Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

MODELING AND SIMULATION BEST PRACTICES FOR WIRELESS AD HOC NETWORKS

Luiz Felipe Perrone

Dept. of Computer Science
Bucknell University
Lewisburg, PA 17837-2029, U.S.A.

Yougu Yuan

Dept. of Computer Science
Dartmouth College
Hanover, NH 03755-3510, U.S.A.

David M. Nicol

Dept. of Elec. and Computer Engineering
University of Illinois Urbana-Champaign
Urbana, IL 61801-2918, U.S.A.

ABSTRACT

This paper calls attention to important practices in the modeling and the simulation of wireless ad hoc networks. We present three case studies to highlight the importance of following well-established simulation techniques, of carefully describing experimental study scenarios, and, finally, of understanding assumptions sometimes unstated in the framework of a simulator. The first case addresses the initial transient problem inherent to mobility and traffic generation sub-models. We quantitatively demonstrate how these transients can affect the simulation. Our second case illustrates the fact that strong scientific contributions can only be made via simulation studies when the models used are unambiguously specified. The example we use are simulations with and without a model for the ARP protocol. Finally, our third case discusses the importance of understanding the simulation tool and any default values used for model parameters. The example used relates to the use of the limited interference model.

1 INTRODUCTION

It is widely known that comprehensive models for wireless ad hoc networks are mathematically intractable. On its own, each individual layer of the protocol stack may be complex enough to discourage attempts at mathematical analysis. Interactions between layers in the protocol stack magnify this complexity. In order to better understand these protocols, many have resorted to an experimental approach with running networks, which can quickly prove to be cumbersome and expensive. For these reasons, computer simulation has been the tool of choice to study wireless networks in a scalable, controllable and repeatable environment.

It is therefore not surprising that computer simulation has been used to develop much of the knowledge we have today in the performance of protocols for wireless networks. The vast majority of this body of work has been developed with the assistance of only a few simulators, namely OPNET, ns-2, and GloMoSim. While these tools are household names in the wireless networking community, other simulators have been developed to address specific research questions or to take the capabilities of the tool in directions not covered by their predecessors. One such example is our own SSF-based Simulator for Wireless Ad Hoc Networks (SWAN) (Liu et al. 2001), developed at the Institute for Security Technology Studies, at Dartmouth College. This simulator is being constructed to study the security exposures of protocols for wireless networks.

In the development of SWAN, we have looked for starting points in the literature and in existing open source simulators. While much of what we have found was of great assistance, not all that we discovered was positive. In what regards model development, there seems to be a long lag between the time when a model is embraced by the community and the time when that model becomes well-understood. Recent developments in the analysis of mobility models serve to corroborate this observation (Yoon et al. 2003, Bettstetter et al. 2002). Also, concerning the simulation methodology used in experimental studies in this community, we have come across evidence of ample room for improvement. Discounting the eventual cases of less-than-rigorous analyses of simulation output that end up published, there are also instances when established simulation techniques are ignored resulting in studies with compromised credibility. Our goal in this paper is to propose steps toward averting the crisis of credibility that has been building up around simulations of computer networks, as pointed out by Pawlikowski et al. (2002).

The study presented in this paper results from efforts to validate the models created for SWAN and to establish our own experimental methodology. In order to construct consistent and meaningful scenarios for our experiments, we set out to determine parameters values for each sub-model in our simulations. Surveying the literature, we discovered that, while certain parameter settings have been widely used, the reason for their choosing is most often not explained and their effects not understood. As we attempted to evaluate the motivation for these choices, we have chosen to investigate how they impact our simulations.

Note that although this paper has points in common with works such as Takai et al. (2001) and Heidemann et al. (2001), our main goals are quite different. We do not intend so much to evaluate the effects of detail as to call attention to the necessity of precisely and completely stating the simulation scenarios used in any experimental works with wireless networks. This is not to say that the results presented here don't shed light into the matter of analyzing the sensitivity of wireless network simulation models to their various parameters. We present these results partly to demonstrate this sensitivity phenomenon. Our results illustrate that without the detailed description of experiments, it is unlikely that anyone will be able to successfully replicate or build upon them.

Understandably, many publications reporting the simulation of wireless networks do not disclose all the parameter settings for their studies. The media used for scientific communications, conference proceedings or journals, does not allow one to dedicate much space to listing parameter values. The obvious solution, which has been gaining popularity quickly, is to make this kind of information available in a web page associated with the paper (though it is not obvious how to make the information persistent). Our work follows this practice and the detailed descriptions of our experiments and result data are available at `http://www.eg.bucknell.edu/˜perrone/research/`. We hope that this data can help the users of our simulator to become cognizant of the characteristics of our models.

