Test bed for network protocols optimization

Afan Ceco, Member IEEE; Sasa Mrdovic, Senior Member IEEE

Abstract— **This paper describes the test platform for verifying the functionality of network protocols and for optimization of their parameters. The test bed is made using combined OPNET simulator and MATLAB development environment. This test platform connects OPNET network protocols simulator with MATLAB development environment in the way that OPNET runs simulations of network traffic, with the predetermined parameter values, while MATLAB executes the script with a mathematical algorithm, which optimizes parameters listed in OPNET simulator.**

Keywords— **test platform, OPNET, MATLAB, network protocols, optimization**

I. Introduction

It is very often necessary to test network protocols in the simulation environment. The reason why it is necessary is the impossibility of testing in real networks, so as not to disturb the real traffic. Network simulators (e.g. OPNET) have no support for non-standard parameters or their dynamic change. On the other hand, MATLAB is suitable for testing ideas and concepts, because it represents a high level programming language and a development environment. Enabling these two components connection together provides the best of both worlds.

The network simulators such as NS-2 [1], the NS-3 [2], OPNET [3], OMNET++ [4] or others, allow checking the validity of networking concepts, as well as changes in the network protocols. However, they lack the power of specialized software for mathematical calculations, such as MATLAB [5], when it comes to complex mathematical operations (optimization, 3-D models, etc.). In this paper, a test platform is presented, which can be used to verify the functionality of the new network protocol, that would include optimization techniques, or verification of the existing protocols, in addition to optimization of their parameters. It consists of a connected simulator OPNET Modeler 14.5 and a development environment MATLAB R2012.

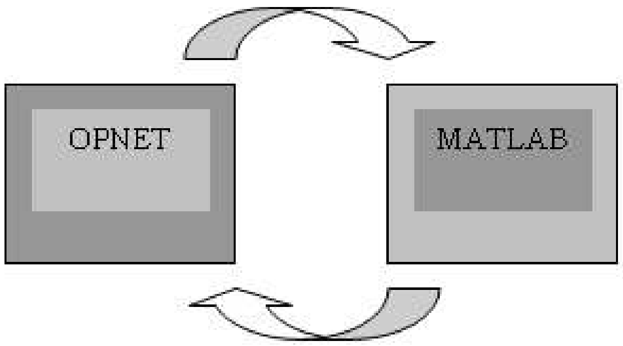


Fig. 1. Link between OPNET and MATLAB

This test platform connects OPNET simulator of the network protocols with MATLAB development environment in the way that OPNET runs simulations of network traffic, with the predetermined parameter values, while MATLAB executes the script with a mathematical algorithm, which optimizes the parameters listed in OPNET simulator.

The first part of this paper presents the situation in the subject area. The second part describes OPNET simulator. In the third part, MATLAB development environment is presented. In the fourth part, the test platform is explained, consisting of a paired OPNET and MATLAB, and its implementation. At the end of this paper, the results of simulations are presented, obtained by optimizing the parameters, and executed on the subject test platform.

II. Review of the Subject Area

In the research papers which can be found in this area, the models for linking different simulators are described, as well as for linking simulators and other software tools. Very often,

Afan Ceco, BH Telecom d.d. Sarajevo, Masarykova 46, 72000 Zenica, B&H (e-mail: afan.ceco@bhtelecom.ba)
Sasa Mrdovic, Faculty of EE, University of Sarajevo, Zmaja od Bosne bb, 71 000 Sarajevo, B&H (e-mail: sasa.mrdovic@etf.unsa.ba)

