

## Design and Implementation of Co-Simulation Platform for Wireless Networked Control Systems Based on Simulink and OPNET

<sup>1</sup>Lin Zhang, <sup>2</sup>Jianming Liu, <sup>2</sup>Xin Wang, <sup>1</sup>Hongzhou Li and <sup>1</sup>Zhiyong Peng

<sup>1</sup>School of Electronic Engineering and Automation,

<sup>2</sup>School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China

**Abstract:** In this study, we have a research of the design and implementation of co-simulation platform for wireless networked control systems based on Simulink and OPNET. The real time simulation platform for wireless networked control systems were designed and realized by using Simulink and OPNET. Simulink and OPNET were adopted to simulate the real time control and network performance separately. The communication between Simulink and OPNET was realized by using the technology of MATLAB engine interface. Combining the virtues of Simulink and OPNET, Co-simulation for WNCS was implemented. A simulation example of wireless networked system was given. It clearly shows that the data transmission of wireless channel was impacted on Simulink model. Simulation results became more authentic and idealistic simulation environment of OPNET was overcome. Thus the effectiveness of the Co-simulation platform based on Simulink and OPNET was proved.

**Keywords:** Engine interface, OPNET, simulink, simulation platform, wireless networked control systems

### INTRODUCTION

In recent years, a new type of control system has been widely concerned on. It can achieve the exchange of information between the sensor of geographical distribution and mobile field, controller and actuator and achieve real-time feedback control of the controlled object by using wireless communication network. We called it "Wireless Networked Control Systems (WNCS)". Compared with the traditional point-to-point or wired networked control systems, wireless networked control systems have the advantages of resource sharing, high diagnostic capabilities, increased system flexibility and reliability. However, information transmission of the network has the disadvantages of time delay, packet loss, timing disorder and so on due to network bandwidth, packet size and other physical constraints. This makes it more complex for system analysis, design and simulation. The networked control systems are related to control theory, computer networks and optimization theory. We cannot get a satisfactory control quality by using traditional delay control theory, because of the existence of random variable network delay. In order to study such problems more in-depth and verify the research results of WNCS, we must carry out simulation experiments (Wang *et al.*, 2011). Therefore, design and implementation of a simulation platform for WNCS has important significance to further development of the wireless networked control systems.

In this study, we have a research of the design and implementation of co-simulation platform for wireless networked control systems based on Simulink and OPNET. The real time simulation platform for wireless networked control systems were designed and realized by using Simulink and OPNET. Simulink and OPNET were adopted to simulate the real time control and network performance separately. The communication between Simulink and OPNET was realized by using the technology of MATLAB engine interface. Combining the virtues of Simulink and OPNET, Co-simulation for WNCS was implemented. A simulation example of wireless networked system was given. It clearly shows that the data transmission of wireless channel was impacted on Simulink model. Simulation results became more authentic and idealistic simulation environment of OPNET was overcome. Thus the effectiveness of the Co-simulation platform based on Simulink and OPNET was proved.

### SIMULATION OF THE WIRELESS NETWORKED CONTROL SYSTEMS

Simulation of WNCS is an important part of the research areas of the wireless networked control systems. The simulation of WNCS should not only support the modeling of the control system, but also carry out the simulation of the network. However, there is little software of support function in these two areas

