IEEE TCSIM Newsletter

IEEE computer society

Issue 05 June 2010

Inside this issue:

Recent Events	2
Upcoming Events	2
TCSIM Student Award	2
Smart Grid	3
Interference Modeling in NS-3	6
Call for Papers	8

Chair's Message

By Dr. Dave Cavalcanti, Chair of TCSIM

Dear TCSIM colleagues,

Welcome to Issue No. 5 of the TCSIM newsletter. This is the second issue in 2010 and I'd like to take this opportunity to thank all the authors, reviewers and the editorial team for making the extra-effort in increasing the circulation frequency of the newsletter. I hope you all can take advantage of this opportunity to share not only technical information, but also to keep up to date with the TCSIM activities. We count on your contributions to make this publication even more interesting and beneficiary.

This issue features two invited articles. The first article aims to present one of the early attempts to simulate Smart Grid Systems. The second article addresses the present state-of-art in packet-level interference modeling of IEEE 802.11b Wireless Local Area Networks in NS-3.

Hope you enjoy the reading and thanks for your collaboration!

New TCSIM mailing list

In order to facilitate communication and information sharing with TC members, a new TCSIM mailing list has been created using the IEEE Listserv system.

To send a message to the new TCSIM list, just send mail to TCSIM@LISTSERV.IEEE.ORG

If you are not currently subscribed to the list please send a message to *Prof. Kaushik Chowdhury* at krc@ece.neu.edu who is currently managing all the subscriptions to the new TCSIM list. You can also search for the TCSIM list at http://listserv.ieee.org/

We hope you can make use of this new list to share simulation related information with the TCSIM community.

Recent Events

(PADS 2010) Principles of Advance and Distributed Simulation
17 May 10, - Georgia Institute of Technology, Atlanta, GA (USA)
http://www.pads-workshop.org/pads2010/

(ISCC 2010) IEEE Symposium on Computers and Communications
22 Jun 10, Riccione, Italy
http://www.ieee-iscc.org/2010/

Upcoming Events

IEEE SMARTGRIDCOMM 2010 – The 1st IEEE International Conference on Smart Grid Communications, October 4 – 6, 2010 National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA
http://smartgrid.ieee.org/ieee-smartgrid-news/53-ieee-smartgridcomm-2010

(MASS 2010) IEEE The 7th International Conference on Mobile Adhoc and Sensor Systems, November 2010, - Intercontinental Mark Hopkins (tentative), San Francisco, CA (USA)
https://mass2010.soe.ucsc.edu/

TCSIM Student Award

"TCSIM supports students presenting ideas on exciting research frontiers through performance appreciation and travel awards."

TCSIM sponsors several Student Support Awards every year. The awards are intended to encourage and motivate student participation in conferences. The TCSIM awards are only given to students with accepted work in selected conferences. The awards selection process and distribution are fully managed by the organizing committees of the events.

Congratulations to the winners of the TCSIM sponsored student best paper award in SECON 2010:

- Lap Kong Law, paper: "On the Uplink Capacity of Hybrid Cellular Ad Hoc Networks"

Additional TCSIM awards in 2010 will be given during the following events:

- CGAMES 2010 (Student Best Paper Award)
- MASS 2010 (Student Travel Grants)

For more information on the 2010 TCSIM Student Awards, please visit the TCSIM webpage: http://tab.computer.org/tcsim

Smart Grid

Mostafa El-Said, Mark Dornbush and Yue Li
School of Computing and Information Systems

Grand Valley State University
Grand Rapids, MI 49401, USA

Introduction

A smart grid is a network that delivers electricity from power companies to consumers. The "smart" part of the grid is that electrical usage is monitored and sent back to the power company. With this feedback, the power companies can better regulate the power grid. A smart grid can even regulate energy usage by scheduling appliances to run during off peak hours or turning off an appliance when a brown-out may occur. In the USA, the power grid's supply of electricity is taken for granted. Power outages are rare and usually last less than a day; brown-outs are rare but are becoming more common. Renewable energy is growing but only provides a fraction of the demand. By implementing a smart grid in America resources can be saved without affecting people's quality of life. For the power company to receive the monitoring information, the devices will need to be networked. There are several proposals for the smart grid topology [1].