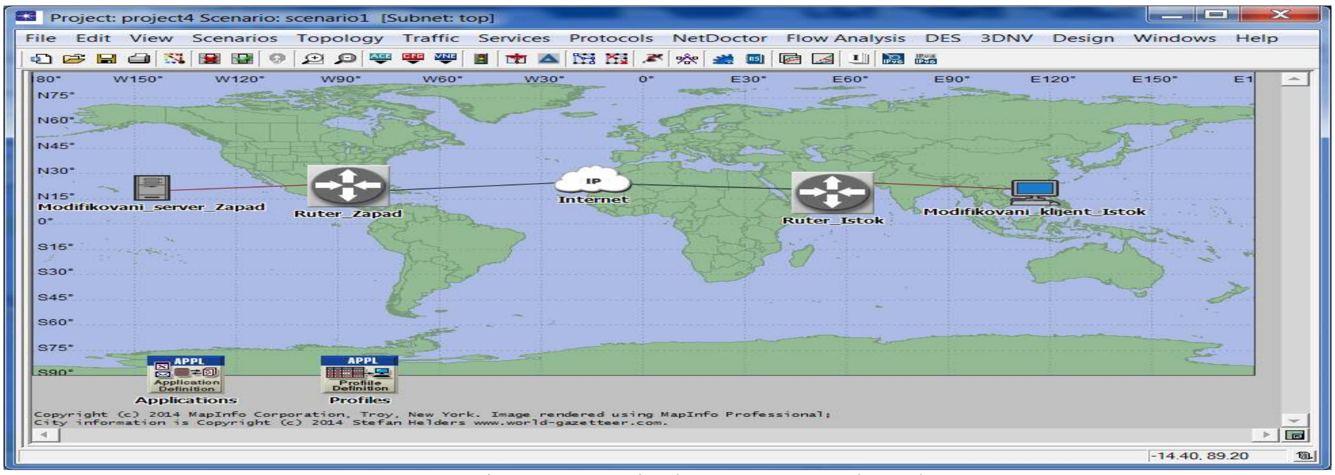

Project: project4 Scenario: scenario1 [Subnet: top]
File Edit View Scenarios Topology Traffic Services Protocols NetDoctor Flow Analysis DES 3DNV Design Windows Help
Modifikovani_server_Zapad
Ruter_Zapad
Internet
Ruter_Istok
Modifikovani_klijent_Istok
Applications
Profiles

Fig. 2. OPNET Simulator – Test network topology

it is the case of a so-called co-simulation, which represents a problem division into two or more parts, which are simultaneously simulated on a variety of simulators. The second most common use is to connect two simulators to validate the results from one simulator, by checking them on the second simulator (from another manufacturer). Co-simulation, as well as the use of two different simulators together, is the topic of Paper [6]. The authors describe a platform for co-simulation, which consists of connected simulators OPNET and SIMULINK. SIMULINK environment for simulation is based on block diagrams and on design of models [7]. Both simulators are executed in parallel, and interact with the synchronization, whereby the main simulator is OPNET. In Paper [8], the authors present the implementation of connectivity of OMNET++ simulator and MATLAB development environment, with an emphasis on co-simulation. Paper [9] presents an example of connecting OPNET and MATLAB, provided that it is done for the purpose of validation on MATLAB of the results obtained on OPNET simulator.

Unlike the above described models of linking the simulators, we herein present the connection of OPNET and MATLAB in a common test platform, where they act as a single combined simulator, which utilizes the advantages of both software environments. This test platform uses an iterative principle in order to reach the best possible results, and through a number of (automated) starting and running of simulations, it leads to the optimization of the simulation parameters.

III. OPNET SIMULATOR

OPNET is a simulator of network protocols and communications, a product of the company named Riverbed, and there are several different variants of this product. A very popular variant is called “OPNET IT Guru Academic Edition”, especially in the academic community, because it was free for educational use. However, in this version, simulator users were not allowed to modify the source code of the simulator. At present, the current replacement for this variant is called “OPNET Modeler Academic Edition”, and was presented in 2014. In this paper, we used a commercial version of OPNET Modeler, mostly because this version of the simulator has the option of changing the source code. This allows modification of the complete behavior of network protocols being a part of the simulator. Figure 2 shows a GUI of OPNET simulator with a built topology network that is used to test the proposed linking of OPNET and MATLAB. The topology consists of a server, which is connected via router to the internetwork, then the internetwork itself (Internet), and a client, who is also connected via router to the internetwork.

