# Effects of Detail in Wireless Network Simulation[*]

Nirupama Bulusu, Jeremy Elson, Deborah Estrin,
Ramesh Govindan, John Heidemann, Chalermek Intanagonwiwat,
Nader Salehi, Kun-chan Lan, Ya Xu, Wei Ye

*USC/ISI TR-2000-523*

## Abstract

Experience with wired networks has provides guidance about what level of detail is appropriate for simulation-based protocol studies. Wireless simulations raise many new questions about approriate levels of detail in simulation models for radio propagation and energy consumption. This paper describes the trade-offs in more detailed or abstract simulation models. We evaluate the effects of detail in four case studies of wireless simulations for protocol design. Ultimately the researcher must judge what level of detail is required for a given question, but we suggest two approaches to cope with varying levels of detail. When error is not correlated, networking algorithms that are robust to a range of errors are often stressed in similar ways by random error as by detailed models. We also suggest visualization techniques that can help pinpoint incorrect details and manage detail overload.

## 1 Introduction

Selecting the correct level of detail (or level of abstraction) for a simulation is a difficult problem. Adding detail requires time to implement, debug, and later change, it slows down simulation, and it can distract from the research problem at hand, but too little detail can produce simulations that are misleading or incorrect. Designing simulations to study a protocol inherently involves making choices in which protocol details to implement or use.

Although a number of network simulation packages are available, they do not remove this burden from the designer. In custom simulators, researchers typically include only the minimum possible details outside the immediate area of study. Existing simulators (such as ns-2 [3], Parsec [2], and S3 [7]) provide detailed protocol implementations, but what level of detail is required in new protocols, or in adapting existing protocols to model new hardware? Some simulators ease the cost of changing abstraction with multiple, selectable levels of detail (for example, ns [16]), but the design choice must still be made.

Choices about detail are particularly difficult for wireless network simulations. Wide experience with the important components of wired networks over the last 30 years allows significant abstraction. For example, point-to-point links are often represented as a simply by bandwidth and delay with a queue; framing, coding, and transmission errors are simply ignored or mathematically modeled. The younger field of wireless networking provides less guidance on what abstractions are appropriate. Low-level details can have a large effect on performance, but detailed simulations can be very expensive (for example, radio propagation).

This paper explores the question of what level of detail is needed for simulations of network protocols in wireless domains. We begin by looking at the trade-offs in different levels of detail in simulations. We then consider four case studies: energy consumption in ad hoc routing, radio-based outdoor localization, communications-driven robot following algorithms, and visualization of wireless simulations. These studies illustrate times when we have been misled by too much or too little detail in our models, and they have led to two approaches that allow simulation to tolerate ranges of detail.

---

[*] This research is supported by the Defense Advanced Research Projects Agency (DARPA) through the VINT project under DARPA contract ABT63-96-C-0054, the SCADDS project under DARPA contract DABT63-99-1-0011, and the SCOWR project through NSF grant ANI-9979457.

For questions concering this paper, please contact the authors in care of John Heidemann, `johnh@isi.edu`.

## 2 Trade-offs of Detail in Wireless Simulation

We next consider the trade-offs of more detailed or abstract simulations.

A common goal is to infuse the simulation with as much detail as possible to provide a "realistic" simulation. This approach is attractive: a fully realistic simulation ought to be able to reproduce the results of laboratory experiments or network use by end-users. Failing to implement details guarantees that they won't be reflected in a simulation; for example a wireless propagation model that doesn't consider concurrent transmissions will not model the hidden terminal effect. Furthermore, details at multiple protocol levels can reveal important interactions between layers. For example, router synchronization was first studied in simulation [13].

Yet a "fully realistic" simulation is not possible—does one stop at the network layer? they physical layer? electrons or optics? Simulation designers must limit the level of detail somewhere. The challenge is to identify what level of detail does not affect answers to the design questions at hand. For example, we know of no network simulator that considers details of a CPU's instruction set or memory hierarchy—these do not affect design questions relevant to wireless simulations (although they can be critical to the design of very large routers [10]). Balancing the tendency to additional detail are several penalties: Simulation run-time is adversely affected by detail. Implementation and debugging time is increased, and undetected bugs in distant layers can produce inaccuracies. Even if debugged, protocol details change over time. For example, an extremely detailed implementation of WaveLAN from a few years ago would today be superceded by the 802.11 standard today. Sometimes getting all the details may be impossible, either because they are left unspecified, as are some of the parameters of 802.11[1], or when trying to predict future behavior with protocols not yet implemented or standardized. Finally, for many of these reasons simulations often mix levels of detail in different components. A very detailed, microsecond-level MAC simulation may be forced to use a more abstract propagation model (because all objects in the terrain were not specified) and an older TCP implementation (perhaps not including SACK or recently standardized extensions). Simulations with detailed hardware models may have abstract (perhaps randomized) scenarios of node placement, transmission, and movement.