A *mesh network* could be used where data hops from device to device (or from power meter to power meter) eventually reaching the power company's router or gateway. Another approach would be a more *traditional model* with homes providing information to a server located in transformers which then would be uploaded to the power company's systems.

The US Department of Energy has proposed an "Advanced Metering Infrastructure" as an open standard and their preferred approach to implementing the smart grid. With AMI, a home's primary energy users, washing machines, dryers, heating and cooling would have some intelligence added to control when they would operate. Electricity pricing information is sent to the devices or a central hub in the home. This pricing information allows the consumer to decide when or how much to use a device. Operation can be reduced or scheduled to run at a different time. Whirlpool corporation showed a smart grid compatible clothes dryer at the January 2010 Consumers Electronic Show in Las Vegas. Whirlpool also committed to producing one million smart grid compatible appliances by the end of 2011. Unfortunately, Whirlpool put a significant catch on their promise to produce these appliances, but it is understandable that they felt the need. The catch for Whirlpool is that they need a completed, open, and global standard for smart grid devices by the end of 2010 [2].

Among the first attempts to implement a Smart Grid Simulator application, we developed a Smart Grid Simulation (SGS V.1) monitoring system that implements the following functions:

1. Creates a power plant simulator (server side of the simulation application)
   a. Accepts connections from consumer devices
   b. Based on power plant loading, will allow the consumer device to run
   c. Communicates to consumer device the best time to run based on loading and electrical costs
   d. Commands consumer devices off to prevent brown-outs
2. Create a smart consumer appliance controller simulator (client side of the simulation application)
   a. Communicates with the power plant simulator
   b. Schedules best time to run device based on price and load
   c. Turns device on and off based on data from power plant or consumer's feedback.

The simulation software has been tested using a power plant simulator and two consumer device controllers. The consumer devices connect to the power plant server to receive loading and pricing information. The consumer devices upload their power requirements to the power plant with their desired run time and schedule and will present the pricing information to the consumer. The consumer can choose to pay more for the electricity by running the appliance during peak hours or reschedule a run time during a less expensive time.

Simulation Architecture

The server side of the application was written is Java and is used to simulate a power company's interface to consumers such as given in figure 1. It is a multithreaded application to accept and service multiple connections from clients at the same time. The operator can set electricity prices for two time windows and simulate a brown out condition. The server periodically sends the time (for clock synchronization) and pricing information to all attached clients. A proprietary protocol is used between the server and client applications. The "Set Price" button can be used at any time to adjust the electricity prices for the client application.