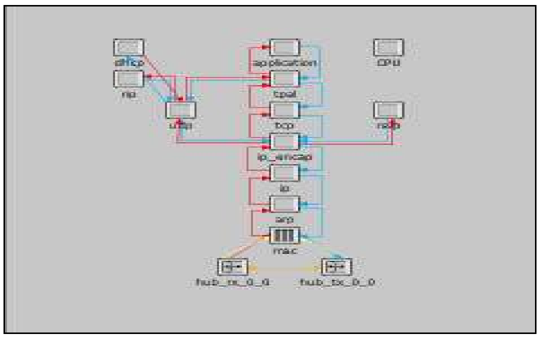


Fig. 3. OPNET Simulator – Display of protocols for client node

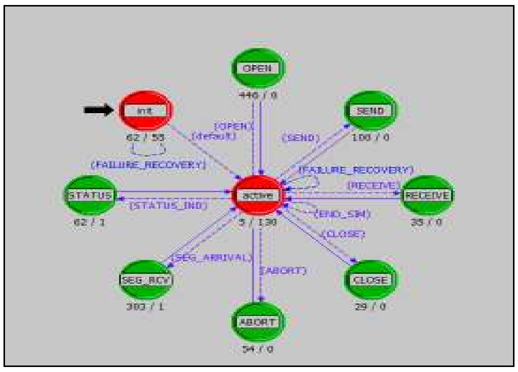
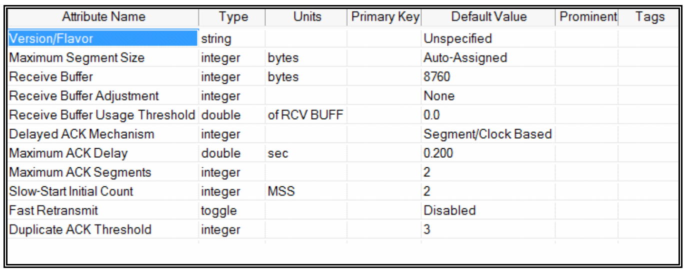


Fig. 4. OPNET Simulator – State diagram for TCP protocol

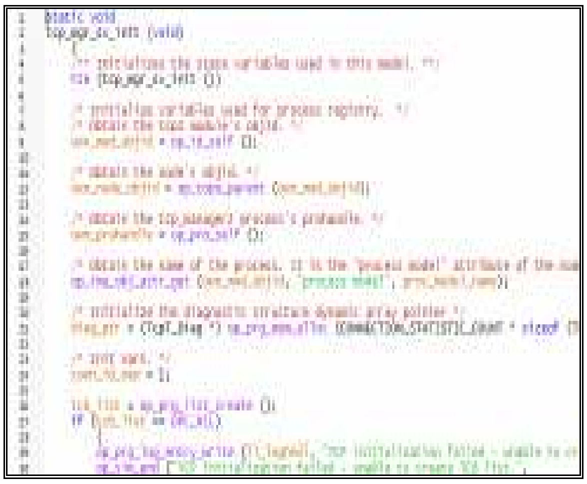
Figure 3 presents the protocols for the client node, i.e. all protocols that are available on the simulator for that node and that can be changed. Figure 4 shows the state diagram for one of the protocols on the client node, and in this case it is the TCP protocol. It is important to point out the fact that each of the states in the diagram can be opened, with its code in C/C++ displayed, which can be changed, if necessary.

Figures 5 and 6 show the parameters of the TCP protocol, as well as the function block with the C/C++ code that can be changed. The parameter values in Fig. 5 can be changed programmatically by changing the code as shown in Fig. 6. Once we perform the change in the code of the simulator, it is necessary to recompile the simulator.



Attribute Name	Type	Units	Primary Key	Default Value	Prominent	Tags
Version/Flavor	string			Unspecified		
Maximum Segment Size	integer	bytes		Auto-Assigned		
Receive Buffer	integer	bytes		8760		
Receive Buffer Adjustment	integer			None		
Receive Buffer Usage Threshold	double	of RCV BUFF		0.0		
Delayed ACK Mechanism	integer			Segment/Clock Based		
Maximum ACK Delay	double	sec		0.200		
Maximum ACK Segments	integer			2		
Slow-Start Initial Count	integer	MSS		2		
Fast Retransmit	toggle			Disabled		
Duplicate ACK Threshold	integer			3		

Fig. 5. OPNET Simulator – Display of parameters



```
static void
tcp_mgr_sv_init (void)
    {
    [illegible]
    }
```

Fig. 6. OPNET Simulator – Function block

In the presented case, we have used MS Visual C++ 9.0 (MS Visual Studio 2008) on MS Windows 7. It is necessary to set the environment variables for the compiler and simulator to be bound. All the details related to the changing code of OPNET Modeler are explained in Book [10]. It should be noted that the operating system, and simulator, and compiler, should be from the same generation, so that they appeared on the market in about the same time, because otherwise some compatibility issues arise.