There are several reasons for intentionally choosing a high level of abstraction for simulation. Distillation of a research question to its essence can provide insight not colored by arbitrary details of specific proposed solutions. For example, although multiple resource reservation and quality-of-service protocols have been proposed, Breslau and Shenker use a very abstract service model to focus on the central issue of the benefits of reservations [5]. When exploring a new area where many issues are unclear, the need to quickly explore a variety of alternatives can be more important than a detailed result for a specific scenario. For this kind of *nimble* simulation, relative comparisons of alternatives are often more important than a single detailed quantitative result. A more abstract simulation can also make the effects of a change in algorithm distinct, where they would be obscured by other effects in a more detailed simulation. Finally, omission of simulation detail can improve performance by multiple orders of magnitude [16]. Memory and run-time improvements due can offer results sooner, or allow larger or longer experiments, revealing different aspects of protocol behavior. For example, the relative performance of ad hoc routing protocols differs at higher scales [9].

The primary risk of simulation abstraction is the unknown. Would additional detail change the conclusions of the simulation study? This problem is particularly challenging when entering a relatively unexplored area where researcher's intuitions may be underdeveloped. Validation of simulations against more detailed simulations and experimental measurements can answer this question. But the cost of validation is fairly high: careful experiments require implementing the details in question or purchasing sufficient hardware for real-world experiments.

Over time, the results of validation experiments will allow the community to build an understanding of what details are important. The community has begun sharing this information through workshops such as the DARPA/NIST Network Simulation Validation Workshop [8]. We next consider several case studies that have arisen in our research as further examples.

## 3 Energy Consumption in Ad Hoc Routing

Our first case study considers energy consumption when routing data in ad hoc networks. We examine two recent studies in this area: an evaluation of data diffusion [17], and a study of an energy-saving variations of on-demand ad hoc routing protocols [21]. Choice of ap-

---

[1] David B. Johnson, personal communication.

propriate models of radio energy consumption and MAC protocols make can completely change the conclusions of these studies.

Several models of energy consumption for wireless communication have been used in literature:

- Successfully sent or received packets incur an energy cost.
- MAC-level costs can be considered—MAC-level retransmissions, CTS/RTS, and packets that are unsuccessfully sent or received incur a cost.
- Listening (having the radio powered on) can also be modeled.
- Non-radio system costs can be considered (display, CPU, disk drive).
- Battery internals (non-linearity, temperature sensitivity, battery memory, etc.) can be considered.

Selecting the right level of detail depends on the research question being considered. For most research questions about networking protocols, non-radio components (for example, the display) can be factored out as a fixed overhead, although in some cases CPU-intensive work must be considered (for example, software radios [4], or MPEG playout). Similarly, for rough comparisons of protocols, detailed battery models are not required—a reasonable simplifying assumption is that memory or temperature will affect all protocols equally.

We have found modeling idle time makes a large difference in protocol comparisons. We studied energy consumption of four ad hoc routing protocols (AODV, DSR, DSDV, and TORA) with a simple traffic model where a few nodes send data over a multi-hop path [21]. Using a simple energy model that does not consider idle-time costs, we found that on-demand protocols such as AODV and DSR consume much less energy than a priori protocols such as DSDV and TORA/IMEP. A priori protocols are constantly expending energy pre-computing routes, while nodes that do not source data do not use these routes. These differences vanish, however, when we adopt a more detailed energy model that considers idle-time energy consumption. WaveLAN radios have a 1:1.05:1.4 ratio of idle:receive:send energy costs [20]. With this radio model all ad hoc routing protocols considered consume roughly the same amount of energy (within a few percent). In this scenario, idle time completely dominates system energy consumption, so an insufficiently detailed energy model (not considering idle time) completely changes the study results.