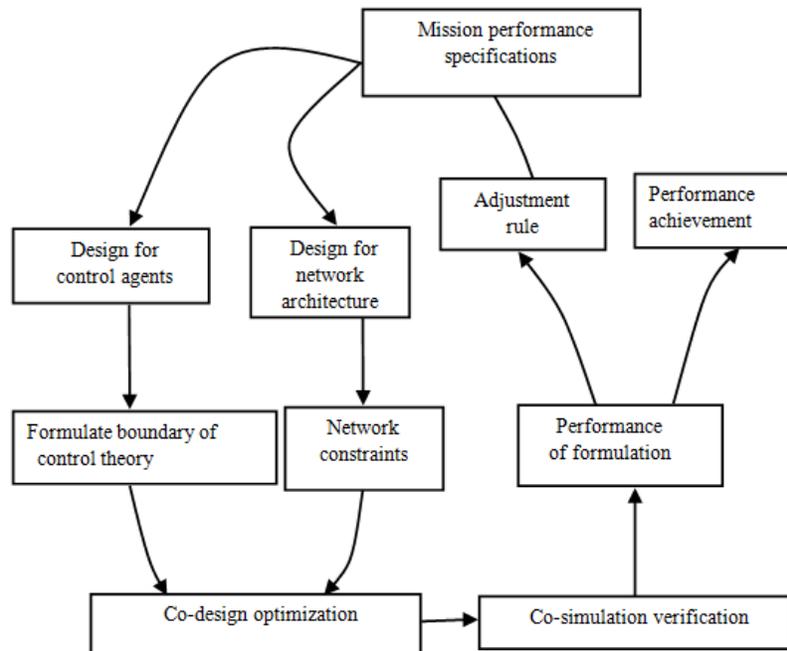


Fig. 1: Overall WNCs design loop

at present. So it becomes a development direction of the WNCs simulation study that how to combine effectively between control system simulation software and network simulation software to constitute WNCs Co-simulation platform. The ideal WNCs simulation platform should have the following quality (Li *et al.*, 2006):

- **Authenticity:** In order to do in-depth study, the WNCs simulation platform should be able to truly reflect the characteristics of the entire system.
- **Flexibility:** WNCs is a class of distributed systems and it uses a different structure of the network, this platform should be able to easily achieve the simulation for WNCs with different structures and networks.
- **Scalability:** This platform should have good scalability, so that we can add new algorithms or modify the network topology and so on.

Based on the above analysis, this study designed a real-time simulation platform based on Simulink and OPNET. It is combined with conventional control systems' simulation and network simulation. As a widely used system development platform, OPNET uses discrete event-driven mechanism. It has very high simulation efficiency; and it is easy to model the real network by using Object-oriented modeling approach and providing a graphical user interface, so it can provide a good development environment for the modeling of network systems. In short, OPNET provides us an excellent network simulation platform,

but it does not have the ability to simulate the control systems. It will be a very significant thing to combine the Simulink which has a powerful capability to generate the control systems model and the OPNET which has a powerful capability of network simulation by using the technology of interface. The simulation platform will have strong modeling capability and realistic network environment. It can achieve comprehensive simulation to the control systems and network parameters and it provides many experiments for the theoretical study of stability and real-time of networked control systems. The overall design of WNCs' simulation is shown in Fig. 1.

### CO-SIMULATION ARCHITECTURE

Generally, there are three methods to achieve the data exchange between Simulink and OPNET, as follows:

- Technology of external application program interface by MATLAB provided
- The High-Level Architecture module (HLA) by OPNET provided
- Invoke the external interface module of OPNET (Esys, External System) directly

**Interface technology of MATLAB:** MATLAB has many interfaces so that it can interact with external program (External Interfaces/API), such as Engine Interface of MATLAB, Active method, MATLAB Compiler (mcc/add-in), invokes MATLAB through

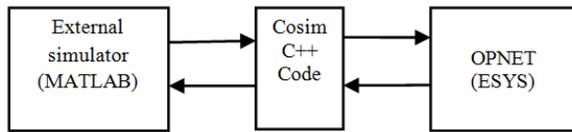


Fig. 2: Structure of Cosim between external system and OPNET

invoke MATLAB C/C++ math library, MATCOM and MEX interface technology (Li and Ma, 2003). This study intends to use the technology of MATLAB Engine Interfaces to realize the connection of OPNET and MATLAB since it is easy to use and has no limited of function. MATLAB Engine is a set of interface functions which is provided by MATLAB. It supports for the language of C, C++, FORTRAN, etc. The users can control MATLAB in other programming environment through this interface functions. MATLAB Engine uses a client/server computing model in a specific application process. MATLAB can run in the background as an ActiveX server and application program can run as a front-end client by interacting with MATLAB through Engine. To achieve data exchange, we need to use the external interface (mx interface) provided by MATLAB. Its purpose is to allow programs written in C language to invoke a library of functions written in MATLAB. After starting the OPNET simulation, at the same time MATLAB simulation is invoked. The pointer which is provided to corresponding memory space of OPNET simulation system will send MATLAB command to the MATLAB Engine. As long as this pointer in the header files of OPNET' process model is declared, this MATLAB engine pointer can be shared in different process model and it can pass the variable values between OPNET and MATLAB by using specific functions. Then the co-simulation is implemented (Chen *et al.*, 2011).

**High-level architecture:** HLA is flexible and reusable software architecture and it is the result of a framework to create distributed simulation based on components (Chen, 2004). It supports co-founded and run of multiple emulators. Each emulator can simulate some aspect of a complex system. OPNET-HLA interface provides the necessary mechanism of sharing common objects, interacting messages, as well as maintaining time synchronization for multiple emulators.