The remainder of this paper is structured as follows. Section 2 discusses the importance of "warming up" models to avoid transients in the simulations. Although this is an age old practice, it is hardly referred to in most experimental studies of wireless networks. Next, in Section 3, we illustrate the importance of a thorough structural description of the simulation model. The case studied considers the sensitivity of the simulation to the use of the Address Resolution Protocol (ARP) in the protocol stack model of the wireless node. Section 4 illustrates the dangers of using a wireless network simulator without fully understanding its implementation. The case studied showcases the effects of the radio propagation model on networking metrics. Finally, Section 5 presents final thoughts and directions for future research.

2 THE INITIAL TRANSIENT PROBLEM

It is surprising that although basic methodology for computer simulation was established decades ago, many recent studies still pay little attention to it. The vast majority of papers on the simulation of wireless networks seems to ignore the fact that one or more of their sub-models do require *warm-up* time to avoid transient behavior. In this section we explore two different ways in which the initial transient problem manifests itself in the simulation of wireless networks. The first regards sub-models in the simulation environment that "drive" the networking nodes. The second regards directly the sub-models that compose the simulated networking node.

Our first case study is illustrated by the application of the *random waypoint model* (RWM). Although our case study focuses on RWM, the issues we investigate are relevant to other mobility models as well. We hope that efforts such as ours will serve to motivate the further analysis of other mobility models either through experimental or mathematical means. Next, we present an experimental exploratory analysis of RWM's sensitivity to a few parameters.

The algorithm behind RWM is very simple and is based on only three parameters: *pause_time*, *min_speed*, and *max_speed*. The time a mobile remains stationary between spurts of movement is deterministic and specified by *pause_time*. All mobiles start out paused and begin to move at the same time: the end of the initial pause. Each mobile then chooses a destination (waypoint) independently and uniformly, and travels toward it with a speed sampled uniformly from *min_speed* to *max_speed*, inclusively. Upon reaching a waypoint, the mobile pauses again and the algorithm repeats.

As shown by Yoon et al. (2003), although RWM is the most popular mobility model in the simulation analysis of mobile wireless networks, it has significant perils. First, if *min_speed* is set to zero or a very small value, the instantaneous node speed (a metric that quantifies the aggregate level of mobility) converges to zero as time advances. The consequence is obvious and severe: after a sufficiently long time, what is meant to be a *mobile network* becomes *stationary*. Under these conditions, the simulation analysis of protocols for mobile wireless networks is likely to produce misleading results. Second, their work also exposed the fact that for RWM, the level of mobility goes through oscillations before settling down onto a "steady state".

In general, if the statistics collected in a simulation run include the initial transient period, it is likely that the results will exhibit considerable error. This effect, known as the *initial transient problem*, is classically mitigated with the application of *data deletion*: statistics collected during the transient are discarded and therefore have no influence in the final results. The application of data deletion hinges on the identification of the instant of time when the transient has ended (known as *settling time*). The drawback of this approach is that compute time is "wasted" in the process of warming up the model since data is being generated only to be thrown away. When data deletion is used, it is therefore important to attempt to minimize settling time.

In the specific case of mobility models, the first effective measure one can take is to start the simulation with the mobiles positioned in space according to the same probability distribution that arises in the simulation of each specific mobility model. As indicated in Camp et al. (2002), for RWM, it is counterproductive to initially deploy mobiles uniformly in the simulation space. Deployment according to a bi-dimensional triangular distribution leads to a better approximation of the distribution of mobiles that arises from RWM. The second effective measure is to choose a sufficiently large *min_speed* guaranteeing that RWM will eventually reach a steady state. The question of what other measures one should take remains open, however. Our investigation exemplifies, through additional experiments, other parameters in the overall model that can affect the settling time for RWM. We refer the reader to Navidi and Camp (2003) and Bettstetter et al. (2002) for in-depth mathematical analyses of RWM.

Our first experiments used a square arena with 1000m of side, a pause time of 60s and 40 mobiles. At this stage, we did not simulate networking at all, only mobility, and so these experiments took very little time to complete. For this setting, we produced Figure 1 assigning four values to *max_speed*. What we show here is not the data collected directly, but rather the curves smoothed out by Welch's algorithm of moving averages as described by Law and Kelton (2000). In these plots we chose a window sufficiently large to produce smooth curves and, at the same time, sufficiently small not to completely remove the wide oscillations. We see that, for different values of *max_speed*, the wide oscillations subside at different instants in time.