By changing the code in OPNET simulator, the existing protocol can be changed to work differently, or we can even create an entirely new network protocol. The precondition is, of course, that we have installed C/C++ compiler, and that the environment variables are set.

IV. MATLAB DEVELOPMENT ENVIRONMENT

MATLAB is a development environment and a high level programming language for numerical and matrix computation. It is a product of the company named MathWorks, which offers a SIMULINK simulator as well.

MATLAB represents all data in the form of arrays. The routines serving to manipulate the arrays in MATLAB are starting with the prefix mx. Import and export of data to and from MATLAB can be made in several ways. Below are presented some of them.

MATLAB can invoke functions and procedures written in the C programming language or in Fortran. In this, the so-called MEX files are commonly used, which represent the interface between MATLAB and subroutines written in C, C++ or Fortran. When compiled, MEX files are dynamically loaded and allow other code to be called within MATLAB, as if it were its own functions. With a MEX file, of course, we can read or write data. A detailed explanation can be found in the manual issued by the MATLAB development environment manufacturer [11].

In contrast, the so-called MAT files allow the import and export of data to or from MATLAB environment in a direct way. MAT files are actually files that MATLAB uses to store data on the disk. MAT files provide an easy and convenient mechanism for transporting data between different platforms. Also, they allow the import and export of data into MATLAB from some other MATLAB or stand-alone applications. In order to simplify the use of MAT files from other applications, a library of access routines is offered, and they all begin with the prefix mat, which makes reading and writing of MAT files very easy using C or Fortran program [11].

It is possible to load data into MATLAB also from ASCII files. Such files must have data written in ASCII format, with rows of fixed length ending with the transition to a new line, as well as spaces that separate the numbers. Data is retrieved from this file by using the load command, and are recorded using the save command [11].

V. COUPLING OF OPNET AND MATLAB

In this paper, we present a test platform that connects OPNET simulator of network protocols with MATLAB development environment. OPNET runs simulations of network traffic, with the predetermined values of parameters, such as the percentage of a packet loss, and other parameters of the TCP connection. Upon completion of the simulation, the values of the variables affecting the parameters we have been looking to improve are recorded, and the script with the mathematical algorithm, which optimizes the above parameters, is executed in MATLAB, for the purposes of OPNET simulator. In Fig. 7, a flow diagram of the execution of the simulation and optimization is shown, which can be applied to any communication protocol. The initial parameter values (x1, y1) are used in OPNET simulator as input parameters. After the simulation ends, the result obtained is shown on the diagram as the value (R). This value (R) is imported into MATLAB in order to carry out the effectiveness check of the initial parameters values. Subsequently, the optimization algorithm in MATLAB is changing the initial parameters (x1, y1) to (x2, y2), which is presented in Fig. 7 by operator λ. These parameters (x2, y2) are recorded now as the initial parameters (x1, y1). During the next start-up of simulation in OPNET, the new input parameters are being loaded. The result (R) is returned back to MATLAB and is compared with the previous result. If the result is better, the optimization algorithm continues in this direction with the further modification of parameters. If the result is worse, the optimization algorithm rejects the previously caused modification. The link between OPNET and MATLAB can be accomplished in several ways, and these methods can be divided into two groups: using files or software methods for direct applications linking. In this case, the connection is achieved using ASCII files and by recording the variables in the file from one software, and then loading the same variables from the given file in another software.