Choice of MAC protocol is also closely tied with radio energy consumption. We have studied data diffusion protocols, evaluating the power consumption of data diffusion as compared to simple flooding and an idealized multicast [17]. The goal of these experiments was to provide energy-conserving protocols for long-lived sensor networks. Again we had trouble with inappropriate models of radio energy consumption; all protocols behaved similarly when idle costs were considered. In this case, the problem was an inappropriate MAC protocol. Wireless networks designed for very long will chose energy-optimized MAC protocols such as TDMA [19], not the 802.11 protocol we began with. This example illustrates problems from incorrect detail in the MAC protocol—our choice of the off-the-shelf protocol already in our simulator was inappropriate.

These examples suggest that idle-time and MAC protocols are important details for wireless communication studies with PC-like network nodes. We have not seen evidence that further details (power consumption of other system components or models of battery internals) alter research results in this domain. Additional experience is needed to validate this assumption. These assumptions may not hold for studies of increasingly tiny (dust-mote-sized) nodes [18]. We hypothesize that as node and radio power consumption shrinks, and as node lifetime increases, additional details will become important.

# 4 Radio Propagation Models

Our next two studies consider the problems of radio-based localization (determining a node's location) and robot following. In both cases, we found the level of detail of the radio propagation model important.

Even more than energy models, many levels of detail are employed in radio propagation models with a single sender and receiver:

- The simplest models consider only propagation distance from sender to receiver with a fixed formula for signal loss.
- Slightly more detailed models might use different models for near and far receivers (for example, the Friss and two-ray ground reflection approximations).
- A more detailed model might consider signal attenuation from large obstacles.

- More detailed models might model antenna geometries (orientation, distance off ground) and perform detailed radio ray-tracing to estimate reflection.

In addition, models may or may not take in the relative power of interfering transmissions.

Radio propagation varies greatly, especially indoors, motiving very detailed propagation models. Unfortunately, accurate models become very computationally expensive and require much more detail about the environment than is typically available.

An attractive alternative is to couple a simple model with some level of statistical loss, but there has been limited experience with how less detailed models change network behavior. We have evaluated this question in two case studies, one where a very simple model proved surprisingly effective in a restricted domain, and then a robotics-inspired approach to designing software to be robust to model error.

## 4.1 Radio-based outdoor localization

Sometimes simple radio propagation models can be quite effective for the purposes of a problem. We are exploring the task of *spatial localization*, determining a node's approximate location, using only radio connectivity to a set of beacons with well known locations [6]. This approach would be important for nodes too small or inexpensive to use GPS.

Radio propagation is a critical aspect of this kind of network-based localization. We began this work using a simple, idealized radio model—we assume each radio has an identical, spherical propagation. We selected this model because it was simple to reason about and evaluate mathematically. We expected that this model, at best, would allow us to select algorithms and establish performance bounds. To our surprise, it compares quite well to experimentally measured propagation in open, outdoor areas. Not so unsurprisingly, it does not model indoor propagation well at all.

We evaluate the effectiveness of this model both by comparing its accuracy to experimental measurements and then by considering its effect on our estimates of localization accuracy. First, to compare its accuracy to measurements, we evaluated propagation between two Radiometrix radio packet controllers (model RPC-418) operating at 418 MHz. A node periodically sent 27-byte beacons; we define a 90% packet reception rate as "connected" and empirically measured an 8.94m spherical range for our simple model. To evaluate how well this simple model compares to a real-world scenario we placed a radio in the corner of an empty parking lot then measured connectivity at 1m intervals over a 10m square quadrant. Figure 1 compares these measurements with connectivity as predicted by the model. Among the 78 points measured, the simple spherical model matches correctly at 68 points and mismatches at 10, all at the edge of the range. Error was never more than 2m.

