30000	5000 x 5000	100

minimal because the repeat of numbers does not occur with the simulator in the exact same state as the previous time. However, according to [76], the dimensionality of the numbers is likely to cause a problem in correlation. Thus, before publishing results, a researcher should validate the PRNG to ensure the PRNG did not cause correlation in the results. If the cycle length is an issue with NS-2, Akaroa-2 [31] offers an NS-2 compatible PRNG with a cycle of $2^{191} - 1$. The Akaroa-2 [31, 77] PRNG provides several orders of magnitude more numbers and is valid to 82 dimensions.

Variable Definition NS-2 uses hundreds of configurable variables during an execution in order to meet its general wired and wireless network simulator requirements. For example, there are 538 variables defined in the `ns-default.tcl` file of NS-2.1b7a and there are 674 variables defined in the `ns-default.tcl` file of NS-2.27. The large number of variables makes it difficult to track each variable's default setting. Additionally, an increase in the number of variables between the different NS-2 versions indicates there is a rising number of variables with each new version of NS-2. Our review of the Tcl driver files from our protocols, as well as the examples provided by NS-2, show that many simulation driver files leave key parameters undefined. For example, three out of 12 of the wireless examples in NS-2 do not define the transmission range of a node. The transmission range is a key variable in MANET performance. If the transmission range default is changed from one NS-2 version to the next, the results of a simulation would be significantly different. The researcher should define all of the variables by using his or her own configuration file or Tcl driver file [11]. See [81] as an example of how to define variables and reference them on a website, providing more detail than can be written in a published paper.

Table 2.6. Derived scenario parameter definitions and formulas.

Parameter	Description	Formula
Node Density	Density of nodes in the simulation area.	$\dfrac{n}{(w \times h)}$
Node Coverage	Area covered by a node's transmission.	$\pi \times r^2$
Footprint	Percentage of the simulation area covered by a node's transmission range	$\dfrac{(\pi \times r^2)}{(w \times h)} \times 100$
Maximum Path	The maximum linear distance a packet can travel from source to destination.	$\sqrt{(w^2 + h^2)}$
Network Diameter	The minimum number of hops a packet can take along the maximum path from source to destination.	$\dfrac{\sqrt{(w^2 + h^2)}}{r}$
Neighbor Count	The number of neighbor nodes based on transmission and simulation area. It does not account for the edge of the simulation area.	$\dfrac{(\pi \times r^2)}{\left(\dfrac{w \times h}{n}\right)}$
Neighbor Count Edge Effect	The average number of neighbor nodes accounting for the edge of the simulation area reducing the node's coverage. For example, a node in the corner of the simulation area only has neighbors in 25% of its coverage area.	Simulation with n, r, and $(w \times h)$
w = width, h = height r = transmission range, n = # of nodes		

illustrating the different shapes used in MANET simulation scenarios. Additionally, Tables 2.4 and 2.5 reflect that the parameter values are often very specific, e.g., a 1981.7 m squared simulation area. The survey results highlight the wide range of simulation scenarios used to conduct MANET research and the lack of uniform rigorous testing of MANET protocols.

We validated the wide range of input parameters by comparing the derived parameters of each scenario. Table 2.6 shows a list of several derived parameters, including definitions and formulas. The derived parameters aggregate multiple input parameters to further characterize a scenario. The derived parameters also provide a common basis for comparison across scenarios. Figure 2.2 is a scatter plot of all the derived parameters for the 61 sets of input parameters. The plot shows every variable plotted against all the others. For example, the upper right plot is simulation area versus neighbor count with edge effect. The scatter plot reflects the wide range of scenarios and the lack of correlation between parameters.

Figure 2.2 also shows the lack of independence between parameters, such as node density and node coverage. In addition, the lack of multiple groupings in each plot illustrates that the community is not covering the range of values in a consistent organized manner. For example, if there were benchmark scenarios for small, medium, and large sized simulations, then there would be three groupings of values in each of the simulation area plots. Finally, the extreme values in the derived parameters do not correlate with the extreme input parameters. For example, the highest number of nodes (30,000) is the 6th lowest value for the neighbor count derived parameter.

The MANET community needs a way to characterize simulation scenarios in order to evaluate and compare protocols and performance, and ensure protocols are rigorously tested. For example, from Tables 2.4 and 2.5, scenario #8, the simulation area is 3000 m x 3000 m, but the transmission range of 1061 m lowers the average hop count to only 1.67 hops. This hop count means most source and destination pairs are direct neighbors and the rest have only one intermediate node. (See Section 2.4 for existing tools that aid in scenario evaluation and characterization.) We also note that there have been several emails on the NS-2 mailing list [34] asking what a valid
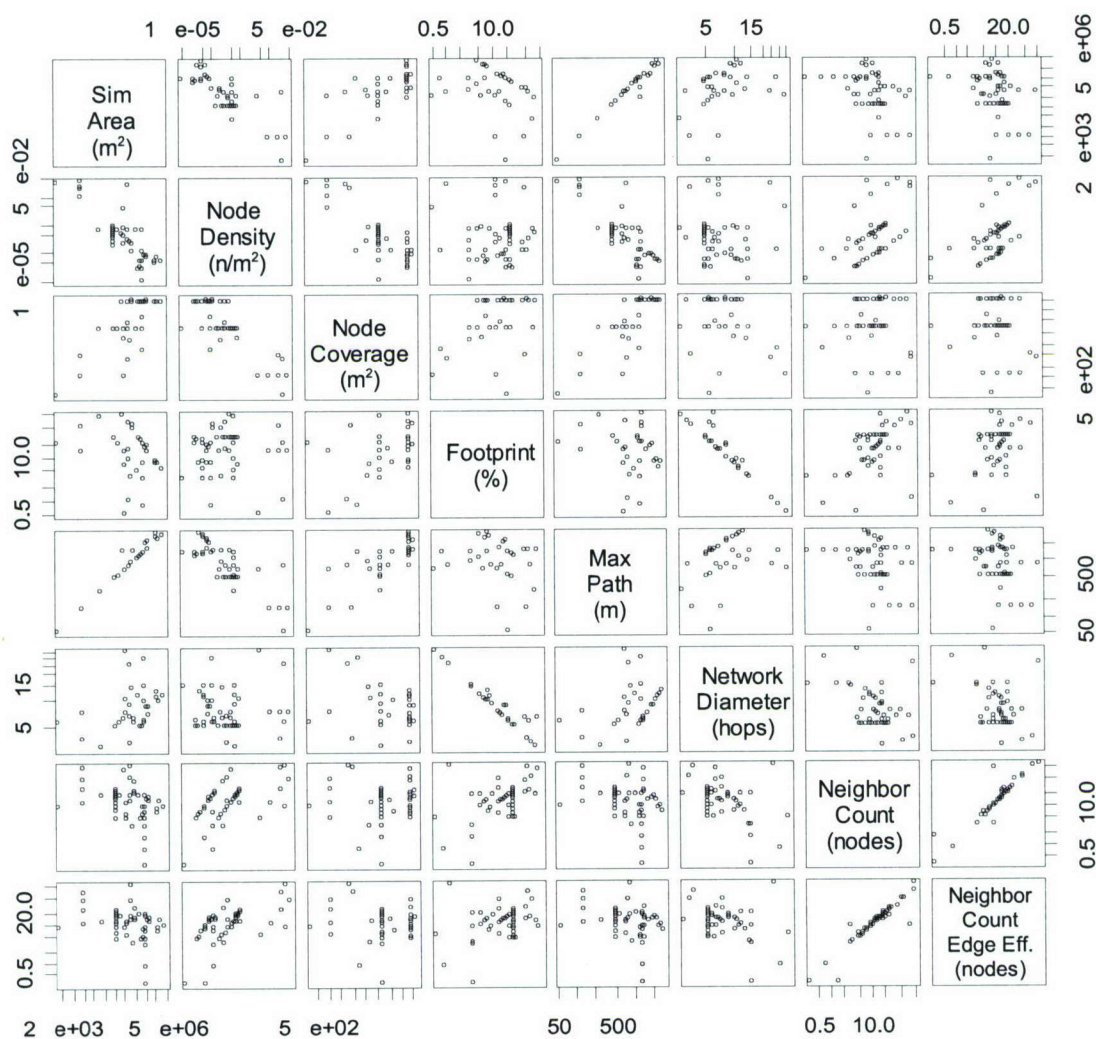
Figure 2.2. A scatter plot with each of the eight derived scenario parameters plotted against the other derived scenario parameters.

scenario is for MANET research, but until our work there is no single benchmark of MANET scenarios to test a protocol. See Chapter 3 for our proposed standards for MANET scenarios.

2.3.2 Simulation Execution

Executing the simulation is where a lot of time is spent. Therefore, it is important to conduct the execution portion correctly. There has been a large amount of research on how best to execute the simulator in a simulation-based study. There are several factors that influence the execution method, such as time to complete, resources available, and number of repetitions required. Because most simulation studies demand a large amount of computation time, one of the biggest categories of research is in parallel execution of simulations [5, 41, 46, 49, 75, 96]. We highlight several execution pitfalls we have discovered; these pitfalls impact data output, analysis, and ultimately results.

Setting the PRNG Seed   One mistake we have seen in NS-2 simulation studies concerns not setting the seed of the PRNG properly. NS-2 uses a default seed of 12345 for each simulation run [70]. Thus, if an NS-2 user does not set the seed, each simulation will produce identical results. Additionally, if the seed is not set or is set poorly, it can negate the independent replication method which is typically used in analysis. Introducing correlation in the replications prevents the use of common statistical analysis techniques. In our MobiHoc survey, none of the 84 simulation papers addressed PRNG issues. The researcher should ensure the seed is set correctly in his or her Tcl driver file and that the NS-2 `Random` class is used for all random variables.

Scenario Initialization   Another pitfall is not initializing the scenario correctly. This pitfall usually occurs from a lack of understanding of the two types of simulation. In terminating simulations, the network is usually started in a certain configuration that represents the start of the simulation window. For example, if the researcher is trying to simulate a protocol's response to a failure event, he or she needs

to have the failure as the initialization of his or her analysis. Likewise, steady-state simulations require that the researcher address initialization bias [65, 95]. Most simulations start with empty caches, queues, and tables. The simulation fills the caches, queues, and tables until a steady-state of activity is reached. Determining and reaching the steady-state level of activity is part of the initialization. Data generated prior to reaching steady-state is biased by the initial conditions of the simulation and should not be used in the analysis. For example, in protocols that maintain neighbor information, the size of the neighbor table should be monitored to determine when the table entries stabilize, because the protocol will perform differently with empty neighbor tables. Akaroa-2 [31] is a tool that monitors variables during execution to determine steady-state (see Section 2.4).

Metric Collection   Another area of concern is the metric measurements collected during execution. If the simulation executes properly, but the researcher does not obtain the data he or she needs from the simulation, the simulation is worthless [75]. Appropriate output is especially critical if output has to be correlated. For example, if the researcher is trying to track delivery ratio for data packets and control packets, each type of packet must be identified along with the source and destination to determine the number of each type of packet sent and successfully received. Outputting only the number of packets sent and the number of packets received will not provide the granularity required in the measures. The researcher needs to include output analysis in his or her practice runs of the simulation to ensure the correct metric is being collected. See [51] for an example of a MobiHoc paper describing and defining the statistics used in calculating results.

2.3.3 Output Analysis

Typically the preceding steps take longer than planned, which means sufficient time is not provided for output analysis at the end of the schedule. Whether it is the publication deadline, or a thesis defense date, proper analysis is often compromised. As a result, output analysis is the downfall of many simulation studies.

Single Set of Data   This pitfall is taking the first set of results from a simulation and accepting the results as "truth." The decision to take the first set is not a plausible way to conduct research. With a single result the probability is high that the single point estimate is not representative of the population statistics. A single execution of a discrete-event simulation is not accounting for the model's innate randomness in the experiment. Executing the simulation once will produce results, maybe even good results [49]; however, the single point estimate produced will not give the researcher sufficient confidence in the unknown population mean. The researcher needs to determine the number of runs necessary to produce the confidence levels required for his or her study. In our MobiHoc survey, only 39 of the 109 MANET protocol simulation papers (35.8%) stated the number of simulation runs executed. See [39] for an example of a MobiHoc paper using multiple replications to achieve high confidence and [27] for an example of a MobiHoc paper documenting the number of replications used and how the quantity was chosen.

Steady-State Initialization Bias   Steady-state simulations have all of the issues associated with initialization bias. If the researcher has selected steady-state simulation, he or she should only use steady-state information to calculate his or her measures. If the startup data is included in the results calculation, the findings will contain a bias [91]. Unfortunately, only eight of the 114 simulation papers in our

MobiHoc survey (7.0%) addressed initialization bias, and all eight used the unreliable method of arbitrarily deleting data. The arbitrary discard periods ranged from 50 seconds to 1000 seconds.

The common trend in addressing initialization bias is to discard the first portion of the output data [104]. Another method to address initialization bias is executing the simulation longer; thus the influence of the initialization bias is minimized. Neither of these options are credible approaches. For discarding data, the question is how much data to discard [23, 30, 49, 56, 73]. The problem with arbitrarily discarding data is that if the point is selected too early in the data, the bias will remain. Conversely, if the point is picked too late in the data, good observations of rare events might be deleted. For longer simulations, there are additional problems of cycling and PRNG limits that effect results [91].

There needs to be statistical rigor in determining a simulation has truly reached steady-state. The researcher should monitor convergence for the steady-state portions of his or her protocol. Fortunately, there are a series of tests that can be conducted to detect the presence of initialization bias and determine where its influence is no longer an impact. For example, in [49] and [102], the authors use a 4-step test to determine the point to stop discarding initial data. For more information on statistically sound methods of addressing initialization bias see [14, 49, 91, 95]. Discussion on measuring and discarding initialization bias is also included in [73]. As an example, Akaroa-2 [31] has implemented initialization bias tests. See [27] for an example of a MobiHoc paper that addressed scenario initialization.

Statistical Analysis This pitfall concerns not using the correct statistical formulas with the different forms of output. (See Appendix B for formulas to use with terminating and steady-state simulations.) For example, it is not correct to use

the standard formulas for mean and variance without ensuring the data is independent and identically distributed (*iid*). Use of *iid*-based formulas on correlated data can reduce reliability by producing biased results. The researcher needs to use batch means or independent replications of the data to ensure *iid* and prevent correlated results [30]. The simulation model is based on randomness, therefore the output will vary. Simulation output will not satisfy the independent and identically distributed required for traditional statistical analysis [30]. The survey in [75] shows that 76.5% of the papers did not discuss the statistical methods used in analysis. See [90] for an example of a MobiHoc author that described the analysis and data used to calculate the results.

Confidence Intervals This pitfall is a culmination of several of the previous analysis issues. Confidence intervals are a tool to provide a range where we think the population mean is located relative to the point estimate [16, 91]. Confidence intervals account for the randomness and varied output from a stochastic simulation. However, in our survey, 98 of the 112 simulation papers using plots (87.5%) did not show confidence intervals on the plots. See [106] for an example of a MobiHoc paper that used confidence intervals.

2.3.4 Publishing

When publishing simulation results a researcher needs to identify the following information at a minimum. Otherwise the experiment can not be repeated [61], and repeatable simulations are necessary in a credible simulation study.

1. Type of simulation - whether it was terminating or steady-state.

(a) For terminating simulations, identify the time frame and setup for the start of the scenario.

(b) For steady-state simulations, identify the definition of steady-state and the initialization removal technique.

2. Tools employed - all validation, execution, and analysis tools used.

3. PRNGs used - describing the cycle length, dimensions, and seeds.

4. Methods of statistical analysis - batch means, replications, etc.

5. Statistical errors associated with the result - errors calculated in analysis.

Table 2.7. Example list of input parameters to document [13]

Parameter Name	Parameter Value
Simulator	NS-2.1b7a
Simulation Time	1000 s
Simulation Area	300 x 600 m
Number of Nodes	50
Transmission Range	100 m
Movement Model	random waypoint
Speed Range	4.4-44 m/s
Average Speed Range	5-40 m/s
Pause Time	0-50 s
CBR Sources	20
Data Payload	64 bytes
Packet Rate	4 packets/sec
Traffic Pattern	peer-to-peer

In addition to the preceding information, [13] provides a list of technical parameters to document in a MANET simulation. Table 2.7, which is a slight modification

of the table from [13], illustrates these parameters to document. The authors of [70] also recommend documenting any patches that have been applied to NS-2.

As an alternative to formal statistical inference, a graphical display can be used to describe simulation output. A well-constructed picture may be worth a thousand words if it reveals clear patterns that might go undetected in standard numerical summaries. Several graphical techniques for describing simulation output are described in detail in [91].

Tables 2.1 and 2.2 list all the data from our MobiHoc paper survey. The lack of consistency in publishing simulation-based study results directly impacts the trustworthiness of these studies. In addition, the inconsistency prevents the direct comparison of results, limiting research advancements. The publishing pitfalls prevent the MANET community from taking advantage of new researchers interested in these studies. A new researcher cannot repeat the studies to start his or her own follow-on research.

Publishing is a big part of breaking the "repeatable" criteria for credible research, because much of the simulation study is unknown to the paper reader. As stated earlier, there are 674 variables defined in the `ns-default.tcl` file of NS-2.27. To ensure repeatability the researcher must document the `ns-default.tcl` file used and any changes made to the settings of the variables in the file. When publishing, the authors need to state if the code is available and how to obtain the code. There should be a code statement even if the code's release is restricted by copyright or third party ownership. See [81] as an example of how to properly define variables without using a large portion of the published paper.

At the bottom of Table 2.1 are publishing specific statistics. Plots of simulation results are common, i.e., 112 of the 114 simulation papers (98.2%) used plots to

describe results. However, 12 of the 112 simulation papers with plots (10.7%) did not provide legends or labels on his or her charts. Additionally, 28 of the 112 simulation papers with plots (25.0%) did not provide units for the data being shown. The lack of labels and units can cause readers of these papers to misinterpret or misunderstand the results.

Several of the results in Tables 2.1 and 2.2 are significant inefficiencies in publishing simulation-based results. For example, 47 of the 109 MANET protocol simulation papers (43.1%) did not state the transmission range of the nodes. Also, 78 of the 109 MANET protocol simulation papers (71.6%) did not mention the packet traffic type used in the simulation. Although both of these parameters were set to execute the simulation, neither were documented nor referenced in these papers.

A final area of concern in publishing results, one that was not quantified in our survey, is supporting the text with charts and graphs and vice versa. Many papers had charts that were not discussed in the text or the text referenced a chart as supportive, but it was not clear in the chart how it supported the work.

These publishing pitfalls directly impact the credibility of the research conducted in the MANET community. The best simulation-based studies can be lost behind a biased, unrepeatable, and unsound document describing the work.

2.4   Community Resources

There is some research in developing techniques and processes to aid credible simulation studies. This research is often found in the general simulation community, not the MANET community specifically; however, many groups and authors, such as [4, 9, 37, 93], have outlined steps applicable to MANET research. These methods

aid in validation, verification, output analysis, etc. for a simulation-based study, and give the overall study more credibility.

Although there has been work on techniques and processes, we have found very few tools that aid researchers in conducting credible simulation studies. Simulation tools are needed to understand the large amount of data produced during network simulations. Tools can analyze the input data as well as aid in validation, verification, initialization, and output analysis. The few tools available include:

- RunJobs is a script developed by Nick Bauer, a Toilers [32] graduate at Colorado School of Mines, that manages the execution of multiple copies of a simulation on multiple machines. If the NS-2 random number generator is initialized properly in the Tcl driver file (see Section 2.3.2), RunJobs uses the microseconds of the local machine's clock to seed the random number generator of each individual simulation. RunJobs checks the utilization of the machines, selecting the potential candidate machines for the simulations. RunJobs also initiates each of the simulations.

- The Akaroa-2 [31] suite is a package that manages the executions of distributed stochastic simulations. Similar to RunJobs, Akaroa-2 manages the start and seeding of multiple simulation runs on multiple machines. In addition, Akaroa-2 provides other services, such as monitoring and determining the initialization period for each simulation to ensure the proper discarding of initialization data [65]. Akaroa-2 does the initialization calculation using a sequential version of the statistical test in [95], implemented in [73]. Akaroa-2 can be configured to stop the steady-state simulation when a configurable threshold has been reached by each simulation.

- The Simulator for Wireless Ad Hoc Networks (SWAN) [52] enables a researcher to create a virtual environment for conducting experiments with MANETs. SWAN is based on the Scalable Simulation Framework (SSF) [26], and is designed to be easy to use, fast, and scalable.

- We have developed SCORES, (a SCenariO characteRizEr for Simulation) tool. SCORES evaluates the rigor with which a scenario tests a MANET protocol by characterizing the scenario. SCORES calculates the derived parameters as well as average shortest-path hop count and average network partitioning (see Chapter 3).

- We have also developed the interactive NS-2 protocol and environment confirmation tool (iNSpect), which visualizes the trace file of an NS-2 simulation. The visualizations can be used for scenario development, model validation, protocol verification, and results analysis (see Chapter 5).

- More recently, since the development of iNSpect, the authors of [94] have created a 3-D visualization tool for NS-2 called *Huginn*. *Huginn* provides visualization of NS-2 trace files and filtering at different levels of the network layers to generate various simulation results. Currently, *Huginn* is not available to the community.

To aid the community in learning about current and future tools available for use with MANET simulation studies, we have created an on-line list. The current list of tools can be found on our research website at *http://toilers.mines.edu.*.

2.5 Conclusions

Summarizing the four areas of credibility, we found less than 15% of the published MobiHoc papers are repeatable. It is difficult, if not impossible, to repeat a

simulation study when the version of a publicly available simulator is unknown, and only seven of the 58 MobiHoc simulation papers that use a public simulator (12.1%) mention the simulator version used. It is also difficult, if not impossible, to repeat a simulation study when the simulator is self-developed and the code is unavailable. In addition, only eight of the 114 simulation papers (7.0%) addressed initialization bias and none of the 84 simulation papers addressed random number generator issues. Thus, we are concerned that over 90% of the MobiHoc published simulation results may include bias. With regard to compromising statistical soundness, 70 of the 109 MANET protocol simulations papers (64.2%) did not identify the number of simulation iterations used, and 98 of the 112 papers that used plots to present simulation results (87.5%) did not include confidence intervals. Hence, only approximately 12% of the MobiHoc simulation results appear to be based on sound statistical techniques.

MANET simulation-based research is an involved process with plenty of opportunities to compromise the credibility of the study. In this chapter, we have identified several pitfalls throughout the simulation lifecycle. Each of the pitfalls discussed in Section 2.3 takes away from the goals of making the research repeatable, unbiased, rigorous, and statistically sound. Documenting these pitfalls and sharing knowledge about how to address these common issues will increase the reliability of studies in the MANET community. Our survey of MobiHoc papers showed the current state of MANET research and the lack of consistency, re-enforcing the need for simulation study guidance.

Chapter 3

SCENARIO STANDARDS FOR RIGOROUS MANET ROUTING PROTOCOL EVALUATION

3.1 Introduction and Standards

In Chapter 2, we discussed the issue of a lack of credibility in network protocol evaluation. This lack of credibility covers all areas of simulation-based research and is sometimes attributed to a lack of standards. Standards establish a baseline for rigorous evaluations and can cover the entire simulation-based study process, from simulation scenario creation to random number generation to results analysis. Many standards are needed to improve the quality and credibility of MANET simulation research. In this chapter we focus on two of these standards as they apply to generic MANET routing protocols[1] and to the simulation scenarios used to evaluate their performance.

To execute a MANET simulation, the researcher must create a simulation scenario. In addition to the mobility model, important parameters of a simulation scenario include the number of nodes, width and height of the simulation area, shape of the simulation area, node speed, node pause time, and transmission range of the node. The values chosen for simulation parameters determine the rigor of a scenario

---

[1]We define generic MANET routing protocols as those protocols that are used for direct end-to-end communication without any specific distinctive quality or application. The goals of these protocols are typically to minimize end-to-end delay, minimize control overhead, and/or maximize delivery ratio.

in assessing the performance of the protocols being evaluated. Appropriate choices for these parameters have long been the subject of debate.

We propose two standards for rigorous evaluation of generic MANET routing protocols. Our proposed standards are not individual parameter settings, but a definition of two metrics that should be calculated and recorded with any simulation-based research that desires credit for rigorously testing a generic MANET routing protocol.

Standard 1: To rigorously evaluate generic MANET routing protocols, the average shortest-path hop count needs to be large.

A scenario with an average shortest-path hop count of 1 or 2 is a scenario in which many packets are only sent between neighbors. In this environment, the generic MANET routing protocol's routing capability is not rigorously tested. Most protocols, even poor protocols, perform well in scenarios that have low average shortest-path hop counts.

Standard 2: To rigorously evaluate generic MANET routing protocols, only a small amount of network partitioning should exist.

Since no routing protocol is able to route between a pair of nodes that is partitioned, most protocols, even good ones, perform poorly in scenarios that have a large amount of network partitioning. In other words, a large amount of network partitioning prevents rigorous evaluation of a generic MANET routing protocol.

The main contribution of this chapter is to provide algorithms that researchers can use to create scenarios that meet our standards proposed. We first precisely define average shortest-path hop count and average network partitioning, the two metrics for our proposed standards, how we estimate these metrics in simulation, and our notation. We then explore the relationship between average shortest-path hop count

and network partitioning, and develop algorithms for generating scenarios that meet our standards, using any values for the metrics that the researcher finds appropriate.

The rest of this chapter is organized as follows. Section 3.2 defines terms used in this chapter, describes the metrics we use, discusses our mobility model selection, and describes the notation used in this chapter. Section 3.3 explores the relationship between average shortest-path hop count and network partitioning, as well as the impact of parameters on these two metrics. In Section 3.4, we develop the algorithms that allow a researcher to calculate the required number of nodes and simulation area to produce scenarios with their desired metric levels. Section 3.5 illustrates some example scenarios based on metric levels we have picked for our two standards, and Section 3.6 presents our conclusions.

3.2 Background

To sufficiently exercise a generic MANET routing protocol, packet destinations need to be several hops from the source, and packets must have the opportunity to be delivered to the destinations. In this section, we consider one metric that is needed to obtain a large number of hops from the source to the destination (i.e., high average shortest-path hop count[2]) and one metric that is needed to give packets the opportunity for delivery (i.e., low network partitioning[3]). Other metrics could be considered; for example, a high average neighbor count metric. We chose, however, to focus our standards on average network partitioning and average shortest-path hop count, because they are both intuitive and can be employed to ensure long routes are

---

[2]The shortest-path hop count is the smallest number of links needed to allow two nodes to communicate. The average shortest-path hop count is the average of all shortest-path hop counts for all node pairs. See Section 3.2.1 for details.

[3]Network partitioning exists when some pair of nodes has no route between them and thus cannot communicate with each other. See Section 3.2.2 for details.
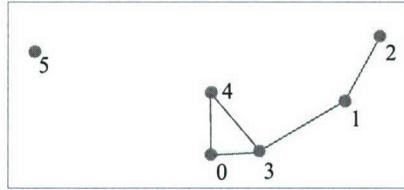
Figure 3.1. Example network with six nodes at a certain point in time. The lines represent communication links between nodes.

available and used between sources and destinations. When long routes are available and used, then routing in a generic MANET routing protocol is rigorously tested.