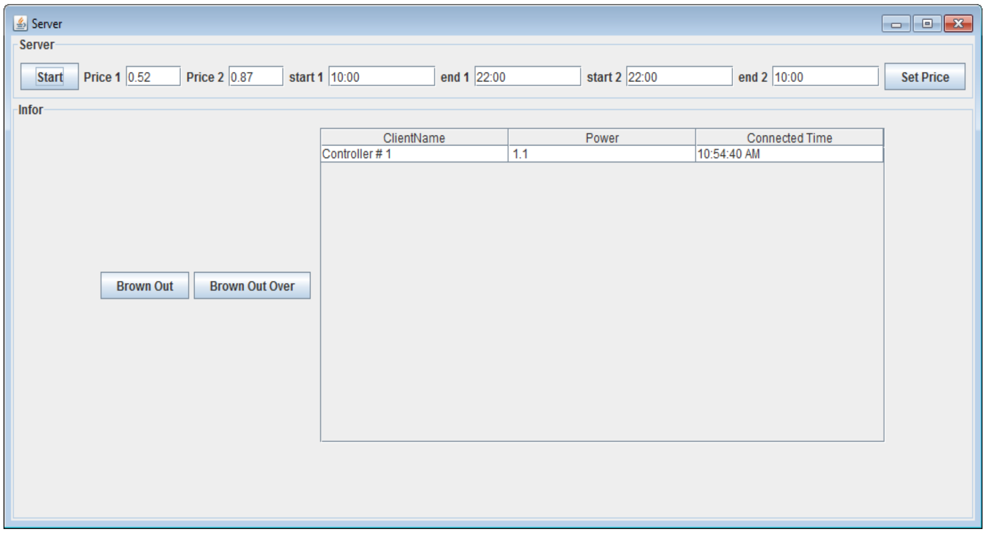

Figure 1. Power Plant Simulator-Server Side

The client side of the simulation application was written in C# such as described in figure 2. The client software simulates a smart consumer's local control over the appliances/devices in their home. For each attached device, the consumer can choose to have manual control, run at an off peak time, schedule a run time, or always be off. With manual control, the attached devices can be turned off and on locally. The remote controlled relays are always on. With the run off peak and schedule run options, the relays are turned on and off based on the time parameters. When the control option is set to always off, the control relays are off and the attached devices will not have power.

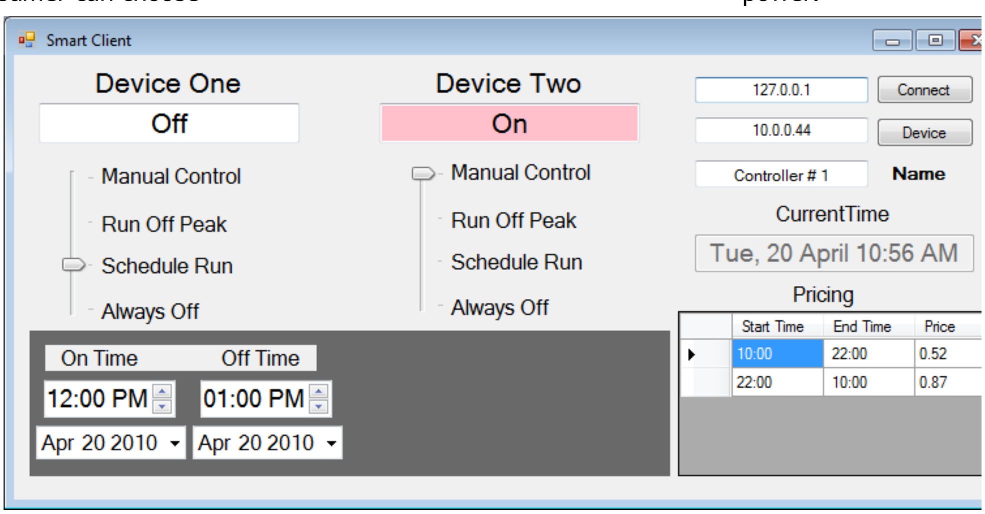

Figure 2. Smart Consumer's Local Control Simulator-Client Side

The client program synchronizes time with the server program so there won't be a miss-match or incorrect billing by running an appliance at the incorrect time. The client also receives updated electricity pricing information from the server at regular intervals. The pricing information is used to schedule a time for the appliance to run when electricity is cheaper.

When the power plant simulator indicates to the client that a brown out condition is occurring, the client software will turn off the attached devices until the power plant indicates the brown out is over. The appliances controlled in the simulation experiments are light bulbs such as shown in figure 3. The bulbs have local switches to mimic usage in the real world.

To interface with the appliances, Wago remote IO was used [3]. The Wago IO acts as a server and uses the Modbus protocol over TCP/IP for control. A separate thread in the client program reads the input status and writes the desired relay status out. A 750-406 module was used as the ac input and a 750-513 module relay.

