International Journal of Engineering Education Vol. 32, No. 5(A), pp. 2098–2106, 2016
Printed in Great Britain
0949-149X/91 $3.00+0.00

Evaluation of Simulators for Teaching Computer Networks*

NENAD JOVANOVIC[1], ZORAN JOVANOVIC[2] and ALEKSANDAR JEVREMOVIC[3]
[1] Faculty of Technical Sciences, University of Pristina, Serbia. E-mail: nenad.jovanovic@pr.ac.rs
[2] Business School of Professional Studies, Kralja Petra 70, Blace, Serbia. E-mail: zoranjov2004@gmail.com
[3] Department of Computing and Informatics, Singidunum University, Bulevar Zorana Djindjica 44, Belgrade, Serbia.
E-mail: ajevremovic@singidunum.ac.rs

This paper provides a short description of thirteen computer network simulators that are mainly used for educational purposes. Criteria for the evaluation and comparison of the simulators were presented. The criteria are applied in order to compare the simulators being described. This comparative analysis is useful when selecting a suitable simulator for teaching a specific computer networks course.

Keywords: network architecture and design, simulation and modeling, computers and education

1. Introduction

Computer networks present a very important area in computer science and informatics. Therefore, courses in this area must reach several goals. Their primary goal is to provide an overview of basic concepts of computer networks and to give insight into the functioning of a typical computer network. Besides this, they should emphasize all important issues in computer networks that engineers are encountering in practice.

Being successful in teaching Computer Networks is often a hard task, mainly because students do not tend to maintain acceptable levels of motivation. Computer networking concepts cover many details about protocols and configuration techniques and, thus, both proper education methodology and proper resources are needed to enhance students' learning [1].

Performance analysis and evaluation have important roles in Computer Networking and they represent difficult aspects of designing this course. Textbooks in this area indicate a trend toward greater integration of performance analysis into such courses. There are difficulties in including a significant performance analysis component in a Computer Networking course, particularly when it is not feasible to require a probability course as a prerequisite.

Using a network simulator can increase the emphasis on performance in an undergraduate networking course.

Practical applications with theoretical parts of any subject domain during the process of higher education of engineering may help to increase the quality level of the graduates. Universities should invest more in hardware and software of computer labs for training prospective engineers and scientists of computer science in a better way, to simulate real cases in a more economical way [2].

Using a network simulator has several advantages. A well constructed simulator can avoid the limiting assumptions needed by many analytic models which do not require expensive hardware. It may also be possible to offer students experience with a wider range of network systems through simulation than it would be possible with an actual hardware [3, 4]. The paper [5] provides information about the use of virtualization technology in computer network education, the utilization of real network devices, and defines a new generation network training platform.

Introductory laboratories are somewhat directed to reinforce the concepts presented in class. Therefore, activities in the laboratory are mainly directed to demonstrate specific phenomena or behavior, and provide experiences with measuring and studying about the desired characteristics. However, intermediate and advanced laboratories include problems that are more open-ended. As a result, students are requested to design and implement solutions, to design experiments to acquire data needed to complete the design or to measure various characteristics. All these activities in the laboratory are most effectively done using appropriate simulators of a computer network [6].

Simulators are particularly important for professionals seeking to obtain and upgrade professional certification.

Research in the field of computer networks with technology of IP (Internet Protocol) is based on analytical methods, simulation, experiments and measurements. While measurement and experimentation provide a means for testing real world net-

* Accepted 12 June 2016.

works, simulation and analysis are limited to testing designed, abstract models.

The limitations of measurement and experimental methods are that they can be applied only to the existing system or partly to new environments. On the other hand, although analytical methods are essential to the understanding of network behavior, there is a risk of application of simplified models that lose the essential features of a network. Simulation is complementary to the analysis because it allows the verification of the accuracy of analysis and testing of complex models that would be difficult or impossible to solve analytically. Therefore, it is advisable to use familiar and acknowledged tools, and one of the best ways of solving the problem of simulation resulting in verification is for simulators and associated scripts to be available for free, so that other researchers can easily test the effects of changes in initial assumptions about the network scenario.

The key property that generally makes modeling and simulation of IP networks difficult is the heterogeneity [7], which is reflected in several aspects: network topology, choice of simulated applications, generating traffic, dynamic traffic routing, the differences in versions of the protocol etc. This means that there is no predefined set of simulation scenarios which is sufficient for verifying the proposed solutions.