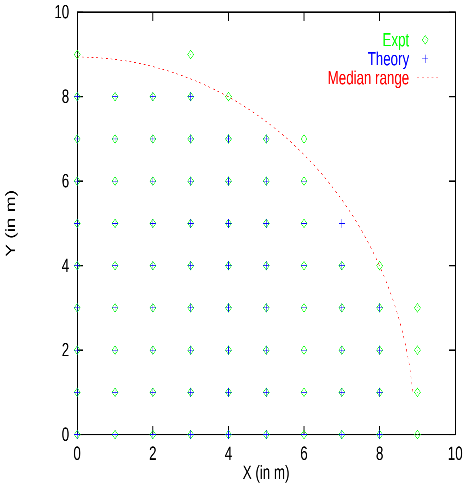

Figure 1: 90% radio connectivity for a transmitter at (0,0)

Although we have evaluated the accuracy of our radio model, a more important metric is the influence that model has on the accuracy of localization and our evaluation of alternative localization algorithms. We evaluated our network localization algorithms by placing beacons at the corners of a 10m square in an outdoor parking lot. We then estimated a node's position at 1m intervals within this square both experimentally and using our spherical model. Localization algorithms typically evaluate the error between predicted and actual position. Figure 2 shows this metric from the model and experiment. They track each other closely, including plateaus is the error levels, although spherical model is consistently slightly optimistic.

From these experiments we conclude that very simple propagation models can be effective when simulating protocols in restricted domains. We caution that this approximation is not appropriate for indoors (as would be expected) where reflection and occlusion is common. Our indoors measurements of propagation range varied widely from 4.6–22.3m depending on walls and exact node locations and orientations. The validated outdoor

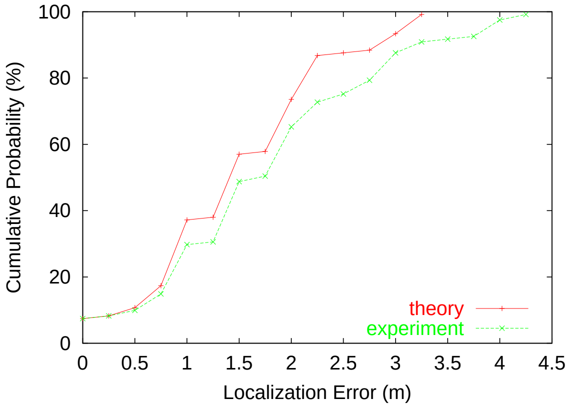

Figure 2: A comparison of localization error with spherical and experimental propagation.

model allows us to explore a much wider range of scenarios through simulation than could be done through physical experimentation.

## 4.2 Radio-based robot following

Robotics is a domain filled with error. Autonomous robots interacts with the real world through noisy sensors and inaccurate actuators. To accommodate the many sources of environmental error, roboticists design very robust algorithms. Instead of trying to develop very detailed models the physics of robot movement, robotics simulators instead introduce large amounts of random error. We believe this philosophy is also applicable in networking: networking algorithms must be robust to network dynamics; robust algorithms can often allow random error to replace detailed models in simulation. (When error is not correlated.)

We evaluate this hypothesis in a hybrid scenario: we have designed and simulated an algorithm to get one robot to follow another at constant distance [11]. The lead robot circles a large rectangular corridor while emitting periodic radio beacons. The follower adjusts its speed to keep a constant distance with the leader. The follower listens to beacon messages and increases speed when the loss rate is high and decreases it when loss rate is low. This algorithm assumes a short-range radio where loss rate corresponds to distance. Figure 3 shows an idealized radio propagation model.

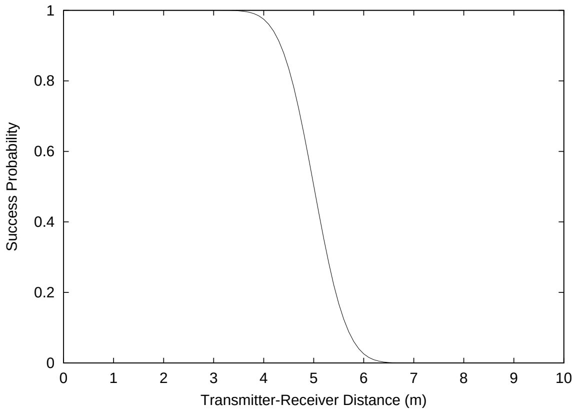

Figure 3: Idealized radio propagation model with a nominal transmission radius of 5m.