In this section, we first precisely define our two metrics that are the basis for our standards: average shortest-path hop count and average network partitioning. We also describe how we estimated these values for various scenarios using simulation. In addition, we describe both our selection of the steady-state Random Waypoint Model and our transmission range notation.

3.2.1 Average Shortest-path Hop Count

A hop in a MANET is the transition of a packet from one node to the next (within transmission range) on a communication link between two nodes. The hop count of a path between a pair of nodes is defined to be the number of communication links on the path. As an example, Figure 3.1 presents a snapshot in time of a network with six nodes. In Figure 3.1, the hop count between node 0 and node 2 is three.

When a protocol is being evaluated, it is common to calculate the average number of hops by counting the total number of hops of all successfully delivered packets, then dividing by the number of successfully delivered packets. This metric is not appropriate for our needs, because it is protocol dependent. We need a metric that measures the potential for a scenario to evaluate protocols in general, rather than

	0	1	2	3	4	5
0	-	2	3	1	1	0
1	2	-	1	1	2	0
2	3	1	-	2	3	0
3	1	1	2	-	1	0
4	1	2	3	1	-	0
5	0	0	0	0	0	-

Figure 3.2. Multi-hop connectivity matrix for the simulation scenario in Figure 3.1.

one which depends on the performance of a particular protocol. For this reason we base our metric on the shortest-path hop count, which is the smallest number of hops along any path between the two nodes.

To calculate the average shortest-path hop count, we use a multi-hop connectivity matrix [63, 88] which stores the shortest-path between two nodes in the matrix. Figure 3.2 presents the multi-hop connectivity matrix for the network in Figure 3.1. Each non-zero entry in the matrix represents the shortest-path hop count for a particular pair of nodes. The zero entries represent partitioned pairs. The instantaneous average shortest-path hop count for the network in Figure 3.1 is found by summing the non-zero entries in the matrix, then dividing by the number of non-zero entries, i.e., $34/20 = 1.7$.

Our metric is the average shortest-path hop count, where the average is taken over all communicating node pairs over all points in time. We denote this by A_{sp}Hops. In practice, A_{sp}Hops is estimated by generating a large number of realizations of a scenario at various points in time, using the multi-hop connectivity matrix to compute the average shortest-path hop count at each point in time, and averaging. Specifically, A_{sp}Hops is calculated using the equation

$$A_{sp}\text{Hops} = \frac{\sum_{i=1}^{T} hops_i}{\sum_{i=1}^{T} paths_i}, \tag{3.1}$$

where T is the number of multi-hop matrices constructed, $hops_i$ is the total number of hops in the multi-hop matrix at time i, and $paths_i$ is the number of cells in the multi-hop matrix at time i that contain a non-zero entry.

3.2.2 Network Partitioning

We define the degree of network partitioning at any given time to be the proportion of node pairs between which no path exists. In Figure 3.1, there are a total of 15 ($6 \times 5/2$) pairs of nodes, and of these pairs, five (the ones involving node 5) have no path between them. To calculate the degree of network partitioning, we use the multi-hop connectivity matrix [63, 88]. Using the multi-hop connectivity matrix in Figure 3.2, the degree of partitioning is the proportion of the matrix with entries equal to 0. Thus, the degree of partitioning in this network, at this point in time, is $5/15 = 33.3\%$.

Our metric is the average amount of network partitioning over all points in time, and is referred to as "average network partitioning" or ANP. In practice, ANP is estimated by generating a large number of realizations of a scenario at various points in time, computing the degree of partitioning for each, and averaging. Specifically,

$$\text{ANP} = \frac{z}{n(n-1)T} \tag{3.2}$$

where z is the total number of zeros in all the matrices constructed, n is the number of nodes, $n(n-1)$ is the potential number of links, and T is the number of multi-hop connectivity matrices constructed.

3.2.3 Mobility Models

We conducted a survey (see Chapter 2) of MANET research published in the 2000-2005 proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) [28]. As mentioned, simulation is an often-used tool; 114 of the 151 MobiHoc papers published (75.5%) reported simulation studies. In addition, each of these studies using mobility required a mobility model.

There are many mobility models available for the MANET community to use to generate node position and movement [18]. Figure 3.3 shows the distribution of the mobility models identified in our survey. As shown, 32 out of the 50 simulation papers (64%) that stated which mobility model was used in the study used the Random Waypoint Model (RWM) [43]. Because the RWM was the most popular, we used the RWM to generate simulation scenarios that meet our two standards; thus, the scenarios developed herein have the broadest application. We note, however, that our method can be modified to produce scenarios with any mobility model that is considered appropriate by the researcher. Herein, we used a steady-state version of the RWM [68] that starts all nodes in the steady-state distribution of the RWM. Use of the steady-state RWM allows us to analyze a simulation scenario from time zero, without initialization bias associated with initial node movement.

3.2.4 Using Transmission Range as the Unit of Distance

There are five main simulation parameters in the steady-state RWM [68]: the number of nodes, the width and height of the simulation area, which affect both the shape and size of the simulation area; and the node speed and pause time. One additional simulation parameter important to simulation scenarios, but not required by the steady-state RWM, is the transmission range of a node. The transmission range is
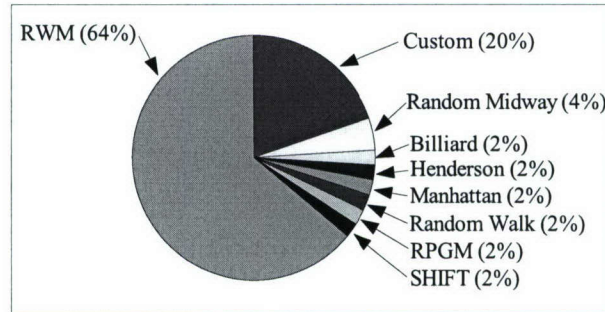
Figure 3.3. MobiHoc survey results for mobility model usage.

Table 3.1. Simulation scenario parameters expressed in meters, and transmission range units (R), for two scenarios.

Parameter	Scenario 1		Scenario 2	
	Meters	R	Meters	R
Trans. Range	40 m	1 R	100 m	1 R
Width	200 m	5 R	500 m	5 R
Height	80 m	2 R	200 m	2 R
Node Speed	10 m/s	0.25 R/s	25 m/s	0.25 R/s

the maximum distance at which the radio signal from a node can be received. When distances are measured in absolute units, such as meters, it is difficult to determine from the five main simulation parameters whether a scenario will effectively test a protocol. The reason for this is that the effect of distance is not determined by its absolute size, but by its size relative to the transmission range. For example, consider a simulation scenario with a 500 m × 500 m area. Then consider the different values of A_{sp}Hops if the transmission range is 500 m versus 50 m; a 50 m transmission range would require considerably more routing by a protocol than a 500 m transmission range. For this reason, it is appropriate to express distances in terms of the transmission range (R).

Using transmission range as the unit of distance, a simulation scenario with a 80 m × 200 m area, a node speed of 10 m/s, and a transmission range of 40 m, would be described as having a 2 R×5 R area and a node speed 0.25 R/s. Table 3.1 presents an example to show that a simulation scenario with a 80 m × 200 m area, a node speed of 10 m/s, and a transmission range of R = 40 m is equivalent to one with a 200 m×500 m area, a node speed of 25 m/s, and a transmission range of R = 100 m. In the rest of this chapter, we express all distances in terms of an arbitrary transmission range R. Our results are, therefore, valid for any choice of transmission range.

3.2.5 Propagation Modeling

Several articles (e.g., [87]) in the literature have discussed the problems associated with specifying the transmission range of a node as a uniform circular representation of the transmission range [22]. We, therefore, use the Two Ray Ground propagation model as implemented in NS-2 [33]. The Two Ray Ground model used the Friis Free Space model [33] (factor of d^2) for nodes close to the source (less than the cross-over distance). For nodes farther from the source (greater than the cross-over distance), it uses a two ray reflection model with a factor of d^4, lowering the probability of a packet being received at the node's neighbor. The cross-over distance for the Two Ray Ground model to switch from d^2 to d^4 with an omnidirectional antenna at 1 m height is 38.6 m.

3.3 Effect of Parameters

In this section we explore the relationship between our two metrics described in Section 3.2, transmission range, and the input parameters of the RWM. First, we evaluate the impact of node speed and node pause time on A_{sp}Hops and ANP. Second,

Table 3.2. Simulation scenario parameters for our speed and pause time study.

Parameter	Value(s)		
# of Nodes	100	150	200
Width	6.75 R	7.25 R	8 R
Height	6.75 R	5.25 R	8 R
Avg. Speed (R/s)	0.075, 0.25, 0.5, 0.75, 1, 1.25		
Pause Time (s)	2, 5, 10, 20, 30, 40		

we look at A_{sp}Hops, which is the metric that, when high, provides the best indicator of rigorous evaluation of a routing protocol. Third, we look at the relationship between A_{sp}Hops and ANP.

3.3.1 Effect of Speed and Pause Time

In this section, we show that speed and pause time have relatively little effect on A_{sp}Hops and ANP for the range of scenarios we evaluated. Using 36 different combinations of speed and pause time, and three combinations of number of nodes, width, and height, we created a total of 108 scenarios for this study. Table 3.2 presents the parameter values used in these scenarios. We then constructed multi-hop connectivity matrices to compute A_{sp}Hops and ANP for each scenario. We generated 200 independent iterations of each scenario and averaged the results. By varying only the node speed and node pause time we can isolate the impact of node speed and node pause time on A_{sp}Hops and ANP.

Table 3.3 shows results from 12 of the 36 different simulation scenarios for 100 nodes; the other 24 scenarios for 100 nodes produced similar results. Neither ANP nor A_{sp}Hops vary greatly over the range of values of speed and pause time; the results for ANP vary less than 1% and the results for A_{sp}Hops vary less than 0.1 hops. Although

Table 3.3. Partial results from our speed and pause time study for the 100 node scenario. Fixed parameters were 100 nodes, 6.75 R width, and 6.75 R height.

Spd (R/s)	Pause (s)	ANP	A_{sp}Hops
0.075	5	4.22	4.04
0.075	30	4.26	4.04
0.25	5	4.37	4.05
0.25	10	4.44	4.03
0.5	20	4.96	4.05
0.5	30	4.99	4.05
0.75	2	5.15	4.07
0.75	5	5.01	4.04
1.0	10	4.64	4.05
1.0	30	4.63	4.09
1.25	10	4.81	4.10
1.25	20	4.63	4.04

not presented, we obtained similar results from the 150- and 200-node scenarios. We, therefore, conclude that node speed and node pause time do not greatly affect ANP or A_{sp}Hops for the scenarios (see Table 3.2) tested.

3.3.2 Effect of Number of Nodes

Due to the performance limitations of some simulators and the need to execute simulation studies quickly, researchers often conduct research studies with scenarios containing 200 nodes or less. This is validated by our MobiHoc survey (see Chapter 2). Specifically, as shown in Tables 2.4 and 2.5, even though the number of nodes varies from 10 to 30000, the majority of scenarios have 200 nodes or less.

We note that scenarios with a small number of nodes are scenarios with low A_{sp}Hops, especially when network partitioning is low [36]. To illustrate, we generated 200 independent scenarios using the steady-state RWM for numbers of nodes from 10
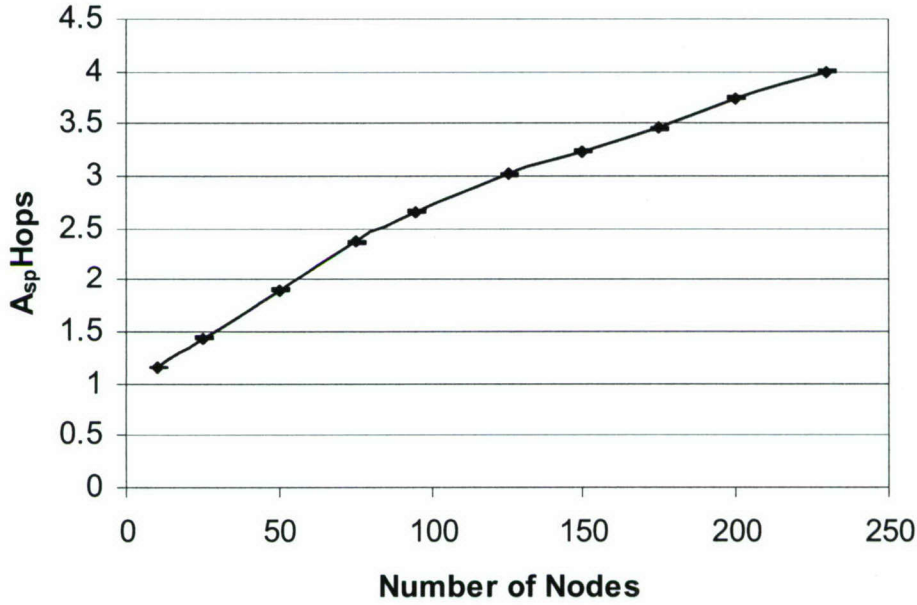
Figure 3.4. A_{sp}Hops versus number of nodes with 95% confidence intervals, and ANP ≈ 0. Each point in the plot is the average result from 200 realizations of the given simulation scenario.

to 230. We adjusted the area of the scenarios to achieve nearly no network partitioning (ANP $< 0.2\%$) for each number of nodes. Figure 3.4 shows A_{sp}Hops versus number of nodes, with 95% confidence intervals. We note that scenarios with 50 nodes or less and small ANP (i.e., ANP $< 0.2\%$) means A_{sp}Hops is less than 2 hops. And, as previously mentioned, a scenario with an average shortest-path hop of 1 or 2 is a scenario in which many packets are only sent between neighbors. Thus, the generic MANET routing protocol's routing capability is not rigorously tested. Most protocols, even poor protocols, perform well in scenarios that have low average shortest-path hop

Table 3.4. Simulation scenario parameters for the square A_{sp}Hop versus ANP study. Average node speed is $0.25\,\text{R/s}$ and pause time is $10\,\text{s}$.

# Nodes	Width & Height
50	4R, 5R, 6R, 7R, 8R, 10R
100	4R, 6R, 6.8R, 7R, 8R, 9R, 10R, 11R, 12R, 14R
150	4R, 6R, 8.1R, 9R, 10R, 11R, 13R, 15R, 16R
200	4R, 8R, 9R, 9.3R, 10R, 12R, 15R, 16R, 18R

counts. Of course, scenarios with more nodes result in poor simulator performance and longer times to generate results. An alternative is to introduce some level of network partitioning that allows scenarios with fewer nodes and larger A_{sp}Hops. We begin to explore the relationship between A_{sp}Hops and ANP in the next section.

3.3.3 Relationship between A_{sp}Hops and ANP

Using the descriptions and equations of Sections 3.2.1 and 3.2.2, we explored square areas over the full possible range of ANP. Table 3.4 contains the values for each of the input parameters (i.e., number of nodes, width, and height of the simulation area) used to cover the partitioning range (0% to $\approx$90%). We fixed the node speed and node pause time parameters at $0.25\,\text{R/s}$ and $10\,\text{s}$, respectively. Additionally, we paired equivalent width and height parameters to maintain square simulation areas. Our analysis was based on 36 different simulation scenarios, which are shown in Table 3.4.

Figure 3.5 and Table 3.5 show the average network partitioning and A_{sp}Hop results of the 34 simulation scenarios. We note that the full range of no network partitioning (0%) to large network partitioning (90%) is shown. The results illustrate

that a scenario with near-zero ANP (less than 1%) means A_{sp}Hop is low (less than 2.5 hops). As the percentage of ANP increases, A_{sp}Hop initially increases. However, as the percentage of partitioning continues to increase, A_{sp}Hop begins to decrease. At large network partitioning percentages (i.e., over 80%), the only nodes that are not partitioned are in close proximity to each other. As a result, the overall A_{sp}Hop is low.

Figure 3.5 shows that although the peak A_{sp}Hop value increases with number of nodes, the overall trend of each result is similar. Figure 3.5 also shows some standard metrics are impossible, e.g., a scenario with 50 nodes and at least 4 A_{sp}Hop is not possible. We continue to explore the relationship between A_{sp}Hop and ANP in Section 3.5.

In the next section, we develop algorithms that take the average shortest-path hop count and average network partitioning desired for a simulation scenario as inputs. The algorithms then output the number of nodes and simulation area required to generate a simulation scenario that meets the inputs desired.

3.4 Generating Rigorous Scenarios

For generic MANET routing protocol evaluation, Standard 1 and Standard 2 should be followed. To follow the standards, a researcher needs to be able to predict the average shortest-path hop count and average network partitioning for a scenario a priori. We have developed several models that take the desired values of A_{sp}Hops and ANP as inputs, and outputs the area and number of nodes required to create a scenario with the standards specified. We consider square simulation areas in Section 3.4.1, and rectangular simulation areas in Section 3.4.2.

Table 3.5. Results for 34 simulation scenarios in square area study.

Nodes	Width	Height	Part %	Avg Hops
50	4R	4R	0.72	2.41
50	5R	5R	4.67	3.06
50	6R	6R	18.37	3.64
50	7R	7R	38.42	4.07
50	8R	8R	59.91	3.87
50	10R	10R	85.40	2.67
100	4R	4R	0.04	2.30
100	6R	6R	1.61	3.51
100	6.8R	6.8R	4.55	4.07
100	7R	7R	6.10	4.27
100	8R	8R	14.71	4.93
100	9R	9R	25.20	5.87
100	10R	10R	46.48	5.81
100	11R	11R	64.64	5.35
100	12R	12R	76.04	4.86
100	14R	14R	90.59	3.50
150	4R	4R	0.01	2.26
150	6R	6R	0.50	3.35
150	8.1R	8.1R	4.58	4.72
150	9R	9R	8.65	5.45
150	10R	10R	16.75	6.20
150	11R	11R	28.68	6.81
150	13R	13R	60.53	7.10
150	15R	15R	81.16	5.98
150	16R	16R	90.19	4.35
200	4R	4R	0.0	2.24
200	8R	8R	1.55	4.52
200	9R	9R	4.14	5.21
200	9.3R	9.3R	4.86	5.40
200	10R	10R	7.48	5.95
200	15R	15R	61.77	8.35
200	16R	16R	73.60	7.78
200	18R	18R	89.03	5.73

54
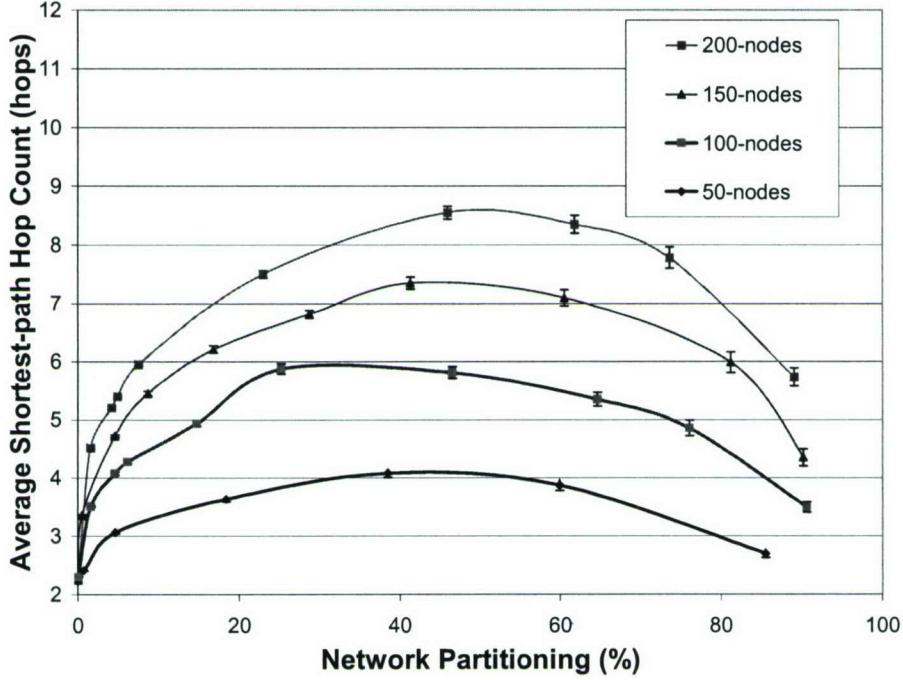
Figure 3.5. ANP versus A_{sp}Hop with 95% confidence intervals. Each point in the plot is the average result from 500 realizations of the given simulation scenario.

3.4.1 Square Simulation Areas

We used linear regression to construct models that predict the values of A_{sp}Hops and ANP for a given scenario. We considered square networks in which nodes move according to the Random Waypoint Model. The input variables are number of nodes, simulation area, node speed, and node pause time. We considered 21 values for number of nodes, the simulation area, and node pause time, and we considered 26 values for node speed. These values are presented in Table 3.6. Our parameters provided a total of $21^3 \times 26 = 240,786$ scenarios. We randomly chose 1,200 of these scenarios. For each of these 1,200 scenarios, we generated 200 independent snapshots

Table 3.6. The parameters and their values used in the square study. NOTE: We do not consider scenarios with less than 50 nodes, as Figure 3.5 illustrates scenarios with less than 50 nodes will not meet Standard 1.

Parameter	Levels
Nodes	50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250
Area (R^2)	40, 43, 46, 49, 53, 56, 59, 62, 66, 69, 72, 75, 79, 82, 85, 88, 92, 95, 98, 101, 105
Speed (R/s)	0.01, 0.02, 0.05, 0.10, 0.20 0.25 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25
Pause (s)	1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40

of the network, computed the average shortest-path hop count and average degree of network partitioning for each snapshot, and averaged over the 200 snapshots.

To construct the models, we set A_{sp}Hops or ANP as the dependent variable, and considered the input variables (number of nodes, simulation area, node speed, and node pause time) as potential predictors. We found that models using the logarithm of the predictors and dependent variables provided a better fit (i.e., goodness of fit of 98.8% with logarithms versus 71.2% without logarithms); however, to consider a value of zero pause time (constant motion), we did not use the logarithm of pause time. We also found that the linear relationship between ANP and the predictor variables is less strong for large values of ANP, making its prediction more difficult

(i.e., goodness of fit drops from 99.3% for ANP < 40% to 86.2% for ANP > 40%). Thus, we constructed our models using only those scenarios with ANP < 40%. We expect that scenarios with ANP < 40% is satisfactory for most MANET routing protocol research.

The fitted models are:

$$
\begin{aligned}
\ln(\text{ANP}) = \ & - \ 2.3774 - 3.04714 \ln(\text{nodes}) \\
& + \ 3.4626 \ln(\text{area}) \\
& + \ 0.00425 \ln(\text{speed}) \\
& - \ 0.00068(\text{pause})
\end{aligned}
\tag{3.3}
$$

and

$$
\begin{aligned}
\ln(\text{A}_{sp}\text{Hops}) = \ & - \ 0.33827 - 0.10941 \ln(\text{nodes}) \\
& + \ 0.5847 \ln(\text{area}) \\
& + \ 0.00015 \ln(\text{speed}) \\
& + \ 0.00014(\text{pause}),
\end{aligned}
\tag{3.4}
$$

where *nodes* is the number of nodes, *area* is the R^2 simulation area, *speed* is the node speed, and *pause* is the node pause time. These models fit well; the coefficient of determination is 99% for Equation 3.3 and 99.1% for Equation 3.4.

Equations 3.3 and 3.4 can be used to construct scenarios that have any desired values of A_{sp}Hops and ANP. Specifically, a researcher provides values for A_{sp}Hops and ANP, along with any two of the independent variables. Equations 3.3 and 3.4

then become two equations with two unknowns, which can be solved to yield values for the two remaining independent variables. However, we note that node speed and node pause time have little effect on the values of A_{sp}Hops and ANP in Equations 3.3 and 3.4. For example, pause time of 40 seconds instead of pause time of 0 seconds decreases ANP by a factor of $e^{-0.00068*40} = 0.973$, a decrease of only (approximately) 5%. Similarly, node speed of 1.25R instead of node speed of 0.25R increases ANP by a factor of $(1.25/0.25)^{0.00425} = 1.007$, an increase of only (approximately) 6%. We, therefore, removed node speed and node pause time from the models and refit. The resulting models are:

$$
\begin{aligned}
\ln(\text{ANP}) = & - 2.39377 - 3.04704 \ln(\text{nodes}) \\
& + 3.46258 \ln(\text{area})
\end{aligned}
\tag{3.5}
$$

and

$$
\begin{aligned}
\ln(\text{A}_{sp}\text{Hops}) = & - 0.33795 - 0.1094 \ln(\text{nodes}) \\
& + 0.5848 \ln(\text{area}),
\end{aligned}
\tag{3.6}
$$

where *nodes* is the number of nodes and *area* is the R^2 simulation area.

Equations 3.5 and 3.6 enable a researcher to input a desired level of A_{sp}Hops and ANP, and then solve for the number of nodes and the simulation area. Of course, with two equations and two unknowns, we can solve the equations for number of nodes and simulation area. The solved equations are:

$$
Nodes = e^{-0.1637} \times ANP^{-0.4168} \times A_{sp}Hops^{2.468}
\tag{3.7}
$$

Table 3.7. Speed and pause time parameters for the model error analysis.

Parameter	Levels
Speed (R/s)	0.075, 0.25, 0.50, 0.75, 1.25
Pause (s)	1, 5, 10, 15, 20

and

$$Area = e^{0.567} \times ANP^{-0.0769} \times A_{sp}Hops^{2.159}. \tag{3.8}$$

For example, to generate a scenario in which the average network partitioning is approximately 5% and the average shortest-path hop count is approximately 4, the number of nodes should be $e^{-0.164} \times 0.05^{-0.417} \times 4^{2.468} \approx 91.6$, and the R^2 area of the simulation should be $e^{0.567} \times 0.05^{-0.0769} \times 4^{2.159} \approx 44.2$. These results confirm our results presented in Section 3.5.2, in which we showed a network with 95 nodes and an area of 44.2 R^2 would have an average shortest-path hop count of approximately 4 and an average network partitioning of approximately 5%. We checked the accuracy of our results for 5% ANP and 4 A_{sp}Hops with a simulation. Specifically, we generated 25 scenarios with 92 nodes, an area of 44.4 R^2, and different values for node speed and node pause time (see Table 3.7). For each scenario, we generated 200 independent snapshots within the scenario (or 5,000 snapshots) and then computed the shortest-path hop count and network partitioning for each snapshot. We then averaged over these 5,000 snapshots to estimate the resulting A_{sp}Hops and ANP for a scenario with 92 nodes and area of 44.4 R^2. The resulting ANP was 0.052 and the resulting A_{sp}Hops was 4.01, which are close to the target values of 0.05 and 4, respectively.

We repeated this accuracy check for a total of 25 combinations of A_{sp}Hops and ANP, which are shown in Table 3.8. For each combination of A_{sp}Hops and ANP, we used Equations 3.7 and 3.8 to compute the number of nodes and simulation area

Table 3.8. A_{sp}Hops and ANP targets for the model error analysis.

Parameter	Levels
A_{sp}Hops	2, 3, 4, 5, 6
ANP	1%, 3%, 5%, 10%, 20%

needed to obtain the specified values of A_{sp}Hops and ANP. We then generated 5,000 independent snapshots of networks with these values for number of nodes and simulation area, using various values for node speed and node pause time (see Table 3.7). Finally, we estimated the resulting A_{sp}Hops and ANP by averaging over the 5,000 snapshots. The results are presented in Figure 3.6. In general, the resulting values of A_{sp}Hops and ANP are close to the target values. The accuracy is best for target values of A_{sp}Hops of 4 and above or ANP less than 10%.

