NCTUns 6.0: A Simulator for Advanced Wireless Vehicular Network Research

Shie-Yuan Wang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
Email: shieyuan@cs.nctu.edu.tw

Chih-Che Lin
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
Email: linjc@cs.nctu.edu.tw

***Abstract*—NCTUns is a novel network simulator/emulator that has many unique features over traditional network simulators/emulators. It runs on Linux and many papers/projects have used it for wireless vehicular network research. According to the NCTUns website (http://NSL.csie.nctu.edu.tw/nctuns.html), as of December 24, 2009, more than 16,164 people from 137 countries in the world have registered and downloaded this software since its first public release on November 1, 2002. In its 6.0 release, NCTUns adds new supports for large-scale microscopic wireless vehicular network (WVN) simulation. In this paper, we introduce the important capabilities of NCTUns 6.0 for wireless vehicular network research.**

***Index Terms*—NCTUns, network simulation, 802.11p, IEEE 1609, vehicular network**

I. Introduction

The NCTUns network simulator/emulator [1][2] is a powerful and valuable tool for network research, planning, and education. It is an open-source software running on Linux and provides an easy-to-use integrated GUI environment for users to conduct simulation/emulation. Since its first release on November 1, 2002, it has been widely used by many researchers in the world. According to its website (http://NSL.csie.nctu.edu.tw/nctuns.html), as of December 24, 2009, more than 16,164 people from 137 countries in the world have registered and downloaded this software.

By adopting an innovative kernel re-entering simulation methodology, NCTUns provides many unique and important advantages over traditional network simulators and emulators. For example, NCTUns directly uses the real-life TCP/IP protocol stack in the Linux kernel to conduct simulations. This enables it to generate high-fidelity simulation results. In addition, NCTUns can directly run real-life application programs during simulation. This enables it to generate realistic traffic to drive a simulation and obtain more realistic simulation results.

Recently, the emergence of wireless vehicular networks (WVNs) has raised many research issues in both academia and industry. Tools for studying such new complicated networks are highly desired. To satisfy such needs, NCTUns supports the simulation and emulation of WVNs with the IEEE 802.11(p)/1609 protocol suite in its 6.0 release. NCTUns supports 1) agent-based and module-based mobility control for vehicles; 2) distinct vehicle mobility models; 3) road network construction; and 4) simulation and emulation of road side unit (RSU) and on-board unit (OBU) devices that are equipped with a radio, such as an IEEE 802.11(b) infrastructure/adhoc mode, GPRS, DVB-RCST, and IEEE 802.11(p)/1609 radio. As compared with other WVN simulators (e.g., SUMO [3]), NCTUns 6.0 provides a tightly-integrated simulation environment with a closed loop between network simulation and traffic simulation. As a result, many advanced researches such as “active safety” can be easily studied on NCTUns. This novel feature is presented in more details in our another article [4].

In the rest of this paper, we explain the features of NCTUns for WVNs in Sections II-V and then summarize this paper in Section VI.

II. Vehicle Mobility Control

NCTUns supports two modes of vehicle mobility control: agent-based and module-based. The agent-based vehicle mobility control (AVMC) uses individual user-level processes to control the movements of vehicles via the APIs provided by NCTUns. AVMC allows users to develop their own vehicle mobility model by writing a simple C/C++ program. In addition, using AVMC one can easily develop novel DSRC applications that simultaneously uses multiple (can be heterogeneous) network interfaces via the standard socket system calls.

However, AVMC controls the movement of each vehicle using an individual process. Each individual process inevitably consumes a certain amount of memory space for maintaining its execution. Such memory consumption may not be affordable on a single machine when the number of simulated vehicles becomes huge.

To solve the scalability problem with AVMC, NCTUns 6.0 provides a module-based vehicle mobility control (MVMC). MVMC, as indicated by its name, uses a special protocol module to control the movement of a simulated vehicle. This special protocol module is essentially a C++ object inside the simulation engine, which consumes minimum memory space to operate. In addition, as compared with AVMC, which has to use inter-process communication APIs with the simulation engine to control vehicle movements, MVMC uses intra-process (i.e., inside the simulation engine process) function calls to achieve the same goal. For these two reasons, in large-scale vehicular network simulations, MVMC is faster and more memory-efficient in comparison with AVMC.

The drawback of MVMC is that, because protocol modules are inside the simulation engine, it is difficult for them to use sockets for transmitting/receiving messages via standard socket interface APIs. (That is, the vehicle mobility control module cannot transmit/receive messages using the complex TCP protocol.) In the current implementation, MVMC only supports UDP message transmission and reception for simulated vehicles.