**OPNET external system access:** External System Access (ESA) provides the programming interface of the co-simulation. It is mainly to solve the problem of interactive communications and simulation synchronization between OPNET internal and external systems and the external system is to participate

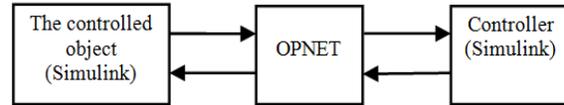


Fig. 3: Connective structure between Simulink and OPNET

forwardly or passively in entire co-simulation by the form of exe or dll.

The structure of co-simulation based on OPNET is shown in Fig. 2; it consists of MATLAB, co-simulation module Cosim, OPNET and its expansion. In order to run co-simulation, we must establish the code of the external system. Cosim, as an external module of the OPNET, is a bridge between demand-side program and OPNET. The idea of co-simulation (Harding *et al.*, 2007) is using external interfaces and interrupt of the OPNET external system-defined to achieve the data exchange between OPNET and Cosim through ESYS external module of OPNET. Communication model in OPNET and Cosim code was statically compiled according to SD document by OPNET. At last, a simulation program which can run independently is formed.

**Synchronization mechanism of co-simulation:** In this study, we built a co-simulation platform of WNCS. It achieves the data exchange between OPNET and MATLAB by using the technology of MATLAB engine interface. According to the design ideas of WNCS, the design of network and control section is completely independent in the system of WNCS, the controlled object and controller is modeled in Simulink and the network architecture is designed in OPNET. If the network model and the plant model are known, the simulation platform can simulate the system with different controllers without affecting the network model and plant model. The structure of connection between Simulink and OPNET is shown in Fig. 3.

During the simulation process, it needs to interact with Simulink and OPNET. But Simulink is used in time-driven mechanism and OPNET is used in discrete event-driven mechanism. How to make both the simulation clock to achieve synchronization is particularly critical. In co-simulation, simulation time is led by OPNET. OPNET simulation pauses at time 0. At the same time, the OPNET node models invoke the corresponding Simulink models. After initialization, the Simulink control object model pauses at time 0. OPNET continues execution and pauses at sampling time T1. It passes a command to the Simulink control object model to excite until time T1. When Simulink suspended at time T1, the OPNET node plant reads the plant state from the Simulink model and generates a sample packet. The total closed-loop delay is denoted as t1, which is the

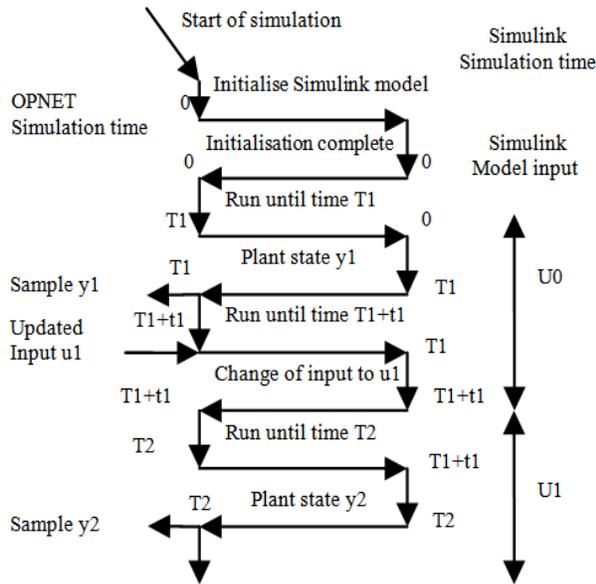


Fig. 4: OPNET-Simulink synchronization

delay of sensor to controller plus the delay of controller to actuator. The control object node will receive a control packet at time  $T1 + t1$ , then OPNET will issue a command to the Simulink control object model to execute until time  $T1 + t1$  with the previous input  $u0$ . The Simulink control object model suspended and the input is changed from  $u0$  to  $u1$  at time  $T1 + t1$  (Hasan *et al.*, 2009). OPNET continues to run the simulation in this fashion when Simulink finishes execution. The synchronization mechanism between Simulink and OPNET models is explained in Fig. 4.