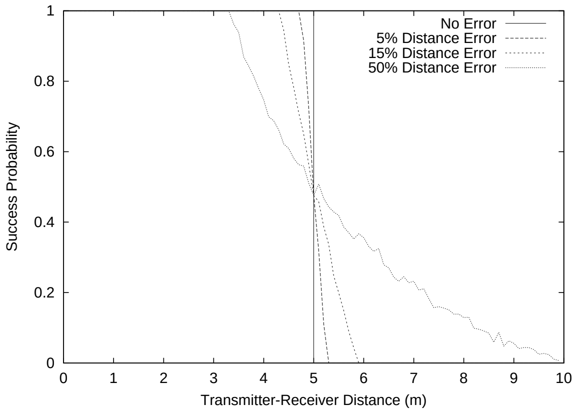

Figure 4: The "$(r + \text{percent-error})^2$" propagation model used for simulation.

Indoor radio propagation is much less than ideal due to multipath reflections. To investigate these effects without extremely detailed models of the interior of our building, we add a random error component based on an "$(r + \text{percent-error})^2$" model. With this model, a packet is always received by nodes within radius $r$, but we add a random error to this radius before thresholding. This error is uniformly chosen within some percentage of actual distance; for example, at 25% error, $r' = r + .25ru$ where $u$ is a random number between $-1$ and 1. Figure 4 shows our adjusted propagation model at 0, 5, 15, and 50% error levels. Note that 0% error is actually better than our idealized propagation model.

We evaluate the quality of distance keeping with each of these error models in Figure 5. We were surprised that distance keeping performance is essentially the same for all propagation models. This argues that, for this experiment, additional detail in the propagation model would not offer additional insight into the tracking algorithm. This result is independent of the underlying model for

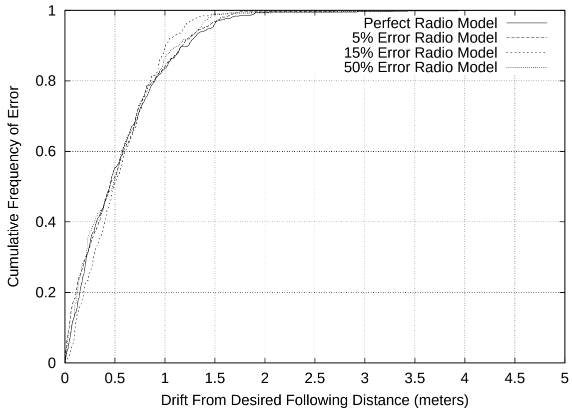

Figure 5: Cumulative distribution of error in following distance for the four radio models.

two reasons. First, the algorithm is robust to error; its decisions are simple and return it to steady distance. Second, our expectations in evaluating this algorithm allow error; a following distance within a meter of expected 90% of the time is good.

This experiment suggests that that qualitative evaluations of this class of robust algorithms can tolerate abstract models of underlying layers. We would like to further verify this claim by repeating this experiment with physical robots.

This result is not specific to robotics; we have observed similar results in experiments involving wired networks and the SRM protocol [14]. SRM has the same properties as our robot-following algorithm: it uses randomized algorithms to repair lost messages, and it can be evaluated by counting numbers of duplicate repair messages. We have found that the number of duplicate repairs is similar both with detailed hop-by-hop network simulations and with abstract simulations that simulate only end-to-end delay [16].

## 5 Visualization of Wireless Simulations

Finally, we consider the effect of details in visualization. We have developed *nam* as a generic tool for visualizing the output of network simulations [12]. We find visualization a very important tool for protocol debugging, but there is need to control the amount of detail presented to the user. In this suggestion we examine ways we use visualization to control details, and ways that visualization is helpful at selecting the right level of detail for wireless simulation.

Easy-to-use visualization alone provides a huge step providing a large amount of detailed information in a manageable fashion. Visual representations of packet flow succinctly capture high-level information about traffic rates, congestion, sources and destinations, and interactions for many nodes and links. Determining the same information from textual packet traces for a single node or link is much more difficult. Once hot spots or problem areas are visually identified, traces can be examined to extract specific information. We strongly encourage simulation authors to visualize their protocols early in development to aid debugging, and the use of a generic tool like nam can reduce this effort.

Recent work in data diffusion provides one example of the importance of visualization [17]. Our early experiments with data diffusion employed a very high traffic load (a large fraction of network capacity). This resulted in timeouts and anomalous behavior completely unrelated to the protocol we were studying simply because we were out of an acceptable operating region. This status would have quickly and easily been determined from a protocol visualization, but was lost in the statistics we considered.

