BUCKNELL UNIVERSITY
UNDERGRADUATE RESEARCH PROPOSAL
SUMMER 2009

DEVELOPING AN XML LANGUAGE FOR CREDIBLE WIRELESS NETWORK SIMULATION

ANDREW W. HALLAGAN
CLASS OF 2011
BOX C3184
BU ID: 105 916 72
andrew.hallagan@bucknell.edu

LUIZ FELIPE PERRONE (FACULTY MENTOR)
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE
perrone@bucknell.edu

PART A

PROJECT DESCRIPTION

Computer networks have proven themselves to be an extraordinary and powerful means of exchanging information in just about any scenario one can imagine. It is hard to overstate how indispensable networks of every scale have become in our continuing race for more information, more data, and more connectivity.

There are two basic network types: hard-wired and wireless. Hard-wired networks are the fastest and most reliable kind of network, however, these networks do not allow for any mobility and require a lot of initial configuring before they can work. Wireless networking offers a bit more freedom, since users are free to move around anywhere as long as they can rely on radio signals to communicate with an *access point,* which itself connects to the Internet by physical cable. This implementation is often adequate for homes, schools, and office buildings, where users do not tend to roam too far from this access point.

Not all scenarios have such a straightforward network implementation, however. "*Ad-hoc"* is the name given to a second kind of wireless network, one in which individual computers act as links in a chain to relay data from computer, to computer, to Internet access point. For a mental picture, imagine three students with laptops outside, but only Student A is close enough to the nearest building to receive Internet access. Student B cannot receive the signal directly, but he is close enough to A to allow him to talk to A's laptop, which can hand off data between B's laptop and the Internet. Student C is further out yet, but close enough to connect with Student B. Information is sent from C, to B, to A, to the access point, and then back again so that even Student C has Internet access. There are many possible applications for this kind of network, many of which are in defense and homeland security.

These kinds of networks are still being developed and analyzed, and there is much work to be done to find out if they will actually behave as we predict. Those predictions can either come from mathematical models, which are faster at producing answers, but often harder to derive, or from computer simulation. If executed correctly, simulation can give us a lot of reliable information, even from the mathematically difficult scenarios we would like to study.

Unfortunately, using network simulation in research has its own set of problems. The fact that different researchers use different simulators, and follow different methodologies, makes it difficult for one to confirm someone else's results. There is something of a mismatch between the power of the newest simulators and the lack of experience many researchers have with them. Researchers eager to simulate their own models of an ad-hoc network may not be fully aware of how a particular simulator runs, and end up with flawed data. Perrone et al. [6] describe this situation when they say, "The process [of simulation] demands time, effort, and skills that potential users cannot always contribute and, commonly, results in solutions that aren't reusable or accurate."

The rather alarming lack of simulation credibility is a recurrent theme in the papers written by researchers in the field. Kurkowski et al. [3] have written a harsh but honest evaluation of the state of ad-hoc network simulation in which they sifted through 151 published papers regarding mobile computing, 114 of which used simulation to support their findings. Their verdict was that ad-hoc network simulation has been just plain sloppy. There are too many simulators, too many parameters to describe, and too many methods of processing output data (i.e. computing statistics such as means, variances, and confidence intervals) for researchers to keep track of. It is the opposite of the scientific method; confounding variables that researchers may not even be aware of flaw their data, and when another group tries to replicate those findings, they may have changed or forgotten details in their own simulation which alters their results as well! The authors have found that virtually every step of the simulation process has ample room for both syntactic and semantic error. Prof. Perrone and I will focus on improving one of the initial steps: simulation setup.

Setup starts with formulating the right kind of *model description.* This is simply a file that contains information such as, the number of "nodes" in the network (which represent individual computers), node density, coverage area, geographical layout, etc. This model description is meant to convey as much detail about our imagined scenario that we can think of, and it is then fed to a program that will simulate the model as it is described. The credibility of a simulation depends on following strict rules of procedure on a number of different steps in the process; the generation of the model description file is no exception. From the beginning, it needs to be complete and widely understood by different parties [3] The consistency of the model description should also be easy to verify so that one is not allowed to combine incompatible components in the same model (many simulators will not catch this type of semantic error, and the researcher is never warned that the data produced by the experiment is completely meaningless.)

Prof. Perrone and I propose to provide a solution to these problems with the creation of a universal language in which to describe models for wireless network simulation. In order to make simulations reproducible and credible, we will provide way to standardize the model description and automate the verification of that model. Ultimately, our language and processing tools will ensure any given simulation is set up correctly and consistently, so that not only will researchers be confident in their results, but others will also be able to easily replicate and validate their findings. We will use the Extensible Markup Language (XML), which was devised for annotating and organizing textual data [2] It is similar to the Hypertext Markup Language (HTML) that is used to code many of the pages on the World Wide Web.