We also verified the accuracy of our original model and our assumption that node speed and pause time have little impact. We executed the same verification tests from Tables 3.7 and 3.8 with our original model that included node speed and node pause time (Equations 3.3 and 3.4). That is, we included node speed and node pause time in each equation, and then solved for A_{sp}Hops and ANP to produce two equations and two unknowns. We then estimated the resulting A_{sp}Hops and ANP by averaging over the 5,000 snapshots. The results are presented in Figure 3.7. Although Figures 3.6 and 3.7 differ, the difference is not significant enough to warrant the use of our original model with two more parameters. The mean squared error between the two models is 0.0001. As a result, we recommend using Equations 3.7 and 3.8 over our original model.

We note that both A_{sp}Hops and ANP measure average behavior of the network in the long run. Thus, scenarios constructed by our method will exhibit approximately
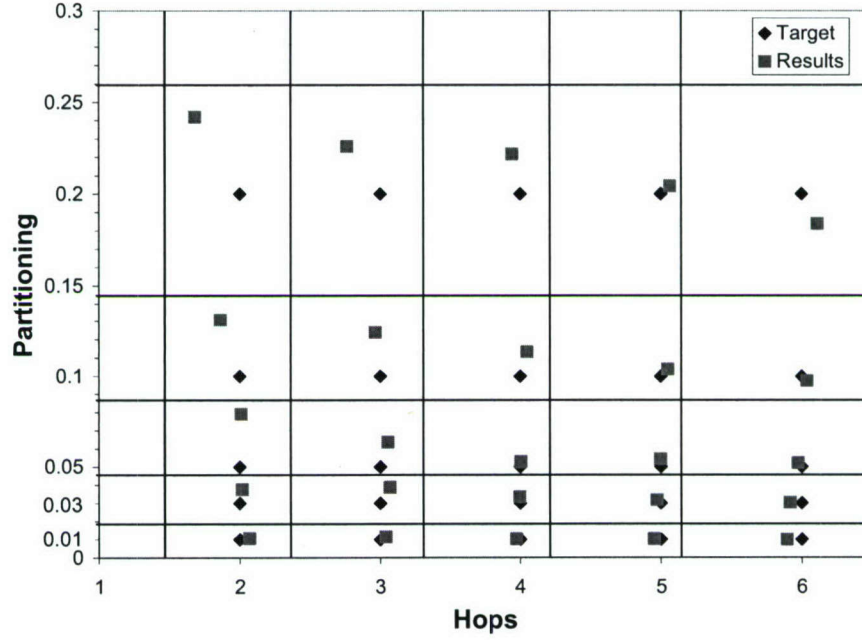
Figure 3.6. Plot of A_{sp}Hops and ANP for both the target values and the resulting simulated values for square simulation areas using our recommended model (Equations 3.7 and 3.8).

the target shortest-path hop count and degree of network partitioning on the average over the long run. The shortest-path hop count and degree of network partitioning will vary around these averages when measured at specific time points, or when measured over short periods of time. This is appropriate, as one would not expect the average number of hops and degree of partitioning to be constant over time in a realistic network scenario.

3.4.2   Rectangular Simulation Areas

In our MobiHoc survey, a majority of MANET simulation studies, 49 of the 59 scenarios (83%) (see Chapter 2), used square simulation simulation areas; 10 of the 59
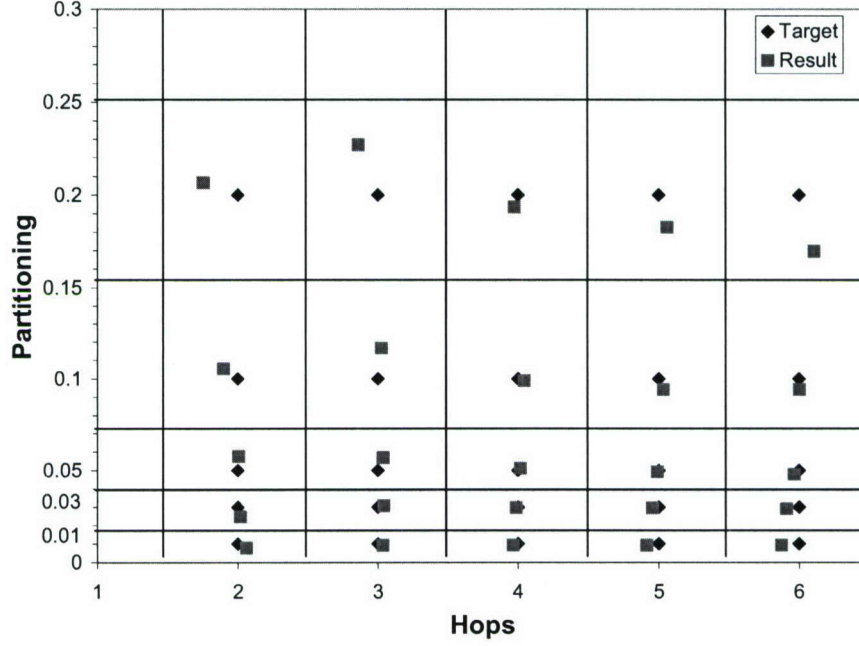
Figure 3.7. Plot of A_{sp}Hops and ANP for both the target values and the resulting simulated values for square simulation areas using our original model (Equations 3.3 and 3.4).

scenarios used rectangular simulation areas. In this section, we consider rectangular scenarios with aspect ratios[4] of 1×2, 1×3, and 1×4.

Similar to our square simulation area study (in Section 3.4.1), we constructed our rectangular models using only those scenarios with ANP < 40%. Also, similar to our square simulation area study, we used linear regression to construct models that allow us to input target values for A_{sp}Hops and ANP. We used the Random Waypoint Model with input values for number of nodes, node speed, and node pause time from Table 3.6. In addition, similar to our square simulation area study, we

---

[4]The aspect ratio is the ratio of the shorter side of the simulation area to the longer side of the simulation area. For a square simulation area, the aspect ratio is 1×1.

used 21 values for the simulation area for each aspect ratio; however, due to results presented in Section 3.5.2, we set the simulation areas for the rectangular simulation area study to be slightly less than the simulation areas for the square simulation area study. Specifically, the simulation areas for the 1×2, 1×3, and 1×4 aspect ratio studies ranged from 35-100R^2, 30-95R^2, and 25-90R^2, respectively.

1×2 Simulation Areas: Similar to our square simulation area study, we found that node speed and node pause time have relatively little effect on A_{sp}Hops and ANP for a 1×2 aspect ratio simulation area. Initially we created models with number of nodes, area, speed, and pause time. However, the p-values for the speed and pause time predictors were statistically insignificant at $\alpha = 0.05$. We, therefore, removed node speed and node pause time from our initial models and refit. The resulting models are:

$$\ln{(\text{ANP})} = -1.9439 - 3.1156\ln{(\text{nodes})}$$
$$+3.4639\ln{(\text{area})} \tag{3.9}$$

and

$$\ln{(\text{A}_{sp}\text{Hops})} = -0.3264 - 0.0802\ln{(\text{nodes})}$$
$$+0.5623\ln{(\text{area})}, \tag{3.10}$$

where *nodes* is the number of nodes and *area* is the R^2 simulation area. Equations 3.9 and 3.10 enable a researcher to input a desired level of A_{sp}Hops and ANP, and then solve for the number of nodes and the simulation area for a 1×2 aspect ratio. The solved equations are:

$$Nodes = e^{0.025} \times ANP^{-0.381} \times A_{sp}Hops^{2.35} \tag{3.11}$$

and

$$Area = e^{0.584} \times ANP^{-0.0544} \times A_{sp}Hops^{2.114}. \tag{3.12}$$

For example, to generate a scenario in which the average network partitioning is approximately 5% and the average shortest-path hop count is approximately 4, the number of nodes should be $e^{0.025} \times 0.05^{-0.381} \times 4^{2.35} \approx 83.4$, and the R^2 area of the simulation should be $e^{0.584} \times 0.05^{-0.0544} \times 4^{2.114} \approx 39.5$. These results confirm our results presented in Section 3.5.2, in which we showed a network with 83 nodes and an area of $40\,R^2$ would have an average shortest-path hop count of approximately 4 and an average network partitioning of approximately 5%. We checked the accuracy of our results for 5% ANP and 4 A_{sp}Hops for a 1×2 rectangle with simulation. Specifically, we generated 25 scenarios with 83 nodes, an area of $39.4\,R^2$, and different values for node speed and node pause time (see Table 3.7). For each scenario, we generated 200 independent snapshots within the scenario (or 5,000 snapshots) and then computed the average shortest-path hop count and average network partitioning for each snapshot. We then averaged over these 5,000 snapshots to estimate the resulting A_{sp}Hops and ANP for a scenario with 83 nodes and area $39.4\,R^2$. The resulting ANP was 0.062 and the resulting A_{sp}Hops was 4.07, which are close to the target values of 0.05 and 4, respectively.

We repeated this accuracy check for a total of 25 combinations of A_{sp}Hops and ANP, which are shown in Table 3.8. For each combination of A_{sp}Hops and ANP, we used Equations 3.11 and 3.12 to compute the number of nodes and simulation area needed to obtain the specified values of A_{sp}Hops and ANP. We then generated 5,000 independent snapshots of networks with these values for number of nodes and simu-

Figure 3.8. Plot of A_{sp}Hops and ANP for both the target values and the resulting simulated values for 1×2 aspect ratio simulation areas using our recommended model (Equations 3.11 and 3.12).

lation area, using various values for node speed and node pause time (see Table 3.7). Finally, we estimated the resulting A_{sp}Hops and ANP by averaging over the 5,000 snapshots. The results are presented in Figure 3.8. In general, the resulting values of A_{sp}Hops and ANP are close to the target values. The accuracy is best for target values of A_{sp}Hops greater than 4 or ANP less than 5%.

1×3 Simulation Areas: The results for the 1×3 aspect ratio study were similar to the results for the 1×2 aspect ratio study. Specifically, we found that node speed and node pause time were not statistically significant for A_{sp}Hops and ANP for a 1×3 aspect ratio simulation area. We, therefore, removed node speed and node pause time

from our initial models and refit. The resulting models are:

$$\ln\,(\text{ANP}) = \; - \;\; 1.2178 - 3.14696\ln\,(\text{nodes})$$
$$+ \;\; 3.3705\ln\,(\text{area}) \tag{3.13}$$

and

$$\ln\,(\text{A}_{sp}\text{Hops}) = \; - \;\; 0.3051 - 0.0455\ln\,(\text{nodes})$$
$$+ \;\; 0.5364\ln\,(\text{area}), \tag{3.14}$$

where *nodes* is the number of nodes and *area* is the R^2 simulation area. Equations 3.13 and 3.14 enable a researcher to input a desired level of A_{sp}Hops and ANP, and then solve for the number of nodes and the simulation area for a 1×3 aspect ratio. The solved equations are:

$$Nodes = e^{0.245} \times ANP^{-0.350} \times A_{sp}Hops^{2.197} \tag{3.15}$$

and

$$Area = e^{0.590} \times ANP^{-0.030} \times A_{sp}Hops^{2.051}. \tag{3.16}$$

For example, to generate a scenario in which the average network partitioning is approximately 5% and the average shortest-path hop count is approximately 4, the number of nodes should be $e^{0.245} \times 0.05^{-0.35} \times 4^{2.197} \approx 76.6$, and the R^2 area of the simulation should be $e^{0.59} \times 0.05^{-0.03} \times 4^{2.051} \approx 33.8$. These results confirm our results presented in Section 3.5.2, in which we showed a network with 75 nodes and an area of 33 R^2 would have an average shortest-path hop count of approximately 4 and an average network partitioning of approximately 5%. We checked the accuracy of our

results for 5% ANP and 4 A_{sp}Hops for a 1×3 rectangle with simulation. Specifically, we generated 25 scenarios with 76 nodes, an area of $34\,R^2$, and different values for node speed and node pause time (see Table 3.7). For each scenario, we generated 200 independent snapshots within the scenario, and then averaged over these 5,000 snapshots to estimate the resulting A_{sp}Hops and ANP. The resulting ANP was 0.060 and the resulting A_{sp}Hops was 4.15, which are close to the target values of 0.05 and 4, respectively.

As before, we repeated this accuracy check for a total of 25 combinations of A_{sp}Hops and ANP, which are shown in Table 3.8. For each combination of A_{sp}Hops and ANP, we used Equations 3.15 and 3.16 to compute the number of nodes and simulation area needed to obtain the specified values of A_{sp}Hops and ANP. We then generated 5,000 independent snapshots of networks, using various values for node speed and node pause time (see Table 3.7), and estimated the resulting A_{sp}Hops and ANP by averaging over the 5,000 snapshots. The results are presented in Figure 3.9. In general, the resulting values of A_{sp}Hops and ANP are close to the target values. The accuracy is best for target values of A_{sp}Hops greater than 4 or ANP less than 5%.

1×4 Simulation Areas: The results for the 1×4 aspect ratio study were similar to the results for the 1×2 and 1×3 aspect ratio studies. Specifically, we found that node speed and node pause time were not statistically significant for A_{sp}Hops and ANP for a 1×4 aspect ratio simulation area. We, therefore, removed node speed and node pause time from our initial models and refit. The resulting models are:

$$\ln{(\text{ANP})} = \;-\;\; 0.55665 - 3.2157 \ln{(\text{nodes})}$$
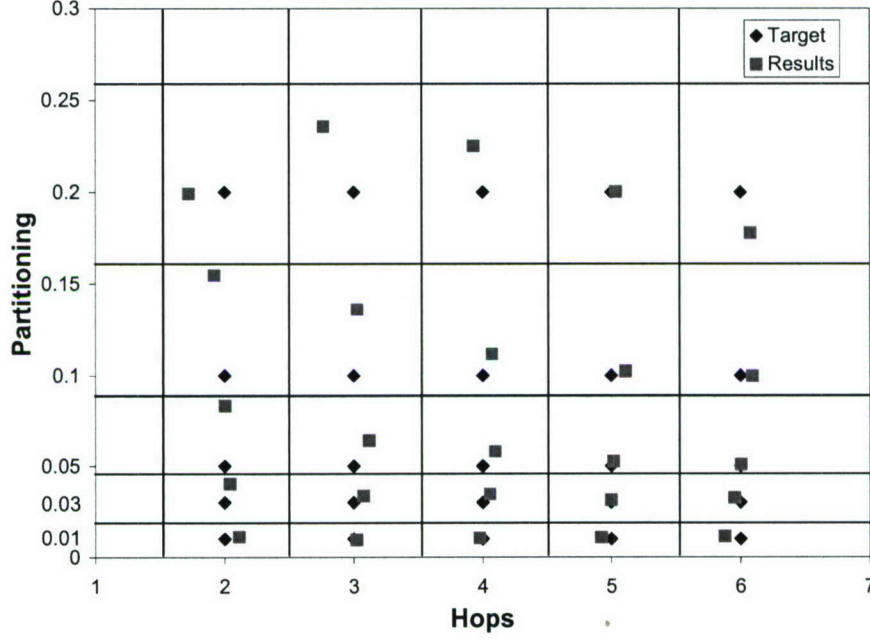$$+ \;\; 3.3385 \ln{(\text{area})} \tag{3.17}$$

Figure 3.9. Plot of A_{sp}Hops and ANP for both the target values and the resulting simulated values for 1×3 aspect ratio simulation areas using our recommended model (Equations 3.15 and 3.16).

and

$$\ln\left(A_{sp}\text{Hops}\right) = \quad - \quad 0.2850 - 0.0161\ln\left(\text{nodes}\right)$$
$$+ \quad 0.5149\ln\left(\text{area}\right), \tag{3.18}$$

where *nodes* is the number of nodes and *area* is the R^2 simulation area. The solved equations are:

$$Nodes = e^{0.415} \times ANP^{-0.321} \times A_{sp}Hops^{2.08} \tag{3.19}$$

and

$$Area = e^{0.566} \times ANP^{-0.01} \times A_{sp}Hops^{2.01}. \tag{3.20}$$

For example, to generate a scenario in which the average network partitioning is approximately 5% and the average shortest-path hop count is approximately 4, the number of nodes should be $e^{0.415} \times 0.05^{-0.321} \times 4^{2.08} \approx 70.8$, and the R^2 area of the simulation should be $e^{0.566} \times 0.05^{-0.01} \times 4^{2.01} \approx 29.4$. These results confirm our results presented in Section 3.5.2, in which we showed a network with 70 nodes and an area of 28 R^2 would have an average shortest-path hop count of approximately 4 and an average network partitioning of approximately 5%.

As before, we checked the accuracy of our results for 5% ANP and 4 A_{sp}Hops for a 1 × 4 rectangle with simulation. We averaged the results over 5,000 snapshots, to estimate the resulting A_{sp}Hops and ANP for a scenario with 71 nodes and an area of 29.1 R^2. The resulting ANP was 0.046 and the resulting A_{sp}Hops was 3.99, which are close to the target values of 0.05 and 4, respectively.

As before, we repeated this accuracy check for a total of 25 combinations of A_{sp}Hops and ANP, which are shown in Table 3.8. For each combination of A_{sp}Hops and ANP, we used Equations 3.19 and 3.20 to compute the number of nodes and simulation area needed. We then generated 5,000 independent snapshots of networks, using various values for node speed and node pause time (see Table 3.7), and estimated the resulting A_{sp}Hops and ANP. The results are presented in Figure 3.10. In general, the resulting values of A_{sp}Hops and ANP are close to the target values. The accuracy is best for target values of A_{sp}Hops greater than 4 or ANP less than 5%.
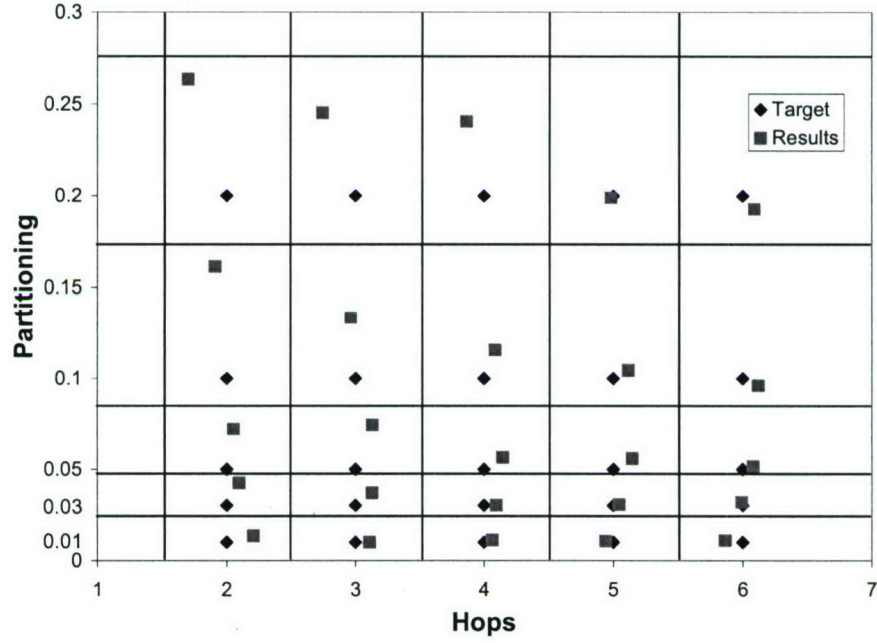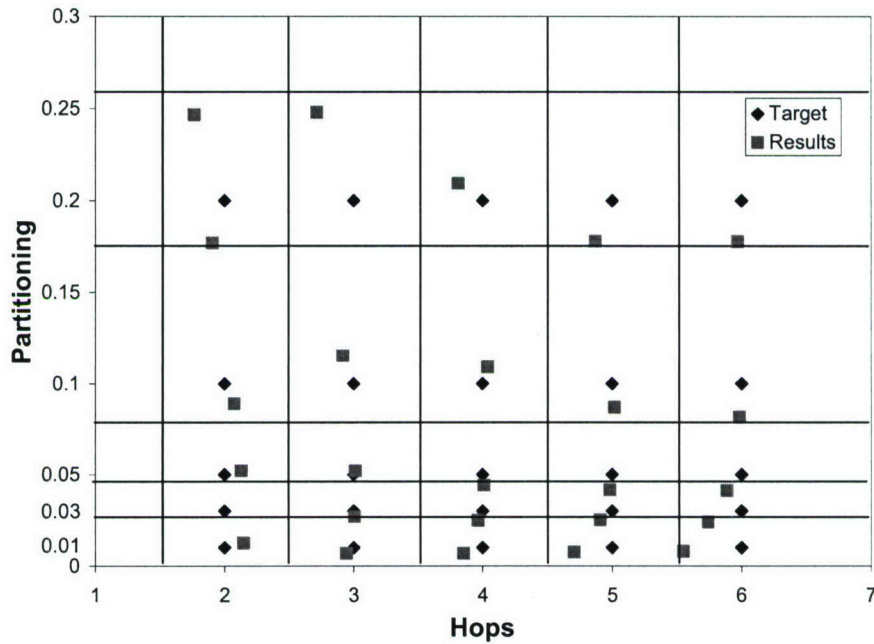
Figure 3.10. Plot of A_{sp}Hops and ANP for both the target values and the resulting simulated values for 1×4 aspect ratio simulation areas using our recommended model (Equations 3.19 and 3.20).

3.5 Scenarios with Standards

We note that all our equations in Section 3.4 will output the simulation area and number of nodes that approximately meet the inputs for A_{sp}Hops and ANP. Instead, the researcher may prefer to consider the range of scenarios that have A_{sp}Hops greater than a minimal value and ANP smaller than a maximum value. We explore a range of scenarios that have A_{sp}Hops greater than a minimal value and ANP smaller than a maximum value in this section.

Imagine a fixed number of nodes in a small square simulation area, such that the number of nodes is larger than the minimum needed to meet our standards. Imagine

that these nodes are tightly packed, so that all nodes are within a single transmission range. This configuration will have no partitioning (ANP = 0), since every node will be within one hop of every other node. However, A_{sp}Hops will be equal to 1 hop, which does not meet our standard for hops (Standard 1).

To increase A_{sp}Hops, imagine gradually expanding the simulation area, retaining its square shape. As the area increases, both A_{sp}Hops and ANP will increase. At some point, the value of A_{sp}Hops will reach the researcher's desired value for A_{sp}Hops (suppose x hops). If, at that point, ANP is still less than the desired degree of network partitioning (suppose $y\%$), then this simulation scenario will meet our two standards (A_{sp}Hops $\geq x$ hops and ANP $\leq y\%$); in fact, this scenario will be the smallest simulation area that meets our two standards for the given number of nodes. Now, suppose that ANP is less than $y\%$ and imagine expanding the simulation area still further. At some point, ANP will reach $y\%$; the resulting simulation scenario will be the largest simulation area that meets our two standards for the given number of nodes. If one expands the simulation area further ANP will be greater than $y\%$. In summary, given a target value of A_{sp}Hops, a target value of ANP, and enough nodes, there will be a range for the simulation area size that meets our two standards. Furthermore, some standard metrics are impossible. For example, Figure 3.5 shows that a scenario with 50 nodes and at least 4 A_{sp}Hop is not possible. Thus, for a given aspect ratio, a minimum number of nodes is needed to ensure our two standards can be met. We investigate this result further in Section 3.5.2.

In the rest of this section, we first justify the standard metrics that we have chosen in our own research. We note, however, that any value a researcher finds appropriate for either metric could be used. Then, given different values for number of nodes, we consider the minimum and maximum simulation area sizes needed to

meet our two standards with our chosen targets for square and rectangular simulation areas.

3.5.1 Our Chosen Targets

In scenarios with A_{sp}Hops of 3 hops or less, there are at most 2 intermediate nodes on average between source and destination. Four hops for A_{sp}Hops ensures that there are at least 3 intermediate nodes on average, which increases the frequency with which packets are routed beyond immediate neighbors. Thus, we have chosen an average of 4 shortest-path hops as the standard in our research. Again, however, any value a researcher finds appropriate for average shortest-path hops could be used.

To determine a value for average network partitioning (ANP), we measured the delivery ratio of the Location Aided Routing (LAR) [48] protocol on NS-2.1b7a [33] using the steady-state Random Waypoint Model (RWM) [68]. We tested several scenarios with values of ANP ranging from 0 to 28% in 100 node scenarios. Each scenario had 20 source and destination nodes, with constant bit rate traffic of four packets per second from each source for 100 seconds. From the simulation data, we conclude that delivery failures occur when network partitioning is present, and many of these failures do not reflect on the performance of generic MANET routing protocols. While it is unrealistic to insist on no network partitioning [36], we desired to keep the average amount of network partitioning low in order to rigorously evaluate our MANET routing protocols. While any low level of ANP that a researcher finds appropriate could be used, we chose ANP < 5% as the standard in our research.

In Section 3.5.2, we describe a variety of scenarios that meet both our standard for hops (A_{sp}Hops $\geq$ 4 hops) and our standard for average network partitioning (ANP $\leq$ 5%). In other words, the scenarios presented in the next section meet our

standards with the standard targets that we have chosen (i.e., A_{sp}Hops ≥ 4 hops and ANP $\leq 5\%$); however, our models in Section 3.4 can be used to construct other scenarios that meet Standard 1 and Standard 2 with any target value for each metric that a researcher finds appropriate.

3.5.2   Our Standard Simulation Scenarios

For the results presented in this section, we calculated the combinations of number of nodes and simulation area width and height using the area and node equations from Section 3.4. We based our results on 500 independent realizations of the scenario using the steady-state RWM.

Square Simulation Scenarios: We now present numerous simulation scenarios with square areas that meet our two standards with our chosen targets. Figure 3.11 presents results for a 150-node square simulation area using the RWM with node speed $0.25\,R/s$ and pause time $10\,s$. There are two curves in Figure 3.11, one of which plots simulation area versus A_{sp}Hops and one of which plots simulation area versus ANP. The solid horizontal line represents both our standard for hops (A_{sp}Hops ≥ 4 hops) and our standard for partitioning (ANP $\leq 5\%$). Figure 3.11 illustrates that areas less than about $7.05\,R \times 7.05\,R$ ($\approx 49\,R^2$) have A_{sp}Hops < 4 hops; in other words, these simulation areas are too small to meet our standard for hops. Areas greater than about $8.2\,R \times 8.2\,R$ ($\approx 67\,R^2$) have ANP $> 5\%$; in other words, these simulation areas are too large to meet our standard for partitioning. Finally, areas between approximately $49\,R^2$ and $67\,R^2$ have A_{sp}Hops ≥ 4 hops and ANP $< 5\%$; simulation scenarios with areas between these two values will, in most cases, meet our two standards.

Figure 3.11. For 150 node square scenarios, the dashed curve plots A_{sp}Hops versus simulation area. Areas for which the curve is above the horizontal line meet our hops standard ($\geq$ 4 hops). The solid curve plots ANP versus area. Areas for which the curve is below the horizontal line meet our ANP standard ($\leq$ 5%). Therefore, for a square 150 node scenario, simulation areas between $49\,R^2$ and $67\,R^2$ meet our two standards. The results assume a steady-state RWM, node speed $0.25\,R/s$, and pause time $10\,s$.

