

OPNET IN AN EDUCATION ENVIRONMENT: THE SIMON FRASER UNIVERSITY EXPERIENCE

C. Cheng, M. Jiang, E. Lo, R.H.S. Hardy, and Lj. Trajkovic

School of Engineering Science

Simon Fraser University

Burnaby, British Columbia

Canada V5A 1S6

chao, zjiang, edward, steve, ljilja@cs.sfu.ca

<http://www.ensc.sfu.ca/research/cnl>

ABSTRACT

In this paper we discuss the network systems support for the OPNET (version 5.1.D) application at the School of Engineering Science at Simon Fraser University (SFU), we describe a research project in wireless data networks using OPNET (version 5.1.D), and we discuss the teaching experiences using OPNET (version 4.0.A) at the University College of the Fraser Valley (UCFV).

INTRODUCTION

OPNET (version 5.1.D) [1] is an integral part of the laboratory exercises in our senior undergraduate networks course in the School of Engineering Science at Simon Fraser University (SFU), as well as an optional part of the student projects in our graduate course in network protocols and performance. OPNET (5.1.D) is used as a simulation tool for experimental research in wireless data networks at the graduate level. One ongoing project involves the simulation of wireless data network performance using traffic traces that exhibit long range dependence. Genuine traffic traces are used in a trace-driven simulation. At the University College of the Fraser Valley (UCFV), an affiliated college to SFU, OPNET (version 4.0.A) is used in an undergraduate data communications course at the third year level.

INSTALLATION AND MANAGEMENT OF OPNET (5.1.D) APPLICATION

The OPNET (5.1.D) network system at SFU consists of two servers, and Unix and NT workstations connected together on a switched 10/100BaseT network.

To ease installation, upgrade and license management, OPNET (5.1.D) is installed on a server and the users have access based on their needs and location. The system consists of a Sun Microsystems Ultra 10 Server

and two subnets (LabNet and ResearchNet) on a 100BaseT switched backbone.

On the LabNet, the Server is also used as a CPU server. Undergraduate students use PC's (Dell Dimension) to open X-window connections via Hummingbird Exceed X Window to the Server. The OPNET (5.1.D) sessions run on the Server and the OPNET (5.1.D) windows are displayed on the PC monitors. As most of OPNET (5.1.D) tasks are run on the Server, the server load can be heavy at times. Because all the PC's are located in publicly accessible open labs, we use this setup to ensure the OPNET (5.1.D) files and license keys are centrally maintained. Therefore, we do not need to reinstall OPNET (5.1.D) and lose the license key in the event of a local drive crash. The 100BaseT switched connection between the PC's and the Server ensures that the performance approaches that of a local copy of OPNET (5.1.D).

On the ResearchNet, the Server is the file and license server for OPNET (5.1.D). Graduate students and researchers use Sun Workstations (Ultra 10's and Ultra 30's) to mount the OPNET (5.1.D) files, check out licenses from the Server, and run OPNET (5.1.D) on the local workstations. The tasks of NFS and license daemon add very little load on the server. Most OPNET (5.1.D) tasks are run on the high end Sun Workstations. A group of Dell Dimension PC's has recently been purchased and will also be used for OPNET (5.1.D). The Faculty and system staff workstations (both Sun Microsystems Ultra and Dell Dimension) are connected to the ResearchNet. These users have the choice to run OPNET (5.1.D) on the Sun workstations or on the PC's. While OPNET (5.1.D) for Solaris is mounted from the Server, OPNET (5.1.D) for Windows NT is installed locally on each PC.

Most system administration is done centrally on the server. The system staff installs and maintains the OPNET (5.1.D) releases. Local scripts are issued to help the users. Based on their research, teaching and studying

requirements, the scripts ease the startup process (set paths for the Sun Workstation users and X-windows connections for the PC users). As a result, the users are able to quickly use OPNET (5.1.D) for their projects. The system staff also obtains licenses from <http://www.mil3.com> and keeps track of the centrally installed Solaris licenses and locally installed NT licenses.