The problem is solved partially with the identification of invariant characteristics of IP traffic, the process of arriving calls, duration of sessions, as well as regular approximations of models and examining the behavior of the network in conditions of changing simulation set parameters in a wide range.

The primary motivation in the use and design of a network simulator has been to offer tractable hands-on networking exercises providing visual feedback. The design objectives of the networking environment are to provide an environment for the design and operation of practical exercises for a large number of students [8].

This paper attempts to give an overview of network simulators suitable for teaching courses in a computer network, to establish the evaluation criteria and to evaluate selected simulators according to the criteria.

This paper is organized in the following way: Section II gives an overview of simulators suitable for teaching in a computer network. Section III introduces the criteria for their evaluation. Section IV presents the comparison of the simulators' characteristics. Conclusion is given in Section V.

2. An overview of network simulators

This overview includes the following network simulators: OPNET, SSFNet, ns-2, ns-3, GNS3, GTNetS, GloMoSim, KivaNS, MIMIC, CNET, NetSim, Nessi and WnetSim (Table 1).

OPNET Modeler is a commercial simulator that supports a variety of LAN and WAN networks (ATM, IP, mobile networks, wireless LAN, etc.) and implements more than 400 functions for modeling protocols, network elements and the dynamic behavior of the network. OPNET modeler belongs to the class of events conducted in discrete time simulators, and the implementation of the simulator is based on an object-oriented finite state machine combined with the analytical model. It is available for Windows and Unix platforms. High quality graphic user interface and extensive documentation with examples of simulation studies are the main advantages of this simulator. Source programs aren't freely available—they are available only for generating simulation scenarios with varying set of input parameters [9].

SSFNet is a simulator with a free distribution of the source code and freely available programs. It is designed for modeling scalable global IP network and the Internet. SSFNet simulator is driven by discrete time events which are implemented using the Java SSF (Scalable Simulation Framework) module. It is available for Unix and Windows plat-

Table 1. Selected simulators

System	Author	Address
OPNET Modeler	OPNET Technologies suite	http://www.opnet.com/
SSFNet	SSF Research Network	http://www.ssfnet.org/homePage.html
NS-2	Berkeley University	http://www.isi.edu/nsnam/ns/
NS-3	Berkeley University	http://www.nsnam.org/
GNS3	EPITECH Innovative Project	http://www.gns3.net/
GTNets	Georgia Institute of Technology	http://pcl.cs.ucla.edu/projects/glomosim/
GloMoSim	UCLA University	http://pcl.cs.ucla.edu/projects/glomosim/
KivaNS	University of Alicante	http://www.disclab.ua.es/kiva/
MIMIC	Gambit Communications	http://www.gambitcomm.com/site/products/mimic_simulator.shtml
CNET	University of Western Australia	http://www.csse.uwa.edu.au/cnet/
NetSim	Tetcos	http://www.tetcos.com/netsim_gen.html
NESSI	HES-SO, Western Switzerland	http://www.nessi2.de/
WNetSim	VPS, Blace -Serbia	http://weblab.vpskp.edu.rs/

forms. For a description of abstract simulation models, it uses standardized syntax of the DML (Domain Modeling Language) language, by which it can create simulation models [10].

NS-2 is a discrete event network simulator which provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. NS-2 is licensed under the GNU GPLv2 license, and is available for research and development. This simulator is implemented in C++ and the simulator exposes a front-end API to the users. This front-end API is in oTCL language; therefore, the network model and the simulation control are implemented in this language. As oTCL is an interpreted language, this approach significantly reduces compilation time of the network models. However, it also reduces the performance and resource requirements of the simulator hampering the scalability of ns-2 [11].

NS-3 is also an open sourced discrete-event network simulator which targets primarily at research and educational use. NS-3 is licensed under the GNU GPLv2 license, and is available for research and development. NS-3 is designed to replace the current popular NS-2. However, NS-3 is not an updated version of NS-2 since that NS3 is a new simulator and it is not backward-compatible with ns-2. The basic idea of NS-3 comes from several different network simulators including ns-2, yans, and GTNetS. The major difference between ns-3 and ns-2 is that the core of ns-3 is written in C + + and with Python scripting interface (compared to OTcl in ns-2). Several advanced C + + design patterns are also used. One of the fundamental goals in the ns-3 design was to improve the realism of the models; i.e., to make the models closer in implementation to the actual software implementation that they represent [12].