**Debug and configuration of co-simulation platform:**

To achieve the joint commissioning of MATLAB and OPNET, we need to set accordingly the compilation and link in OPNET; the purpose is to allow the compiler of OPNET to join some of the header files and libraries in MATLAB. Software version is OPNET Modeler 14.5 and MATLAB R2007a in the simulation experiments. It is set up before compile and run, as follows (Gao, 2010):

Open the menu of Edit-Preferences in OPNET and find the parameter named `commp-flags-common`, add to the parameter value `"/I<MATLAB header file directory>"` in the back of the parameter value, for example, `"/ IC: \MATALB\ the extern\include"`, so the modified value is `"/W3/D-CRT-SECURE-NO-DEPRECATE/IC:\PROGRA~1\OPNET\145~1.A\M odels\std\include/IC: \MATLAB\extern\include"`. Its role is that make the OPNET contains some necessary header files of MATLAB. Find the parameter named `bind-shobj-libs` and add to the `"libmx.lib libmat.lib libeng.lib"` in the back of the parameter value. These three files are also indispensable library files. During

operation process, we should write these parameters in accordance with the installation of MATLAB directory.

**Design of WNCS simulation scheme:** WNCS is a wireless networked control system. We use OPNET to simulate wireless LAN. The simulation will inevitably involve the modeling of wireless channel. OPNET wireless channel modeling is composed of 14 pipeline stages; they are (Gong and Cao, 2010): receiver group, transmission delay, physical accessibility, channel match, transmitter antenna gain, propagation delay, receiver antenna gain, received power, background noise, interference noise, signal to noise ratio, bit error rate, error allocation, error correction. OPNET Modeler uses a framework of pipeline process for Wireless Channel Modeling, although this framework can reflect correctly the process of the wireless transmission, the realization of the default model library is not quite accurate.

On the one hand, the implementation of model library ignores the difference between the channel information rate and air rate. It can only be applied to the channel of both same rates, such as the channel which uses binary modulation, like BPSK, QPSK. On the other hand, the implementation of transmission of multi-path, attenuation, background noise, thermal noise and the simulation of environment. It needs the support of the shared library of propagation model. However, the propagation model of OPNET Modeler is very limited. It has only three kinds: TIREM, Longley-Rice and Frees pace. And it requires user to define other commonly model, such as Rayleigh model and AWGN model. However, the Rayleigh model and AWGN model are the most basic models. They have a profound impact on the transmission performance of communication channel. If we can import these models into the channel, there will be a great help to the expansion of simulation applications and the improvement of the correctness of simulation results. Meanwhile, it overcomes the inadequate of idealistic simulation environment of OPNET. Fortunately, there are many off-the-shelf resources in MATLAB. They can be easily combined with other programming languages or simulation software and achieve complex scientific computing and simulation. If we can effectively combine OPNET and MATLAB, then we will be able to take advantage of these resources. This will undoubtedly be of great help to improve the authenticity and credibility of simulation results.

The pipeline process established in this study can identify the specific modulation mode. It is able to make the data bits translate into a number of symbols. The rate of the transceiver can reflect accurately the rate of the air and it can pass the current modulation mode and the conditions of signal to noise ratio to MATLAB. It can

use the analog modulation to add white Gaussian noise and the process of calculate Rayleigh attenuation and demodulation, then it can get a certain number of bit error of the packet and continues to accumulate. Finally, we can get the specific number of bit error of the entire packet. The number of symbols will translate into bits of packet when it receives data. According to this number of bit error, we can calculate the bit error rate and determine whether beyond the correction threshold, the excess packets were not received.

This design scheme can be divided into four stages; they are the preparation of the realization functions of pipeline stage, the preparation of the process model, the establishment of the node model, the preparation of scenarios:

- **The preparation of the realization functions of pipeline stage:** Here, it needs to write three functions of pipeline stages and an external file, respectively named as “MATLAB-err”, “MATLAB-txdel”, “MATLAB-ecc” and “pipe-support”. The role of external file “pipe-support” is to control the functions of MATLAB, it will distinguish the interact functions of MATLAB engine and the realization functions of pipeline stages. The realization functions of “MATLAB-err” stage should be declared to quote the external file “pipe-support” before compiling. At first, these functions should be placed to the folder of op-models and then they were compiled one by one.
- **The preparation of the process model:** In order to control the opening and closing of the MATLAB engine, we need to create a process model; it is named “engine-connect”. And its Header Block was defined as follows:

```
#include “engine.h”
Engine *ep;
```

At the beginning of simulation, the program of opening the MATLAB engine in “Enter Execs” of “run” state was completed:

```
If (op_intrpt_type () ==
OPC_INTRPT_BEGSIM)
{
if (!(ep = engOpen (“0”)))
{
op_sim_end (“t start MATLAB engine”,
“”, “”, “”);
}
}
else if (op_intrpt_type () ==
OPC_INTRPT_ENDSIM)
{
```

```
engClose (ep);
}
```