SIMULATION OF MOBILE NETWORKS USING OPNET (5.1.D)

Simulation of the CDPD network

Cellular digital packet data (CDPD) [2] is a standard protocol developed for mobile data networks. It is a multiple access protocol in which stations that want to transmit data into the network must compete for access in the shared communications medium. CDPD shares some characteristics of more familiar multiple access networks, such as Ethernet (IEEE 802.3), while still having significant differences. CDPD differs from other multiple access protocols in two ways: the wireless transmission medium, and the mechanism for collision detection.

In most networks the data is transmitted on a wire or via the fiber-optics cable. In mobile data networks, radios transmit the data over the air. This so-called air-interface, in contrast to wires or fiber-optics, is prone to bit errors introduced into the transmitted data. A typical bit error rate (BER) for wired and fiber-optics networks is 10^{-10} , and is fairly static. The wireless BER can be rather dynamic and can be as high as 10^{-2} , or even higher. This highly dynamic environment presents a unique challenge to the implementation of CDPD networks.

A second difference between CDPD and multiple access protocols is in the collision detection. In wired networks, stations can listen to their own transmissions. If another station is transmitting at the same time, stations will detect that a collision has occurred. When using radios to transmit data, mobile end systems (M-ES) cannot listen to their own transmissions, and, therefore, must rely on the forward channel and the mobile data base station (MDBS) to determine if the transmission was successful.

The topology of a simple CDPD network is shown in Figure 1. Only the Media Access Control (MAC) layer of the CDPD protocol is presented. Also shown are a mobile data intermediate system (MD-IS) or mobile

router, and several fixed end stations (F-ES) connected to the wired network.

CDPD network model

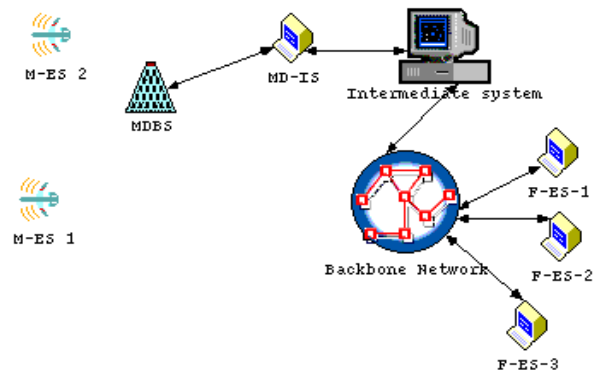


Figure 1. Model of a CDPD network. Each mobile end system (M-ES) is connected with the backbone network through the mobile data base station (MDBS).

In the network model of the Airlink interface shown in Figure 2, the two M-ES compete for the bandwidth (825 MHz \sim 825 MHz + 30 KHz). Each M-ES generates data packets according to the genuine trace collected from the operational CDPD network. Other traditional traffic models from the OPNET (5.1.D) libraries, such as Poisson and Gaussian models, are also available. The MDBS arbitrates activities on the channels it hosts at the MAC sub-layer, much like an Ethernet hub.

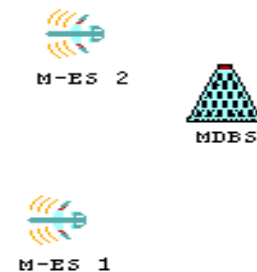


Figure 2. Network model for the Airlink interface. The two M-ES send packets to the MDBS via the reverse channel at a minimum frequency of 825 MHz. They receive packets from the MDBS via the forward channel at a minimum frequency of 870 MHz.

Node models

The M-ES model shown in Figure 3 is composed of a radio receiver and a radio transmitter, both working at 19.2 Kbps. The minimum frequency of the receiver (forward channel) is 870 MHz, with a bandwidth of 30 KHz. The minimum frequency of the transmitter (reverse channel) is 825 MHz, with a bandwidth of 30 KHz.