GNS3 is a Graphical Network Simulator that allows emulation of complex networks. GNS3 is extensively supported with a range of associated video tutorials, ranging from suitable for beginners to advanced practising professionals. It allows emulation using Cisco Internet work Operating Systems [13]. GNS3 allows users to design a network topology based on specific models of different network devices along with an associated CLI window. One of the main limitations of GNS3 is obtaining a network device operating system to be used with the simulator. Without the proper devices, operating system GNS3 cannot perform any significant tasks. Normally, operating software is bundled with the network equipment and it is not purchased individually. Another limitation is its high consumption of processing resources. However, as it is an open source software, further functionality and modification may become available. The methods and experiences of using GNS3 in the process of teaching are introduced in [14].

GTNets (The Georgia Tech Network Simulator) is a full—featured network simulation environment that allows researchers in computer networks to study the behavior of moderate to large scale networks, under a variety of conditions [15]. The design philosophy of GTNets is to create a simulation environment that is structured much like actual networks are structured. GTNetS is implemented in C++. The GTNets simulator consists of a large number of C++ objects which implement the behavior of a variety of network elements. Building and running a simulation using GTNets, requires creating a C++ main program that instantiates the various network elements to describe a simulated topology, and the various applications and protocols used to move simulated data through the topology. GTNetS provides a better representation of computer networks by implementing a clear distinction between protocol layers. The nodes in GTNetS are simple containers where the networking interfaces, protocols and applications are "installed." The server applications are bound to ports and the client applications connect to the servers. As these representations mimic actual computer networks, the users can implement network models easily [11].

GloMoSim (Global Mobile Simulator), is a scalable simulation environment for wireless and wired network systems. GloMoSim is being designed using the parallel discrete-event simulation capability provided by PARSEC Compiler. GloMoSim supports various layers like Mobility, Radio Propagation, Radio Model, Packet reception models, Data Link, Network (Routing), Transport, Application. GloMoSim targets mainly wireless networks [16, 17].

KivaNS is a free open-source application programmed with Java, which is focused on the simulation of computer networks interconnected with the TCP-IP architecture. This application has been developed as a virtual laboratory, which allows the students who attend courses on computer networks to carry out experiments about IP routing, without the necessity of complex and expensive real equipment. KivaNS has been evaluated by students of University of Alicante by means of practical cases for resolving network problems with ARP, IP, ICMP protocols [18]. The main aims of KivaNS are to simulate how IP works and especially to study the different techniques for routing the data packets through different networks. KivaNS also includes the simulation of auxiliary protocols such as ARP, ICMP in addition to IP, and emulates the basic working of link layer technologies such as PPP, Ethernet, or switched Ethernet. The architecture of KivaNS is designed to have the simulation and the user interface services well differentiated. KivaNS is composed of two main blocks, both implemented in Java. The first block is an API that offers a simulation engine for data networks. The second block is a complete graphical user interface, which enables users to design data networks schemes, as well as to simulate and analyze them by using the API block in a transparent way [19].

MIMIC simulator provides an interactive hands-on lab for quality assurance, development, sales presentation, evaluation, deployment and training of enterprise management applications. Users create a customizable virtual environment populated with simulated routers, hubs, switches, WiFi/WiMAX/LTE devices, probes, cable modems, servers and workstations. MIMIC IOS Simulator fully supports Cisco IOS software and SNMPv1, v2, v2c, v3. It is pre-packaged with data to setup out-of-the-box virtual labs quickly. It is shipped with large libraries of simulated devices, networks, and MIBs from the leading networking companies [20]. If a user is planning to teach about Network Management and use a network management application to show knowledge units, then MIMIC can be useful for most of the topics.

CNET is an open source network simulator developed at the University of Western Australia designed specifically for teaching. It is a network simulator which enables experimentation with various data-link layer, network layer, routing and transport layer networking protocols in networks consisting of any combination of wide-area-networking (WAN), local-area-networking (LAN), or wireless-local-area-networking (WLAN) links. With reference to the OSI/ISO Networking Reference Model, CNET provides the Application and Physical layers [21].