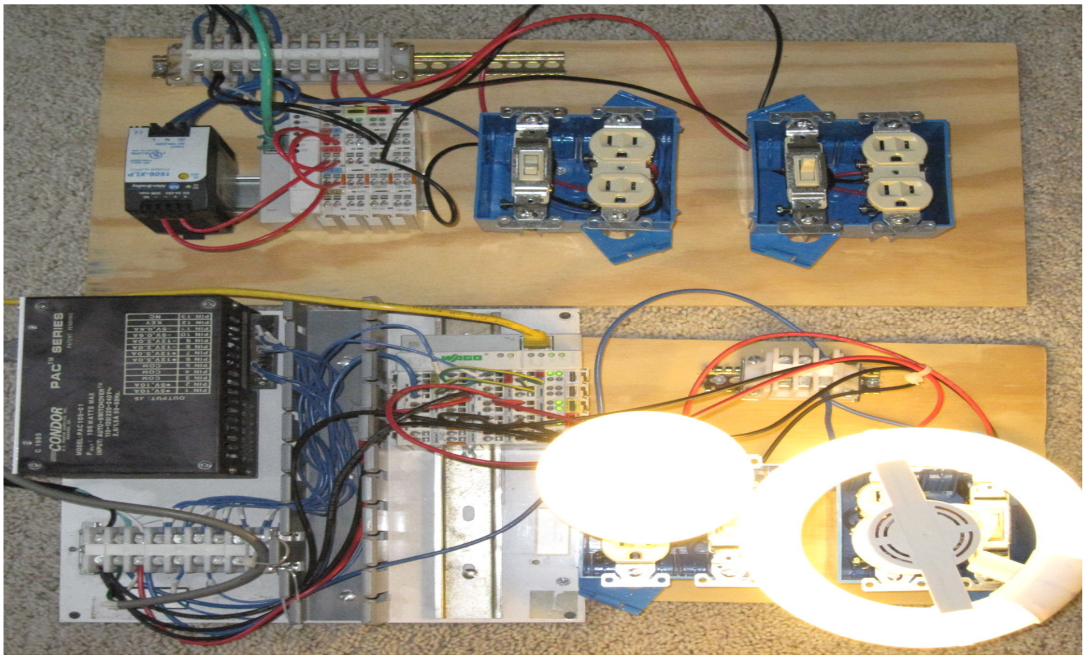

Figure 3. Controlled Appliances

References:

1. US Department of Energy - Smart Grid http://www.oe.energy.gov/smartgrid.htm
2. GreenTechGrid, "Whirlpool Plans 1M Smart Dryers by 2011". resource accessed on April 2010 at http://www.greentechmedia.com/articles/read/whirlpool-plans-to-make-1m-smart-dryers-by-2011/
3. Wago remote IO devices, www.wago.com

Interference Modeling in NS-3

Trevor Bosaw and Chittabrata Ghosh,
Electrical Engineering Department
University of Washington, Seattle, WA-98125

Introduction

Despite the rapidly increasing research interest of interference in the 802.11 transmission frequencies, interference modeling in network simulators remains limited. Many simulators model interference from a simple threshold, while others, such as NS-3, rely on a clear channel validated model for interference. In this article, three main types of interference on packets considered in NS-3 are explained in the context of IEEE 802.11b Wireless Local Area Networks. These behaviors are: the physical layer capture effect, single packet preamble interference, and single packet payload interference. These three aspects consist of the majority of packet-level interference, and the incorporation of these characteristics will improve the accuracy of the NS-3 interference model to a large extent.

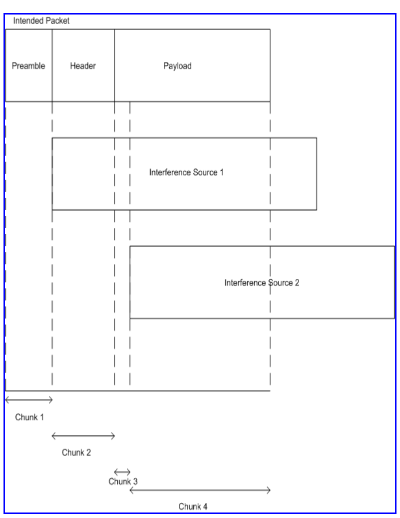

Figure 1: *Visual description of where NS-3 splits a packet into chunks for a typical interference scenario*

Physical Layer Capture Effect

The NS-3 physical layer is composed of a state machine, split into four states: TX, RX, IDLE, and CCA_BUSY. Each node initializes in the IDLE stage, and if the power in the channel is measured to be above a clear channel assessment (CCA) busy threshold, the state can switch to CCA_BUSY. In both of these stages, any packet above a packet receipt threshold can be received at any time. When the node begins to transmit a packet, the state is automatically changed to the TX state, regardless of the current state of the node (it is the medium access control (MAC) layer's responsibility to prevent a transmission while the node is in the RX state, so the physical layer does nothing to avoid this). While the node is in the TX state, no packets can be received. Finally, the node can switch to the RX state from the IDLE or CCA_BUSY state whenever a packet is received (again, assuming the packet's power is above the packet receipt threshold). When in the RX state, the node cannot receive any further packets.