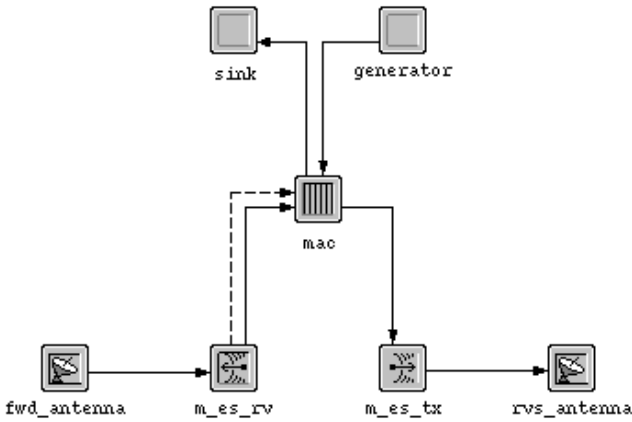


Figure 3. Node model for M-ES. The radio transmitter is operating at a minimum frequency of 825 MHz. The radio receiver is operating at a minimum frequency of 870 MHz. Both have bandwidth of 30 KHz.

The processor (mac) includes the digital sense multiple access (DSMA) logic. It retrieves the information about the reverse channel from the forward channel (*busy/idle* and *decode status* flags). If the reverse channel is busy or a collision occurs, the M-ES will back up for a random time period and try to retransmit again.

The generator generates packets according to our genuine traffic trace. It can also use the traditional traffic models supported by OPNET (5.1.D) libraries. The M-ES model also destroys the packets it has received.

The MD-BS model shown in Figure 4 is more complex. It is connected to the external network (the wired part of a CDPD network) with a pair of transmitters and receivers. It receives packets from the M-ES via two radio receivers operating at a minimum frequency of 825 MHz (reverse channel), and transmits packets via two radio transmitters at a minimum frequency of 870 MHz (forward channel).

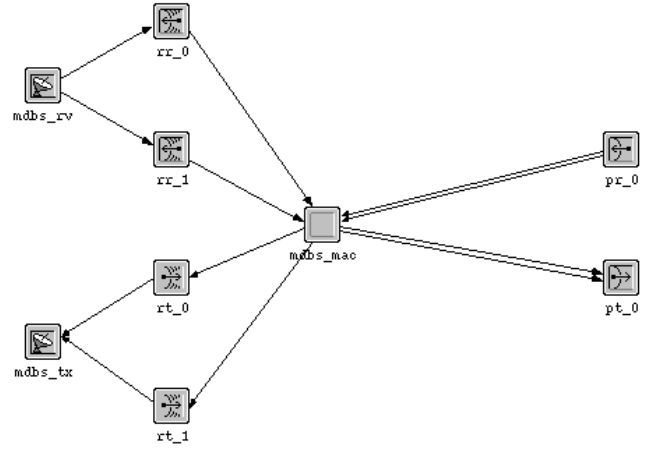


Figure 4. Mobile data base station (MD-BS) node model. This is the interface between the wireless part and the wired part of a CDPD network.

The MD-BS model also includes a processor (mdbs_mac), which implements the DSMA logic for the MD-BS. It sets the *busy/idle* and *decode status* flags on the forward channel according to the status of the reverse channel. The MD-BS also receives the packets and measures certain statistics such as BER, the RF power level, the signal-to-noise ratio, and the end-to-end delay.

The model is highly parameterized. In each M-ES, users may modify the parameters such as transmission rates, packet length (which can be fixed, read in from a trace, or described stochastically), and the radio transmission modulation method. Where applicable, parameters are set to the default values specified by the CDPD standard.