- **The establishment of the node model:** In the simulation scene, we need to create two kinds of node model. They are respectively named “MATLAB-starter” and “MATLAB-node”. The first node model contains a process model, it is “engine-connect”. The second node model changes from the “wlan-wkstn-adv” model in OPNET. As follows, in the “wlan-port-rx0” module, the attribute of “error model” was changed from “wlan-err” to “MATLAB-err”, the attribute of “ecc model” was changed from “wlan-ecc” to “MATLAB-ecc”. In the “wlan\_port\_tx0” module, the attribute of “txdel model” was changed from “wlan-txdel” to “MATLAB-txdel” and the attribute value of “ecc threshold” was changed from 0.0 to 0.5.
- **The preparation of scenarios:** Place a “MATLAB-starter” node, 50 “MATLAB-node” nodes and a server in the range of 10×10 m. Each node adopts Ad hoc mode. The routing protocol adopts DSR protocol and business generation adopts the ON-OFF mode.

## SIMULATION RESULTS AND ANALYSIS

During the operation process, the taskbar will display the icon of MATLAB engine and shut it down automatically after the end of run. We have taken before and after comparison of the co-simulation experiment. It is compared with these following performance indicators. They are Wireless LAN Delay, Load, Media Access Delay, throughput, in Ftp. Upload Response Time, in Ftp. Download Response Time, in DSR. Total Traffic Sent, in DSR. Total Traffic Received. The simulation time is set to 1 h. The results were shown in following figures.

Figure 5 to 6 is a comparison chart of the network topology before and after the co-simulation. It is “wlan-wkstn-adv” node in Fig. 5 and it is “MATLAB-starter” and “MATLAB-node” node in Fig. 6. We achieve the data exchange between Simulink and OPNET by using the technology of MATLAB engine interface in Fig. 6. The comparison chart of simulation results before and after co-simulation was shown in Fig. 7 to 14. The blue line stands for the simulation results before co-simulation and the red line stands for the simulation results of co-simulation. As can be seen from these figures, there is great difference of the two curves. This shows that the introduced models such as Rayleigh model and AWGN model of Simulink through co-simulation are indeed affecting the data transmission of wireless channel. It changes all aspects of simulation