We note that if the number of nodes is too small, then no simulation scenario will meet our two standards. Figure 3.12 presents results for a 50-node square scenario. In order to meet our standard for partitioning, the simulation area must be less than about $27\,R^2$. However, in order to meet our standard for hops, the simulation area must be greater than $43\,R^2$. Therefore, no square scenario with 50 nodes will meet our two standards.
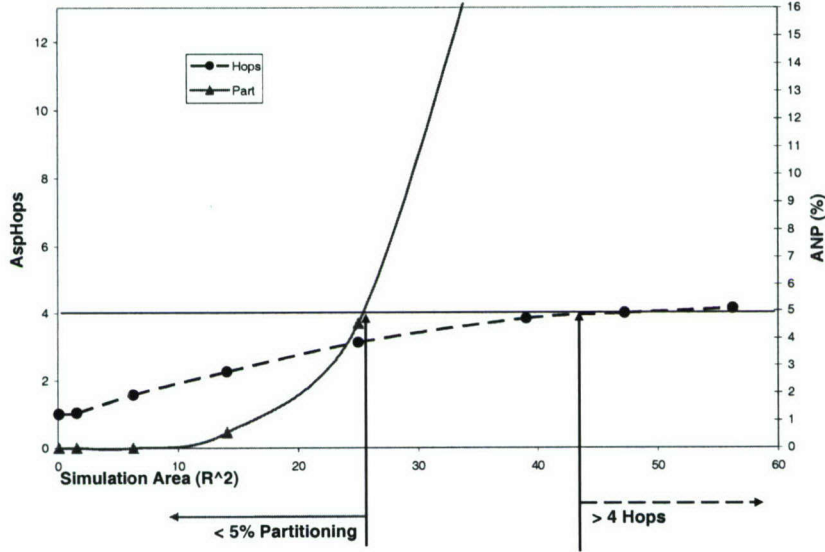
Figure 3.12. For 50 node square scenarios, the dashed curve plots A_{sp}Hops versus simulation area. Areas for which the curve is above the horizontal line meet our hops standard ($\geq$ 4 hops). The solid curve plots ANP versus area. Areas for which the curve is below the horizontal line meet our ANP standard ($\leq$ 5%). Therefore, for a square 50 node scenario, there is no area that meets both standards. These results assume a steady-state RWM, node speed 0.25 R/s, and pause time 10 s.

The smallest number of nodes that can be used to meet our two standards in a square scenario is about 95, which follows our result from Section 3.4.1. Figure 3.13 presents results for a 95-node square scenario. An area of about 6.65 R×6.65 R ($\approx 44\,R^2$) just meets both standards, A_{sp}Hops $\geq$ 4 hops and ANP $\leq$ 5%. Smaller simulation areas with 95 nodes will fail to meet our hops standard, and larger simulation areas with 95 nodes will fail to meet our partitioning standard.

To estimate the minimum and maximum simulation areas that will meet our two standards for various numbers of nodes in square scenarios, we used Equations 3.7 and
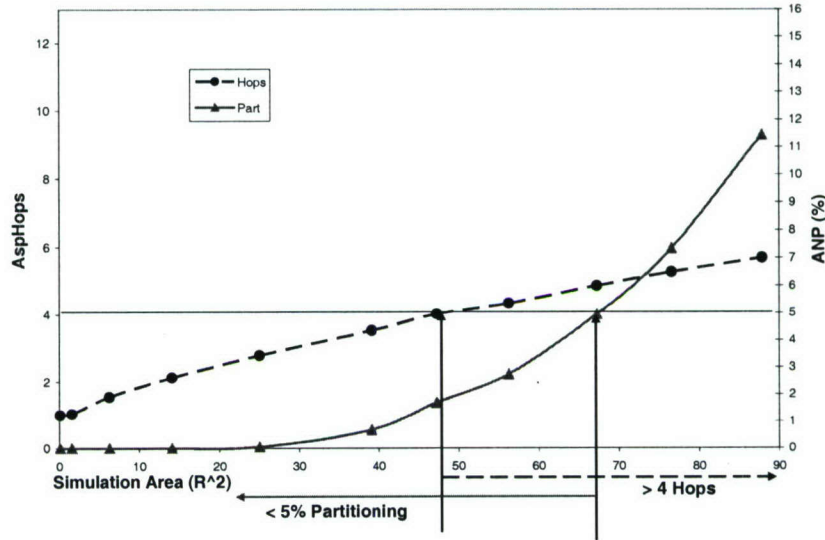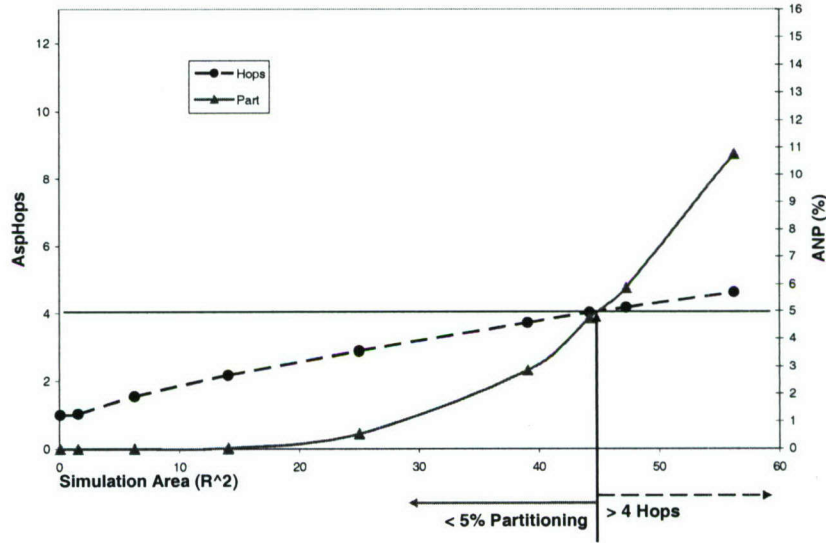
Figure 3.13. For 95 node square scenarios, the dashed curve plots A_{sp}Hops versus simulation area. Areas for which the curve is above the horizontal line meet our hops standard ($\geq$ 4 hops). The solid curve plots ANP versus area. Areas for which the curve is below the horizontal line meet our ANP standard ($\leq$ 5%). Therefore, for a square 95 node scenario, both standards are just met in a simulation area of about $44\,R^2$. These results assume steady-state RWM, node speed $0.25\,R/s$, and pause time $10\,s$.

3.8. We first set A_{sp}Hops $= 4$ and ANP $= 5\%$ as our targets, and then calculated the smallest number of nodes that will meet our standard targets. We then fixed A_{sp}Hops at 4, and decremented ANP from 5% to 0%, by increasing the simulation area in increments of $25\,R^2$. This process determined the minimum simulation area needed for larger numbers of nodes. We also fixed ANP at 5% and incremented A_{sp}Hops past 4, by increasing the simulation area in increments of $25\,R^2$. This process determined the maximum simulation area possible for larger numbers of nodes.

Table 3.9. Approximate min. and max. simulation areas for square scenarios for numbers of nodes (n). Fixed parameters were speed 0.25 R/s and pause time 10 s.

n	Minimum Area	Maximum Area
95	$6.65\,\mathrm{R}\times6.65\,\mathrm{R}$	$6.65\,\mathrm{R}\times6.65\,\mathrm{R}$
100	$6.70\,\mathrm{R}\times6.70\,\mathrm{R}$	$6.80\,\mathrm{R}\times6.80\,\mathrm{R}$
125	$6.90\,\mathrm{R}\times6.90\,\mathrm{R}$	$7.60\,\mathrm{R}\times7.60\,\mathrm{R}$
150	$7.05\,\mathrm{R}\times7.05\,\mathrm{R}$	$8.20\,\mathrm{R}\times8.20\,\mathrm{R}$
200	$7.20\,\mathrm{R}\times7.20\,\mathrm{R}$	$9.30\,\mathrm{R}\times9.30\,\mathrm{R}$
230	$7.30\,\mathrm{R}\times7.30\,\mathrm{R}$	$10.00\,\mathrm{R}\times10.00\,\mathrm{R}$

For each scenario, we generated 500 independent realizations of a steady-state RWM with node speed 0.25 R/s and pause time 10 s, from which we estimated A_{sp}Hops and ANP. Table 3.9 presents the results. For each number of nodes, the minimum area is the smallest simulation area found that meets our standard for hops (A_{sp}Hops $\geq$ 4 hops), and the maximum area is the largest simulation area found that meets our standard for partitioning ($\leq 5\%$).

Rectangular Simulation Scenarios: We repeated our estimation of minimum and maximum simulation areas for a given number of nodes in rectangular scenarios with aspect ratios of 1×2, 1×3, and 1×4, using our equations for each aspect ratio from Section 3.4. For each of these scenarios, we generated 500 independent realizations of a steady-state RWM with node speed 0.25 R/s and pause time 10 s, from which we estimated A_{sp}Hops and ANP. Table 3.10 presents the results. For a given number of nodes and a given aspect ratio, the minimum area is the smallest simulation area found that meets Standard 1 with our chosen target for hops (A_{sp}Hops $\geq$ 4 hops), and the maximum area is the largest simulation area found that meets Standard 2 with our chosen target for partitioning (ANP $\leq 5\%$).

Table 3.10. Approximate minimum and maximum simulation areas for rectangular scenarios for various numbers of nodes. Fixed parameters were $0.25\,\mathrm{R/s}$ node speed and $10\,\mathrm{s}$ node pause time.

n	Aspect Ratio	Minimum Area	Maximum Area
85	1×2	$4.35\,\mathrm{R} \times 8.7\,\mathrm{R}$	$4.35\,\mathrm{R} \times 8.70\,\mathrm{R}$
90	1×2	$4.40\,\mathrm{R} \times 8.8\,\mathrm{R}$	$4.50\,\mathrm{R} \times 9.00\,\mathrm{R}$
100	1×2	$4.475\,\mathrm{R} \times 8.95\,\mathrm{R}$	$4.725\,\mathrm{R} \times 9.45\,\mathrm{R}$
125	1×2	$4.575\,\mathrm{R} \times 9.15\,\mathrm{R}$	$5.25\,\mathrm{R} \times 10.5\,\mathrm{R}$
150	1×2	$4.65\,\mathrm{R} \times 9.3\,\mathrm{R}$	$5.70\,\mathrm{R} \times 11.4\,\mathrm{R}$
180	1×2	$4.75\,\mathrm{R} \times 9.5\,\mathrm{R}$	$6.225\,\mathrm{R} \times 12.45\,\mathrm{R}$
200	1×2	$4.80\,\mathrm{R} \times 9.6\,\mathrm{R}$	$6.55\,\mathrm{R} \times 13.1\,\mathrm{R}$
220	1×2	$4.85\,\mathrm{R} \times 9.7\,\mathrm{R}$	$6.85\,\mathrm{R} \times 13.7\,\mathrm{R}$
75	1×3	$3.275\,\mathrm{R} \times 9.825\,\mathrm{R}$	$3.275\,\mathrm{R} \times 9.825\,\mathrm{R}$
100	1×3	$3.35\,\mathrm{R} \times 10.05\,\mathrm{R}$	$3.80\,\mathrm{R} \times 11.4\,\mathrm{R}$
125	1×3	$3.40\,\mathrm{R} \times 10.2\,\mathrm{R}$	$4.175\,\mathrm{R} \times 12.525\,\mathrm{R}$
150	1×3	$3.45\,\mathrm{R} \times 10.35\,\mathrm{R}$	$4.55\,\mathrm{R} \times 13.65\,\mathrm{R}$
175	1×3	$3.50\,\mathrm{R} \times 10.5\,\mathrm{R}$	$4.925\,\mathrm{R} \times 14.775\,\mathrm{R}$
200	1×3	$3.55\,\mathrm{R} \times 10.65\,\mathrm{R}$	$5.20\,\mathrm{R} \times 15.6\,\mathrm{R}$
70	1×4	$2.60\,\mathrm{R} \times 10.4\,\mathrm{R}$	$2.60\,\mathrm{R} \times 10.4\,\mathrm{R}$
90	1×4	$2.65\,\mathrm{R} \times 10.6\,\mathrm{R}$	$2.975\,\mathrm{R} \times 11.9\,\mathrm{R}$
100	1×4	$2.675\,\mathrm{R} \times 10.7\,\mathrm{R}$	$3.15\,\mathrm{R} \times 12.6\,\mathrm{R}$
125	1×4	$2.725\,\mathrm{R} \times 10.9\,\mathrm{R}$	$3.50\,\mathrm{R} \times 14.0\,\mathrm{R}$
150	1×4	$2.75\,\mathrm{R} \times 11.0\,\mathrm{R}$	$3.875\,\mathrm{R} \times 15.5\,\mathrm{R}$

As mentioned previously, if the number of nodes is too small, then no simulation scenario will meet our two standards with our chosen targets ($\mathrm{A}_{sp}\mathrm{Hops} \geq 4$ hops and $\mathrm{ANP} \leq 5\%$). In Table 3.10, the smallest number of nodes listed for each aspect ratio is the smallest number of nodes that can be used to meet our two standards in that aspect ratio. Specifically, the smallest number of nodes that can be used to meet our two standards in a 1×2, 1×3, and 1×4 aspect ratio is about 85, 75, and 70 nodes, respectively. (These results follow the results from Section 3.4.2.) Note that as the

aspect ratio goes from 1×1 to 1×4, the smallest number of nodes required to meet our two standards decreases.

3.6 Recommendations and Conclusions

In order to ensure that scenarios provide an effective platform for testing generic MANET routing protocols, we recommend using the average shortest-path hop count and the average amount of network partitioning to characterize simulation scenarios. To calculate A_{sp}Hops, build the multi-hop connectivity matrix at regular intervals throughout the simulation. The value of A_{sp}Hops is found by averaging all non-zero entries in all evaluations of the multi-hop connectivity matrix (Equation 3.1). To calculate ANP, evaluate the connectivity matrix at regular intervals throughout the simulation. The value of ANP is the proportion of entries in all evaluations of the multi-hop connectivity matrix that are equal to 0 (Equation 3.2).

Conclusion #1: We presented algorithms that enable investigators to specify desired values for ANP and A_{sp}Hops, then construct a simulation scenario that meets these target values to a close approximation. Our specific conclusions for this work follow.

A. Our models work when the target value of A_{sp}Hops is between 3 and 6 and the target value of ANP is between 1% and 20%.

B. Node speed and node pause time have little impact on A_{sp}Hops and ANP, if speed is within the range 0.01-1.25 R/sec and pause time is less than 40 seconds.

C. Equations 3.7 and 3.8 can be used to construct scenarios with square simulation areas that meet specified values for A_{sp}Hops and ANP.

D. Equations 3.11 and 3.12 can be used to construct scenarios with simulation areas with 1×2 aspect ratios and specified values for A_{sp}Hops and ANP.

E. Equations 3.15 and 3.16 can be used to construct scenarios with simulation areas with 1×3 aspect ratios and specified values for A_{sp}Hops and ANP.

F. Equations 3.19 and 3.20 can be used to construct scenarios with simulation areas with 1×4 aspect ratios and specified values for A_{sp}Hops and ANP.

Conclusion #2: We note that both A_{sp}Hops and ANP measure long-run average behavior of the network. Thus, scenarios constructed by our method will exhibit approximately the target number of hops and approximately the target degree of network partitioning on the average over the long run. The shortest-path hop count and degree of network partitioning will vary around these averages when measured at specific time points, or when measured over short periods of time. This is appropriate, as one would not expect the average number of hops and degree of partitioning to be constant over time in a realistic network scenario.

Conclusion #3: For a given aspect ratio, there exists a smallest number of nodes that can be used to meet our two standards. The smallest number of nodes that can be used in a 1×1, 1×2, 1×3, and 1×4 simulation area for A_{sp}Hops ≥ 4 hops and ANP $\leq 5\%$ is approximately 95, 85, 75, and 70 nodes, respectively. As the aspect ratio goes from 1×1 to 1×4, the smallest number of nodes required to meet our standards decreases.

Conclusion #4: For a given aspect ratio and a given number of nodes, there exists a smallest simulation area that can be used to meet our standard for hops. For a given aspect ratio and a given number of nodes, there exists a largest simulation area that can be used to meet our partitioning standard.

The main contributions of this chapter are (1) to highlight that we need standards to obtain rigorous evaluation of MANET protocols and to begin defining these standards, (2) to propose two standards that should be employed to ensure long routes are available and used in the evaluation of generic MANET routing protocols, (3) to provide algorithms that researchers can use to determine the number of nodes and area required to generate desired A_{sp}Hops and ANP levels and, therefore, construct scenarios that meet their standards (4) to illustrate our method that others can modify to generate scenarios that use a different mobility model or a different propagation model, with different values for both the minimum average shortest-path hop count and the maximum amount of network partitioning.

Chapter 4

DISCOVERING VARIABLES THAT AFFECT MANET PROTOCOL PERFORMANCE

4.1 Introduction

Much recent research (e.g., [74, 75]) has been concerned with improving the credibility of network simulation studies. Designing a simulation study involves choosing values for a number of variables, and for a study to be credible the researcher must understand how these choices may affect the simulation results. In this chapter, we identify a list of variables whose values can substantially affect the results of a simulation, but which have not received a corresponding amount of attention in the literature.

This chapter is organized as follows. Section 4.2 briefly discusses other work related to MANET simulation-based research. Section 4.3 describes the setup of our study. Section 4.4 describes the methods we used to construct a list of variables that appeared to have a substantial effect on simulation results. Section 4.5 presents a validation study that shows that the variables we selected do indeed have a substantial impact. Finally, Section 4.6 provides our conclusions along with suggestions for future work.

4.2 Related Work

We believe our work is the first to evaluate a *large* number of MANET simulation variables, with the goal of determining which variables have the greatest effect on a typical MANET simulation study. The most related studies to our work are the recent design of experiments (DOE) and variable-based studies on MANETs. We briefly summarize these related works in this section and in Table 4.1.

Vadde and Syrotiuk [101] studied delay sensitive applications. They used a DOE to analyze the impact of quality-of-service (QoS) architecture, routing proto-col, medium access control (MAC) protocol, packet send rate, and node speed on throughput and latency. They looked at both main and two-way interactions.

Totaro and Perkins [100] also conducted a DOE study with MANETs. Their study focused on understanding the relationship between network density, node speed, number of packet sources, network size, MAC protocol, and transport protocol. Their study used a 2^k factorial design to determine both the main effects and the two-way interaction effects on latency, delivery ratio, and overhead.

Perkins et al. [80] also used DOE to quantify the effects of various variables and their interactions on performance. This study quantified the factors of node speed, node pause time, network size, number of sources, and routing protocol to ultimately aid in the design of adaptive protocols. They used throughput, overhead, and power consumption as metrics.

Barrett et al. [10] used an analysis of variance method to study the interaction of routing and medium access protocols as node speed, and injection rate (i.e., packet send rate and packet size) are varied. Their study used sound statistical analysis to quantify the parameter interactions on latency, delivery ratio, throughput, and fairness.

Table 4.1. Factors, levels, and metrics for these studies

	Study			
	[101]	[100]	[80]	[10]
Variables				
	QoS Architecture Routing Protocol MAC Protocol Packet Send Rate Node Speed	Network Density Node Speed # of Packet Sources Network Size MAC Protocol Transport Protocol	Node Speed Node Pause Time Network Size # of Sources Routing Protocol	Routing Protocol MAC Protocol Node Speed Pkt. Send Rate and Pkt. Size
Levels				
Routing	Ad-hoc On-Dem. Dist. Vect. (AODV) [79] Dynamic Source Routing (DSR) [43]	LAR	AODV DSR	AODV DSR Location Aided Routing (LAR) [48]
MAC	802.11 [25] Enh. Dist. Coord. Func. (EDCF) [12]	802.11 w/ RTS 802.11 w/o RTS	802.11	Carrier Sense Multi. Acc. (CSMA) [66] 802.11 Med. Acc. Coll. Avoid. (MACA) [44]
Mobility	Random Waypoint Model (RWM) [43]	RWM	RWM	RWM Uniform Grid Exp. Corr. Rand. Mod. (ECRM) [86]
Metrics				
	Throughput Latency	Latency Delivery Ratio Overhead	Throughput Overhead Power Consumption	Latency Delivery Ratio Throughput Fairness

Table 4.2. Variable Categories

Category	Description
Simulator	Chosen simulator, models, and layers
Scenario	Chosen scenarios and physical characteristics
Traffic	Chosen packet activity and characteristics
Protocol	Chosen protocol and variables

These studies illustrate the trend in the MANET community toward rigorous statistical-based analysis of MANET performance, both inside and outside an individual routing protocol and across layers of the network stack. However, the goals of these previous studies are different than our goals. The goals of the previous work were to determine interactions among a small set of variables across layers or build predictive models. We do not specify variables to study up front, nor do we develop prediction models. Instead, we use statistical techniques on a *large* number of variables to discover the variables that significantly effect performance of a MANET routing protocol.

4.3   Setup

When designing a simulation study, a researcher must make some fundamental choices, such as the network simulator and mobility model to use. These choices can be divided into four categories: simulator variables, scenario variables, traffic variables, and protocol variables (see Table 4.2). We have chosen a fairly typical setup for our study. In this section, we discuss the constant variables that we chose for each of these four categories.

Table 4.3. Study Constants

Item	Setting	Category
Simulator	NS-2	Simulator
Version	NS2.1b7a	Simulator
Channel	WirelessChannel	Simulator
Antenna	OmniAntenna	Simulator
Interface	WirelessPHY CPThresh = 10.0 W CSThresh = 1.559e-11 W RXThresh = 3.652e-10 W Freq = 914e6 MHz L = 1.0 Rb = 2e6 B	Simulator
MAC	802.11	Simulator
Queue	Priority Droptail	Simulator
LinkLayer	LL mindelay 50 μs avgdelay 25 μs	Simulator
Movement model	SS-Random waypoint	Scenario
Duration	100 seconds	Scenario
Class	Peer-to-peer	Traffic
Generation	CBR	Traffic
Routing Protocol	LAR	Protocol

Simulator Of the many discrete-event network simulators available [85], the Network Simulator-2 (NS-2) [33] is the most commonly used simulator in MANET research (see Chapter 2). Thus, in this study, we used NS-2, version NS2.1b7a, with the wireless extensions from the Monarch Project [82].

Table 4.3 lists the eight simulator constants used in our studies. We used the wireless channel with an omnidirectional antenna and configured the interface for the 914 MHz Lucent WaveLAN DSSS radio. This WaveLAN interface has a capture threshold (CPThresh) of 10 and a carrier sense threshold (CSThresh) of 1.559e-11 W

for a node to sense an incoming packet. It then has a receive threshold (RXThresh) of 3.652e-10 W for the receiving power required to both sense an incoming packet and receive it completely. The operating frequency (Freq) for the WaveLAN radio is 914 MHz and the system loss factor (L) is 1. The receiving bandwidth (Rb) capability of the interface is 2 MB. The MAC protocol we chose for our study was 802.11 with a priority drop tail queue. We also used the default link layer object for NS-2 (LL) with a minimum delay of $50\,\mu s$ and an average delay of $25\,\mu s$ added to each packet when it is sent up or down the network stack from the link layer.

Scenario The NS-2 simulator requires a scenario to provide the movement and positions of nodes throughout the simulation. These scenarios are generated from mobility models. There are many mobility models available for the MANET community to use [18]. We used the Random Waypoint Model (RWM) because it is the most commonly used mobility model (see Chapter 2). As shown in Table 4.3, we used the steady-state version of the RWM [68] that starts all nodes in the steady-state distribution of the RWM. Use of the steady-state RWM allows us to analyze a simulation scenario from time zero, without initialization bias associated with initial node movement. All of our simulations started in the steady-state and were executed for 100 seconds; we use several runs to average results and reduce variation from any one result.

Most MANET simulation scenarios are described by the number of nodes, transmission range, and area width and height. In addition to listing these parameters for a scenario, we further characterize the scenario by documenting the derived parameters from Table 2.6 and Table 4.4.

Table 4.5 lists the scenarios we used in our studies, as well as the derived parameters for the scenarios. (We used node speed of 10 m/s and pause time of 20 seconds

Table 4.4. Derived scenario parameters from Chapter 3.

Parameter	Description	Formula
Average Network Partitioning (ANP)	Percentage of node pairs with no available route, thus restricting communication.	$\dfrac{z}{n(n-1)T}$
Average Shortest Path Hops (A_{sp}Hops)	The smallest number of links needed to allow two nodes to communicate. The average shortest-path hop count is the average of all shortest-path hop counts for all node pairs.	$\dfrac{\sum\limits_{i=1}^{T} hops_i}{\sum\limits_{i=1}^{T} paths_i}$
$n = \#$ of nodes, $T = \#$ of matrices, $z = \#$ of 0s in matrix representing no available route, $hops_i$ $(paths_i) = \#$ of hops (paths) in matrix at time i		

to determine the value of two derived parameters, i.e., average network partitioning and average shortest-path hops.) The selection scenario, which determines the variables with the largest impact on delivery ratio, has the following characteristics: 100 nodes, 270 m×270 m simulation area, and a transmission range of 40 m. We used separate validation study scenarios to validate our results with different scenarios and derived parameters. Validation scenario I has the following characteristics: 100 nodes, 270 m×270 m simulation area, and a transmission range of 100 m. Validation scenario II has the following characteristics: 150 nodes, 400 m×400 m simulation area, and a transmission range of 100 m. We note our validation scenarios are similar to scenarios used in MANET simulation studies (see Tables 2.4 and 2.5).

Traffic In the traffic category, we used peer-to-peer traffic throughout our study. In addition, as noted in Table 4.3, all of the traffic generated is constant bit rate (CBR).

Table 4.5. Scenarios in our study.

Parameter	Selection Scenario	Validation	
		Scenario I	Scenario II
Number of Nodes	100	100	150
Dimensions	$270\,\text{m} \times 270\,\text{m}$	$270\,\text{m} \times 270\,\text{m}$	$400\,\text{m} \times 400\,\text{m}$
Transmission Range	40 m	100 m	100 m
Node Density	0.00137 nodes/m^2	0.00137 nodes/m^2	0.00094 nodes/m^2
Node Coverage	5026.5 m^2	31,416 m^2	31,416 m^2
Footprint	6.90%	43.1%	19.6%
Maximum Path	381.8 m	381.8 m	565.7 m
Network Diameter	9.52 hops	3.8 hops	5.6 hops
Neighbor Count	7 nodes	43 nodes	29 nodes
ANP	5 %	0	0
A$_{sp}$Hops	4 hops	1.88 hops	2.581 hops

Protocol Lastly, as noted in Table 4.3, we used the Location Aided Routing (LAR) [48] protocol as a constant in this study, as LAR is a popular routing protocol that performs effectively [19]. Specifically, LAR is an on-demand source routing protocol that uses location information in the route request process. A node includes its location information in each packet sent, allowing other nodes to learn the location of the node and reduce the amount of overhead required to find a route to the node when one is demanded. LAR does not flood route requests to all nodes in the network; instead, route requests are only transmitted by nodes in a *forwarding zone*. Two methods exist for a node to determine this LAR forwarding zone [48].