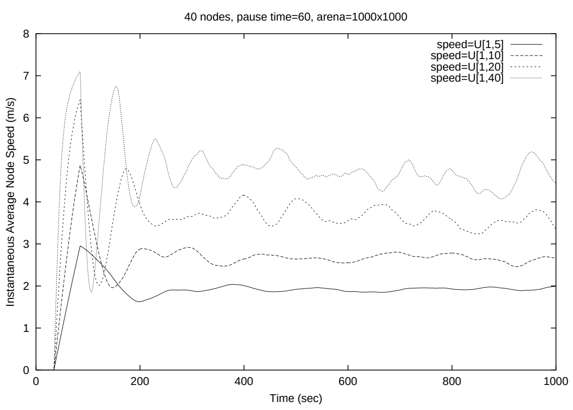

Figure 1: Random Waypoint with Various Values for Maximum Speed

A similar effect is observed in our second set of experiments, in which, keeping all of random waypoint's parameters constant, we varied only the size of the arena. As Figure 2 shows, different dimension values determine transients of different durations. Intuition indicates that in varying the arena size, one also varies the ranges from which the destination points are chosen and thus also the average length of each leg in the trajectory of a mobile. The most important observation that arises from these two figures is the fact that settling time changes with scenario parameters *other* than RWM's parameters. Until the analytical framework has been developed to predict how long the transient will last, it is advisable to do an experimental assessment of the settling time so that data deletion can be safely applied.

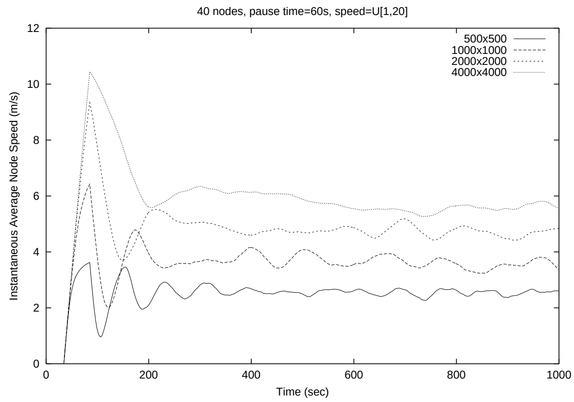

Figure 2: Random Waypoint with Various Values for Arena Size

Having observed these transients in a metric that assesses the level of node mobility in the model, the next point to investigate was, naturally, the impact that these transients would have in the networking metrics computed through simulation. In the experiments that follow in this section and in the remainder of this paper, unless stated otherwise, we have used the same set up described below. In the first 500s of simulation time, networking was disabled; for the parameters used in random waypoint, by this time the transient in instantaneous average node speed was guaranteed to have ended.

Baseline Scenario:

RF propagation: 2-ray ground reflection model, antenna height of 1.5m, transmission power of 15dBm. Each node sees an average of 7 neighbors, so for experiments with varying number of nodes, we vary the arena size accordingly. Signal-to-noise ratio threshold packet reception. Transmitter power of 15dBm.

Mobility: random waypoint model under scenarios that generate a stationary network, and also networks with low and high mobility. The respective parameters were *pause_time=simulation_length* with *min_speed=max_speed=0*, *pause_time=60s* with *speed ˜ U[1,3]*, and *pause_time=0s* with *speed ˜ U[1,10]*.

Traffic generation: a variation of constant bit rate (CBR). Session length is deterministic and equal to 60s. The interval between sessions is also deterministic and equal to 20s. A fixed destination node is chosen uniformly at random for each session. During each session, a constant bit rate is maintained; with packet size fixed at 512 octets, we vary packet interarrival time to produce traffic loads of 16kbps, 56kbps, and 300kbps.

Protocol stack: IEEE 802.11b PHY (modem capture using the message retraining model described by C.Ware et al. 2001), IEEE 802.11b MAC Distributed Coordination Function (DCF), Address Resolution Protocol (ARP), IP, and Ad Hoc On Demand Distance Vector (AODV) routing.

Arena size: adjusted according to the number of nodes in the model so as to maintain an average node density of 7 neighbors per node.

Scale: 20, 30, 40, and 50 network nodes.

Number of runs: 10 runs with different values of random number generator seeds for each stream used in the model. For all the metrics we collected, we estimated intervals of 95% confidence as well as point estimates for their averages.

We compared estimates of end-to-end delay constructed when we correctly skipped the mobility transient. Comparing these observations with the results obtained when the transient is not skipped, we observed that the point estimates for the average relative error varied from as little as 5% to as high as 30%. Figure 3 shows the average relative error that arises in the simulation of a slow-moving network. This plot is produced using only the point estimates from the confidence intervals and, by itself, it indicates the trend that the higher the number of nodes, the higher the average relative error.

The plot in Figure 4 shows the specific case of a 50-node network detailing the difference used in the computation of relative error for average end-to-end delays. The figure shows a growing uncertainty in the estimation of end-to-end delay. It is important to remark that this uncertainty does not arise from the mobility transient. End-to-end delay statistics are collected only for packets that actually arrive at their destinations. As traffic increases, more packets tend to get dropped along their routes and the number of samples used to compute end-to-end delays is reduced. Since the width of the interval estimated is inversely proportional to the square root of the number of samples, the level of uncertainty grows.