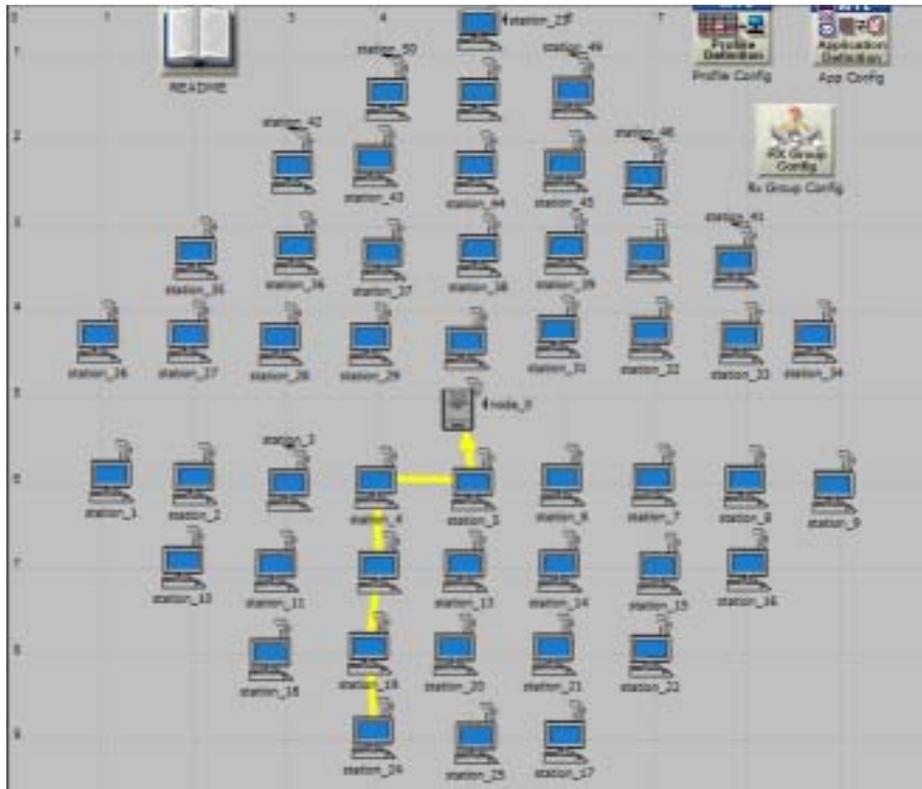


Fig. 5: Network topology before co-simulation

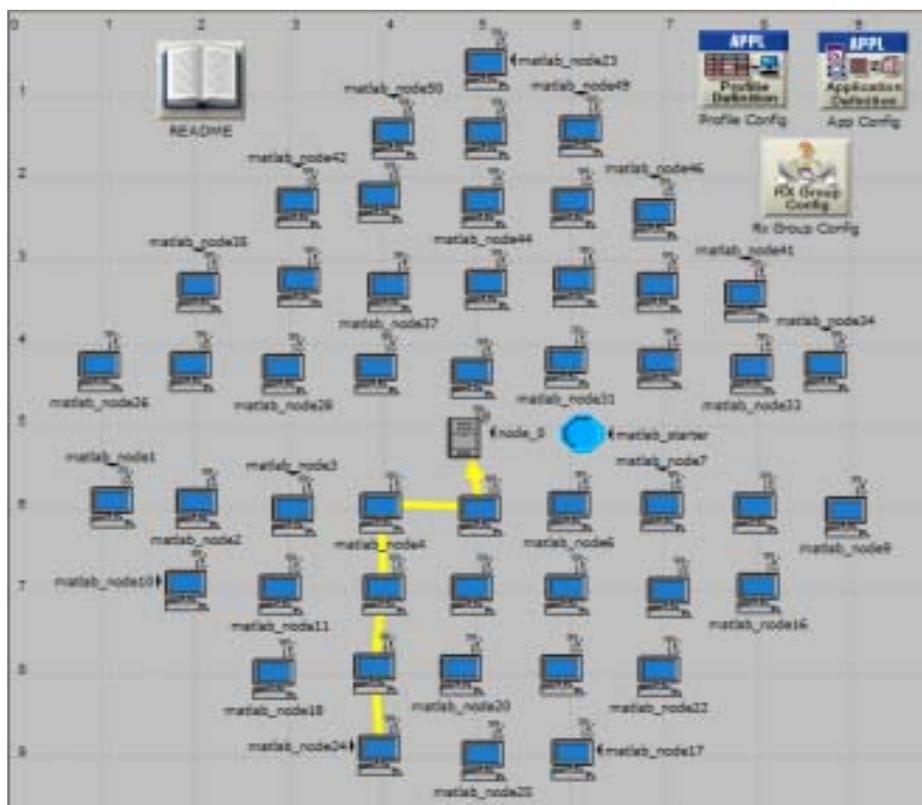


Fig. 6: Network topology of co-simulation

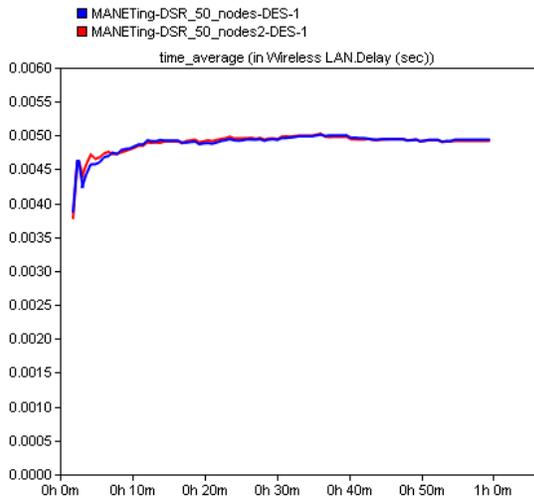


Fig. 7: Delay

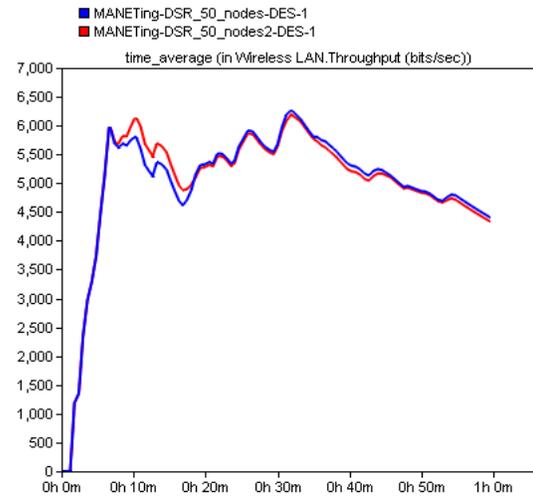


Fig. 10: Throughput

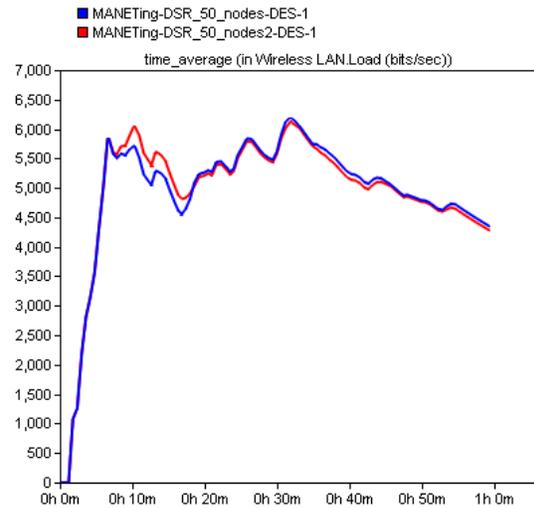


Fig. 8: Load

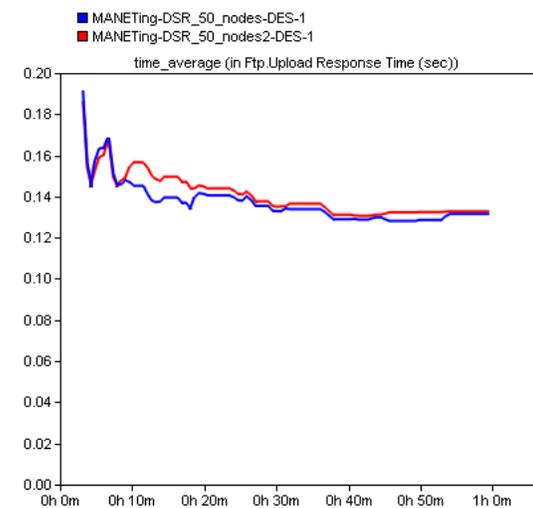


Fig. 11: In Ftp.upload response time

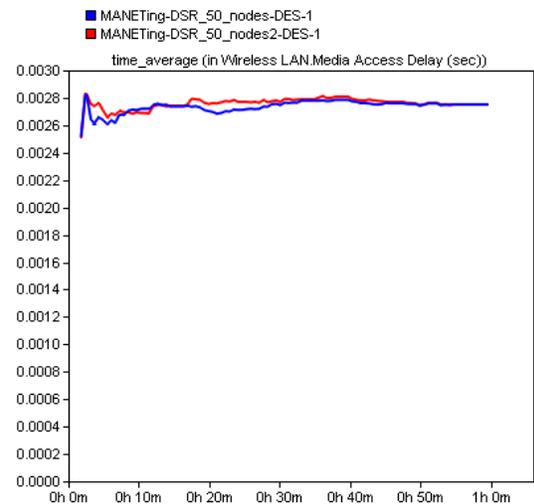


Fig. 9: Media access delay

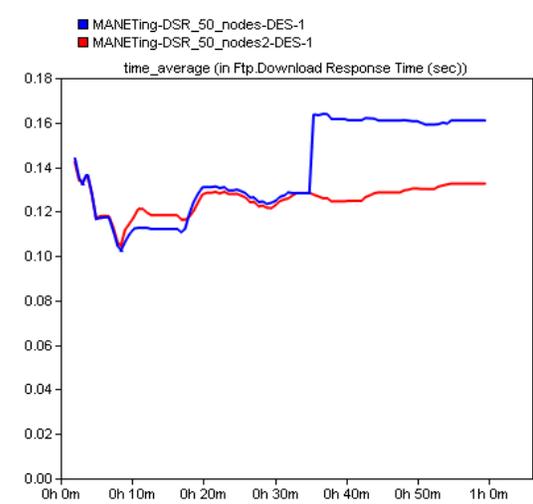


Fig. 12: In Ftp.download response time

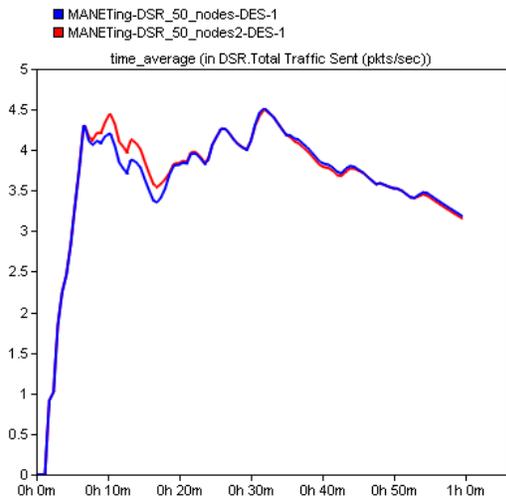


Fig. 13: In DSR.total traffic sent

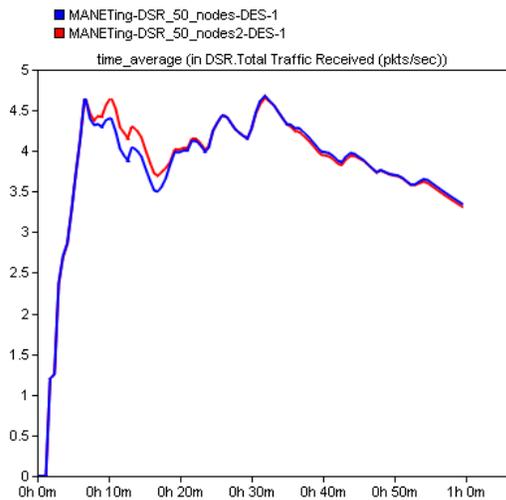


Fig. 14: In DSR.total traffic received

results and makes the simulation closer to the real environment of wireless network. The simulation results more truly reflect the network performance.

### CONCLUSION

As a new topic in control theory, WNCS causes great concern of the majority of researchers because of its unique advantages. An excellent simulation platform plays an irreplaceable role in the course of the study of WNCS. According to the overall design idea of WNCS and combining the virtues of Simulink and OPNET, using the technology of MATLAB engine interface, joint