We are currently working on improving the model by incorporating a Rayleigh fading model into the radio transceiver pipeline to induce bursts of bit errors in the transmitted packets. This type of bit error model will allow a direct specification of the BER and the burstiness of the errors in the channel.

Impact of traffic characteristics on wireless teletraffic engineering

Traffic generated by new services (voice, data, image, and video) available in current data networks differs from traffic patterns observed in circuit switched voice networks. These new services require new traffic models that are different from traditional Poisson models used for modeling voice traffic. Hence, users may experience poor performance due to incorrect assumptions in provisioning and designing data networks.

Interest in self-similar traffic was first stimulated by the measurements of Ethernet traffic at Bellcore [3]. The measurements indicated that “traffic looks the same on all time scales”. Hence, the term self-similar (or “fractal”) traffic. Since then, this feature has been established in many other traffic traces, as, for example, in Transport Control Protocol (TCP) and Motion Pictures Experts Group (MPEG) video traffic. An important characteristic of self-similar traffic is its long-range dependence, i.e., the existence of correlation over a broad range of time scales [3, 4].

Until now, most of the network traffic measurements were performed on wired networks. The question arises whether the traffic in wireless data networks exhibits self-similar behavior as well. If so, it is important to determine if this traffic characteristic affects the provisioning and design of wireless data networks.

To begin answering some of these questions, we measured traffic on a CDPD network of a local mobile data service provider (BC TEL Mobility). Our initial findings indicate that the trace exhibited a certain degree of self-similarity. We also used the genuine trace in our OPNET (5.1.D) CDPD network model. The preliminary results of our simulations indicate that traditional traffic models may indeed be inadequate for wireless data networks.

Self-similar traffic

We give here the following widely adopted definition of self-similarity for stochastic processes. Assume X_k to be a wide-sense stationary process with mean $E(X_k) = \bar{X}$ and autocorrelation function $r(i)$. Consider next the processes $X_k^{(m)}$ ($m = 1, 2, \dots$) that are constructed from X_k as $X_k^{(m)} = \left(\sum_{n=0}^{m-1} X_{km+n} \right) / m$, i.e., by averaging over non-overlapping blocks of size m . The processes $X_k^{(m)}$ are also wide-sense stationary, with mean \bar{X} and autocorrelation function $r^{(m)}(i)$. The process X_k is called (asymptotically) second-order self-similar if $r^{(m)}(i) = r(i)$, (for $m, i \rightarrow \infty$).

An important parameter of a self-similar process is the Hurst parameter-- H . Self-similarity is implied by $0.5 < H < 1$. Our traffic traces indicate that $0.73 < H < 0.91$, which implies that the wireless data network traffic may be self-similar. Longer traffic traces are needed to confirm our initial observations.

USING OPNET (4.0.A) IN AN UNDERGRADUATE DATA COMMUNICATIONS COURSE

At the University College of the Fraser Valley (UCFV), we offer a bachelor degree in Computer Information Systems. One of the required courses is a third year undergraduate data communications course. The course topics include circuit and packet switching, frame relay, asynchronous transfer mode (ATM), and data security. In the last part of the course, we use the OPNET Lab Manual [5], which includes a copy of OPNET (4.0.A), to help students understand the underlying data communications principles through simulations. Therefore, most of the comments in this section apply to OPNET (4.0.A) only. We expect that most of these issues have already been resolved in the latest version of OPNET (6.0.A).

There are five computers running NT and OPNET (4.0.A). During installation, we found that in order to run OPNET (4.0.A) correctly, we had to change the environment variables to include both the directories for OPNET (4.0.A) and the directories for Microsoft Visual C++ LIB. We are using Pentiums running at 100 and 150 MHz with 32 and 48 Mbytes of RAM. The simulations ran very slowly when students tried to perform several exercises provided in the Lab Manual. The simulations would hang occasionally, so users had to run `m3_manfls` and then enter `revoke_permit` to reset the permit. Originally, the swap file size was set to 43 Mbytes, but simulations often ran out of memory. Most simulations ran fine if one resets the swap file size to 128 Mbytes.