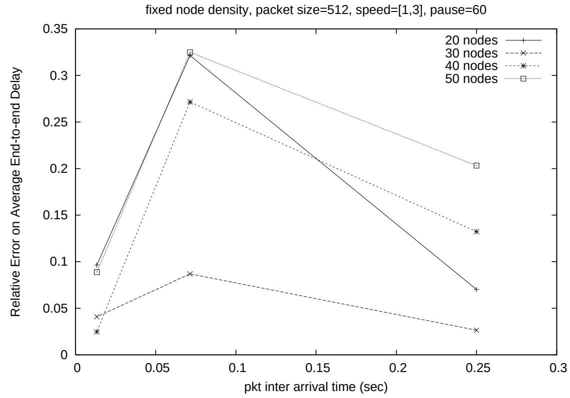

Figure 3: Effect of Mobility Transient on End-to-End Delay with Fixed Node Density in Slow-Moving Network

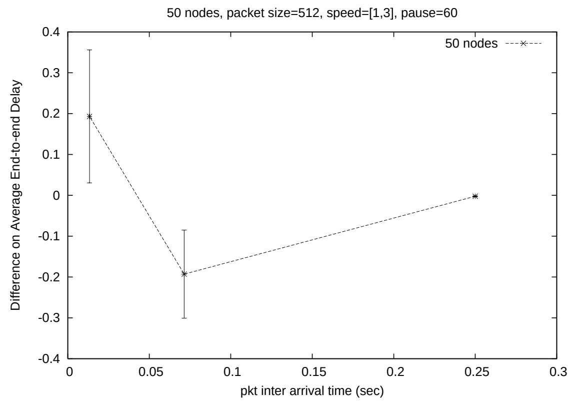

Figure 4: Detail of the Effect of Mobility Transient in 50-Node Slow-Moving Network (95% Confidence Intervals)

Looking at end-to-end delay in the same scenario, but with a more mobile network, we observed similar results. What was consistent among all the variations we investigated, however, was the fact that the mobility transient had a pronounced impact on the accuracy of the estimated values for this metric. We have also observed that tighter confidence intervals are produced for lower traffic loads. This is a direct consequence of the fact that end-to-end delay can only be estimated for packets that arrive at their destination. Since the number of packets that are not dropped due to medium contention or buffer overflows increases fast with increases in traffic load, the number of samples for end-to-end delay is proportionally reduced.

The mobility model transient also showed a marked effect in the ratio of routing packets to application data packets sent. The average relative errors observed were in the same range as those for end-to-end delay. Packet delivery ratio, on the other hand, wasn't affected as much, but still exhibited average relative error as high as 10%.

Next, we turned our attention to the impact of transient effects due to the start up of *network traffic generation*. It would be natural to expect that if buffer queues in protocol models are not pre-initialized so as not to start empty, some kind of transient effect would ensue. To illustrate this point, we ran the simulation of a stationary network with 20 nodes and recorded the evolution of the number of simultaneous ongoing transmissions in time. Figure 5, which has been smoothed out by Welch's algorithm with window size of 200, shows the first 500 seconds of this time series for different values of inter-session time.

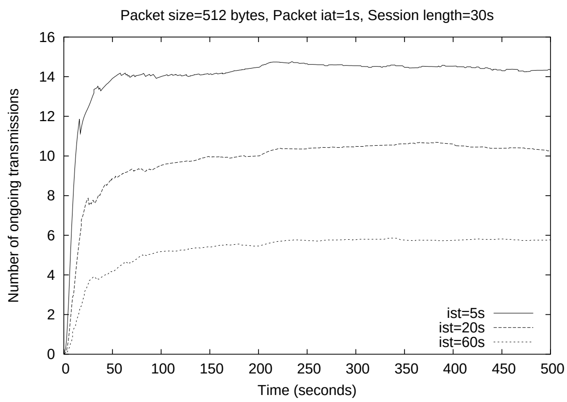

Figure 5: Transient of Traffic Model for 20 Nodes

We have observed that in most cases, a small impact of this transient on network metrics such as packet end-to-end delay, packet delivery ratio (PDR), and the ratio of routing control packets to data packets. In few cases, such as when we measured the average relative error in the number of control packets sent by AODV for each application data packet sent, this metric rose disproportionately to as much as 14%. This observation is sufficient to warrant the suspicion that for certain parameters in mobility, or for various number of nodes, the transient in traffic has the potential to produce more substantial adverse effects.

The most practical solution is to instrument the simulation to avoid collecting data on networking metrics until one can be sure that the traffic transient has abated. Fortunately, in all our observations, this transient is on the order of a few tens of simulated seconds. Even if one overestimates this time for the sake of safety, the length of traffic transient should be small compared to the average length of most simulations. The price paid in terms of the compute time used to warm up the networking models will certainly pay off in the accuracy of the results produced.

We have found it useful in our implementation of SWAN, to postulate that all collected metrics be sub-classed from a statistics base class rather than defined as basic data types such as integer or floating point. This allows us to offer the user a single point to define when data collection should start (this time value is defined the description of the simulation scenario). In addition to defining a settling time for network protocol model transients, the user also specifies the length of the warm up period for the mobility sub-model. Effectively, the collection of statistics begins at a time computed as the sum of the time to warm up the mobility model and the time to warm up the networking models.