In the first method, often called the *Box* method [19], the sending node uses a box to define its forwarding zone. In the box method, an intermediate node determines if it is within the forwarding zone by using the location of the source and expected zone of the destination. This expected zone is a circular area surrounding the most recent location known for the destination node. The average known velocity for the

destination node (V_{avg}) and the elapsed time (t_e) are used to calculate the radius (R) for this circular area, where $R = V_{avg} \times t_e$. The forwarding zone is a rectangle with the sender (source or intermediate node) of the route request packet in one corner and the expected zone for the destination in the other corner. A node that receives a route request and determines it is within the forwarding zone forwards the request.

In the second method, often called the *Step* method [19], an intermediate node determines it is within the forwarding zone if the node is closer to the destination than the node that sent the route request. Once again, intermediate nodes that receive a route request and determine they are within the forwarding zone forward the request. We used the *Box* method for all simulations in our study.

4.4 Parameter Selection Method

In this section we document our method of selecting the variables. We document the initial list and the method used to narrow down the list.

4.4.1 Initial List

To determine a set of important variables, we began by listing a large number of potentially important variables from which to choose. Again, these variables can be divided into four categories (see Table 4.2).

Simulator There are many NS-2 default variables set in the `ns-defaults` file. In our study, we focused on the configurable wireless variables in NS-2. These variables are described in Table 4.6. Antenna position is either on top of the node or in front of the node. The bandwidth is the capacity of the network, and was varied between 1 and 100 MB. The propagation models we selected were the NS-2 Friis Free

Table 4.6. Levels for the simulator, scenario, and traffic variables used in our study.

Category	Variable	Description	Levels
Simulator	Antenna position	Where antenna is on node	front(x), **top(z)**
Simulator	Bandwidth (MB)	Bandwidth capacity of the network	1, **2**, 10, 30, 100
Simulator	Propagation model	Model of radio phenomenon	FreeSpace, **TwoRayGround**
Simulator	Interface queue length (packets)	Length of node's interface queue	10, 20, 50, 75, **100**
Simulator	Queue drop front	Use of drop front queue or not	**FALSE**, TRUE
Simulator	Queue limit (packets)	Size of node's queue	10, 20, **50**, 75, 100
Simulator	Queue blocked	If queue blocked while in use	**FALSE**, TRUE
Simulator	Queue unblock on resume	Remove block at end of use	FALSE, TRUE
Scenario	Node speed (m/sec.)	Speed of node movement	3, 5, **10**, 20, 30
Scenario	Node pause time (secs.)	Time stopped at destination	0, 10, **20**, 30, 50
Traffic	Packet Size (bytes)	Size of data packet	64, 128, **512**, 1K, 2K
Traffic	Packet Send Rate (packets/sec.)	Packets sent per second	1, 4, **8**, 10, 20
Traffic	Number of sources (nodes)	Nodes originating packets	10, 15, **20**, 30, 50
Traffic	Source-destination pairs	How the destination of a source is chosen	**Fixed**, Random

Space model and the Two Ray Ground model [82]. The interface queue length is the maximum number of packets allowed in the network interface queue between the physical layer and the MAC layer. This queue length ranged from 10 to 100 packets. The last four simulator variables describe the queue between the link layer and MAC layer. For our study we use an optional drop front queue with a queue limit (between 10 and 100) and optional blocking during use.

Scenario In the scenario category, we varied the node speed and node pause time. The node speed ranged from 3 m/s to 30 m/s and the node pause time ranged from 0 to 50 secs. Their exact values are listed in Table 4.6.

Traffic   We varied packet size, packet send rate, and the number of sources sending packets in the traffic category. See Table 4.6 for the range of each parameters' values. We also varied the method of choosing packet destinations. (We note the sources are randomly selected at the initiation of the simulation.) The method of choosing packet destinations was either fixed or random. With fixed source-destination pairs, each source's target destination was the same throughout the scenario. With random source-destination pairs, the destination was any random node for each packet sent.

Protocol   As mentioned in Section 4.3, we only used the box method for LAR in our study. We did, however, vary two forwarding zone adjustment variables. First, we varied δ (LARDelta) to slightly increase or decrease the size of the forwarding zone. Second, we separately varied δ for the first hop of a route request (LARFHDelta), which is sent by the source node. See Table 4.7 for the range of these two parameters.

There are 15 other configurable variables in our version of LAR. The configurable LAR variables include several timers and optional intermediate node behaviors, and are defined in Table 4.7. We implemented route request timeouts, an optional timeout to purge pending packets, and an optional route persistent timeout to remove saved routes. For the intermediate node behavior we implemented optional promiscuous listening. Using promiscuous listening enables intermediate route repair and response, i.e., an intermediate node can attempt to repair a broken route and reply to route requests even if the node is not the destination. If LARdropIntermediatePacketsIfNoRoute is true, then an intermediate node drops a packet if there is no route to the destination; otherwise the intermediate node queues the packet and attempts to locate a route at a later time. We also implemented an optional area restriction check (LARuseAreaRestrict and LARuseAreaFallback) to purge packets that have exceeded the area threshold (LARareaThreshold). Using area restrictions reduces

Table 4.7. Protocol variables and their levels used in our study.

Category	Variable	Description	Levels
Protocol	LARDelta	Error factor for increasing or decreasing the forwarding zone size.	**0**, 0.5, 1.0, 1.5, 2.0
Protocol	LARFHDelta	First hop error factor increasing or decreasing the forwarding zone.	1, **2**, 3, 4, 5
Protocol	LARrouteRequestTimeout (secs.)	If a route reply is not received in this amount of time a request is flooded.	0.1, 0.25, **0.5**, 0.75, 1
Protocol	LARpurgePending	Whether to drop packets in the pending queue or not.	FALSE, **TRUE**
Protocol	LARdropPendingPacketsAfter (seconds)	Time to drop pending packets after.	10, **30**, 50, 60, 75
Protocol	LARuseRoutePersistence	Whether or not to use route persistence timeouts.	**FALSE**, TRUE
Protocol	LARroutePersistenceTimeout (seconds)	Time limit for a route to remain valid.	0.1, 0.5, 0.9, 1.0, 1.5
Protocol	LARusePromiscuousListening	Whether to use promiscuous listening or not for route replies.	**FALSE**, TRUE
Protocol	LARuseIntermediateRouteRepair	Whether to allow intermediate nodes to fix routes or not.	FALSE, TRUE
Protocol	LARuseIntermediateRouteReply	Whether or not to allow intermediate nodes to reply to a route request.	FALSE, TRUE
Protocol	LARdropIntermediatePacketsIfNoRoute	Whether intermediate nodes should drop packets with no route while intermediate route repair is on.	FALSE, TRUE
Protocol	LARuseAreaRestrict	Whether to use area restriction or not.	**FALSE**, TRUE
Protocol	LARuseAreaFallback	Whether to use area fallback or not.	**FALSE**, TRUE
Protocol	LARareaThreshold (%)	Percentage of the transmission range to restrict forwarding zone.	0.1, 0.3, 0.5, 0.7, 0.9
Protocol	LARuseJitteronSend	Adds jitter to unicast packets sent.	FALSE, **TRUE**
Protocol	LARuseJitteronBroadcast	Adds jitter to broadcast packets sent.	FALSE, **TRUE**
Protocol	LARpendingPacketQueueLength (packets)	Maximum number of packets allowed in the pending queue.	40, 50, **64**, 100, 150

the forwarding zone at an intermediate node, which means more nodes drop packets and reduce congestion. Finally, we implemented options such as sending unicast and broadcast packets with jitter, and setting queue length limits for pending packets. We note these protocol specific variables are preceded with "LAR".

4.4.2 Selection Methodology

The goal of our variable selection procedure was to determine the variables that substantially affect delivery ratio. In our approach, each variable under consideration is assigned a list of possible values, or levels, as shown in Tables 4.6 and 4.7. In a "full factorial" design, a simulation would be executed for each possible combination of levels. Since we have 16 variables with two levels and 15 variables with five levels, this produces a total of $2^{16} \times 5^{15} = 2 \times 10^{15}$ possible combinations. It is, of course, impossible to execute 2×10^{15} simulations. We therefore chose 1000 of these combinations at random and ran 300 independent simulations for each combination, 100 seconds long for each, using the selection scenario in Table 4.5. We then computed the delivery ratio for each of these 300 simulations and averaged them.

Our variable selection procedure made use of stepwise regression. In the stepwise regression procedure, the first step is to choose the candidate variable that produces the best-fitting linear regression model with the outcome variable, which is delivery ratio herein. In each succeeding step, the candidate variable that most improves the fit of the linear model is added to the model, if the improvement exceeds a predetermined threshold. Then any variable in the model that can be dropped without reducing the fit by more than a predetermined amount is dropped. This process is iterated until no variable that is not in the model will improve the fit by an amount exceeding the threshold, and no variable that is in the model can be dropped without reducing the

94

fit by more than the predetermined amount. See [62] for a more detailed description.

We used the 1000 combinations of levels and their corresponding delivery ratios to construct our list of important variables. Our procedure consisted of three stages. In the first stage, we used stepwise regression to reduce the list of 31 candidate variables to a smaller list containing the more important variables. This step created a list of nine possibly important variables, namely propagation model, interface queue length, node speed, node pause time, source-destination pairs, number of sources, packet send rate, packet size, and LARareaThreshold.

For the second stage, we added to this list of nine possibly important variables all two-way interactions (products of pairs of the nine variables) and squares of those variables with more than two levels. (Variables with two levels are coded 0 and 1, and thus are equal to their squares.) This process gave us a list of 52 candidate variables (9 univariates, 36 pairs, and 7 squares). We then used stepwise regression on this list of 52 candidate variables, which produced a list of 25 variables.

The third and final stage consisted of executing best-subsets regression (which is sometimes called all-subsets regression) on this list of 25 variables. As its name implies, best-subsets regression consists of fitting every possible linear model containing one variable, then every model containing two variables, and so on. When this process ends, one can identify the model with any given number of variables that fits best, using a criterion such as the coefficient of determination (R^2). Our final list of variables contains the variables in the best-fitting subset of four. This list is presented in Table 4.8.

We note that there are many other methods that could have been used to select a final list of important variables. Our purpose is not to propose a method of variable selection; instead, it is to construct a list of variables that have a substantial impact

Table 4.8. Four variables selected for their impact on delivery ratio.

Variables
Number of sources (NSrc)
Source-destination pairs (SD)
Packet send rate (PSR)
Propagation model (PR)

on delivery ratio in a typical MANET simulation study. In the next section, we show that these four variables do indeed have a substantial impact on delivery ratio and, in particular, we show that their impact is considerably greater than that of more frequently considered variables.

4.5 Findings

In this section we document our findings using our selection method. We document our verification study and the significants of the various variables used in our study.

4.5.1 Verifying the Importance of the Selected Variables

To verify that the variables we selected do in fact have a substantial effect on delivery ratio, we performed a validation study. In our validation study, we considered validation scenario I and validation scenario II from Table 4.5. We note these two scenarios have different derived parameters than the selection scenario, which was used in our selection procedure.

We began by studying the effect of varying the number of sources on the delivery ratio. We conducted five simulations, and only varied the number of sources in each. Thus, any systematic differences in delivery ratio were only due to differences in

Table 4.9. Variables and their values used in our validation study.

Variables	Values
Number of sources (NSrc)	10, 15, **20**, 30, 50 nodes
Source-destination pairs (SD)	**Fixed**, Random
Packet send rate (PSR)	1, 4, **8**, 10, 20 packets/sec.
Propagation model (PR)	FreeSpace, **TwoRayGround**

the number of sources. We then performed similar sets of simulations for the other selected variables, i.e., we varied the value of one selected variable while holding the other variables constant. Table 4.9 presents a subset of Table 4.6, and shows the four variables we evaluated in this validation study. Each bold value is the value held constant when the corresponding variable was not under study. We used validation scenario I and validation scenario II (which are described in Table 4.5), the bold values in Tables 4.6 and 4.7 for all the other variables, and the defined constants (which are given in Table 4.3) in this study.

Figure 4.1 shows the delivery ratio for each value of the four selected variables under validation scenario I. For each variable, the bars (read from left to right) show the delivery ratio for the values of that variable in the order they appear in Table 4.9. For example, for source-destination pairs, the delivery ratio was 57.1% when source-destination pairs were fixed and 9.6% when source-destination pairs were random.

Figure 4.1 illustrates that each of the four variables has a substantial effect on delivery ratio. For example, decreasing the number of sources from 20 to 15 sources almost doubles the delivery ratio. For source-destination pairs, the delivery ratio more than quadruples if random source-destination pairs are replaced by fixed ones. The effect of varying the packet send rate from 8 to 10 packets per second is to decrease the delivery ratio nearly in half. The effect of changing the propagation model from
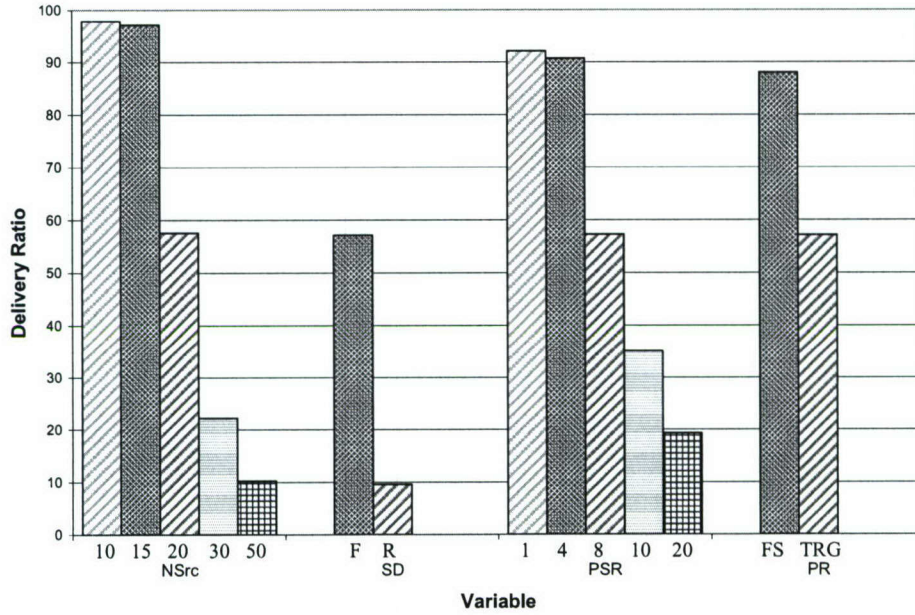
Figure 4.1. Delivery ratio versus the selected variables for validation scenario I. Specifications for the scenario are given in Table 4.5.

FreeSpace to *TwoRayGround* is to decrease the delivery ratio from approximately 88% to approximately 57%.

Similar results hold for validation scenario II, even though scenario II involves a sparser network than validation scenario I. Figure 4.2 shows the delivery ratio varies from 7% to 90.5% over the range of numbers of sources, and triples when source-destination pairs are fixed instead of random. Over the range of packet send rates, the delivery ratio decreases from 73% to 10%. Lastly, changing the propagation model from *FreeSpace* to *TwoRayGround* decreases the delivery ratio by approximately one-fourth.

The reasons for the substantial impact of these selected variables are reasonably clear, at least in hindsight. Increasing the number of source nodes (NSrc) increases

the number of packets in the network, because the packet send rate is held constant. A source in a random source-destination pair is less likely to have a valid route to its destination than a source in a fixed source-destination pair; thus, the number of control packets transmitted in a random source-destination pair is higher. Increasing the packet send rate increases the number of packets in the network, because the number of sources is fixed. In all three of these cases, increasing the number of packets in the network increases the number of collisions and, in turn, reduces delivery ratio. For the propagation model, the Two Ray Ground model used the Free Space model (factor of d^2) for nodes close to the source (less than the cross-over distance). For nodes farther from the source (greater than the cross-over distance), it uses a two ray reflection model with a factor of d^4, lowering the probability of a packet being received at the node's neighbor. The cross-over distance for the Two Ray Ground model to switch from d^2 to d^4 in our validation scenarios (with defined constants in Table 4.3) was 38.6 m. As a result, with a 100 m transmission range, this switch often happened.

4.5.2 Traditional Variables

We have found that several variables traditionally considered, such as node speed, node pause time, and packet size, have less effect on delivery ratio. To illustrate, we performed simulation experiments similar to the experiments we used to validate our selected variables. In this case, we vary the values of node speed, node pause time, and packet size as the other variables are held constant. Table 4.10 presents a subset of Table 4.6, and shows the three variables used in this study. Each bold value is the value held constant when the corresponding variable was not under study. We used validation scenario I and validation scenario II (which are described in Table 4.5), the
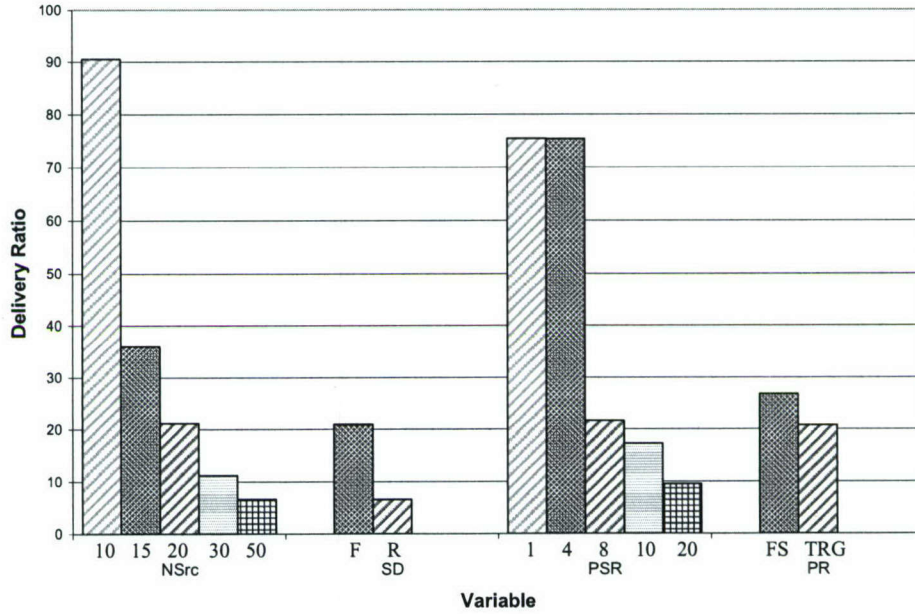
Figure 4.2. Delivery ratio versus the selected variables for validation scenario II. Specifications for the scenario are given in Table 4.5.

bold values in Tables 4.6 and 4.7 for all the other variables, and the defined constants (which are given in Table 4.3) in this study.

For each of these three traditional variables (i.e., packet size, node speed, and node pause time), we considered five values that cover a reasonably wide range. To assess the impact of each variable on delivery ratio, we conducted five simulation experiments and assigned a different value of the variable in each experiment. The values of the other variables were held constant; thus, all systematic differences in delivery ratio were due to differences in the value of the variable under study.

Figure 4.3 shows the delivery ratio for each value of the three traditional variables under validation scenario I. For each variable, the bars (read from left to right) show the delivery ratio for the values of that variable in the order they appear in Table 4.10.

Table 4.10. Variables and their values used in our traditional variable study.

Variable	Values
Node Speed (NS)	3, 5, **10**, 20, 30 m/sec.
Node Pause Time (NPT)	0, 10, **20**, 30, 50 secs.
Packet Size (PS)	64, 128, **512**, 1024, 2048 bytes

Figure 4.3. Delivery ratio versus node speed, node pause time, and packet size for validation scenario I. Specifications for the scenario are given in Table 4.5.

As shown, none of the three values has much effect on delivery ratio. The delivery ratio varies by less than 10% over the range of packet sizes and over the range of pause times, and by less than 20% over the range of node speeds.

Similar results hold for validation scenario II, and are shown in Figure 4.4. The delivery ratios for validation scenario II are smaller than the delivery ratios in vali-
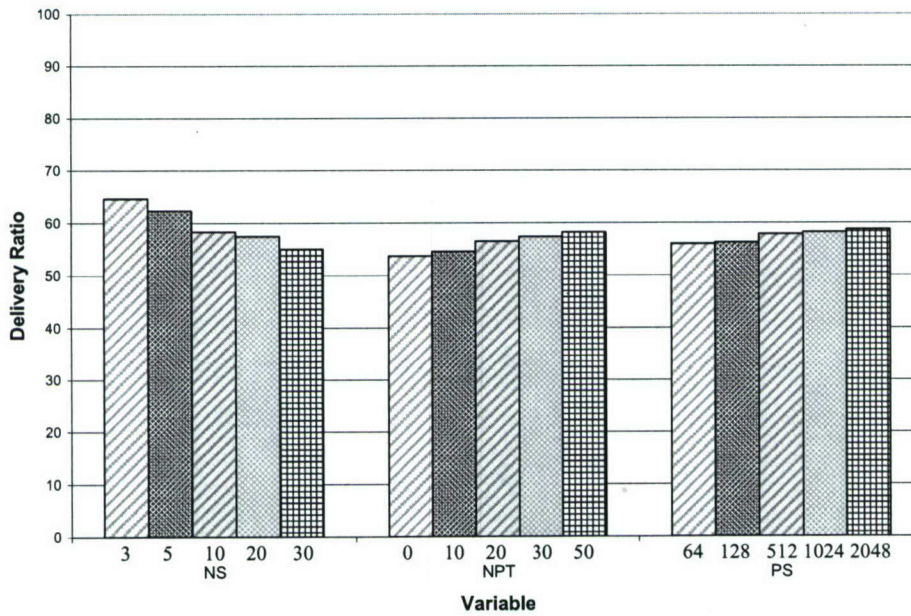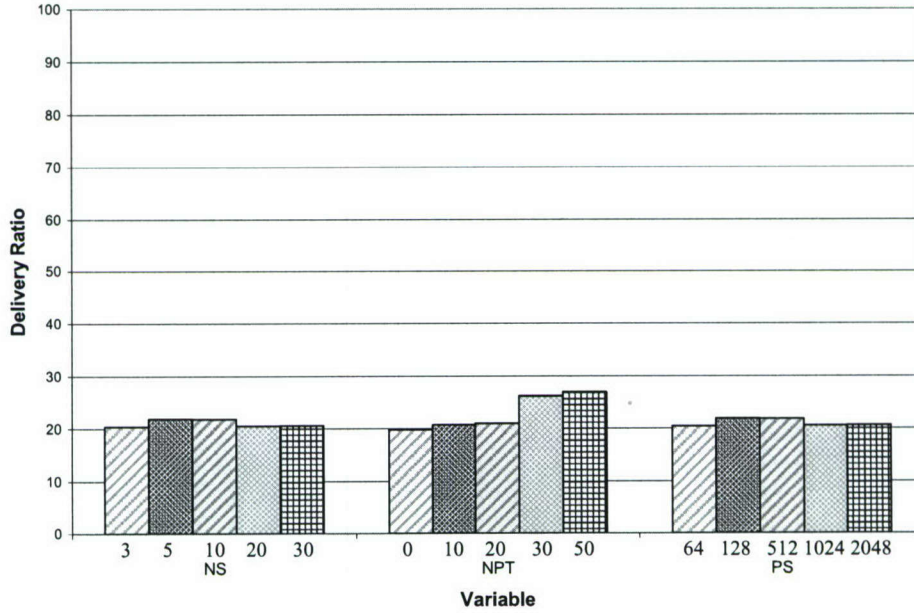
Figure 4.4. Delivery ratio versus node speed, node pause time, and packet size for validation scenario II. Specifications for the scenario are given in Table 4.5.

dation scenario I because the network is sparser, but the values of the three traditional variables continue to have little effect. Comparing Figure 4.3 with Figure 4.1, and Figure 4.4 with Figure 4.2 shows that our selected variables (i.e., number of sources, source-destination pairs, packet send rate, and propagation model) have a much greater impact on delivery ratio than three traditional variables (i.e., node speed, node pause time, and packet size).

4.5.3 Source-destination Pairs

We were particularly interested in our results for whether random or fixed source-destination pairs are used, since this variable is typically not evaluated in the litera-

ture. That is, in our MobiHoc survey (see Chapter 2), 98 of the published MobiHoc papers conducted simulations of end-to-end packet traffic. Of the 98 papers, 21 papers (21.43%) evaluated random source-destination pairs and 39 papers (39.80%) evaluated fixed source-destination pairs. The remaining 38 papers (38.78%) do not mention how the source-destination pairs were chosen. Thus, we conclude that researchers evaluate random source-destination pairs or fixed source-destination pairs, but not both.

In our selection procedure, the destination for each packet from a given source was either fixed ahead of time or chosen at random from among all nodes. In this study, we included two intermediate cases as well. In one case, each packet from a given source chooses its destination at random from among two previously specified nodes; in the other case, the choice is made from among five previously specified nodes. This reflects recent trends, which show that it has become more realistic for researchers to use an intermediate setting between fixed source-destination pairs and random source-destination pairs [84]. For example, a user often sends traffic among a small number of servers or peers.

To further evaluate the effect of the degree of randomness on destination selection, we performed simulations with several additional scenarios. These scenarios are based on validation scenario I in Table 4.5, with a variation of scenario area size. The four areas were square areas with widths of 120 m, 170 m, 220 m, and 270 m. Changing the area size highlights the difference in performance of the different source-destination pairs. We generated five iterations of 100 second simulations and averaged the results for delivery ratio. We used the constants from Table 4.3 and the bold values in Tables 4.6 and 4.7. The results of the simulations are presented in Figure 4.5.

The effect of varying the degree of randomness in destination selection remains strong in the scenarios we considered. The fixed source-destination pairs have the highest delivery ratio. However, the delivery ratio drops as the network becomes more sparse (i.e., larger simulation areas). The delivery ratio also drops as the number of possible destinations increases. In addition, the random pairs had poor delivery ratio in all cases. In summary, varying the type of source-destination pairs is critical in the evaluation of a MANET routing protocol. If a researcher varies the number of possible destinations and the method in which a source selects a destination, the results of the simulation study may be significantly different.

4.6 Future Work and Conclusions

In this section we document our ideas for future work. We also list our conclusions from our work.

4.6.1 Future Work

This study was based on the Location Aided Routing protocol (LAR) [48]. We plan to conduct similar studies with other protocols such as the AODV and DSR protocols. Studying these two unicast protocols will help determine the extent to which the variables that have the greatest impact on protocol performance are the same across protocols, and the extent to which they are protocol-dependent.