NetSim is a network simulation tool used for network lab experimentation and research. The Boson NetSim Network Simulator is an application that simulates Cisco Systems networking hardware and software and is designed to aid the user in learning the Cisco IOS command structure. NetSim is a Windows-based product that simulates a wide variety of Cisco routers. NetSim supports multiple routing protocols, including RIP, IGRP, EIGRP, BGP, and OSPF. It supports different LAN/WAN protocols, including PPP/CHAP, ISDN, and Frame Relay. Netsim provides a simple method for including interactive, discrete-event simulation in WWW sites. It uses graphical interfaces for model entry, editing and manipulation, allows user control of model execution and provides animated and numerical output. Simulator uses Java to provide: object-oriented flexibility, platform independence through the WWW and compatibility with other Java-based graphical and analytical tools [22].

Nessi (Network Security Simulator) is a network simulation tool which incorporates a variety of features relevant to network security, which distinguishes it from general-purpose network simulators. Its capabilities such as profile-based automated attack generation, traffic analysis and support for the detection algorithm plugins allow it to be used for security research and evaluation purposes. Nessi is mainly oriented toward educational use, where it enables students to implement or modify simulation models of protocols with minimal overhead. A second application of nessi is for verification and performance evaluation of new protocols, where it allows the developer to easily explore different options [23–25].

WnetSim is the computer network TCP/IP simulator, which was developed at the Department for Informatics at the Business School of Applied Studies in Blace. The simulator has been used as a teaching tool for the course Computer Networks at the Department for Computer Science. While working with the simulator, standard Windows window appears with the default look and features that are common for them. The system enables editing of

Table 2. The basic characteristics of selected simulators

Simulator	OS	Language	Availability	Open source
OPNET Modeler	Windows, Unix	C/C++	not	No
SSFNet	Windows, Unix	Java	free	Yes
NS-2	Linux, BSD, Solaris, Mac OS, Windows	C++	free	Yes
NS-3	Linux, BSD, Solaris, Mac OS, Windows	C++, Python	free	Yes
GNS3	Windows, Linux, Mac OS	Python	free	Yes
GTNets	Linux, OSX, Solaris, Windows	C++	free	Yes
GloMoSim	Windows, Linux, FreeBSD, Solaris . . .	C, Parsec	free	Yes
KivaNS	Windows, Linux	Java	free	Yes
MIMIC	Win., Solaris, Linux, Amazon Cloud	Tcl, C++, Java, Perl	not	No
CNET	Linux, Mac OS	C	free	Yes
NetSim	Windows	C/C++	not	No
NESSI	Win., MAC OS X, Linux, Unix, BSD	Python	free	Yes
WNetSim	Windows, Linux, Mac OS	Java	free	No

arbitrary topology of computer networks and monitoring processes in the network in real time. Configuring the components in the system can be done in two ways. The first way is to use the standard Windows dialog box and Windows controls, such as when configuring a computer. Another way is to configure using command line interface, for example, when configuring the router [26].

The basic characteristics of selected simulators are shown in Table 2 and include authors, operating systems in which the simulators run, programming languages used for their development and availability.

3. Evaluation criteria

The authors established a set of relevant criteria for simulator evaluation which can be classified in a group called coverage criteria. Coverage criteria estimate which topics of computer network courses are supported by certain simulators.

Other criteria, such as use, effect on learning results, effect on understanding certain teaching materials, transparency of distributed simulations, information hiding, but also including these criteria demands real use of the simulators, which is impractical and, therefore, the criteria are limited only to those which can be analyzed based on available information from books.

The coverage criteria are established using the Computer Engineering Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering [27]. This document represents the final report of the Joint Task Force on Computing Curricula—an undertaking of SIGITE (Special Interest Group on Information Technology Education) of the ACM (Association for Computing Machinery), the ACM, and the IEEE Computer Society—for programs in Computer Engineering and defining broad knowledge areas that are applicable to all Computer Engineering programs. Computer networks are one of these knowledge areas which are comprised of a set of 7 core knowledge units:

- History and overview
- Communications network architecture
- Communications network protocols
- Local and wide area networks
- Client-server computing
- Data security and integrity
- Wireless and mobile computing

These knowledge units are chosen as the coverage criteria and given with detailed lists of their topics in Table 3. First knowledge unit is not taken into consideration.

4. Comparison of simulators characteristics

This section evaluates the network simulators with respect to the established coverage criteria.