simulation of Simulink and OPNET was implemented successfully and a joint simulation platform of WNCS was formed. The platform combines the virtues of control system modeling and real-time network scheduling. It has a strong capability of system modeling and network simulation. The simulation

results also show the superiority of the co-simulation platform. Therefore, we provide an excellent simulation platform for WNCS researchers.

Further work is to establish more complex simulation platform of WNCS. It can convert all program of MATLAB to C language and compile it in OPNET. The speed of simulation will be further improved.

### ACKNOWLEDGMENT

The work has been sponsored by the National Natural Science Foundation of China (61162008), Foundation of Department of Education of Guangxi Province, China (201101ZD006), the National Natural Science Foundation of China (60762002), Open Project of Key Laboratory of Trusted Software of Guangxi Province, China (kx201101), Subsidy Scheme of talents of university of Guangxi Province, China (201065), Innovation Projects of Graduate Education of Guangxi Province, China (2010105950812M25).

### REFERENCES

- Chen, M., 2004. OPNET Network Simulation. Tsinghua University Press, Beijing, China.
- Chen, Y., Y. Song and M.R. Fei, 2011. Interactive cosimulation based on simulink and OPNET. J. Syst. Simul. Technol., 7(3): 242-247.
- Gao, S., 2010. OPNET Modeler Simulation Modeling Decryption. Publishing House of Electronics Industry, Beijing, China.
- Gong, T. and X.Y. Cao, 2010. Study of wireless channel in OPNET and its application in system simulation of the IEEE 802.11a. J. Guangdong Commun. Technol., pp: 66-71.
- Harding, C., A. Griffiths and H. Yu, 2007. An interface between MATLAB and OPNET to allow simulation of WNCS with MANETs. Proceedings of IEEE International Conference on Networking, Sensing and Control, pp: 711-716.
- Hasan, M.S., H. Yu, A. Carrington and T.C. Yang, 2009. Co-simulation of wireless networked control systems over mobile ad hoc network using SIMULINK and OPNET. IET Commun., 3(8): 1297-1310.
- Li, P.F. and R.G. Ma, 2003. Matlab interface and generation of random numbers for simulation. J. Chang'an Univ. (Natural Science Edition), 23(6): 111-114.
- Li, H.B., F.G. Wu, Z.Q. Sun and F.C. Sun, 2006. Design and implementation of simulation platform of networked control system. J. Syst. Simul., 18(6): 1700-1704.
- Wang, Q.F., H. Chen and P. Wang, 2011. Design and implementation of simulation platform for networked control systems based on NS2. J. Syst. Simul., 23(2): 270-274.