3 THE PROTOCOL STACK MODEL

Although common sense dictates that any simulation study of networking protocols should state clearly all the sub-models it uses, this practice is not widely embraced. The literature has numerous cases which leave room for doubt as to what protocols are modeled in the stack of a wireless node. In order for one to *replicate* or *build upon* published simulation studies, it is essential that their readers can unmistakably identify not only all the sub-models used, but also the values of any parameters that can affect their behaviors.

In the course of the development of our own simulator for wireless network (SWAN), we have attempted to use previously published works to verify the correctness of some of our code. Much to our dismay, we quickly discovered that this was a hopeless venture. Most of the studies we found in the literature didn't completely document their experimental scenarios. Invariably, either the value for important parameters was discovered to be missing or the structure of the model was incompletely specified.

In this section we investigate how details in the composition of the protocol stack, might affect the results of a simulation experiment. The example we have used in this study is the Address Resolution Protocol (ARP), which as mentioned in earlier works (Broch et al. 1998, Johnson 1999), produces interactions with non-negligible effects in the simulation of a wireless network.

ARP works in conjunction with the MAC layer to constructs mappings from the IP addresses used by higher protocol layers to the MAC addresses assigned to networking hardware devices. To accomplish this goal, ARP broadcasts queries asking "who has IP address i?". The query is replied by the appropriate node stating that "device m has address i", where m is a MAC address. The replies are kept in ARP's own cache; entries in this buffer are removed if not referenced again after a maximum time. Packets destined to nodes for which an ARP query is in progress are made to wait in a buffer until the query is resolved. ARP buffers only one outstanding packet for each IP address i that is waiting for a query resolution. If more packets are presented for this same address i, they are dropped by ARP.

From this brief description, it is easy to see how ARP can affect network metrics such PDR and packet end-to-end delay. Since few papers explicitly state whether the protocol stack for their wireless node models includes ARP, we have deemed important to quantify its impact in the simulation. Intuitively, the interactions of ARP with other

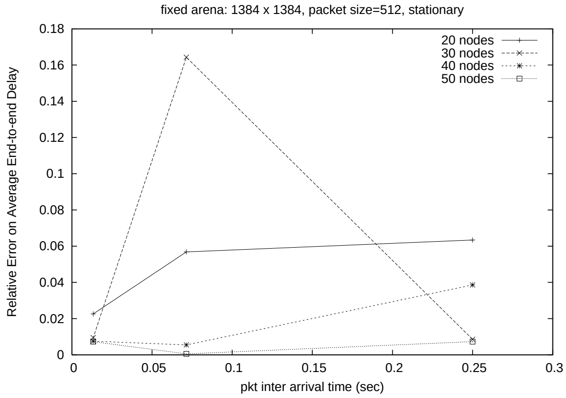

Figure 6: Effect of ARP on End-to-End Delay in a Stationary Network

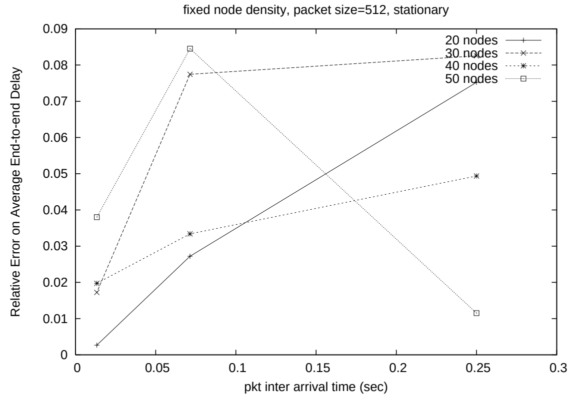

Figure 7: Effect of ARP on End-to-End Delay in a Highly-Mobile Network

protocols seem significant, but we wanted to have a sense of how network metrics would be affected.

After simulating our baseline experiments, we removed ARP from the protocol stack and reran the same experiments with the modified setup. We compared the networking metrics estimated in the two scenarios and found evidence that ARP does indeed affect the operation of the network by introducing an average relative error in end-to-end delays as high as 16% for low and medium traffic loads on mobile networks. As shown in Figure 6, in stationary networks, however, this metric was consistently less than 6%, except for medium traffic load with 30 nodes. Further investigating this data point, we discovered that although the point estimates used to compute the relative error did indeed exhibit the difference quantified, the 95% confidence intervals computed were statistically very close. This led us to believe the higher relative error is likely to have arisen from artifacts in floating point computations.

It is important to note also that the impact of ARP on packet end-to-end delay is much less pronounced at high traffic since the protocol's cache entries tend to survive longer thus reducing the number of queries for the same IP address. Very similar conclusions were drawn from the scenario depicted in Figure 7, which shows the average relative error in end-to-end delay for a highly mobile network.