III. Vehicle Mobility Model

NCTUns 6.0 supports two vehicle mobility models. One is called the "strolling" model, which is commonly used in vehicular network simulators. In the strolling model, each vehicle makes a turn randomly when entering an intersection. The other model is called the "landmark-based movement" model. Using this model, one first divides the road map into grids and then specify several landmarks on each grid. Before moving, each vehicle first selects a landmark as its destination in a probabilistic manner. After selecting the landmark, the vehicle starts to move towards the destination. That is, when entering an intersection, the vehicle makes a turn based on the location of its chosen destination. After arriving at the destination, the vehicle then selects another landmark as its new destination in the same way. Such a process repeats until the simulation ends. The number of landmarks on each grid can be different. For a vehicle, it chooses a landmark as its destination based on a distribution of probabilities. By varying the number of landmarks placed on each grid and the above probability distribution, the generated mobility model can represent many different kinds of traffic pattern that correspond to daily lifes.

IV. Road Network Construction

NCTUns 6.0 provides two methods to create a road network for users. The first method is using the road network components provided by the GUI program to draw a road network. As shown in Fig. 1, one can click the icon of a road component on the GUI panel and then insert it into the field. The second method is importing a road map from a file. NCTUns allows users to import a SHAPE-format map file into the field. When loading a SHAPE-format map, the GUI program will automatically convert it into the road structures used by NCTUns. A snapshot of a road network constructed by loading a SHAPE-format map is shown in Fig. 2.

V. Protocol Implementation for the IEEE 802.11(p)/1609 Network

NCTUns 6.0 supports two types of nodes for the IEEE 802.11(p)/1609 WVN: "802.11(p) RSU" and "802.11(p) OBU." The former is usually deployed at an intersection as a fixed relay node while the latter is usually installed on vehicles as an end-user terminal device. These two types of nodes use the same protocol stack configuration, which is shown in Fig. 3.

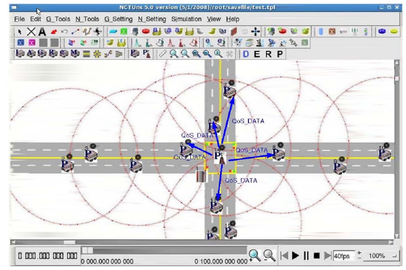

Fig. 1. Creating a road network using road network components

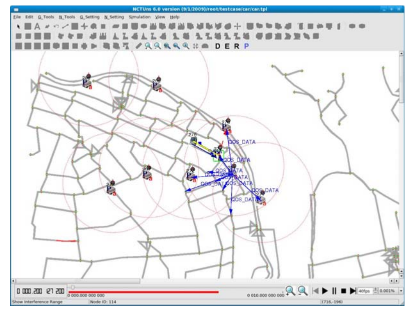

Fig. 2. Creating a road network by importing a road map

VI. Conclusion

NCTUns is powerful network simulator and emulator that uses real-life applications and Linux protocol stack to generate high-fidelity simulation results. In its 6.0 release, NCTUns adds more important functions for advanced wireless vehicular network research. Based on users' suggestions, we are further improving NCTUns to make it even better.

References

[1] S.Y. Wang et al., *"The Design and Implementation of the NCTUns Network Simulator,"* Computer Networks, Vol. 42, Issue 2, June 2003, pp. 175-197.

[2] S.Y. Wang, C.L. Chou, C.C. Lin, *"The Design and Implementation of the NCTUns Network Simulation Engine"*, Elsevier Simulation Modelling Practice and Theory, 15 (2007), p.p. 5781.

[3] The SUMO traffic simulation package, available at http://sumo.sourceforge.net/index.shtml.

[4] S.Y. Wang et al., *"NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic, Communication, and Network Researches,"* 1st IEEE International Symposium on Wireless Vehicular Communications (WiVec 2007), Sept. 30 Oct. 1, 2007, Baltimore, MD, USA.

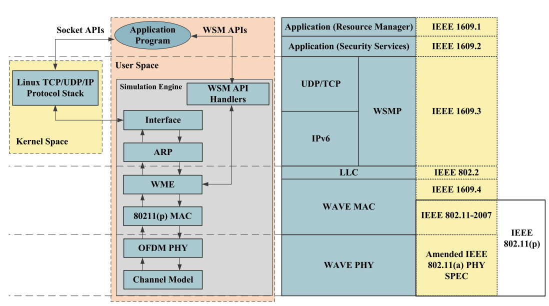

Fig. 3. Protocol implementation for the IEEE 802.11(p)/1609 WVN