4.6.2 Conclusions

The design of a MANET simulation-based study involves setting values for a large number of variables. To conduct a rigorous and credible simulation study of a MANET routing protocol, an investigator must know which variables are likely to
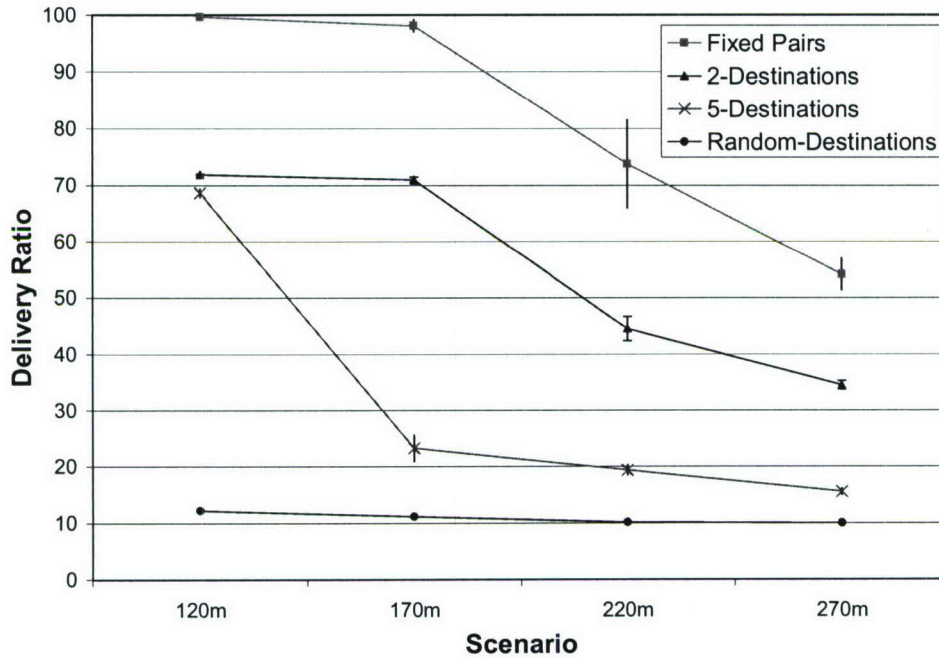
Figure 4.5. Source-destination pairs versus scenario I from Table 4.5 with widths as shown. 95% confidence intervals are shown.

have the greatest impact on protocol performance. In our evaluation of the Location Aided Routing (LAR) protocol in a fairly typical setting, we have reached the following conclusions:

Conclusion #1: A statistical approach can be used to screen a large number of variables and to identify several that are particularly important.

Conclusion #2: Some often-considered variables such as packet size, node speed, and node pause time did not have significant impact on delivery ratio in our study.

Conclusion #3: Some less frequently considered variables such as the number

of sources, random versus fixed source-destination pairs, packet send rate, and the propagation model had a substantial impact on delivery ratio in our study.

Conclusion #4: The selection and pairing of source nodes to destination nodes for packets is an important process that impacted the performance of a MANET routing protocol in our study. Different numbers and types of pairings need to be evaluated and documented in a credible MANET simulation-based study.

Conclusion #5: To improve the credibility of simulation studies, investigators should determine which variables are likely to have a strong impact on protocol performance (i.e., which variables are significant).

Conclusion #6: To show true protocol performance, MANET routing protocols should be systematically tested across a sufficiently wide range of values for each variable found to be significant.

Chapter 5

A VISUALIZATION AND ANALYSIS TOOL FOR WIRELESS SIMULATIONS: INSPECT

Network simulators allow researchers to analyze the behavior of wireless devices at every level. As a result, these simulations are capable of producing very large amounts of data. The simulation community has made available many types of scripts (e.g., tracegraph [54]) to parse and analyze this output data, but visualization of the data is needed to further aid understanding of the output. A good visualization package is important, because the human visual system is unrivaled in pattern recognition and offers the ability to process large amounts of data quickly and clearly [3]. Visualization adds to the understanding gained via statistical analysis. As we show in this chapter, certain erroneous network behaviors could go undetected without visualizations.

There have been many emails on the NS-users mailing list asking for visualization or video support for wireless networks in NS-2 (see [2] and [45], for examples). The increasing complexity of node and protocol behavior is driving the need for a visualization tool.

In this chapter we present our *interactive NS-2 protocol and environment confirmation tool* (iNSpect)[1]. The iNSpect program was developed to allow direct visualization and analysis of NS-2 wireless simulations. Because it can animate a mobile

---

[1]As of June 2006, iNSpect has been requested from and shared with 205 researchers at 165 research labs/universities in 39 countries.

ad hoc network without running NS-2 itself (by reading the mobility file, which is an input to NS-2) and because it can post-process successful NS-2 simulations (by reading the trace file, which is an output from NS-2), iNSpect is an agile tool that can be utilized with minimal overhead. In addition to NS-2, iNSpect can also be used with any simulator or testbed environment which produces output in the iNSpect expected file format (see Section 5.1.3).

We developed iNSpect to work with NS-2 input and output files directly, because NS-2 is the most popular simulator used in the MANET community. However, as Figure 2.1 shows, several other simulators are used for wireless network simulations. In addition, ad hoc network testbeds (e.g., wireless sensor network testbeds) are becoming quite common. Thus, we have developed an iNSpect input option that allows iNSpect to read a specific iNSpect formatted trace file. The *vizTrace* file format allows any simulator or testbed that can generate custom output to use iNSpect.

5.1 iNSpect Overview

The *interactive NS-2 protocol and environment confirmation tool* (iNSpect) is a C++ OpenGL-based [105] visualization tool that allows analysis of simulated wireless networks. The iNSpect program uses a GTK+ [29] graphical user interface (GUI) for direct scene manipulation. The iNSpect program is multi-platform and will execute on Linux, MacOS X, Windows, and Cygwin.

5.1.1 Related Work

As mentioned, a visualization tool is needed to understand the large amount of data produced during network simulations. For these reasons, the Network Animator (NAM) was designed to provide a graphical user interface for the creation of

wired network topologies in NS-2 [57]. It has an extensive environment for wired network development as well as trace file playback. Playback for the wired environment includes the display of links and packet flows.

NAM has not been extended under the Monarch Project [82] to visualize wireless networking at the same level as wired networks. By adding Cartesian support to NAM, it can playback NS-2 generated trace files for wireless simulations. However, playback in the animation screen is limited to node movements and emitting a circular pattern to represent the node's transmission signal. The concepts of links and queues are not supported in the NAM wireless animations, because links and queues represent a permanent relationship between two nodes in NAM which do not exist in mobile wireless networks. In other words, NAM does not show the wireless links. Packet and agent monitors in wireless networks provide the same information as in the wired networks. In summary, NAM is not a visualization tool designed for wireless analysis and validation.

With the increased demand for NS-2 simulations for wireless networks, a robust visualization tool is needed for wireless networks. There was an effort in the late 1990s to develop Ad-hockey [83], a visualization tool for NS-2 wireless simulations, but that effort has not continued. The last supported update of Ad-hockey was 1999. Thus, Ad-hockey does not work with the Tool Command Language (Tcl) versions of NS-2 currently used (since version NS-2.1b7a). The current version of NS-2 is NS-2.29 [33].

The fact that the Monarch Project envisioned Ad-hockey is further justification for the need of a visualization tool that can be used with NS-2. In this chapter we discuss work we have done on building a visualization tool for NS-2 that can be used in current research. The tool is not a replacement for NAM, but a complimenting tool for the wireless community. Because Ad-hockey has not been maintained, *the*

interactive NS-2 protocol and environment confirmation tool (iNSpect) is a useful resource for today's wireless simulation researchers.

5.1.2   iNSpect Visualization

The iNSpect program produces a 3-dimensional visual display of the nodes in a wireless scenario based on Cartesian (x,y,z) coordinates used by NS-2. Unlike NAM, which does not show the transmissions in wireless networks, the iNSpect program shows the wireless routes and the success or failure of wireless packet transmissions. The transmissions are displayed with route lines and color coded nodes. When a node is transmitting to another node, a line is drawn between the two nodes. The line represents the attempt to transmit between the two nodes, similar to the link object in the NAM wired scenarios. The events associated with a node are mapped to colors by a configuration file. A customizable color scheme aids in analysis.

Figure 5.1 illustrates the basic use of iNSpect, with the default color scheme (i.e., blue sending nodes, yellow forwarding nodes, green destination nodes, and red unsuccessful transmissions). In Figure 5.1, node 19 is attempting to send a packet to node 5 via intermediate node 12, but node 12 does not receive the packet. In Figure 5.2, node 19 successfully sends the packet to node 21, which forwards the packet to node 22, which forwards the packet to the destination node 5. The persistence of the lines and status of the packet activity is configurable, allowing for individual route analysis.

In summary, with the default color scheme, blue and yellow nodes along a path leading to a green node shows a successful transmission to a destination node; blue and yellow nodes along a path leading to a red node shows failure of the packet to reach the destination node and at which hop the packet failed. A graphical representation
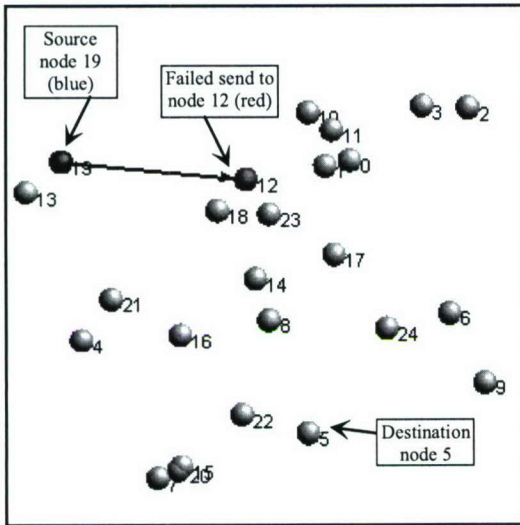
Figure 5.1. iNSpect showing a failed attempt to send a packet from node 19 (blue) to node 5, via intermediate node 12 (red).
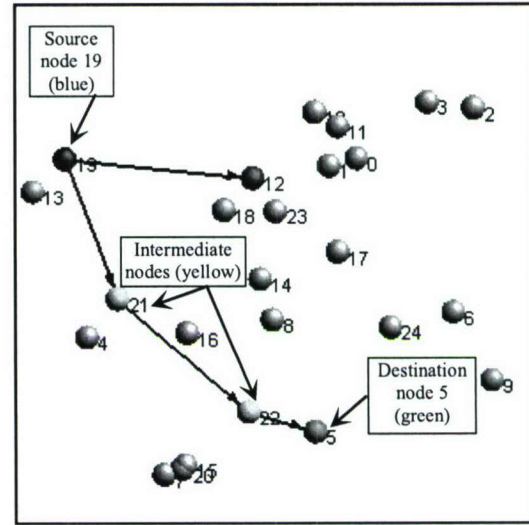
Figure 5.2. iNSpect showing node 19's (blue) successful attempt to send a packet to node 5 (green) via intermediate nodes 21 and 22 (yellow).

Note: Simulation area is $300\,\text{m} \times 300\,\text{m}$ with 25 nodes.

of network activity gives the researcher more clues about the individual success and failure of packets than the overall delivery ratio printed at the end of a scenario.

5.1.3   iNSpect Input

The iNSpect program uses an event builder object to schedule node movement and packet traffic of various types and formats. Because the event builder can translate different inputs, the iNSpect program can animate a stand-alone mobility file (an NS-2 input file), an NS-2 trace file (an NS-2 output file), and a mobility file with a specific iNSpect formatted trace file (a *vizTrace* file).

Stand-alone Mobility File   Our iNSpect program can be used directly with a mobility file generated by an external mobility model. Unlike NAM, there is no requirement to first generate a trace file from NS-2. Mobility file analysis outside of NS-2 is extremely useful for mobility model development and mobility model output verification, eliminating the overhead of additional lengthy simulations in NS-2 to generate a trace file.

NS-2 Trace File   For protocol evaluation, the NS-2 generated wireless trace file can be used by the iNSpect event builder to schedule packet transmissions and to process node movements without a mobility file. An NS-2 trace file created in the Tool Command Language (Tcl) driver file with medium access control (MAC) layer trace (`macTrace`) and movement trace (`movementTrace`) turned on contains all of the node packet and movement activity. The iNSpect program can process this stand-alone NS-2 trace file without using an externally generated mobility file.

Mobility File and vizTrace File   The iNSpect trace file, called a *vizTrace* file, allows iNSpect to be used with any simulator or testbed that produces a trace file in the iNSpect expected format. To build a *vizTrace* file, a researcher records each event in the format of Table 5.1. (The options for an entry are listed below the name of each field.) The six fields of each event are included in each line of the *vizTrace* file: node ID, event time, event title, other node ID, status of event, and packet ID. The user defined *status* of event field enables customized color schemes and statistics (see Section 5.2.3).

Table 5.1 also shows four example events in a *vizTrace* file. The example events show node 2 sending packet number 45 to node 12 via node 6. In the user defined *status* field, node 2 is the *source*, node 6 is the *forwarding* node, and node 12 is the

Table 5.1. Format and example events for iNSpect's *vizTrace* file.

Field and Description

Node	Time	Event	Other Node	Status	Packet
ID	Seconds elapsed	*sending to* or *received from*	*ID* or *-1*	Custom string	ID

Sample data

2	2.34628	sending to	6	source	45
6	2.35677	received from	2	forwarding	45
6	2.35999	sending to	12	forwarding	45
12	2.36125	received from	6	destination	45

destination for packet number 45. These user defined terms can then be associated with specific colors in iNSpect. In summary, iNSpect can process any simulator or testbed output file if the output file is in the *vizTrace* format.

5.2 iNSpect Uses and Results

In this section we highlight our successes with iNSpect as a research tool. We have used iNSpect to develop, analyze and verify mobility models, to find a problem in NS-2.27, to verify protocol development, and to analyze protocol performance.

5.2.1 Topology Analysis and Validation

Visualizing nodes moving can help verify a mobility model. With NS-2, the complete analysis of a mobility file can be done only by running a simulation through NS-2 to produce a NAM trace file. NAM would then be used to visualize the NAM trace file, and ultimately the node movement.

Unlike NAM, iNSpect can process an NS-2 formatted mobility file directly. The iNSpect engine calculates the node movements directly from the mobility file. This ca-
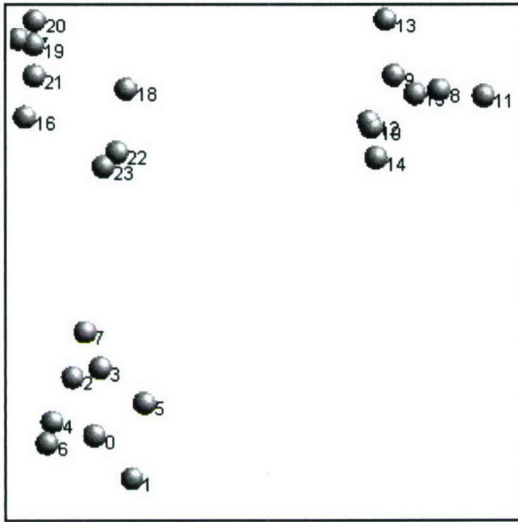
114

Figure 5.3. iNSpect displaying the RPGM model with 60 m reference point separation.
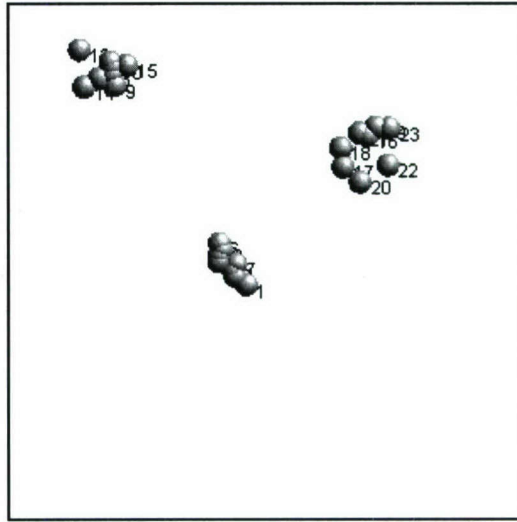
Figure 5.4. iNSpect displaying the RPGM model with 15 m reference point separation.

Note: Simulation area is 300 m x 300 m with 3 groups of 8 nodes each, 20 m/s node speed, and 0 seconds of pause time.

pability streamlines the development of mobility files generated by individual topologies or mobility files generated from an automated script or mobility model, because the nodes can be displayed and animated outside of NS-2. Thus, iNSpect can produce visual verification for a mobility file instantly. The direct processing of the mobility file allows a developer to complete many iterations quickly.

Figures 5.3 and 5.4 illustrate an example verification of the Reference Point Group Mobility (RPGM) Model [18, 78]. To generate a mobility file from the RPGM model, the user must determine numerous parameters including reference point separation distances and individual node separations from the reference point [18]. Figures 5.3 and 5.4 illustrate how a user can analyze these two parameters in iNSpect. Fig-

ures 5.3 and 5.4 show two mobility files with the same dimensions, speed, pause time, number of nodes, and number of groups, but different levels of node separation from the reference point. Immediately the effect of the parameter is seen. The iNSpect program is the only way to see the effect of these parameters from a mobility file directly.

Furthermore, because iNSpect allows the immediate verification of files produced by a mobility model, iNSpect can be used to develop new mobility models. For example, we used iNSpect during the development of a new congestion-based mobility model. In this new model, a node will slow down if its number of neighbors exceeds a threshold. We used iNSpect in two ways. First, iNSpect gave us an instant look at the model results and allowed us to visually see the nodes slow down in congested areas. Second, iNSpect enabled us to discover a movement problem with the implementation of the model. With iNSpect, our implementation problem was debugged and quickly fixed. Without iNSpect the problem might have gone unnoticed.

5.2.2 Simulation Model Analysis and Validation

As stated at the beginning of the NS-2 documentation [33] *"users of NS-2 are responsible for verifying for themselves that their simulations are not invalidated by bugs."* The question is how one ensures a simulation is correct? While there is no way to guarantee correctness, iNSpect can help. The iNSpect program can provide insight into the simulation process that summary statistics cannot provide.

As an example, when we upgraded to NS-2 version 2.27, we noticed a significant drop in the performance of our simulations (e.g., delivery ratio), similar to several accounts on the NS-mailing list (e.g., [20]). Using iNSpect, we discovered an error in the simulator. Specifically, NS-2.27 does not update the position of a node unless

there is an event for that node. (The error was concurrently located by the author of [99].) The NS-2.27 error is shown in Figure 5.5. In Figure 5.5, the simulation area is 600 m x 600 m. Each node's transmission range is 100 m. As shown, node 0 successfully transmits a packet outside the 100 m range (e.g., node 0 to node 26 in Figure 5.5). The actual distance between node 0 and node 26 is 453 meters, which is well over the 100 m transmission range. The blue star in Figure 5.5 shows NS-2.27's incorrect view of node 0's location, explaining the reason NS-2 allowed the transmission to be successful.

In summary, iNSpect quickly illustrated the inconsistencies of the simulation output under NS-2.27. We also note that NAM could not have shown this problem for two reasons. First, NAM's output is based on the NS-2 model; therefore, the nodes shown in NAM would be in the locations seen by NS-2 (e.g., the incorrect location of node 0 in Figure 5.5). Second, NAM does not show the links and packet flows. Thus, even if the nodes were in their correct locations, one would not have seen the transmission over 100 m that succeeded.

5.2.3 Simulation Results Analysis

An entire simulation (node movement and wireless network traffic) can be animated with iNSpect. The iNSpect display shows each transmission, with lines between nodes for transmission attempts and node colors which represent the sending nodes, nodes that receive a transmission successfully, and nodes that do not receive a transmission successfully. The iNSpect display shows the virtual link in a transmission, instead of the transmission ring. The ring, although representative of an omni-directional wireless signal without obstacles, does not help a researcher trace the route of a packet. The iNSpect animation allows quick analysis of packet routes
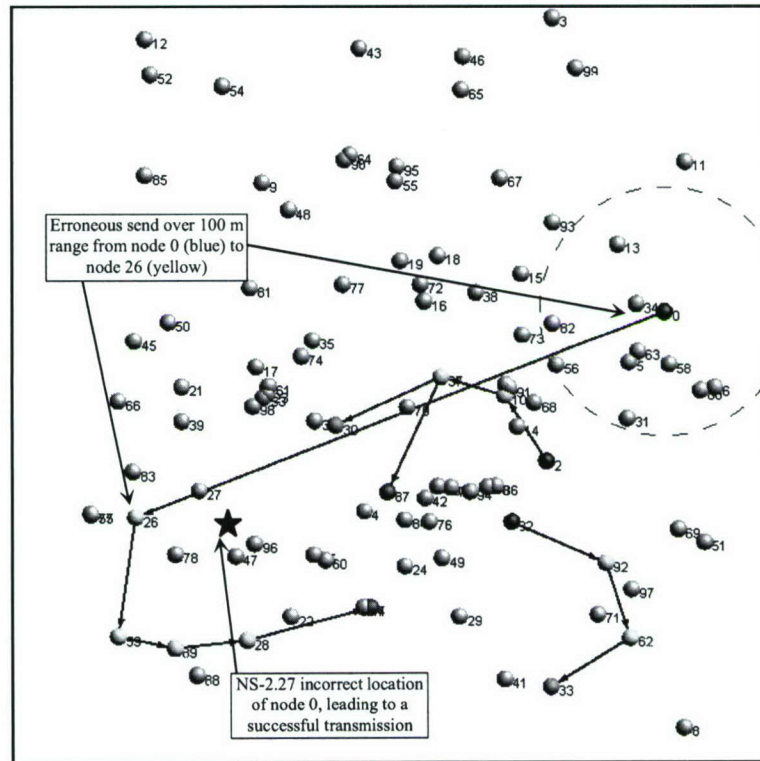
Figure 5.5. iNSpect showing an NS-2.27 model error. Node 0's transmission exceeds the 100 m transmission range, because NS-2's incorrect view of node 0's location places node 0 in range of node 26 (blue star). The simulation area is 600 m x 600 m with 100 nodes.

and node activities. An animation of the results aids understanding of summary performance statistics such as delivery ratio, end-to-end delay, and overhead.

Path Analysis By knowing the path a packet takes from source to destination, we can learn more about the behavior of a protocol. Figure 5.6 shows a snapshot of a Location Aided Routing (LAR) simulation [48]. LAR routing uses knowledge of the destination node's location to build routes for a packet transmission. We can use
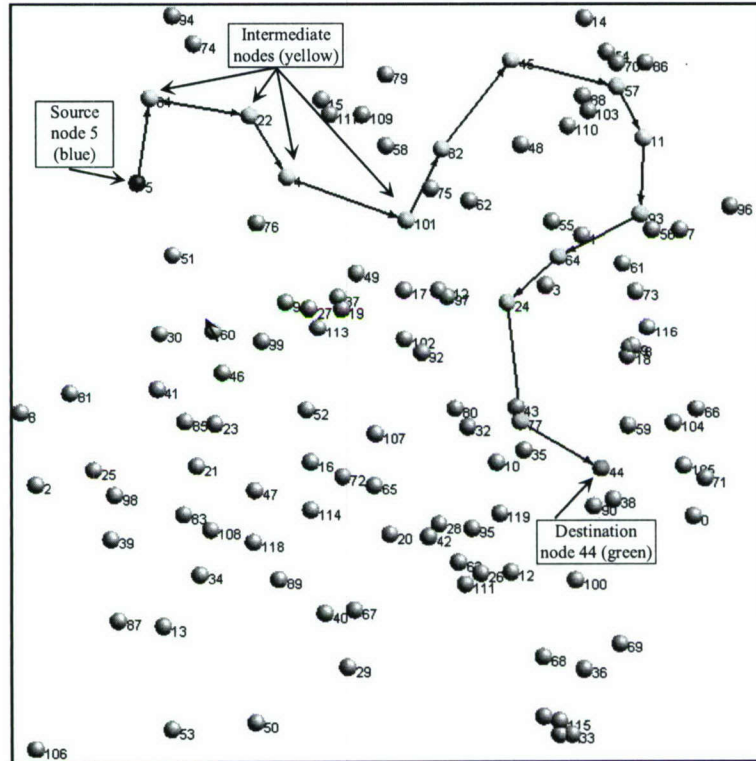
Figure 5.6. iNSpect showing a Location Aided Routing route selection for a transmission from node 5 to node 44 in a $600\,\text{m} \times 600\,\text{m}$ simulation area with a $100\,\text{m}$ transmission range and 120 nodes.

iNSpect to evaluate the number of hops a given successful transmission takes, and whether a protocol can be improved to reduce the number of hops. For example, in Figure 5.6 we see a successful transmission from node 5 to node 44. The path from node 5 goes through nodes 84, 22, 4, 101, 82, 45, 57, 11, 93, 64, 24, 77, and 44 (a total of 13 hops). From the iNSpect program we have the time this event occurs and we see other more direct paths such as the one from node 101 to nodes 112 or 97, to 24. With this knowledge we can look at which routes were in the cache for node 5 and see why the protocol did not discover a shorter route. Individual analysis
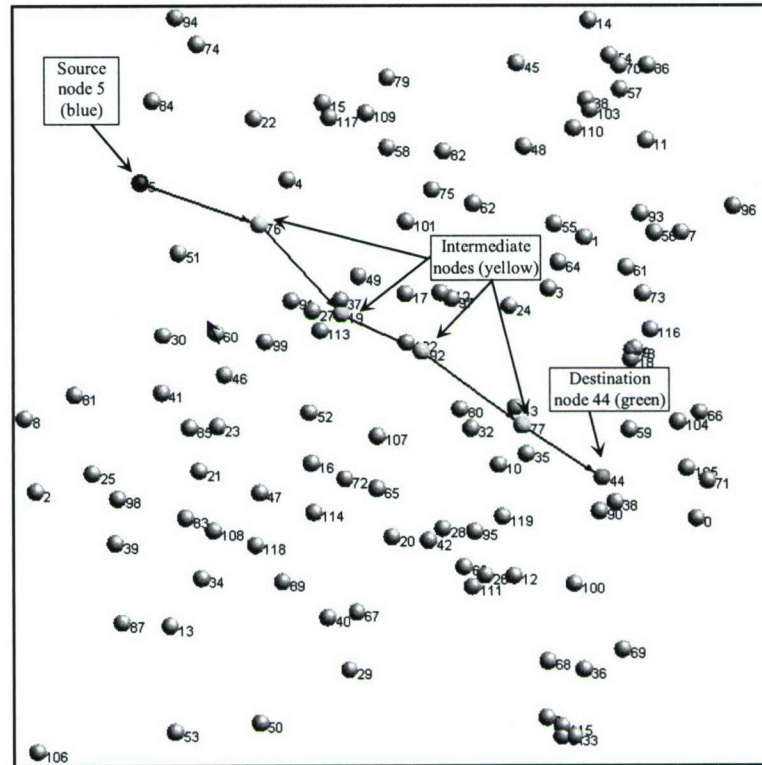
Figure 5.7. iNSpect showing our projection method [24] for the Location Aided Routing protocol. The same scenario as Figure 5.6 is used.

such as this would be impossible with only performance statistics and no iNSpect visualization. With the help of iNSpect, we developed improvements to LAR that utilize the location information disseminated to find more direct routes [24].