The impact of ARP on PDR indicated by our experiments was, on the other hand, markedly small. For a stationary network, the worst observation in average relative error was slightly higher than 1%, for high traffic and 50 nodes. This behavior repeated itself for networks with low and high mobility, where the average relative error went higher to nearly 1.6%. Since the scale of our experiments in terms of number of nodes was small, we cannot state that this metric will rise when the network is scaled up. Further experiments are required to evaluate this hypothesis.

A final point we discuss is the effect of the ARP model on the simulation execution time. Rather than look directly at this metric, which is dependent on the architecture of the computing platform, we chose to observe the difference in the number of events processed with and without ARP. We observed that the use of this protocol did not necessarily increase the number of events in the simulation. Depending on the traffic load and the number of mobile nodes, the difference in events processed may be slightly higher or slightly lower when compared to the simulation of a model without ARP. These effects are due to the fact that in certain settings, ARP will cause slightly more packets (ARP queries) to be injected on the network, and in other settings, ARP will cause data packets to be dropped. These fluctuations in the number of packets pushed around in a simulation have a direct connection with the number of events simulated.

The conclusion that can be drawn from evidence produced in this case study is that ARP does indeed have a marked effect in the protocol stack model for a wireless network. Since the ARP model contributes with only a small computational load to the simulation, there is no drawback to its use other than the small added memory usage to store the model's code.

4 THE LIMITED INTERFERENCE MODEL

Our final case study in this paper regards the model for radio interference in a wireless network simulation. Given the different ways in which each specific simulator may compute radio interference, it is important to know exactly what model drives this computation. The work by Takai, Martin, and Bagrodia (2001) has shown the substantial impact this model has in determining the accuracy of the simulation's results.

The assessment of the strength of interference on a wireless node, however, comes at a high price in terms of computation. The total amount of interference on a node is the summation of all signals that can be picked up at its location which come from a source other than the sender of information. When the number of nodes in a wireless network model grows, not only do the number of terms in this summation grow fast, but also does the number of times the summation has to be computed. Clearly, without any measure to restrain the increase in the complexity of these computations, the scalability of the simulator can be severely impaired.

A common solution to reduce the computational complexity of interference calculations in wireless networking models is to limit the propagation range of interfering signals. In practice, this amounts to defining a *cutoff* value for radio signal propagation. The basic idea is that since interference is computed as the summation of all the "other" signals in a channel, sufficiently small terms in this summation could be discarded without substantially compromising the accuracy of the calculation. The crucial question here lies in determining how faint a signal should be so that it can be discarded from an interference computation without inducing substantial errors. This idea has been studied, in the context of wireless cellular networks, by Liljenstam and Ayani (1998) and by Perrone and Nicol (2000). The error induced by the application of this technique in simulations of IEEE 802.11b channels hadn't been quantified until now.

If a simulator should offer a cutoff parameter in the description of the experimental scenario, one should understand what consequences a chosen value brings. SWAN, our own simulator, offers this modeling option, and thus we felt compelled to investigate this question. This parameter can be interpreted in two different ways. It can be read as the maximum distance between transmitter and receiver that guarantees that the received signal is intelligible. Alternatively, cutoff can be defined as the highest attenuation (or path loss) that a signal may suffer and still be received (measured in decibels). We have taken the latter approach and require the user to enter this value in the configuration of the simulation scenario. Using a function provided by the underlying radio propagation model, the simulator converts this attenuation value to a distance value. Since different radio propagation models determine very different attenuations for the same transmitter-receiver separation, we believe this is the most general and practical solution.

Note that the importance of a cutoff parameter extends beyond just determining the complexity of the interference computation. This parameter is used in the construction of a connectivity graph for the network, which determines what radio links exist between nodes. When a node sends out a radio frame the connectivity graph is inspected to that the simulator knows to what other nodes deliver the information. If the network nodes are mobile, this connectivity graph is updated periodically.

Next, we look at how the limited interference model affected network metrics in experiments with two different values of maximum path loss: 126dB and 106dB. These values correspond, respectively, to 2120m and 670m, when

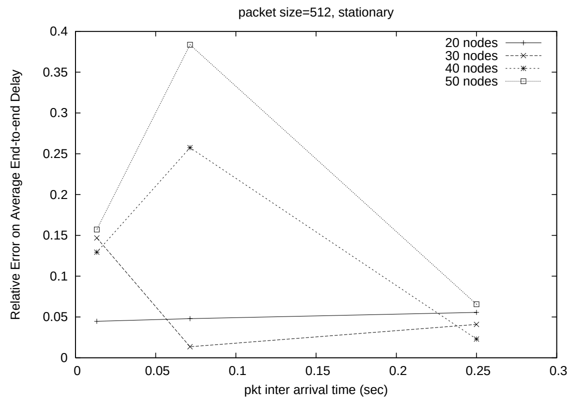

Figure 8: Impact of the Limited Interference Model on End-to-End Delay in a Stationary Network

the propagation model is 2-ray ground reflection, antenna height is 1.5m and transmitter power 15dBm. The experiments we report consider a scenario where the dimensions of the simulation arena are changed according to the number of nodes to maintain a constant node density of 7 neighbors per node. It must be noted that setting the cutoff parameter to 126dB effectively puts all nodes within radio range of each other, for the numbers of nodes we've considered.

