An Investigation of Policies for Caching Radio Propagation Calculations in the Simulation of Wireless Networks

Michael Dippery

mdippery@bucknell.edu

Class of 2008
Box C0206

Luiz Felipe Perrone (Faculty Mentor)

perrone@bucknell.edu
Department of Computer Science

1 Project Description

Wireless networking is so ubiquitous today that it is almost surprising to walk into a public place like a café and *not* find a wireless network. Wireless networking has given us quick, convenient access to information that is unprecedented in human history.

The strength of wireless networking, however, lies in its ability to create so-called *ad-hoc* wireless networks. *Ad-hoc* wireless networks are created spontaneously by intelligent protocols in the network devices. The protocols assume that nodes are mobile and take care of reconfiguring the network so information can be routed to the appropriate destinations. This reconfiguration happens automatically, without the need for infrastructure or human intervention. This property of dynamic self-configuration makes the technology applicable to numerous scenarios, including surveillance, public safety, emergency response, and sensor networks.

Many applications of this technology require networks to scale up to hundreds, thousands, or even hundreds of thousands of components. The challenge in the development of a system of such a scale is that it requires physical experiments that are prohibitively costly and nearly impossible to control and replicate. Computer simulation of large wireless networks can overcome these difficulties, but also introduces challenges of its own as the time to conduct each individual experiment is substantial.

The focus of our research will be the investigation of a particular technique that can accelerate simulations of wireless networks. The Simulator of Wireless Ad-hoc Networks (SWAN), a project started at Dartmouth College and now carried on at Bucknell University, will serve as the platform on which we will construct and evaluate our techniques. In the simulations done with SWAN, a very frequent computation estimates the strength of the signal received by each network device. These are complex and heavy computations, and the values they produce depend upon the positions of the network nodes in the simulated virtual space.

We have observed that the positions of the nodes change very little between successive computations of signal strength because nodes move slowly compared to the speed at which the simulator recomputes received signal strength. Our work will attempt to exploit this characteristic by memorizing the results in a structure commonly called a *cache*. Once we have memorized one round, we can retrieve the values in the cache if we detect that the relative positions of a set of nodes have changed little over a period of time. This will prevent the simulator from recalculating *all* signal strengths, improving its efficiency: as long as the time to look up a result in the cache is smaller than the time needed to recompute the values from scratch, the simulation will be accelerated.

The potential success of this idea relies upon a few key issues that must be investigated. The first is that we can only reuse precomputed values when we can determine that there is little error in not doing the calculations over again. We intend to use the available knowledge of node movement speeds and trajectories to estimate the magnitude of these errors.

The second issue is that, for the cache memory to allow fast look-ups, it will need to be small. This means that we will not be able to store all computed values all the time. The ideal case would be to store only the most frequently needed signal strength values in the cache. Defining what this set of values will be is not easy, and we will investigate policies to determine this set dynamically as the simulation progresses.

My work will extend previous work done by a student under Professor Perrone's tutelage. However, my research will focus specifically on the caching issue, whereas the other student's research was broader in scope. That student was unable to complete work on the data cache; I plan to greatly expand on his work and finish the module.

Ultimately, my research will culminate in a more efficient data storage cache for the radio model component of the SWAN simulator, as described above. A better mechanism for storing and retrieving frequently-used data will greatly enhance the performance of the simulator, allowing large-scale networks with thousands of nodes to be modeled in an efficient manner. This will enhance the productivity of researchers working with the simulator. Since the source code for the simulator will be freely available, this will not only help Professor Perrone's efforts, but the efforts of other computer scientists engaged in research in similar areas.

2 Milestones

The most crucial aspect of my project is the investigation of existing simulation literature and analysis of theoretical aspects of the problem. It is likely that some investigation in this area has already been done, and the research and work of others might very well provide some clues and hints as to the best way to implement the data storage and retrieval mechanism in SWAN.

The next step will be getting acquainted with the SWAN software. Understanding SWAN will require examining the source code, as well as reading documentation and possibly papers about SWAN. I expect this step to take approximately one week.

Some time must then be sent aside for design, which can take a significant amount of time. A good design can make the actual coding go much more smoothly and save a lot of time down the road, so I plan on devoting an ample amount of time to this task—two weeks, likely. This will involve thinking about the goal of the project, and sketching and diagramming the product.

After that, the actual implementation—the coding—of the software data cache will begin. This will likely take the most significant chunk of time. Since this phase will also involve some testing along the way, I'd like to devote three weeks to this part of the project. The majority of this time will be spent writing and checking the code that will be added to the project.

Finally, testing and "tweaking" the new cache will round out the research. This in and of itself could take a fair amount of time, however, as the more strenuous testing procedures might reveal bugs or inefficient code that must be rewritten.

Weeks 1 – 2:	Investigate existing simulation literature and analyze the theoretical aspects of the problem
Week 3:	Familiarize myself with the design of the SWAN software by examining source code and documentation
Weeks 4 – 5:	Design the structure of the new software module
Weeks 6 – 8:	Write the code for the new software data cache module
Weeks 9 – 10:	Run the new module through rigorous testing and make changes and improvements as needed, as well as tie up other loose ends necessary to complete the implementation of the caching module

3 Research Environment

Professor Perrone and I will be in close contact for the duration of the research project. This project is part of Professor Perrone's larger scholarship agenda, and we will work in close cooperation. At the same time, I expect that I will spend most of my time working independently in this project, and will be able to use Professor Perrone's guidance as often as needed. In addition to such impromptu meetings, we plan to meet formally once a week for at least an hour.

All research will be done on-campus at Bucknell University. I will be using the Linux cluster purchased by Professor Perrone's external funding. This cluster computer will be used to run large sets of experiments with the SWAN simulator.

I will also make use of Bucknell's Linux workstations for software development and study of the necessary components of the SWAN simulator.