Comparative analyses of the performance of these simulators are not published in large numbers, and the existing publications usually include a limited set of experiments with a simple simulation scenario.

For example, the scalability of the simulator ns-2 and SSFNet, in terms of speed performance required for simulation and computer memory resources is compared and the analysis results showed that ns-2 is the fastest, and the most demanding in terms of memory resources.

The great advantage of freely available simulator for research in the field of IP network is able to add new or modify existing features (proposed new algorithm, modification of the protocol, etc.).

In teaching with cnet it would be possible for cnet to be used to program and develop protocols that are part of the TCP/IP suite, but cnet does not simply provide them all as part of its APIs. Students may get to code these, or port them (from, say, BSD or Linux) into cnet, and then measure their effectiveness.

The main feature of the KivaNS is that the protocols are programmed in a library independent of the graphical user interface, following an Object Oriented structure equivalent to the layers and PDUs of the network architecture, and this library can be expanded with new protocols and features.

The SSFNet is a network simulation tool using open source software with various network simulation applications. It has been designed for the expansion of network, including topology, protocols, traffic, and etc, and it can support simulation for a large-scale network like the Internet. However, it is not easy for general users to perform network simulation using SSFNet because the SSFNet does not provide users with any supplementary tools for designing network elements and topology, and analyzing simulation results. The network modeling and analysis process must be done manually by users themselves.

GloMoSim has been designed to be extensible and composable. The communication protocol stack for wireless networks is divided into a set of layers, each with its own API. Model of protocols at one layer interact with those at a lower (or higher) layer only via these APIs [16].

In the ns-2, network models are programmed in a front-end language which is an API for the scripting language oTcl. The developers of the simulator explicitly state which C++ class methods and/or attributes can be accessed from oTCL scripts. This

Table 3. List of topics in each knowledge unit with detailed lists of their topics and assessment of simulators

Knowledge units	**Topics**	**OPNET**	**SSFNet**	**NS-2**	**NS-3**	**GNS3**	**GTNets**	**GloMoSim**	**KivaNS**	**MIMIC**	**CNET**	**NetSim**	**NESSI**	**WNetSim**
Communications network architecture	Network line configuration (point-to-point, multipoint)	+	+	+	+	+	+	+	+	–	+	+	–	+
	Networking and internetworking devices: Repeaters, bridges, switches, routers, gateways	+	+	+	+	+	+	+	+	–	+	+	–	+
	Network Topologies (mesh, star, tree, bus, ring)	+	+	+	+	+	+	–	+	–	+	+	+	+
	Connection-oriented and connectionless services	+	+	+	+	+	+	–	–	–	+	+	–	+
Communications network protocols	Network protocol (syntax, semantics, timing)	+	+	+	+	+	+	+	+	+	+	+	+	+
	Protocol suites (TCP/IP)	+	+	+	+	+	+	+	+	+	–	+	+	+
	Physical layer networking concepts; data link layer concepts; internetworking and routing	+	+	+	+	–	+	–	+	+	+	+	+	+
	Network Standards and standardization bodies	–	–	–	–	+	–	–	+	+	–	+	–	+
Local and wide area networks	LAN topologies (bus, ring, star)	+	+	+	+	+	+	+	+	–	+	+	+	+
	LAN technologies (Ethernet, token Ring, Gigabit Ethernet)	+	+	+	+	+	+	+	+	–	+	+	+	+
	Error detection and correction	–	+	–	–	–	+	–	+	–	+	+	+	+
	Carrier sense multiple access networks (CSMA)	+	+	+	+	+	+	+	–	–	+	+	+	–
	Large networks and wide areas	+	+	+	+	+	+	–	–	–	–	+	–	+
	Circuit switching and packet switching	+	+	+	+	–	+	–	–	–	–	+	–	+
	Protocols (addressing, congestion control, virtual circuits, quality of service)	+	+	+	+	+	+	+	+	–	+	+	+	+
Client-server computing	Web technologies: Server-side programs; client-side scripts; The applet concept	–	–	–	–	–	–	–	–	–	–	–	–	–
	Characteristics of web servers	+	+	+	+	–	+	–	–	–	–	–	–	–
	Support tools for web site creation and web management	–	–	–	–	–	–	–	–	–	–	–	–	–
Data security and integrity	Fundamentals of secure networks; cryptography	–	–	–	–	–	–	–	–	–	+	+	–	–
	Encryption and privacy: Public key, private key, symmetric key	–	–	–	–	–	–	–	–	–	+	+	–	–
	Authentication protocols	+	+	–	+	+	–		–	–	+	+	–	–
	Packet filtering	+	–	–	–	+	–	—	–	–	+	+	–	–
	Firewalls	–	–	–	–	–	–		–	–	+	–	–	–
	Virtual private networks	+	–	+	+	+	+	–	–	–	+	+	–	–
	Transport layer security	+	+	+	+	+	–	+	–	–	+	+	–	–
Wireless and mobile computing	Overview of the history, evolution, and compatibility of wireless standards	–	–	–	–	–	–	–	–	–	–	–	–	–
	The special problems of wireless and mobile computing	–	–	–	–	–	–	+	–	–	+	–	–	–
	Wireless local area networks and satellite-based networks	+	+	+	+	+	–	+	–	–	+	–	–	–
	Mobile Internet protocol	+	+	+	+	–	–	+	–	–		–	–	–
	Mobile aware adaptation	–	–	–	–	–	–	–	–	–		–	–	–
	Extending the client-server model to accommodate mobility	–	–	–	–	–	–	+	–	–	+	–	–	–
	Mobile data access: server data dissemination and client cache management	–	–	–	–	–	–	+	–	–		–	–	–
	The software packages to support mobile and wireless computing	–	–	–	–	–	–	+	–	–	–	–	–	–
	The role of middleware and support tools	–	–	–	–	–	–	–	–	–	–	–	–	–
	Performance issues	–	–	–	–	–	–	–	–	–	–	–	–	–
	Emerging technologies	–	–	–	–	–	–	–	–	–	–	–	–	–