In the NT environment, if an administrator has installed the OPNET (4.0.A) application, the User Variables under the Environment need to be added for the individual users. Otherwise, only an administrator may run OPNET (4.0.A) successfully. If an ordinary user tries to install OPNET (4.0.A), the installation will be unsuccessful because of lack of access rights to the NT system directories.

In order to create model files included in the Lab Manual, the file “env_db40” in `<HOME>/op_admin` has to be modified. Otherwise, students will be unable to access the files. Most exercises given in the Lab Manual are based on the models included in the CD accompanying the text [5]. Accessing the model files through OPNET (4.0.A) is important. The ability of a user to access a model from any directory is one of the most convenient features that could be incorporated into the next release of OPNET.

Another observation refers to the listing of files in OPNET (4.0.A). The list is sorted under upper and lower cases. Students unfamiliar with it may not be able to locate the files. It would be useful if the sorted files were listed in the same format as the Windows Explorer used in NT4.

In the ARQ chapter of the Lab Manual, we fail to reproduce the Figure 3.2. It would be helpful if the random number seed of the simulations were included in the figure. Beginners of OPNET usually need some reassurance that they are doing the exercises correctly, and recreating an example given in the Lab Manual will help achieve this goal.

We have also found that one cannot close the windows when running the Process Editor in OPNET (4.0.A). The Action Button panel under the Tool Button panel shows many options of the Process Editor. The Close button is not displayed in the desktop and students have to close OPNET (4.0.A) in order to close the Process Editor. One way to get around the problem is to set the monitor to a very high resolution. That may be difficult with 15-inch monitors. Our monitor resolution was set to 800 by 600 and still could not nicely display the Action Button panel.

Other OPNET products such as ITDecisionGuru would also be excellent tools for teaching OPNET to the computer information professionals. Network administrators and architects may have limited time and expertise to construct models of computer networks. They are more likely to prefer a block-oriented approach and use the off-the-shelf building blocks of computer networks to model specific companies' networks. Furthermore, they may prefer to learn more about how to use what-if scenarios to optimize their networks. Providing these OPNET models with a Lab Manual may prove useful to these students.

Our students provided the following suggestions:

- Include necessary setups for examples at the beginning of each chapter, followed by several operations and scenarios that one could use for that setup.

- Include more online help files. The use of a "Contents and Index" menu item, similar to Microsoft applications, would be invaluable for beginners.

- Incorporate simple animation into the simulation models so that students could see the processing of packets graphically.

CONCLUSIONS

In this paper we have demonstrated a technique for efficiently serving a number of Unix and NT machines with OPNET (5.1.D) application from a central Unix server. Furthermore, OPNET (5.1.D) was used to simulate a realistic wireless network using trace driven simulations with genuine traffic traces from a client network. We also discussed our experience using OPNET (4.0.A) in an undergraduate data communications course. OPNET proved to be an invaluable tool in helping students understand network operations and performance.

The authors acknowledge with thanks the assistance of MIL 3 through the provision of access to OPNET, and the assistance of BC TEL Mobility through the provision of CDPD traffic traces.

REFERENCES

1. OPNET Documentation V.5.1.D, MIL 3, Inc., Washington, D.C.
2. M. Sreetharan and R. Kumar, *Cellular Digital Packet Data*, Artech House, 1996.
3. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. on Networking*, vol. 2, no. 1, pp. 1-15, Feb. 1994.
4. B. B. Mandelbrot and M. S. Taqqu, "Robust R/S analysis of long run serial correlation," *42nd Session of the International Statistical Institute*, Manila, pp. 1-37, Dec. 1979.
5. I. Katzela and MIL 3, Inc., *Modeling and Simulating Communication Networks, A Hands-On Approach Using OPNET*, Prentice-Hall, 1999.