Figure 8 shows the impact on end-to-end delay in a stationary network varying the number of nodes up to 50. The relative errors computed compare the two settings for the cutoff parameters assuming that a cutoff of 126dB is nearly the same as having no cutoff at all. This plot shows that at low traffic, the error incurred with the cutoff model is small, but it increases with the number of nodes. As traffic increases, the error becomes much more significant. For medium traffic, that is, 56kbps per node, the spread in relative error for different numbers of nodes is the highest. With traffic at its highest rate of 300kbps, this spread for 30, 40, and 50 nodes diminishes. This surprising observation has lead us to look at the statistics more carefully.

In Figure 9 we arbitrarily chose to isolate the curve for 40 nodes showing 95% confidence intervals on the difference between measures of end-to-end delay for the two cutoff settings. The first point to observe is that the width of the confidence interval increases with traffic load. Since this plot corresponds to end-to-end delay, the same argument made earlier regarding the reduced number of samples for this metric as traffic increases applies here. It should also be observed that the difference in end-to-end delay estimates becomes negative as a result of the reduction of the interference range. With cutoff at 106dB, the simulation yields less contention for the medium thereby reducing packet end-to-end delay. Clearly, these results motivate continuing this investigation. It would be especially important to develop a feature in the experimental framework to drive the simulation to continue until enough samples

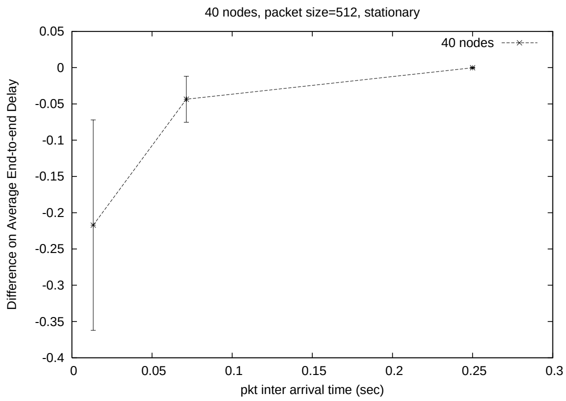

Figure 9: Detail on End-to-End Delay Interval Estimation for 40-Node Stationary Network

of a metric have been taken to yield narrower confidence intervals.

We have observed similar curves for end-to-end delay and also PDR for other mobility scenarios with the limited interference method and have concluded that the errors introduced by the application of the limited interference model in ad hoc networks are comparable to what was previously reported for cellular networks. Considering that the statistical difference in the results is acceptable, the use of the limited interference model is justified by the substantial reduction in the number of events processed in the simulation. Our experiments have shown that this difference can represent savings of as much as 55% in the number of events processed and can substantially enhance the scalability of the model.

5 CONCLUSIONS

The results presented in this paper should motivate efforts in thorough descriptions of the simulation scenarios used in studies of wireless networks. Our investigation has reiterated the importance of detail in these simulation models.

Countless experimental studies already published cannot be replicated because they do not fully report the conditions in which they were carried out. We have shown that it is not enough to simply collect data. One should mind the possible existence of transients in sub-models, determine when it is safe to collect data, and state the value of settling time used for data deletion. We have also shown that it is important to present detailed statistics so that correct conclusions can be drawn.

We are currently extending in breadth and in depth the investigation reported in this paper. First, we are attempting to better explain the observations made in our experiments through investigating other network metrics in these simulations. Second, we acknowledge that the level of detail in other sub-models may impact both accuracy and simulation scalability (in terms of run time and memory occupation). Finally, the need for a sensitivity analysis of sub-models to their various parameters has become evident.

In this work, we have simulated a very large number of possible scenarios. Generating experimental scenarios that were meaningful and consistent across different sub-models required expertise and knowledge in wireless network modeling. This instilled in us the concern that a user with less expertise would find daunting the task of determining the scenario for his or her experiments. This situation is especially troublesome when the number of parameters in the simulator is large. It would be helpful to devise tools and methodologies based on knowledge collected from experts to guide the user through the combinations in parameter space to the desired settings. We are currently starting an investigation in this problem area.

We end this paper with a word of caution to the users of currently available simulators for wireless networks who use them as experimental testbeds for their research. One should be aware of what model parameters have default values hidden within the simulator. First, these defaults may not set the exact scenario the experimenter has in mind. Second, for the sake of disseminating details of a study and for the sake of reproducibility, when publishing a study, the values for these parameters should be made clear. A study which simply states that for a certain simulator, the default parameters have been used does not provide all the information it should.

ACKNOWLEDGMENTS

This work was developed in its entirety at the Institute for Security Technology Studies (ISTS) at Dartmouth College.