allows the developers to expose a simplified API to the ns-2 user, reducing the learning curve involved in developing simulations. The disadvantage of using a full-fledged scripting language is that the interpreter of the language is computationally expensive, which reduces the performance of the simulator.

The ns-3 provides two separate APIs: the "front-end" API and the "backend" API. The "front-end" API, which includes the helper classes that facilitate the initialization and the usage of the classes representing the network appliances.

The "backend" API includes all the details of the simulator. In ns-3, the helper classes can only be called from user programs; that is, using them at the back-end is not permitted and it yields to compilation errors. However, users are not limited to using helper classes; they can use the back-end API for initialization, which can lead to long simulation model codes. The benefit of providing two APIs is that network models can be implemented in the language the simulator is programmed with. Front-end API of ns-3 has bindings for Python programming language. However, the aim of these bindings is not to hide implementation details but to provide a platform for users not familiar with the C++ programming language [11].

Table 4. The share of certain areas in the teaching material

Knowledge units	**Hours per week**	**Percentage share**
Communications network architecture	3	15
Communications network protocols	4	20
Local and wide area networks	4	20
Client-server computing	3	15
Data security and integrity	4	20
Wireless and mobile computing	2	10

Assessment of simulators according to the criteria which are based on [27] is presented in Table 2. Symbol „+" means that a simulator has a specific functionality necessary for covering teaching material in a certain field of computer network and symbol „–" means that a simulator does not have the specific functionality necessary for covering teaching material in a certain field.

Table 4 shows the share of certain areas in comparison to the total number of classes [27].

Table 5 sums up the results of comparing simulators according to area coverage, in comparison to the share of certain areas in the total number of classes.

The results of the evaluation are discussed briefly here.

Topics in Communications network architecture unit are covered to a great extent in almost all simulators. The exceptions are simulators such as MIMIC and Nessi which are developed for specialized needs.

If we are planning to teach Network Management and use a network management application to show all these topics, then MIMIC can be useful for most of these. If SNMP protocol is included, MIMIC Simulator can be very useful.

Nessi is a network simulation tool which incorporates a variety of features relevant to network security, distinguishing it from general-purpose network simulators. While other simulators focus on minimizing the simulation time, Nessi tries to minimize the development time and the difficulties to implement a new simulation model. The advantage of Nessi is that simulation models can be developed in a fraction time necessary for other simulations. Therefore, Nessi allows students to create or modify models of network protocols with minimal overhead and is thus perfectly suited for networking laboratories or semester projects [23].