Even with visualizations, the detail can become overwhelming. We are exploring two ways to control this detail in nam. First, we provide different kinds of visualization for different kinds of wireless communication. Second, we allow the user to control the level of detail nam presents.

Nam has two ways to visualize wireless communications. First, we can visualize packet flow as rectangles that are animated and move directly from the source to destination (the lines from node 1 to nodes 2 and 3 in Figure 6). This representation has evolved from nam's use to visualize wired point-to-point networks where packets flow on links. This approach clearly identifies the sender and receiver of the packet, the direction of packet flow, and the time of transmission and receipt. However, this visualization does not easily adapt to support broadcast traffic. Representing a broadcast packet as multiple rectangles visually suggests multiple packets. This approach also does not easily show when concurrent transmissions from different nodes interfere with each other.

An alternate visualization approach is to show wireless packets as expanding circles (the circles in Figure 6). This clearly shows the packet source and interference with other packets, but it does not show destinations. If the rings disappear or fade with distance, it also shows nominal radio range. Currently we use both

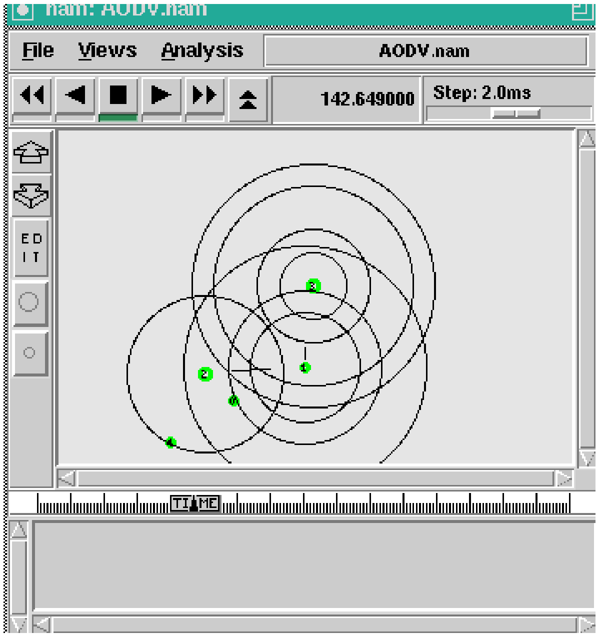

Figure 6: Wireless visualization in nam

approaches in nam: unicast packets are sent using rectangles, while broadcasts are sent with expanding circles.

In addition to choosing between two visualization methods, we allow the user to control the level of detail presented. We are adding support for both transport- and MAC-level trace collection in ns. Transport-level traces show packets traveling from sources to destinations; MAC-level traces add MAC-layer retransmits and losses. Users of nam can also select and filter data at run-time, focusing on data for a particular sender, receiver, flow, packet-type, or similar characteristics.

## 6 Related Work

The wired networking world has depended on years of experience to guide detail in networking simulations. Ahn et al. were the first to suggest explicitly using abstract representations of packet trains to speed simulation [1]. Huang et al. have examined the use of selective levels of detail or abstraction in wired multicast simulations, and demonstrated that abstraction causes minimal changes to SRM evaluations [16].

The difficulty of radio propagation has long forced the wireless networking community to multiple levels of detail. Recently the community has focused on the question of validation and levels of detail in wireless simulations at events such as the DARPA/NIST Network Simulation Validation Workshop [8, 15]. Johnson has compared simulations and experimental results for wireless ad hoc routing, showing that simulation can provide enough detail to model reality in this scenario.

## 7 Conclusions

Choosing the right level of detail for network simulation is difficult. Since the networking community has less experience in the wireless domain than with wired networks, choosing abstractions there is even more difficult.

There are risks both in simulating with too much detail or too little. Too much detail results in slow simulations and cumbersome simulators. A very detailed simulation may accurately predict today's performance, but it may not predict tomorrows protocol variations or be easily adapt to quickly explore alternatives. Simulations which lack necessary details can result in misleading or incorrect answers. Researchers must chose their level of simulation detail with care.