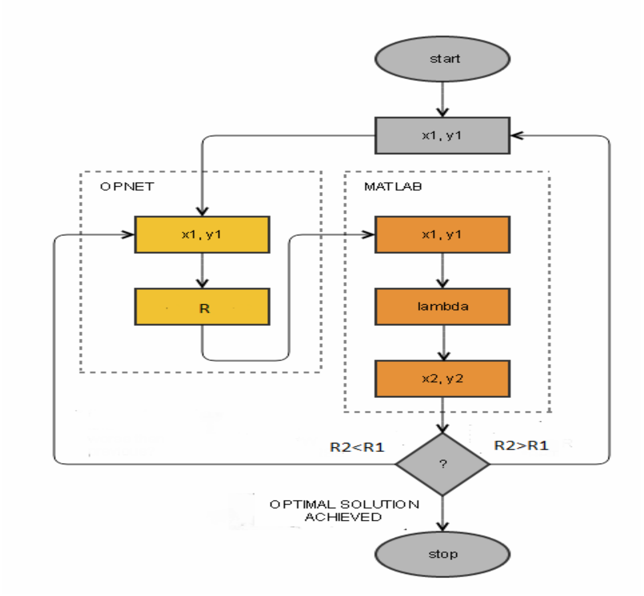


Fig. 7. Flow Diagram – Execution of simulation with the relationship between OPNET and MATLAB

The connection between MATLAB and OPNET can be achieved without the files, using COM (Component Object Model) software components, or a similar method of communication between processes, or for the connection of various applications (such as OLE, ActiveX, etc., all of which are derived from a COM model). Starting OPNET simulation large number of times, along with the execution of MATLAB scripts each time, is provided by the software for actions automation on computer system called "MurGee" [12]. OPNET simulator has also its own options for automating the execution of simulations, except that in this case we should ensure the calls of MATLAB scripts, in each iteration cycle.

VII. Conclusion

This paper shows an example of connecting OPNET simulator and MATLAB development environment in a common test platform, using files as a means of communication between these two software environments. The presented test platform can be used to verify the functionality of the new network protocol that would include optimization techniques, or verification of the existing protocols in addition to optimization of their parameters. OPNET runs simulations of network traffic, with the predetermined parameter values, while MATLAB executes the script with the mathematical algorithm, which optimizes the parameters in OPNET simulator. Operation of this test platform has been tested on the example of optimization of parameters for the TCP protocol, which is shown in the form of results, with the performance obtained by simulation, before and after the application of optimization.

References

[1] NS-2 URL: http://nsnam.isi.edu/nsnam/index.php/ User_Information

[2] NS-3 URL: http://www.nsnam.org/

[3] Riverbed OPNET Modeler URL: http://www.riverbed.com/products-solutions/products/networkplanning-simulation/Network-Simulation.html

[4] OMNeT++ Homepage. Available: http://www.omnetpp.org/

[5] http://www.mathworks.com/products/matlab/

[6] M.S. Hasan, H. Yu, A. Carrington, and T.C. Yang, "Co-simulation of wireless networked control systems over mobile ad hoc network using SIMULINK and OPNET," IET Communications, vol.3, no. 8, pp. 1297 – 1310, Aug. 2009.

[7] http://www.mathworks.com/products/ simulink/index-b.html

[8] Zhi Zhang1, Zhonghai Lu1, Qiang Chen1, Xiaolang Yan2, Li-Rong Zheng1, „COSMO: CO-Simulation with MATLAB and OMNeT++ for Indoor Wireless Networks“, iPack VINN Excellence Center, Royal Institute of Technology (KTH), Stockholm, Sweden, 2: Institute of VLSI Design, Zhejiang University, Hangzhou, China, IEEE Globecom 2010

[9] Gilbert E. Perez, Ivica Kostanic, "Comparing a Real-Life WSN Platform Small Network and its OPNET Modeler model using Hypothesis Testing", Government Communications Systems Division, Harris Corporation Melbourne, FL, USA, Electrical and Computer Engineering Department, Florida Institute of Technology, Melbourne, FL, USA, SYSTEMICS, CYBERNETICS AND INFORMATICS VOLUME 12 - NUMBER 7 - YEAR 2014

[10] Unlocking the Power of OPNET Modeler, Zheng Lu, Hongji Yang, Cambridge University Press, 2012

[11] MATLAB - The Language of Technical Computing (Application Program Interface Guide v5), The MathWorks Inc, 1998

[12] MurGee URL: http://www.murgee.com/