Topics in the Local and wide area networks unit are covered by almost all simulators. The exception

Table 5. Evaluation results for the topic coverage criteria

Knowledge units	**OPNET**	**SSFNet**	**NS-2**	**NS-3**	**GNS3**	**GTNets**	**GloMoSim**	**KivaNS**	**MIMIC**	**CNET**	**NetSim**	**NESSI**	**WNetSim**
Communications network architecture	15.00	15.00	15.00	15.00	15.00	15.00	7.50	11.25	0.00	15.00	15.00	3.75	15.00
Communications network protocols	15.00	15.00	15.00	15.00	15.00	15.00	10.00	20.00	20.00	10.00	20.00	15.00	20.00
Local and wide area networks	17.14	20.00	17.14	17.14	14.29	20.00	11.43	11.43	0.00	14.29	20.00	14.29	17.14
Client-server computing	5.00	5.00	5.00	5.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Data security and integrity	11.43	5.71	5.71	8.57	11.43	2.86	2.86	0.00	0.00	20.00	17.14	0.00	0.00
Wireless and mobile computing	1.82	1.82	1.82	1.82	0.91	0.00	5.45	0.00	0.00	2.73	0.00	0.00	0.00
Overall coverage	65.39	62.53	59.68	62.53	56.62	57.86	37.24	42.68	20.00	62.01	72.14	33.04	52.14

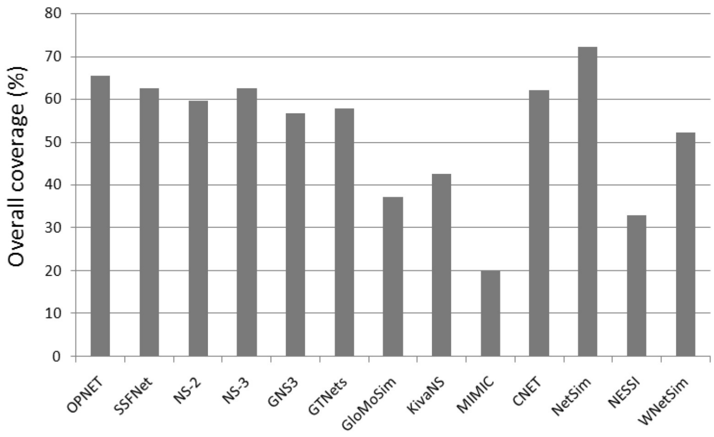

Fig. 1. Evaluation of simulators for teaching computer networks.

is MIMIC, which is not developed to deal with the topics in Local and wide area networks unit.

Topics in the Communications network protocols unit are best covered by KivaNS, MIMIC, NetSim and WNetSim simulators.

Mainly, KivaNS implements the following protocols in detail: Ethernet for IP, PPP, IP and ARP. It also implements IP routing and fragmentation. IP routing tables can be configured in a similar way to being configured in real OS. In addition, it also considers the bridging and switching for Ethernet frames besides bus topology and the Spanning Tree Algorithm (simplified 802.1d). The physical media can simulate errors in frames, and link and network protocols work consequently.

As a result, the Data security and integrity unit has a very good coverage with simulators cnet and NetSim.

Cnet provides many of the basic features for lower level networking, such as data transmission and reception, the probabilistic introduction of errors, accurate timing and statistical functions, a variety of NIC types, and programmable node mobility, but it does not provide all features in its standard distribution.

Topics in the Wireless and mobile computing unit and the Client-server computing unit are poorly supported by almost all simulators. The exception is simulator GloMoSim which is developed for specialized needs. A probable explanation is that most of the simulators analyzed concentrate mainly on topics related to the basic components of a computer network.

There are simulators with lower overall topics coverage, but with high coverage for a specific unit. MIMIC has high coverage for Communications network protocols unit, and cnet and NetSim for Data security and integrity unit. This kind of information is more important than the overall coverage for courses limited to a specific unit.

5. Conclusion

Practical skills are highly regarded but dedicated network teaching laboratories are expensive and an alternative approach is to provide students with network simulators.

Most courses in computer networks used simulators for practical work. In this paper, several computer network simulators are described. They have similar features and usability, so that they can be mutually compared. A survey was made of simulators available in the literature. The coverage criteria have been established for the evaluation of simulators, and these simulators were evaluated.

The results of the evaluation show that there is no single simulator which covers all topics. The outcome is the consequence of the fact that the area of computer network is versatile and that it involves a great number of topics so the development of simulators which covers all topics gives a system which would be big and impractical for use.