In HTML, a programmer uses "tags" to alert the Internet browser on your PC that certain text should be italicized, capitalized, colored red, aligned on the right, etc. We will develop a similar way for organizing the myriad parameters and components that are introduced in a model, and provide a way to automatically check those setup conditions before running the simulation. By creating special constructs in XML, we will make it possible for a program to verify that, in a given model, parameters match their data type and value expectations, and that model components are compatible with one another.

We will design our XML language so that we can use automatic transformations to convert a generalized model description to different simulator languages. For this task we will use an application written in the Extensible Stylesheet Language Transformations (XSLT) language. We will program XSLT to manipulate our XML files and convert them to other formats. Prof. Perrone and I will be able to provide a reliable method of translation, so that setup information can be directly read in by any of the popular simulators used by researchers (e.g. ns-2, ns-3, OMNeT++, GloMoSim). Again, the standardization of the XML language, along with the ability for automatic translation into any number of model description files, will make it possible for researchers using different setups, different systems, and different simulators, to share their XML model descriptions and not have to worry about any information being lost or corrupted in the process.

METHODS

Our first task will be to analyze sample model descriptions written for the SWAN simulator with which Prof. Perrone is already very familiar. Studying these files, we will catalog every piece of data that they give to SWAN, and generalize the information to build the right kind of "vocabulary" for describing these simulations in XML. Once this is done, we will start developing the "rules of grammar" (i.e., XML structures) that our language will use to express these models.

When all of this is completed, we will begin writing the code of an XSLT program that takes our XML files as input, and translates them into the correct model description language for SWAN. After verifying that the process does indeed work for the SWAN simulator, we will look at other popular simulators such as ns-2, ns-3, and OMNeT++. Again, our process will involve analyzing sample input files for these simulators, generalizing from them, and integrating any new information back into our XML language.

OUTCOMES

I am currently taking a semester-long, 1.0 credit independent study with Professor Perrone this spring. In this course, I will become acquainted with the appropriate background knowledge I will need as well as the technical skills for actually executing our project (specifically, I will learn much about XML and XSLT programming). Prof. Perrone and I hope to succeed in establishing a collaboration with Drs. Canonico, Emma, and Ventre from the University of Naples, Italy, whose previous work we plan on extending [1] . We expect that this work will be a strong contribution that will lead to the publication of a conference paper.

TIMELINE

We expect the analysis of sample model descriptions to take approximately one week. With our entire compilation of model description data, the next step will involve coming up with a tentative XML description, which means formulating the actual XML tags that will be used in our language. Thirdly, there will be the coding required to build the right translator to go from XML to the SWAN-compatible model description language. Finally, our last step will be extending the work to include translators for the other simulators mentioned, likely starting with ns-2 and ns-3.

PART B

RESEARCH ENVIRONMENT

Over the course of this project, I will be using the SWAN simulator itself, and other pieces of software such as the XSLT translating environment and the XML language itself. I will make use of Bucknell's own laboratory computers for much of the coding, testing, and trouble-shooting involved.

Prof. Perrone and I will be in close communication throughout the duration of the project. He has specifically mentioned instituting an "open door" policy whereby I will be free to count on him at any time to share important findings, questions, or concerns along the way.

REFERENCES

[1] Canonico, R., D. Emma, and G. Ventre. *"An XML Description Language for Web-Based Network Simulation."* In Proceedings of the IEEE Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2003). Delft, The Netherlands, October, 2003.

[2] Harold, Elliotte R., and W. Scott Means. *XML in a Nutshell*. Ed. Simon St.Laurent. 3rd ed. USA: O'Reilly Media, Inc., 2004.

[3] Kurkowski, Stuart, Tracy Camp, and Michael Colagrosso. *"MANET Simulation Studies: The Incredibles."* SIGMOBILE Mobile Computing and Communications Review 9 (4), 2005, pp.50-61.

[4] Liu, Jason, Felipe L. Perrone, David M. Nicol, Michael Liljenstam, Chip Elliot, David Pearson. *"Simulation Modeling of Large-Scale Ad-Hoc Sensor Networks."* In Proceedings of the 2001 European Simulation Interoperability Workshop (EURO-SIW). London, England, 2001.

[5] Perrone, L. Felipe, Christopher J. Kenna, and Bryan C. Ward. *"Enhancing the Credibility of Wireless Network Simulations with Experiment Automation."* In Proceedings of the IEEE International Workshop on Selected Topics in Mobile and Wireless Computing (STWiMob 2008). Avignon, France, 2008.

[6] Perrone, L. Felipe, Claudio Cicconetti, Giovanni Stea, and Bryan C. Ward. *“On the Automation of Computer Network Simulators,”* In Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools 2009). Rome, Italy, 2009. [In press]