The authors would like to thank Xiaowen “Jason” Liu and Michael Liljenstam for participating in insightful discussions that have helped in this work.

Thanks are also due to Noah H. Miller, from Dartmouth College, who was responsible for integrating the Scripts for Organizing ’Spiriments (SOS) package with SWAN. His work gave us the ability to automate our exploration of the parameter space in our simulations and to organize the large amounts of data we produced.

This research is supported in part by DARPA Contract N66001-96-C-8530, NSF Grant ANI-98 08964, NSF Grant EIA-98-02068, and Dept. of Justice contract 2000-CX-K001. Points of view in this document are those of the authors and do not necessarily represent the official position of the United States Department of Justice.

REFERENCES

Bettstetter, C., H. Hartenstein, and X. Pérez-Costa. 2002. Mobility, modeling, and management: Stochastic properties of the random waypoint mobility model: epoch length, direction distribution, and cell change rate. In *Proc. 5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM.

Broch, J., D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. 1998. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. Fourth Annual International Conference on Mobile Computing and Networking (ACM MobiCom'98)*.

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

C.Ware, T.Wysocki, and J.F.Chicharo. 2001. Modelling capture behaviour in 802.11 modems. In *IEEE International Conference on Telecommunications (ICT2001)*.

Heidemann, J., N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. 2001. Effects of detail in wireless network simulation. In *Proc. SCS Multiconference on Distributed Simulation*, 3–11.

Johnson, D. B. 1999. Validation of wireless and mobile network models and simulation. In *Proc. DARPA/NIST Workshop on Validation of Large-Scale Network Models and Simulation*.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd ed. McGraw-Hill Higher Education.

Liljenstam, M., and R. Ayani. 1998. Interference radius in pcs radio resource management simulations. In *Proc. 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Mannivannan, 1629–1637.

Liu, J., L. F. Perrone, D. M. Nicol, M. Liljenstam, D. Pearson, and C. Elliott. 2001. Simulator for wireless ad hoc networks. In *Proc. 2001 European Simulation Interoperability Workshop (EURO-SIW)*.

Navidi, W., and T. Camp. 2003. Stationary distributions for the random waypoint mobility model. Technical Report MCS-03-04, Dept. Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO.

Pawlikowski, K., H.-D. J. Jeong, and J.-S. R. Lee. 2002. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine* 40 (1): 132–139.

Perrone, L. F., and D. M. Nicol. 2000. Using n-body algorithms for interference computation in wireless cellular simulations. In *Proc. Eight International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*.

Takai, M., J. Martin, and R. Bagrodia. 2001. Effects of wireless physical layer modeling in mobile ad hoc networks. In *Proc. 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*.

Yoon, J., M. Liu, and B. Noble. 2003. Random waypoint considered harmful. In *Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*.

AUTHOR BIOGRAPHIES

LUIZ FELIPE PERRONE is Assistant Professor of Computer Science at Bucknell University. From 2000 to 2003 he was a post-doctoral research associate at the Institute for Security Technology Studies (ISTS) at Dartmouth College. He holds the degrees of Electrical Engineer and M.Sc. in Systems Engineering and Computer Science from the Federal University of Rio de Janeiro, in Brazil, and PhD in Computer Science from the College of William & Mary. His research interests include modeling and simulation of wireless systems, network security and parallel discrete-event simulation. He has served as vice-program chair of *The 9th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2001)*. He is a member of the IEEE Computer Society. His e-mail address is `<lfperrone@ieee.org>`.

YOUGU YUAN is a PhD student with the Dept. of Computer Science at Dartmouth College. He has a B.S. degree from Peking University, in China, and his adviser is Professor David Nicol. He has been one of the developers of SSFNet, a simulator for large scale networks. He is also one of the main authors of the Simulator for Wireless Ad-Hoc Networks (SWAN). His main reasearch interests lie in the intersection of simulation and wireless technologies. His e-mail address is `<Yougu.Yuan@dartmouth.edu>`.

DAVID M. NICOL is Professor of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign, and member of the Coordinated Sciences Laboratory. He is co-author of the textbook *Discrete-Event Systems Simulation*, and served as Editor-in-Chief at ACM TOMACS from 1997-2003. He will serve as the General Chair of the Winter Simulation Conference in 2006. From 1996-2003 he was Professor of Computer Science at Dartmouth College, where he served as department chair, and at the Institute for Security Technology Studies served as Associate Director for Research and Development, and finally as Acting Director. From 1987-1996 he was on the faculty of the Computer Science department at the College of William & Mary; 1985-1987 he was a staff scientist at the Institute for Computer Applications in Science and Engineering. He has a B.A. in mathematics from Carleton College (1979), an M.S. (1983) and PhD (1985) in computer science from the University of Virginia. His research interests are in high performance computing, performance analysis, simulation and modeling, and network security. He is a Fellow of the IEEE.