Using our *projection method*, a node, A, sets its assessment delay (the time it waits before rebroadcasting a route request packet) proportional to the length of the projection of the vector from the sending node, S, to A ($\overrightarrow{SA}$) onto the vector from the source to the destination node, D ($\overrightarrow{SD}$). The longer the projection, the more direct the route is, and the shorter A will set its assessment delay. Figure 5.7 shows

120
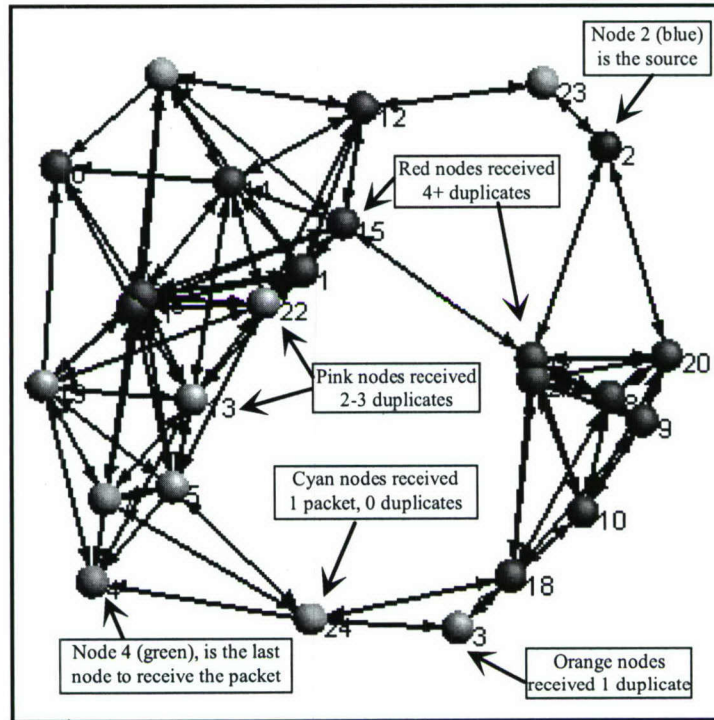
Figure 5.8. iNSpect displaying a snapshot of a simple flooding protocol, at the time when the last node (node 4, green) successfully received the packet. Node 2 (blue) is the source of the broadcast packet.

an example route discovered in our LAR projection method. Our LAR improvement found a route with eight fewer hops, compared to the route shown in Figure 5.6, for the same transmission (from node 5 to 76, 19, 92, 77, to 44). The iNSpect program visualizes the reduced number of hops, and that the resulting path is the shortest between node 5 and node 44. Visualizing the routes with our LAR projection method is a valuable step in designing and analyzing our improvements to the LAR protocol.
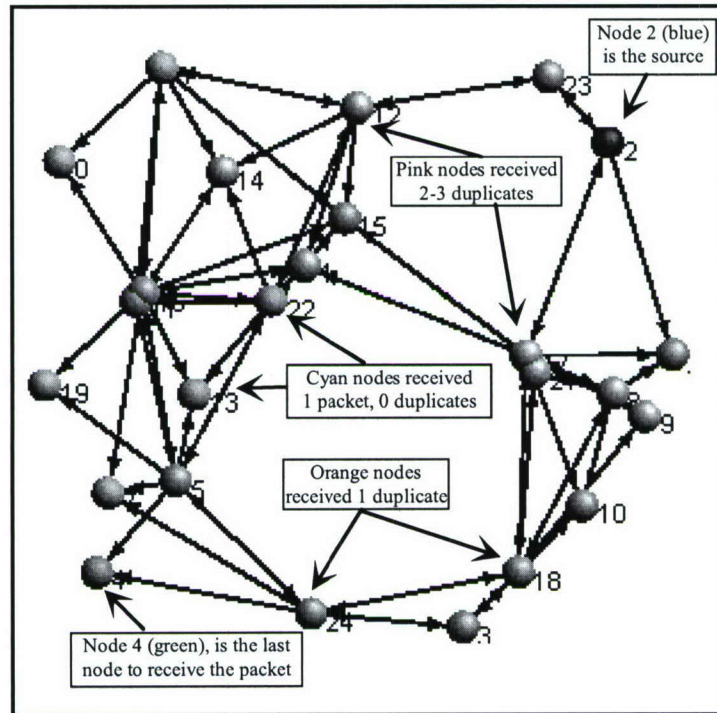
Figure 5.9. iNSpect displaying a snapshot of the Probabilistic Broadcast protocol, for the same scenario as Figure 5.8, at the time when the last node (node 4, green) successfully received the packet. Node 2 (blue) is the source of the broadcast packet.

Node Activity Analysis By knowing the nodes' activities, we can use iNSpect's custom color capability to visualize and monitor the network activity at the nodes. For example, in a simple flooding protocol, packets are broadcast to all of a node's neighbors, all of which rebroadcast the packet [103]. There are, of course, several improvements to simple flooding; see [103] for a comparison of broadcasting protocols. One variation of an improved flooding protocol is the Probabilistic Broadcasting protocol [69]. In the Probabilistic Broadcasting protocol, a node rebroadcasts with probability (P), in order to reduce duplication and collisions. Figures 5.8 and 5.9

Table 5.2. Status field values for Figures 5.8 and 5.9.

Value (color)	Description
source (blue)	Initiated the packet transmission
final (green)	Last node to receive the packet
received (cyan)	Received the packet once
duplicate (orange)	Received the same packet twice
2-duplicates (pink)	Received the same packet at least three times
4-duplicates (red)	Received the same packet at least five times

show a custom color scheme comparing simple flooding and the Probabilistic Broadcasting protocol for the same scenario. The node colors in these two figures were defined in the user defined *status* fields, as shown in Table 5.2.

In Figures 5.8 and 5.9, we see the transmit lines and color coded nodes for a simple flooding and Probabilistic Broadcast ($P = 0.38$) transmission from node 2 at the instant the final node (4, green) receives the packet. The iNSpect display immediately shows the difference between the simple flood and the Probabilistic Broadcast protocols (e.g., no red nodes exist in Figure 5.9). As shown, the Probabilistic Broadcast protocol, compared to simple flooding, sends the packet to all nodes with fewer transmissions (i.e., less lines exist between the nodes) and fewer duplicate packets (i.e., more cyan nodes and no red nodes exist). Visualizing the node activity for simple flooding and the Probabilistic Broadcast protocol is a valuable step in understanding the performance of these two protocols.

Because iNSpect's status field is so flexible, other visualizations of the same data are possible. For example, iNSpect can color a node based on whether it rebroadcast the packet rather than how many packets it received. Visualizing which node rebroadcasts a packet is useful when designing a protocol that approximates the minimum connected dominating set. When designing a protocol that uses neighbor

knowledge, iNSpect can color a node based on the number of one-hop neighbors or two-hop neighbors. Because iNSpect allows the researcher to define the visualization, the most useful data can be highlighted.

Performance Analysis With the suite of calculations that iNSpect provides, researchers now have more analytical techniques available. For example, we executed a protocol with the same scenario multiple times and each time it produced significantly different delivery ratios. Using iNSpect's connectivity graph and partition check tools, which are described in Section 5.3.1, we found the scenario had a large group of nodes isolated from the rest. As a result, when the randomly selected source and destination nodes were in the same non-partitioned set, the delivery ratio was higher than when the source and destination nodes were split across the partition. The iNSpect program made it possible to quickly understand the differing performance statistics.

5.3   iNSpect Details

The iNSpect program offers several calculations and design features that provide the researcher with a powerful visualization and analysis tool. In this section we discuss details of the iNSpect program.

5.3.1   iNSpect Calculations

Connectivity Graph The *Connectivity Graph* tool renders a line between nodes that are within range of each other based on the transmission range of each node. Figure 5.10 shows the iNSpect program with the connectivity graph illustrated. The connectivity lines can be used to determine shortest paths, unavailable paths,
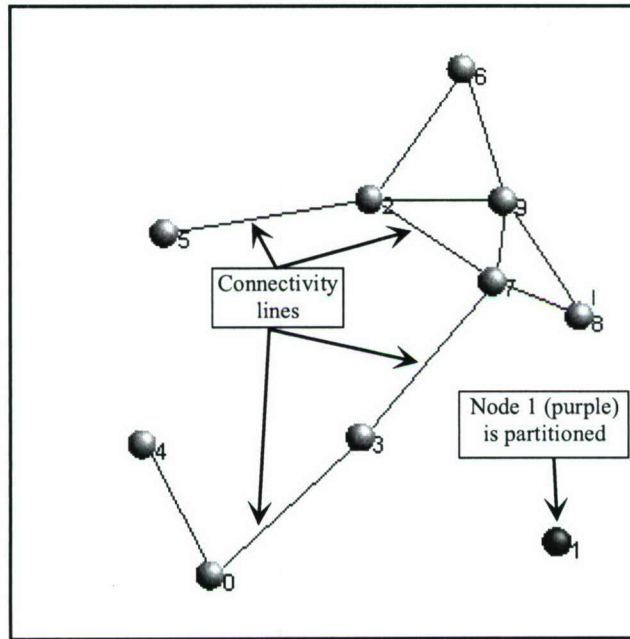
Figure 5.10. iNSpect showing the connectivity graph and partition check calculations for a 300 m x 300 m simulation area with 10 nodes. Transmission range is 100 m.

and potential routing loops. The paths can also be compared to the node's neighbor tables, to determine the accuracy and currency of each node's neighbor information.

Partition Check The *Partition Check* tool identifies isolated nodes in a network. Partitioning occurs when a node is not connected with any other node in the network and, when present, can impact protocol performance. The *Partition Check* tool changes the appearance of any node that is disconnected from the rest of the nodes in the network. Figure 5.10 shows node 1 is partitioned from the other nodes in the network. We use the *Partition Check* tool to check the degree of partitioning present in a network. Generating adjacency and connectivity matrices to check a

```
********************************************************
*                    Node Report                     *
********************************************************
                    Packet Data

Node ID: destination:  dropped:  forwarding:  source:
0           0             0          82         145
1           0            24         106           0
2           0            58         158           0
3           0             6          10           0
4           0            24          86         139
5         114            76          78           0
6           0            13          44           0
7           0             5          30           0
8         115            82          48         138
9           0            17          60           0
10          0            11          54           0
11        127           108         110           0
12          0            31          70           0
13          0            49         174         139
14          0             7          32           0
15          0            16          44           0
16        113            74          44         141
17          0            39         124           0
18          0            29          96           0
19          0            17          40         141
20        112            70          24           0
21          0            24         110           0
22          0            24         100           0
23          0            12          60           0
24        116            84          90           0

Totals:   697           900        1874         843
       -------------------------------------------------
```

Figure 5.11. An iNSpect *Node Report* showing the number of packets received, dropped, forwarded, and sent by each node.

scenario for partitioning is an expensive calculation. However, because iNSpect is already scanning and rendering each node, visualizing partitioning is an inexpensive calculation during the playback.

Node Reports The iNSpect program can generate reports of the performance statistics calculated during the simulation. These reports can be generated anytime throughout the playback. A *Node Report* is shown in Figure 5.11 and includes the total number of packets delivered (destination) to the node, dropped by the node, forwarded by the node, and sent (source) by the node, for each node in the simulation.

The *Node Report* provides more than a summation of the trace file entries. For example, the NS-2 trace file contains only send, receive, and collision packet events. Packets that are sent from a node and never received at the next hop are not recorded as dropped packets. The iNSpect program links a targeted node with a send attempt. This link allows iNSpect to calculate the number of packets forwarded, dropped, and received successfully at the destination and to provide this information in the *Node Report*. The *Node Report* can then be used to identify any unusual trends in certain nodes or areas of the scenario.

5.3.2   iNSpect Features

The iNSpect program provides a feature rich environment to visualize and analyze wireless networks. The iNSpect program has display maneuvering, node locations, node selection, and the following overlay capabilities: geometric shapes and background images. The iNSpect program also provides direct image and movie capture. Additionally, all customizable settings are configurable through the iNSpect control panel or configuration files. We discuss iNSpect's features in this section.

Display Manipulation   The iNSpect display area is a three dimensional rendering area for OpenGL, that allows the researcher to zoom in and out. The display area can also be panned up, down, left, and right. In addition, there is a slider bar and buttons to adjust the time of the simulation playback, allowing the user to move the simulation time forward or backward (see Figure 5.12).

Node Location   The iNSpect coordinate overlay displays node locations in (x,y) coordinates; see Figure 5.13 for an example. Location information can be used to evaluate location-based routing protocols. In location-based protocols, a node's
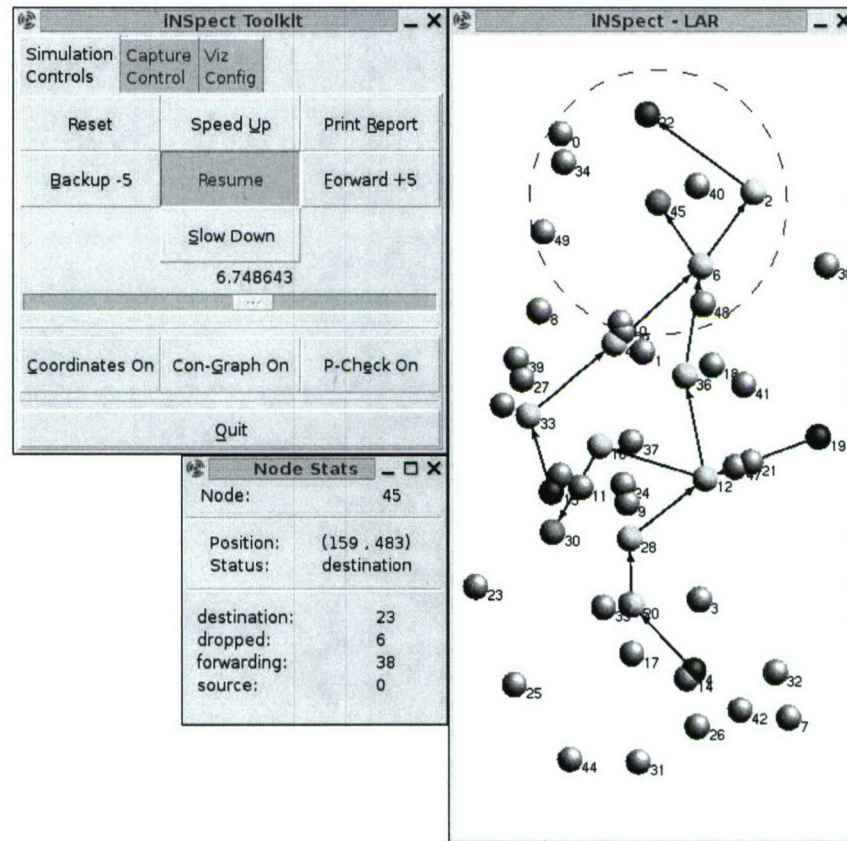
Figure 5.12. The iNSpect graphical user interface. Clockwise from top left: iNSpect control window, network display window and the node status window.

knowledge of a destination's location is used to determine a route to the destination [55]. Using persistent routes and the node locations on the iNSpect display, a researcher can evaluate a protocol's performance on individual routes. This gives the researcher detailed information not available in summary statistics.

Node Selection Clicking on a node in the display selects the node. When an individual node is selected in the scenario, a transmission range ring (dotted line) is

128
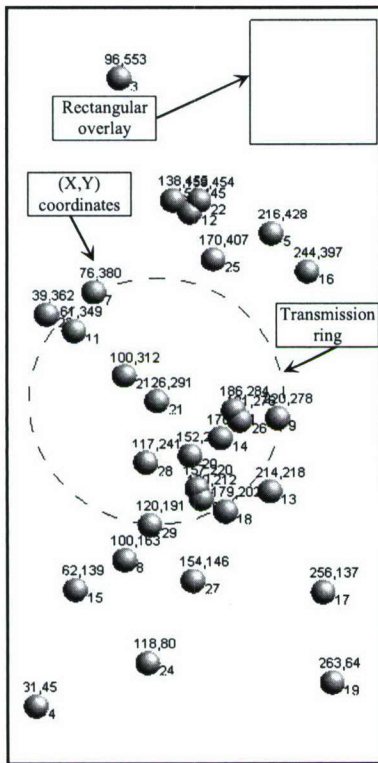
Figure 5.13. iNSpect showing node location, transmission ring, and square area of interest in a 300 m x 600 m simulation area with 30 nodes. Transmission range is 100 m.
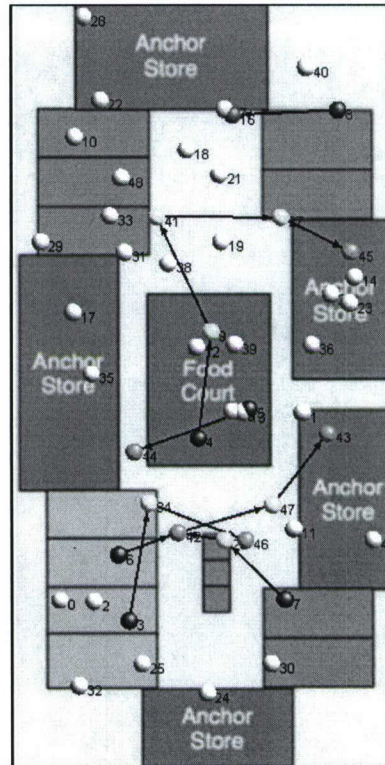
Figure 5.14. iNSpect visualization with a mall as an overlayed background image. Simulation area is 0.5 km x 1 km with 50 nodes.

displayed showing the ideal transmission range of the node (see Figure 5.12, node 45, and Figure 5.13, node 21). Additionally, the *Node Status* window (see Figure 5.12) is updated with the node's number, the node's current x and y location, and the node's current status from the trace file (e.g., source or destination). The *Node Status* window also gives current totals for the different status types in the trace file.

Geometric Shapes   The iNSpect program allows a user to display geometric objects, such as circles and rectangles, which may identify regions of interest. We used circular overlays with a new mobility model that implements congestive movement for nodes in a given area of the simulation. In this new mobility model, a node moves according to the Random Waypoint Mobility Model [18]. (The nodes are initialized in the steady state distribution of the Random Waypoint Mobility Model [67, 68].) Then, as a node moves into the area of congestion, the node slows down. We use the circular overlay in iNSpect to represent the congested area, verifying that the nodes slow in this defined area. The area can represent a food court at a mall or a large intersection in a city. The location and size of the circular objects are configurable within iNSpect.

A rectangular overlay is also available in iNSpect. We used a rectangular overlay in evaluating geocast routing protocols. In geocast routing, packets are forwarded to nodes in a specific geographical area of the simulation [42]. For example, a city dispatcher may need to send emergency information to a certain area of a town to alert citizens of an evacuation. Figure 5.13 depicts a rectangular area of interest with corners at $(200\,\text{m}, 500\,\text{m})$ and $(300\,\text{m}, 600\,\text{m})$. The representation of the rectangle on the display allows visual analysis of a packet's route to the area.

The geometric overlays of iNSpect can be used to represent obstacles as well. As stated in [1], obstacles make mobility models more realistic. The obstacles affect both transmission and movement of nodes. The iNSpect program can be used to observe the affects of the obstacles on the movement of nodes and the transmission of packets.

Background Image Display   The iNSpect program allows the researcher to display background images. The GTK+ toolkit enables iNSpect to support popular image formats (`jpeg`, `gif`, `png`, etc.). The background in the rendering area

is an image. While the default image is white, other images can be loaded for display. Figure 5.14 shows an example of a scenario with a shopping mall map loaded as the background. The image display capability allows a researcher to display networks and nodes on a real background, adding context to scenarios for education and presentation purposes.

Image/Movie Capture The iNSpect program has native screen capture and movie capture capability for presentations and education. The *Capture Control* tab provides facilities to capture screenshots of the display area in both **ppm** and **png** formats. The images are captured directly from the frame buffer for high quality images. The *Capture Control* tab also provides controls to produce **MJPEG** encoded movies. The movie control provides a selector to set the start time, stop time, and frame rate. These screenshots and images can be used for education and presentation purposes. We used these controls to capture the images for this Chapter.

5.3.3 iNSpect Implementation

Graphical User Interface The iNSpect program provides a graphical user interface (GUI) for researchers to interact effectively with the playback environment. The *Simulation Controls* tab of the iNSpect GUI is illustrated in Figure 5.12. The "Speed-up" button doubles the simulation playback speed and the "Slow-down" button halves the simulation playback speed. The "Backup -5" and "Forward +5" buttons move the simulation playback timer backward or forward five seconds, respectively. The Pause/Resume button works as one would expect; Figure 5.12 shows an example of the *Simulation Controls* tab in the paused state. The slider bar allows the user to move to any point (forwards or backwards) in the simulation. The current simulation time is displayed above the slider bar. The three buttons above

Quit (i.e., "Coordinates On", "Con-Graph On", and "P-Check On") were discussed in Section 5.3.1. Finally, all controls can be accessed using a hot key from both the toolkit and simulation windows. As an example, the 'p' key can be pressed once to pause the simulation and then again to resume the simulation.

Configuration File Our iNSpect program is driven by a configuration file, which minimizes the command line arguments while enabling the user to control numerous aspects of the display. All user configurable parameters are defined by defaults in the program or by values provided in the configuration file. The configuration file and system defaults minimize the amount of effort required by a researcher to customize iNSpect for his or her needs. For example, the user can define the start and end time of the playback, which allows a researcher to jump to a specific portion of the playback quickly. If the user does not define the start and end time, the playback will begin at zero seconds and end after the full trace file is played. These values can be changed directly in the configuration file, or the *Viz Config* tab can be used to adjust and save configurable items. Using the *Viz Config* tab allows a researcher to see the immediate impact of a parameter change.

File Parser Input/output processing can be a performance issue for NS-2 simulation playback due to the large size of NS-2 trace files. (A typical NS-2 simulation can generate trace files over 1 GB for a 1000 second simulation.) The iNSpect program utilizes a threaded parser and a read-ahead scheme to keep data flowing to the display portion of iNSpect. The iNSpect program starts the file parser early in the startup sequence. The parser signals the display to begin rendering when the parser has read sufficient data for each node in the display. The file parser then continues to read-ahead in the background while the display is rendering the scenario to the

researcher. The file parser is implemented as a thread to eliminate blocking between the file read and display. This approach enables the researcher to view the NS-2 simulation scenario quickly and smoothly, and avoids a several minute wait to pre-process a large trace file.

5.3.4 Additional uses

The iNSpect program can also be used to verify propagation models and transmission range behavior. In this case, the researcher places nodes at varying distances around a test node and has the test node transmit to each node. The resulting trace file and iNSpect can verify the communication successes of nodes within the transmission range and communication failures of nodes outside the transmission range.

Furthermore, because iNSpect is C++, GTK+, and OpenGL code, it is easy to write front-end processing units. The straightforward code can easily be extended to process different types of trace files, mobility files, and events. The overlay patterns present in the current code can be extended to include other OpenGL-based rendering functions.

5.4 Conclusions

With the increase in wireless network research, visualization and analysis of node behavior, simulations, and results are necessary to engage in productive development. By using the iNSpect tool, a researcher can discover anomalies in topology files, the simulation model itself, or even in the results of a particular protocol. As we have seen in our own research, iNSpect can reveal issues that summary statistics cannot and has saved us hours of detailed detective work trying to verify results. From analyzing node movement to packet routing, iNSpect can provide insight not available from totals and

averages. Our tool is useful for simulations of large sensor networks, a simple wireless LAN, or a mobile ad hoc network. The iNSpect program works directly with NS-2 input and output files, and can read a specific iNSpect formatted trace file from other simulators or testbeds. In short, iNSpect lets the human visual system participate in the analysis of wireless simulation results. For details on obtaining iNSpect, go to *http://toilers.mines.edu/iNSpect.*

Chapter 6

CONCLUSIONS

MANET simulation-based research is an involved process with plenty of opportunities to compromise the credibility of the study. Our survey of MobiHoc papers showed the current state of MANET research and the lack of consistency, re-enforcing the need for simulation study guidance (see Chapter 2). In general, results published on MANET simulation studies lack believability. There are several factors involved in conducting trustworthy simulation-based research. We focused on the following four areas of credibility in simulation research.

1. Repeatable: A fellow researcher should be able to repeat the results for his/her own satisfaction, future reviews, or further development.

2. Unbiased: The results must not be specific to the scenario used in the experiment.

3. Rigorous: The scenarios and conditions used to test the experiment must truly exercise the aspect of MANETs being studied.

4. Statistically sound: The execution and analysis of the experiment must be based on mathematical principles.

Based on this credibility criteria, we identified several simulation lifecycle pitfalls. Each of the pitfalls discussed in Chapter 2 takes away from the goals of making the research repeatable, unbiased, rigorous, and statistically sound. Documenting these

136

pitfalls and sharing knowledge about how to address these common issues will increase the reliability of studies in the MANET community.

In Chapter 3, we expanded on the pitfalls of simulation setup in the area of scenario generation. We highlighted that we need standards to obtain rigorous evaluation of MANET protocols, and we began to define these standards and proposed two standards that should be employed to ensure long routes are available and used in the evaluation of generic MANET routing protocols. Our two proposed standards are not individual parameter settings, but a definition of two metrics that should be calculated and recorded with any simulation-based research that desires credit for rigorously testing a generic MANET routing protocol.

Standard 1: To rigorously evaluate generic MANET routing protocols, the average shortest-path hop count needs to be large.

Standard 2: To rigorously evaluate generic MANET routing protocols, only a small amount of network partitioning should exist.

Additionally, we provided algorithms that researchers can use to determine the number of nodes and area required to generate desired A_{sp}Hops and ANP levels and, therefore, construct scenarios that meet their standards. Then, we illustrated our method that others can modify to generate scenarios that use a different mobility model or propagation model, with different values for both the minimum average shortest-path hop count and the maximum amount of network partitioning.

The algorithms we presented in Chapter 3 enable investigators to specify desired values for ANP and A_{sp}Hops, then construct a simulation scenario that meets these target values to a close approximation. We developed algorithms for four different aspect ratios.

- Equations 3.7 and 3.8 can be used to construct scenarios with square simulation areas that meet specified values for A_{sp}Hops and ANP.

- Equations 3.11 and 3.12 can be used to construct scenarios with simulation areas with 1×2 aspect ratios and specified values for A_{sp}Hops and ANP.

- Equations 3.15 and 3.16 can be used to construct scenarios with simulation areas with 1×3 aspect ratios and specified values for A_{sp}Hops and ANP.

- Equations 3.19 and 3.20 can be used to construct scenarios with simulation areas with 1×4 aspect ratios and specified values for A_{sp}Hops and ANP.

For each of the algorithms developed for a specific aspect ratio and number of nodes, there is no guarantee that a scenario exists that meets the standards used by the researcher. In fact, there exists a smallest simulation area that can be used to meet our standard for hops and a largest simulation area that can be used to meet our standard for partitioning. Additionally, we concluded that within each number of nodes and width/height combination that we tested, varying node speed and node pause time had little effect on A_{sp}Hops and ANP.

We showed in Chapter 4 that the design of a MANET simulation-based study involves setting values for a large number of variables, beyond the scenarios. To conduct a rigorous and credible simulation study of a MANET routing protocol, an investigator must know which variables are likely to have the greatest impact on protocol performance. In our evaluation of the Location Aided Routing (LAR) protocol, we reached the following conclusions:

- A statistical approach can be used to screen a large number of variables.

- Some often-considered variables do not have the most significant impact on delivery ratio.

- Some less frequently considered variables have a substantial impact on delivery ratio.

- The selection and pairing of source nodes to destination nodes for packets is an important process.

- Investigators should determine which variables are significant.

- MANET routing protocols should be systematically tested across each variable found to be significant.

Our desire to provide researchers with guidance, standards, and knowledge of significant variables continued through the simulation study lifecycle with the development of a visualization and analysis tool. In Chapter 5 we showed that with the increase in wireless network research, visualization and analysis of node behavior, simulations, and results are necessary to engage in productive development. The interactive NS-2 protocol and environment confirmation tool (iNSpect) program enables a researcher to animate the results and reveal issues that summary statistics cannot. The iNSpect program lets the human visual system participate in the analysis of wireless simulation results.