Prior research has shown that such a model does not completely match actual behavior. According to [1], a receiver can exhibit behavior referred to as the 'physical layer capture effect', which is the case in which a receiver can synchronize to a new packet even if it is already synchronized to a current packet.

In [1], it was found that this situation occurred whenever the following packet was of higher signal strength than the original packet. While it is difficult to truly determine whether this effect matches threshold based behavior, or if more complicated algorithms are necessary, this study shows that this effect exists, and that it makes a big difference on packet reception behavior. Also, there are two possible cases for the capture effect: one in which the second packet arrives while the receiver is correlating the first packet's preamble, and a case where the second packet arrives while the receiver is synchronized to the first packet's payload.

While the NS-3 main repository currently has no model for this capture effect, an NS-3 side project has this functionality implemented. This project includes wifi enhancements to the current NS-3 model, with a focus on the wifi physical layer (located at: http://code.nsnam.org/timob/ns-3-wifiex/). The basic physical layer included in NS-3 is preserved here, with simple code additions to the RX state in order to include the capture effect. In this implementation, if a node is in the RX state (receiving an initial packet), it can drop the initial packet in favor of a second packet in the case when the second packet's signal strength is above a certain threshold.

There is also the ability to set two different thresholds for the two capture effect cases (as detailed above, when the receiver is correlating the preamble of the first packet, or when the receiver is synchronized to the payload of the first packet), or even to turn off one or both of these cases.

In order to improve the NS-3 wifi physical layer to allow for capture effect, it is suggested that this wifi physical layer enhancement be merged with the main repository for NS-3. For a threshold based capture effect functionality, these wifi physical layer enhancements perform as a beneficial addition to physical layer modeling in NS-3.

Preamble Interference Model

The interference modeling implementation in NS-3 currently has no functionality for the calculation of the effects of interference during the preamble of a received packet. Although there has been little research to the true effects of this preamble interference, the findings of at least one paper [2] show that the interference during the preamble of a packet influences the total packet error rate by a great amount. Due to the lack of research on this subject, it is difficult to find a suitable algorithm to model this interference. As an initial step, the results from [2] can be used as a lookup table to model single packet preamble interference. While multiple packet interference is outside of the scope of this lookup table, the majority of interference events occur with single packet interference, reducing the importance of this problem.

To implement this, code has been added to NS-3's interference-helper class that detects the signal to interference ratio during the preamble, as well as the location in the interference of the preamble. The lookup table [3] is then referred to, and a packet error rate is determined from the preamble interference. This only serves as a preliminary solution to the full calculation of preamble interference effects, but it is still an improvement over the previous functionality, which ignored interference during a received packet's preamble.

Payload Interference Model

In the current NS-3 implementation, payload interference is determined by calculating a specific bit error rate for a specific region, and then applying this bit error rate as a binomial distribution on the amount of bits in the region. To do this, it first constructs an event array of the power changes in the channel over the time period that the intended packet is received. This array is able to keep track of changes of interference for multiple packets, as well as noise or interference existing before the receipt of the intended packet. In order to actually calculate the packet error rate, the interference helper first splits the time segment that the packet is received into chunks. Each chunk represents a section of the received packet (either the header or payload) with a specific interference power level. Each chunk can only have a single interference power level, and so a received packet with many changes in interference will have many chunks, while a packet with a constant level of interference over its time of receipt will only be split into two chunks (header and payload).