We have offered several case studies in wireless network simulation to offer guidance for when detail is or is not required. Even when examples are not directly applicable, similar validation approaches may be. We have also suggested two approaches to cope with varying levels of detail. When error is not correlated, networking algorithms that are robust to a range of errors are often stressed in similar ways by random error as by detailed models. Finally, visualization techniques can help pinpoint incorrect details and control detail overload.

## Acknowledgements

We would like to thank Gaurav Sukhatme and Maja Matarić for their involvement in the robotics aspects of this work. We would like to thank members of the SCADDS project and especially Lewis Girod for their contributions to the SCADDS testbed and our Radiometrix experiments. We would like to thank members of the VINT project for support of our simulation platform, and the CMU Monarch project and Sun's AODV group for their contributions to ns's wireless networking.

## References

[1] Jong-Suk Ahn, P.B. Danzig, D. Estrin, and B. Timmerman. Hybrid technique for simulating high bandwidth

delay computer networks. In *Proceedings of the ACM SIGMETRICS*, pages 260–261, Santa Clara, CA, USA, May 1993. ACM.

[2] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu-an Chen, Xian Zeng, Jay Martin, and Ha Yoon Song. PARSEC: A parallel simulation environment for complex systems. *IEEE Computer*, 31(10):77–85, October 1998.

[3] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving simulation for network research. Technical Report 99-702b, University of Southern California, March 1999. revised September 1999, to appear in IEEE Computer.

[4] Vanu Bose, David Wetherall, and John Guttag. Next century challenges: Radioactive networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, August 1999. ACM.

[5] Lee Breslau and Scott Shenker. Best-effort versus reservations: A simple comparative analysis. In *Proceedings of the ACM SIGCOMM*, pages 3–16, Vancouver, Canada, September 1998. ACM.

[6] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, xxx(xxx):to appear, August 2000.

[7] James H. Cowie, David M. Nicol, and Andy T. Ogielski. Modeling the global internet. *Computing in Science & Engineering*, pages 30–36, January 1999.

[8] DARPA/NIST. *DARPA/NIST Network Simulation Validation Workshop.* Fairfax and Virginia, USA, May 1999. Proceedings at `http://www.dyncorp-is.com/darpa/meetings/nist99may/index.html`.

[9] Samir Das, Charles Perkins, and Elizabeth Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Infocom*, pages 3–12, Tel Aviv, Israel, March 2000. IEEE.

[10] Mikael Degermark, Andrej Brodnik, Svante Carlsson, and Stephen Pink. Small forwarding tables for fast routing lookups. In *Proceedings of the ACM SIGCOMM*, pages 3–14, Cannes, France, September 1997. ACM.

[11] Jeremy Elson. Simulation of constant-distance robot following using radio proximity beacons. Available as `http://www.circlemud.org/~jelson/writings/constant-following.ps`, December 1999.

[12] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, and Haobo Yu. Network visualization with the VINT network animator nam. Technical Report 99-703b, University of Southern California, March 1999. revised November 1999, to appear in IEEE Computer.

[13] S. Floyd and V. Jacobson. The synchronization of periodic routing messages. *ACM/IEEE Transactions on Networking*, 2(2):122–136, April 1994.

[14] Sally Floyd and Van Jacobson. Link-sharing and resource management models for packet networks. *ACM/IEEE Transactions on Networking*, 3(4):365–386, August 1995.

[15] John Heidemann, Kevin Mills, and Sri Kumar. Expanding confidence in network simulation. Research Report 00-522, USC/Information Sciences Institute, April 2000. Submitted for publication, IEEE Computer.

[16] Polly Huang, Deborah Estrin, and John Heidemann. Enabling large-scale simulations: selective abstraction approach to the study of multicast protocols. In *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 241–248, Montreal, Canada, July 1998. IEEE.

[17] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. Submitted for publication, May 2000.

[18] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for “smart dust”. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, page to appear, Seattle, Washington, USA, August 1999. ACM.

[19] Gregory J. Pottie and William J. Kaiser. Embedding the internet: wireless integrated network sensors. *Communications of the ACM*, 43(5):51–xxx, May 2000.

[20] Mark Stemm and Randy H. Katz. Measuring and reducing energy consumption of network interfaces in handheld devices. *IEICE Transactions on Communications*, E80-B(8):1125–1131, August 1997.

[21] Ya Xu, John Heidemann, and Deborah Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Submitted for publication, May 2000.