Mobile ad hoc network simulation-based research will continue, and so will the opportunities to compromise credibility. However, with our research that raises awareness of the issues, and provides standards and tools, compromises can be reduced. As a result, the guidance, standards, and tools in this dissertation can help increase the credibility of MANET studies community-wide. Information on obtaining the code used in this study can be found at *http://toilers.mines.edu*.

REFERENCES

[1] K.C. Almeroth, A. Jardosh, E. M. Belding-Royer, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the Nineth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 217–229, 2003.

[2] K. Amallada. NAM with wireless simulation. http://mailman.isi.edu/piper mail/ns-users/2004-April/041770.html. Page accessed on July 5, 2006.

[3] E. Angel. *Interactive computer graphics: a top-down approach with OpenGL*. Addison Wesley, 1997.

[4] R. Bagrodia and M. Takai. Position paper on validation of network simulation models. Computer Science Department of University of California, Los Angeles, 1999.

[5] F. Bai, A. Helmy, and N. Sadagopan. User manual for important mobility tool generators in NS-2 simulator. Version 1.0, beta, 2004.

[6] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala. Improving simulation for network research. Technical Report 99-702, University of Southern California, 1999.

[7] O. Balci. Validation, verification, and testing techniques throughout the life-cycle of a simulation study. In *Proceedings of the 26th Conference on Winter Simulation*, pages 215–220, 1994.

[8] O. Balci. Guidelines for successful simulation studies. In *Proceedings of the 22nd Conference on Winter Simulation*, pages 25–32, 1997.

[9] O. Balci. Verification, validation and accreditation of simulation models. In *Proceedings of the 29th Conference on Winter Simulation*, pages 135–141, 1997.

[10] C. Barrett, M. Drozda, A. Marathe, and M. Marathe. Characterizing the interaction between routing and MAC protocols in ad-hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 92–103, 2002.

[11] R. Barton. Design of experiments: designing simulation experiments. In *Proceedings of the 33rd Conference on Winter Simulation*, pages 47–52, 2001.

[12] M. Benveniste, G. Chesson, M. Hoeben, A. Singla, H. Teunissen, and M. Wentink. Enhanced distributed coordination function EDCF. IEEE Working Document 802.11-01/131r1, March 2001.

[13] J. Boleng and T. Camp. Adaptive location aided mobile ad hoc network routing. In *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 423–432, 2004.

[14] J.-Y. Le Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proceedings of the 38th IEEE Symposium on Simulation*, pages 72–79, 2005.

[15] G. Box, S. Hunter, and W. Hunter. *Statistics for Experimenters - An Introduction to Design, Data Analysis, and Model Building*. John Wiley and Sons, Inc., 1978.

[16] L. Brakmo and L. Peterson. Experiences with network simulation. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, pages 80–90, 1996.

[17] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 45–50, 2000.

[18] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC) Journal: Special Issue on Mobile Ad Hoc Networking*, 2(5):483–502, 2002.

[19] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance comparison of two location based routing protocols for ad hoc networks. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1678–1687, 2002.

[20] J. Chen. DSR performance is too bad in ns-2, why? URL: http://mailman.isi.edu/pipermail/ns-users/2004-March/040565.html. Page accessed on July 5, 2006.

[21] T. Cioppa. *Dissertation on Efficient Nearly Orthogonal and Space-Filling Experimental Designs for High-Dimensional Complex Models*. PhD thesis, Naval Postgraduate School, 2002.

[22] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[23] M. Cobb, K. White, and S. Spratt. A comparison of five steady-state truncation heuristics for simulation. In *Proceedings of the 32nd Conference on Winter Simulation*, pages 755–760, 2000.

[24] M. Colagrosso, N. Enochs, and T. Camp. Improvements to location-aided routing through directional count restrictions. In *Proceedings of the International Conference on Wireless Networking (ICWN)*, pages 924–929, 2004.

[25] IEEE Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *IEEE 802.11 Standard*. IEEE, New York, 1996. ISBN 1-55937-935-9.

[26] J. Cowie. Scalable simulation framework API reference manual. Version 1.0, 1999.

[27] T. Dyer and R. Boppana. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 56–66, 2001.

[28] Association for Computing Machinery. The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). URL: http://www.sigmobile.org/mobihoc. Page accessed on July 5, 2006.

[29] GIMP. The GNU image manipulation program GIMP toolkit. http://www.gtk.org. Page accessed on July 5, 2006.

[30] D. Goldsman and G. Tokol. Output analysis: output analysis procedures for computer simulations. In *Proceedings of the 32nd Conference on Winter Simulation*, pages 39–45, 2000.

[31] Akaroa Research Group. Akaroa2. URL: http://www.tkn.tu-berlin.de/research/ns-2_akaroa-2/ns.html. Page accessed on July 5, 2006.

[32] The Toilers Research Group. The Toilers, an ad hoc networks research group. URL: http://toilers.mines.edu/. Page accessed on July 5, 2006.

[33] The VINT Group. The network simulator - ns-2. URL: http://www.isi.edu/nsnam/ns/. Page accessed on July 5, 2006.

[34] The VINT Group. Ns-users mailing list. http://mailman.isi.edu/mailman/listinfo/ns-users. Page accessed on July 5, 2006.

142

[35] I. Haddad and D. Gordon. Linux style: Network simulator 2: a simulation tool for Linux. *Linux Journal*, October 20 2002.

[36] J. Hahner, D. Dudkowski, P. Marran, and K. Rothermel. A quantitative analysis of partitioning in mobile ad hoc networks. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, pages 400–401, 2004.

[37] J. Heidemann, S. Kumar, and K. Mills. Expanding confidence in network simulations. *IEEE Network*, September/October:58–63, 2001.

[38] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 53–60, August 1999.

[39] Y. Hu and D. Johnson. Implicit source routes for on-demand ad hoc network routing. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 1–10, 2001.

[40] R. Jain. *The Art of Computer System Performance Analysis*. John Wiley & Sons Inc, 1991.

[41] Z. Ji, J. Zhou, M. Takai, J. Martin, and R. Bagrodia. Optimizing parallel execution of detailed wireless network simulation. In *Proceedings of the Eighteenth Workshop on Parallel and Distributed Simulation*, pages 162–169, 2004.

[42] X. Jiang and T. Camp. A review of geocasting protocols for a mobile ad hoc network. In *Proceedings of the Grace Hopper Celebration (GHC)*, pages 1–5, 2002.

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

[44] P. Karn. MACA - a new channel access method for packet radio. In *Proceedings of the 9th ARRL/CRRL Amateur Radio Computer Networking Conference*, pages 134–140, 1990.

[45] H. Kee. NAM support for wireless traffic. http://mailman.isi.edu/pipermail/ns-users/2004-May/042046.html. Page accessed on July 5, 2006.

[46] C. Kelly and R. Manohar. An event-synchronization protocol for parallel simulation of large-scale wireless networks. In *Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications*, pages 110–119, 2004.

[47] J. Kleijnen, S. Sanchez, T. Lucas, and T. Cioppa. A user's guide to the brave new world of designing simulation experiments. URL: http://diana .gl.nps.navy.mil/šusan/. Page accessed on July 5, 2006.

[48] Y. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 66–75, 1998.

[49] A. Law and W. Kelton. *Simulation modeling and analysis*. McGraw-Hill, 2000.

[50] P. L'Ecuyer and R. Simard. On the performance of birthday spacing tests for certain families of random number generators. *Mathematics and Computers in Simulation*, pages 131–137, 2001.

[51] S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 37–44, 2000.

[52] J. Liu, L. F. Perrone, Y. Yuan, and D. Nicol. The simulator for wireless ad hoc networks (SWAN). URL: http://www.eg.bucknell.edu/swan/. Page accessed on July 5, 2006.

[53] J. Liu, Y. Yuan, D. Nicol, R. Gray, C. Newport, D. Kotz, and L. Perrone. Simulation validation using direct execution of wireless ad-hoc routing protocols. In *Proceedings of the 18th Workshop on Parallel and Distributed Simulation (PADS)*, pages 7–16, 2004.

[54] J. Malek. Trace graph - network simulator ns-2 trace files analyser. URL: http://www.trace graph.com. Page accessed on July 5, 2006.

[55] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, 2001.

[56] D. McNickle, R. Lee, and K. Pawlikowski. Initial transient period detection for steady-state quantile estimation. In *Proceedings of the Summer Computer Simulation Conference (SCSC)*, pages 1–6, 2000.

[57] J. Mehringer. The NAM editor: A graphical tool for creating ns scripts. A presentation for the CONSER retreat, slide 3, 2001. URL: http:// www.isi.edu/nsnam/nam/nam-editor.ps. Page accessed on July 5, 2006.

[58] M. Meketon. The variance time-curve: Theory, estimation, and application. Master's thesis, Cornell University, 1980.

[59] M. Meketon and B. Schmeiser. Overlapping batch means: Something for nothing. In *Proceedings of the 16th Conference on Winter Simulation*, pages 227–230, 1984.

[60] T. Melodia, D. Pompili, V. Gungor, and I. Akyildiz. A distributed coordination framework for wireless sensor and actor networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 99–110, 2005.

[61] S. Midkiff. Simulation: Methodology. Presentation to ECE590 class. URL:http:// fiddle.visc.vt.edu/courses/ece5984-net2/lectures/09_sim.pdf. Page accessed on July 5, 2006.

[62] A. Miller. *Subset Selection in Regression, 2nd ed.* Chapman and Hall, 2002.

[63] L. Miller. Multihop connectivity of arbitrary networks. Technical report, Wireless Communication Technologies Group, National Institute of Standards and Technology (NIST), 2001.

[64] D. Montgomery. *Design and Analysis of Experiments*. Wiley, 2005.

[65] E. Mota, A. Wolisz, and K. Pawlikowski. Experience with Akaroa: speeding up stochastic simulation using parallel replications. In *Proceedings of the High Performance Computing and Advanced Simulation Technologies Conference*, pages 296–301, 1998.

[66] A. Nasipuri, J. Zhuang, and S. Das. A multichannel CSMA MAC protocol for multihop wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1402–1406, 1999.

[67] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, pages 99–108, 2004.

[68] W. Navidi, T. Camp, and N. Bauer. Improving the accuracy of random waypoint simulations through steady-state initialization. In *Proceedings of the 15th International Conference on Modeling and Simulation (MS'04)*, pages 319–326, 2004.

[69] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 151–162, 1999.

[70] *The NS manual: formerly known as notes and documentation*. URL:http://www.isi.edu/nsnam/ns/. Page accessed on July 5, 2006.

[71] Canadian Department of National Defence Synthetic Environment Coordination Office. *Modeling and Simulation Verification, Validation and Accreditation (VV&A) Guidebook.* URL:http://www.drdc-rddc.gc.ca/seco/documents/VVA _Guidebook_DND_SECO_May_2003_e.html. Page accessed on July 5, 2006.

[72] D. Patterson. How to have a bad career in research/academia. URL: http:// www.cs.berkeley.edu/pattrsn/talks/BadCareer.pdf. Page accessed on July 5, 2006.

[73] K. Pawlikowski. Steady-state simulation of queuing processes: survey of problems and solutions. *ACM Journal of Computing Surveys*, pages 123–170, 1990.

[74] K. Pawlikowski. Do not trust all simulation studies of telecommunications networks. *International Conference on Information Networking (ICOIN)*, pages 899–908, 2003.

[75] K. Pawlikowski, J. Jeong, and R. Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, pages 132–139, 2001.

[76] K. Pawlikowski, J. Jeong, and R. Lee. Letters to the editor. *IEEE Communications Magazine*, pages 132–139, 2002.

[77] K. Pawlikowski and V. Yau. *Akaroa-2 Manual.* Department of Computer Science, University of Canterbury, ChristChurch New Zealand. URL: http: //www.tkn.tu-berlin.de/research/ns-2_akaroa-2. Page accessed on July 5, 2006.

[78] G. Pei, M. Gerla, X. Hong, and C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1538–1542, 1999.

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

[80] D. Perkins, H. Hughes, and C. Owens. Factors affecting the performance of ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 2048–2052, 2002.

[81] L. F. Perrone, Y. Yuan, and D. Nicol. Modeling and simulation best practices for wireless ad hoc networks. In *Proceedings of the 35th Conference on Winter Simulation*, pages 685–693, 2003.

[82] The CMU Monarch Project. The CMU monarch extensions to the ns simulator. URL: http://www.monarch.cs.cmu.edu/. Page accessed on July 5, 2006.

[83] The CMU Monarch Project. Monarch Project: Ad-hockey for Perl/Tk800.015. http://www.monarch.cs.rice.edu/cmu-ns.html. Page accessed on July 5, 2006.

[84] The CRAWDAD Project. A community resource for archiving wireless data at Dartmouth CRAWDAD. URL: http://crawdad.cs.dartmouth.edu/index.php. Page accessed on July 5, 2006.

[85] The SEACORN Project. SEACORN simulation tools. URL: http://seacorn. ptinovacao.pt/sim_tools.html. Page accessed on July 5, 2006.

[86] S. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile networks for multimedia support. *ACM/Baltzer Mobile Networks and Applications*, 3(1):101–119, 1998.

[87] T. Rappaport, editor. *Wireless Communications - Principle and Practice.* Prentice-Hall, 1996.

[88] J. Rodrigue. The notion of accessibility. URL: http://people.hofstra.edu/geo trans/eng/ch1en/meth1en/ch1m2en.html Page accessed on July 5, 2006.

[89] E. Royer, P.M. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 857–861, 2001.

[90] N. Sadagopan, F. Bai, B. Kishuamachori, and A. Helmy. PATHS analysis of path duration statistics and their impact on reactive MANET routing protocols. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 245–256, 2003.

[91] S. Sanchez. Output modeling: abc's of output analysis. In *Proceedings of the 33rd Conference on Winter Simulation*, pages 30–38, 2001.

[92] R. Sargent. Validation and verification of simulation models. In *Proceedings of the 31st Conference on Winter Simulation*, pages 39–48, 1999.

[93] R. Sargent. Verification, validation, and accreditation of simulation models. In *Proceedings of the 32nd Conference on Winter Simulation*, pages 50–59, 2000.

[94] B. Scheuermann, H. Füßler, M. Transier, M. Busse, M. Mauve, and W. Effelsberg. Huginn: A 3D Visualizer for Wireless ns-2 Traces. In *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 134–150, 2005.

[95] L. Schruben. Detecting initialization bias in simulation output. *Operations Research*, pages 569–590, 1982.

[96] The Georgia Tech Parallel/Distributed simulation project. Parallel/distributed extensions for NS-2. URL: http://www.cc.gatech.edu/computing/compass /pdns/index.html. Page accessed on July 5, 2006.

[97] Network Simulator-2. The network simulator NS-2: Validation tests. URL: http://www.isi.edu/nsnam/ns/ns-tests.html. Page accessed on July 5, 2006.

[98] N. Steiger and J. Wilson. Experimental performance evaluation of batch means procedures for simulation output analysis. In *Proceedings of the 32nd Conference on Winter Simulation*, pages 627–636, 2000.

[99] K. To. Bug in ns-2.27 wireless channel. URL: http://mailman.isi.edu/pipermail /ns-users/2004-April/041388.html. Page accessed on July 5, 2006.

[100] M. Totaro and D. Perkins. Using statistical design of experiments for analyzing mobile ad hoc networks. In *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 159–168, 2005.

[101] K. Vadde and V. Syrotiuk. Factor interaction on service delivery in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, pages 1335–1346, 2004.

[102] P. Welch. *The Computer Performance Modeling Handbook*, chapter The Statistical Analysis of Simulation Results, pages 268–328. Academic Press, 1983.

[103] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 194–205, 2002.

[104] J. Wilson, A. Alan, and B. Pritsker. A survey of research on the simulation startup problem. *Simulation*, pages 55–58, 1978.

[105] M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide: The official guide to learning OpenGL*. Addison Wesley, 1997.

[106] R. Zheng, J. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 35–45, 2003.

APPENDIX A

DESIGN OF EXPERIMENTS

A.1 DOE Overview

Design of Experiments (DOE) is a large research area with a rich history in a variety of fields outside of computer simulation [15]. Recent research shows DOE is an excellent approach for evaluating input parameters (factors in DOE) of simulations [47]. DOE is recommended for iterative experiments where the simulation-study designer wants to determine a short list of impacting factors from a long list of potential factors. The short list of factors is used to eliminate unimportant factors. The short list of factors is also a starting point to conduct other detailed DOEs on the impacting factors [47]. The initial design can measure a few levels for each factor. Then the detailed design, with fewer factors, can measure a larger range of factor levels. If a researcher tried a large range of levels with tens of factors, with a simple design, the number of experiments would be in the millions. DOE provides a means to conduct credible studies in an executable number of experiments.

The authors of [10, 80, 100, 101] describe DOE techniques to evaluate factors affecting MANET performance. The goal of the study in [80] was to determine which factors most affected MANET protocol performance. As a result, a protocol could be

Table A.1. Partial sample factorial design matrix

Design Point	1	2	3	4	5	6	7	8	Result
1	-	-	-	-	-	-	-	-	R_1
2	+	-	-	-	-	-	-	-	R_2
3	-	+	-	-	-	-	-	-	R_3
4	+	+	-	-	-	-	-	-	R_4
5	-	-	+	-	-	-	-	-	R_5
6	+	-	+	-	-	-	-	-	R_6
7	-	+	+	-	-	-	-	-	R_7
8	+	+	+	-	-	-	-	-	R_8
...	...	...	...	...	...	...	...	...	...
255	-	+	+	+	+	+	+	+	R_{255}
256	+	+	+	+	+	+	+	+	R_{256}

designed to adapt with the most impacting factor. The authors of [80] use node speed, pause time, network size, number of traffic sources, and routing protocol as factors. They present the concept of design of experiments to determine factor interactions and impact, specifically, using a 2^k factorial design. In the following two subsections we discuss the DOE technique in more detail.

A.2   Factor Analysis

The 2^k factorial design tests two levels for each factor: a low level represented by a "-" sign and a high level represented by a "+" sign [40, 49]. The 2^k factorial design executes each factor's low and high levels against the other factor's low and high levels. Table A.1 shows the partial list of *design points* for an eight factor experiment. The design points are individual simulation executions with the factors

set as indicated in the design point's row. The *result* is the output measure's value at the end of the simulation execution (e.g., the average delivery ratio for a design point's simulation). Table A.1 is called the *design matrix* for a factorial design. The design matrix associates each factor's setting with a design point and result number.

To measure the effect of a single factor the researcher calculates the change in the result due to changing from the factor's low level to the high level while holding the other factors' levels constant [49]. For example, design points 1 and 2 fix factors 2 - 8 while changing factor 1 from a "-" sign (factor 1's low level) to a "+" sign (factor 1's high level). The difference between the results for design point 1 (R_1) and design point 2 (R_2) is the effect of factor 1 when all other factors have a "-" sign (their low level). The main effect for a factor is the summation of the differences in results due to changing from the factor's "-" sign (low level) to the "+" sign (high level), with all the other factors' combinations. The summation is divided by 2^{k-1} to produce an average effect (e). The main effect for factor 1 (e_1), from Table A.1, is given by

$$e_1 = \frac{(R_2 - R_1) + (R_4 - R_3) + (R_6 - R_5) + (R_8 - R_7) + ... + (R_{256} - R_{255})}{128} \quad (A.1)$$

where the denominator is 2^{k-1}, to average the result differences, and R_x is the result for design point x.

The main effect can be rewritten by applying the signs in the numerator and reordering the results in increasing order. The rewritten expression for factor 1's

main effect (Equation A.1) is given by

$$e_1 = \frac{-R_1 + R_2 - R_3 + R_4 - R_5 + R_6 - R_7 + R_8 - \ldots - R_{255} + R_{256}}{128} \qquad (A.2)$$

where R_x is the result for design point x. The rewritten form (Equation A.2) provides a more general form of the expression. In the general form the results for the design points are reordered in increasing order and the sign of each result follow the sign of the factor whose effect is being calculated. For design point 1, all factors are " - " including factor 1. Therefore, the sign of design point 1's result R_1 is a "-" sign. For design point 2, factor 1 is a "+" sign, making the second result $+R_2$. In other words, if the "+" signs and "-" signs are thought of as 1 and -1, the average is the dot product of the factor column with the result column divided by 2^{k-1} [49]. Using the general form, the main effect equation for factor 2 (e_2) is given by

$$e_2 = \frac{-R_1 - R_2 + R_3 + R_4 - R_5 - R_6 + R_7 + R_8 - \ldots + R_{255} + R_{256}}{128} \qquad (A.3)$$

where the signs of the results follow the sign of factor 2's column in the design matrix.

The main effect accounts for the impact of the individual factor, but does not account for the interaction of multiple factors. The interaction of two factors is called the *two-interaction effect* [49]. The two-interaction effect looks at the combined impact of two factors on the performance results. For example, the two-interaction effect of factor 1 and factor 2 is given by 1x2. Table A.2 adds a partial list of the

Table A.2. Partial sample factorial design matrix of two-interaction effects

Design Point	1	2	3	4	5	6	7	8	1x2	1x3	...	7x8	Result
1	-	-	-	-	-	-	-	-	+	+	...	+	R_1
2	+	-	-	-	-	-	-	-	-	-	...	+	R_2
3	-	+	-	-	-	-	-	-	-	+	...	+	R_3
4	+	+	-	-	-	-	-	-	+	-	...	+	R_4
5	-	-	+	-	-	-	-	-	+	-	...	+	R_5
6	+	-	+	-	-	-	-	-	-	+	...	+	R_6
7	-	+	+	-	-	-	-	-	-	-	...	+	R_7
8	+	+	+	-	-	-	-	-	+	+	...	+	R_8
...	...	...	...	...	...	...	...	...	...	...	...	...	...
255	-	+	+	+	+	+	+	+	-	+	...	+	R_{255}
256	+	+	+	+	+	+	+	+	+	+	...	+	R_{256}

two-interaction effects to the design matrix shown in Table A.1. The signs shown in the two-interaction effect columns in Table A.2 are the signs used for the design point's result, based on the multiplication [49].

The impacts for the two-interaction effects are calculated as half the difference between the average effect of the first factor when the second factor is a "+" sign and the average effect of the first factor when the second factor is a "-" sign. The two-interaction effect for factors 1 and 2 ($e_{1,2}$) from Table A.2 is given as

$$e_{1,2} = \frac{1}{2}\left[\frac{(R_4 - R_3) + ... + (R_{256} - R_{255})}{64} - \frac{(R_2 - R_1) + ... + (R_{254} - R_{253})}{64}\right] \quad (A.4)$$

The effect is the difference between the average when the two factors are at the same level and the average when the two factors are at different levels. Equation A.4

can be rewritten by applying the signs, reordering the results, and distributing the 1/2. The rewritten two-interaction effect equation for factors 1 and 2 is given by

$$e_{1,2} = \frac{(R_1 - R_2 - R_3 + R_4 + R_5 - R_6 - R_7 + R_8 + ... - R_{255} + R_{256})}{128} \quad (A.5)$$

The sign is the multiplication of the signs from factors 1 and 2. The sign on the result is a "+" sign when the two factors' signs are the same and a "-" sign when the signs are different. The two-interaction effect provides an advantage in understanding impact over other comparison techniques that assume the factors are independent.

Once all of the main effects and two-interaction effects have been calculated, various plots can be used to determine the factors causing the most negative and positive effect on the results. If there are no clear impacts from the factors or a few factors with a majority of the impacts, more analysis can be done with a range of factors' values in place of the high and low only values in the 2^k factorial design.

A.3 Other DOEs

In [21] the authors use DOE with an orthogonal Latin Hypercube Sampling (LHS) design to evaluate simulation factors. Orthogonal LHS allows the evaluation of factors with a range of settings. The 2^k factorial designs only allow a factor to have a low and high value.

APPENDIX B

ANALYSIS DETAILS

B.1 Analyzing Terminating Simulations

In terminating simulations, the statistical analysis of the discrete or continuous output is based on estimating the expected value of the sample mean. The sample mean is given by

$$\bar{X}(n) = \frac{\sum_{i=1}^{n} X_i}{n} \tag{B.1}$$

and the sample variance is given by

$$S^2(n) = \frac{\sum_{i=1}^{n} [X_i - \bar{X}(n)]^2}{n - 1} \tag{B.2}$$

where n is the number of observations, $S(n)$ is the standard deviation, and the data, X_i, is *iid*. If equations (B.1) and (B.2) were used without *iid*, the mean would still be an unbiased estimator, but the variance would be highly biased [30]. To achieve *iid* output, terminating simulationists use independent replications [49]. These replications are made independent by initializing each simulation with a different seed for the PRNG.

Also, with random-based simulations, confidence intervals should be calculated to see the range required to cover the sample mean. The confidence interval is given by

$$\bar{X}(n) \pm t_{1-\alpha/2,n-1}\sqrt{\frac{S^2(n)}{n}} \tag{B.3}$$

where $100(1-\alpha)\%$ is the confidence level and $1-\alpha/2$ is the upper critical point of the t distribution with $n-1$ degrees of freedom. As noted in [49] the correctness of the confidence interval depends on the assumption that the X_i values are normal random variables. The confidence interval is the most common method to approximate the unknown mean, because the "normal" assumption rarely occurs in simulation output.

B.2   Analyzing Steady-State Simulations

Once the initialization bias has been removed from the data set, there are several ways to analyze the steady-state output. See [73] for a survey of steady-state output analysis techniques. The trade off between the different ways of handling the data is a balance between replication and initialization bias. The first technique is similar to terminating replication, but contains a higher risk of including initialization bias in the output. The other techniques are based on a single longer simulation, where only one instance of initialization bias is included.

Independent Replication: Independent replication for steady-state simula-
tion is the same as that discussed for terminating simulations (see Section B.1), once
the initialization bias is removed. Unlike batch means, replication output is indepen-
dent by construction because the issue of correlation is addressed [30]. The idea is to
execute the simulation 5 to 10 times [49]. The disadvantage is that each replication
must be initialized at startup, which means the initialization bias is an issue with
each replication.

Batch Means: Batch means, unlike replication, is based on a single execution
of the simulation. The single execution is sufficiently long to discard the initialization
bias and contains enough output data to divide it into batches [98]. All of the output
is divided into equal sized batches. Based on the central limit theorem, the batch
sample means are approximately *iid* with normal distribution. Because the batches
are *iid* and normal, the equations (B.1), (B.2), and (B.3) can be used. The advantage
of batch means is that initialization bias is handled once. One caution with batch
means is the risk of correlation among the batch means. For example, if an unusual
event happens in a batch, it probably happens to the data around the event, skewing
the mean for that batch.

Overlapping Batch Means: Another technique proposed in [58] and shown
in [73] is the concept of overlapping batch means. The data, after removing initial-

ization bias, is divided into equally sized batches. The advantage is a researcher can increase the size of the batch, without reducing the number of batches for a single set of data. The authors of [59] show that overlapping to collect larger batches will reduce the variance of the variance estimator.

Regenerative method: The regenerative method is another batching type method, except the batches are of varying length. The regenerative method is based on the idea that simulations cycle through events (starting conditions, especially) [73]. For example, one regenerative batch is started each time the queues of a network are empty. The advantage of regenerative cycles is that initialization bias is gone, because each cycle starts with the same regenerative point. The difficult part is clearly identifying the regenerative point and determining when it occurs in the simulation. Additionally, because the batches are of random length, equations (B.1), (B.2), and (B.3) cannot be used. See [73] for a description of all the special estimators needed for regenerative cycle analysis.