Figure 1 illustrates an example of the chunk distribution for a typical scenario where a packet is received with interference. After splitting the received packet into chunks, chunk success rates are found for each section. To do this, the bit rate of the packet, interference power, and time of chunk are used. The bit rate of the packet is used to decide what base error rate function is used (bit error rate is different between 1 Mbps and 5.5 Mbps, for example, because the latter uses CCK coding), the interference power is used to calculate the signal to noise ratio of the chunk, and the time segment length is used (in conjunction with the bit rate) to determine how many bits are in the chunk. Finally, to get chunk success rate, the bit error rate is used in a summing binomial function [3] which is then raised to the power of however many bits are in the chunk. Taking all of these chunk success rates and multiplying them together then gets the packet error rate, which is sent up to the physical layer to determine whether the packet should be dropped or not.

The majority of the NS-3 interference modeling occurs in the calculation of payload interference. This model takes into account multiple packet interference, as well as different methods for every common 802.11b bit-rate, but it does not corroborate with actual receiver behavior. Since there exists a lack of research into the specific effects of payload interference, data from CMU emulator experiments was used to provide an alternate payload interference implementation. This data is implemented as a lookup table, similar to the lookup table used for [2] preamble interference results.

References

1. A. Kochut, A. Vasan, A. U. Shankar, and A. Agrawala, "Sniffing out the correct Physical Layer Capture model in 802.11b," 12th IEEE International Conference on Network Protocols, 2004.
2. G. Judd and Peter Steenkiste, "Characterizing 802.11 wireless link behavior," Wireless Networks, 2008.
3. T. Bosaw, An improved 802.11b Interference Model for Network Simulation. M.S. Thesis, University of Washington, Seattle, June 2010.

Call for Papers – IEEE TCSIM Newsletter

The IEEE TCSIM Newsletters will publish short technical papers. The submissions should emphasize modeling, design, and analysis of computational methods for simulations and its applications in various areas, including, but not limited to, computer science, engineering, communications, and simulation applications. The submissions are invited covering, but not limited to, the following topics:

- Simulation architecture modeling and prototyping
- Simulation algorithm design, implementation, and analysis
- Simulation complexity in computing
- Parallel and distributed simulation
- Design and usage of simulation tools
- Real-time simulation monitoring
- Simulation tools for communications and networks
- Simulation of computer systems and applications
- Agent-based simulation tools focus on the use of agents in engineering, human and social dynamics, military applications
- Systems and process simulation
- Simulation of ubiquitous networking and computing
- Simulation of transportation systems
- Automotive simulation applications
- Building and energy management simulations
- Machine learning
- Virtual reality systems
- Knowledge and data systems
- Systems optimization
- Web-based simulation and applications
- Department of Defense Architecture Framework (DoDAF)-based network simulations
- DoDAF-based vulnerability assessment

Submission

All papers must be submitted to elsaidm@gvsu.edu in four pages or fewer, including all figures, tables, and references. A manuscript submitted for publication should be original work that should not have been previously published and should not be under consideration for publication elsewhere. If an author uses charts, photographs, or other graphics from previously printed material, he/she is responsible for obtaining written permission from the publisher to use the material in his/her manuscript. The maximal number of figures and tables are five, and the number of reference is limited to ten. Submissions exceeding this length will be returned without review. Papers should use 7.875in x 10.75 in (20cm x 27.30cm) trim size and the IEEE transactions two-column format in 10-pt. font. Please submit electronically in PDF file, and ensure that the submitted file can be viewed in Acrobat Reader 8.0. No hard copy is necessary. A standard IEEE copyright release will also be required before full acceptance.

All papers must include the authors' affiliation and e-mail addresses of all authors. All papers will be fully refereed for accuracy, technical content, and relevance. Contact Dr. El-Said at elsaidm@gvsu.edu with any questions concerning the paper submission and review process, or questions regarding the relevance of a paper to the IEEE TCSIM Newsletters.

Editor-in-Chief
Mostafa El-Said, Ph.D.

Director of Data Communication Center (DCC)
School of Computing and Information Systems
Grand Valley State University
C-2-100 Mackinac Hall
1 Campus Drive
Allendale, MI 49401

Phone: 616-331-8686
Fax: 616-331-2106
Email: elsaidm@gvsu.edu