The best overall topics coverage is achieved with simulators NetSim (72.14%). Most of the simulators achieve more then 50.00% coverage. The only exception is a MIMIC simulator, where the coverage percentage is 20% and it is specialized to work with an SNMP protocol. Other simulators achieve about 30.00% coverage.

The coverage percentage does not offer information about the advantages of using certain simulators but only about the number of covered topics, that is, topics which can easily be covered by the given simulator.

Thereafter, use of a simulator is beneficial to student learning about computer networks.

Acknowledgements—The authors express their gratitude for the help, advice and information received from their colleagues: Chris McDonald for cnet, Pankaj Shah for MIMIC, Francisco Andrés Candelas Herías for KivaNS, Juergen Ehrensberger for nessi and Brian Scheibe for NetSim.

References

1. P. Gil Vázquez, F. A. Candelas Herías, C. A. Jara Bravo, G. J. García Gómez and F. Torres Medina, Web-based OERs in Computer Networks, *The International Journal of Engineering Education*, **29**(6), 2013, pp. 1537–1550.
2. Bulent Gursel Emiroglu and Seda Sahin, Analysis of Students' Performances during Lab Sessions of Computer Networks Course, *Educational Technology & Society*, **16**(3), 2013, pp. 329–346.
3. L. Barnett, NetSim: A Network Performance Simulator, http://scholarship.richmond.edu/, Accessed 15 July 2015.
4. M. Minovic, M. Milovanovic, I. Kovacevic, J. Minovic and D. Starcevic, Game design as a learning tool for the course of computer Networks, *The International Journal of Engineering Education*, **27**(3), 2011, pp. 498–508.
5. N. Yalcin, Y. Altun and U. Kose, Educational material development model for teaching computer network and system management, *Computer Applications in Engineering Education*, **23**(4), 2015, pp. 621–629.
6. B. Nikolic, Z. Radivojevic, J. Djordjevic and V. Milutinovic, A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization, *Education, IEEE Transactions on*, **52**(4), 2009, pp. 449–458.
7. S. Floyd, V. Paxson, Difficulties in Simulating the Internet, *IEEE/ACM Trans. On Networking*, **9**(4), 2001, pp. 392–403.
8. C. McDonald, Network simulation using user-level context switching, *Proceedings of the Australian UNIX Users' Group Conference*,1993, pp. 1–10.
9. Introduction to Using OPNET Modeler, OPNETWORK 2002, Simulation and Modeling, http://www.sce.carleton.ca/faculty/lambadaris/courses/5001/opnet_tutorial.pdf, Accessed 7 January 2016.
10. Scalable Simulation Framework and SSFNet, http://www.ssfnet.org, Accessed 15 March 2015.
11. S. Ciraci and B. Akyol, An evaluation of the network simulators in large-scale distributed simulations, *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid*, ACM, 2011, pp. 59–66.
12. F. G. Riley and R. T. Henderson, *The ns-3 network simulator, Modeling and Tools for Network Simulation*, Springer Berlin Heidelberg, 2010, pp. 15–34.
13. GNS3: graphical network simulator, http://www.gns3.net, Accessed 31 July 2009.
14. Chun-yan Peng, and Bing Liu, Application of GNS3 at Computer Network Teaching, *Theory Research*, **20**, 2010, pp. 136.
15. F. G. Riley, The Georgia tech network simulator, *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, ACM, 2003, pp. 5–12.
16. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia and M. Gerla, Glomosim: A scalable network simulation environment, http://citeseerx.ist.psu.edu/, Accessed 15 July 2015.
17. X. Zeng, R. Bagrodia and M. Gerla, GloMoSim: a library for parallel simulation of large-scale wireless networks, *Parallel and Distributed Simulation*, 1998. PADS 98. Proceedings. Twelfth Workshop on. IEEE, 1998. pp. 154–161.
18. F. A. Candelas Herías and P. Gil Vázquez, Practical experiments with KivaNS: A virtual laboratory for simulating IP routing in computer networks subjects, *Research, Reflections and Innovations in Integrating ICT in Education*, **1**(3), 2009, pp. 1414–1418.
19. P. Gil Vázquez, F. A. Candelas Herías and C. A. Jara Bravo, Computer networks e-learning based on interactive simulations and SCORM, *International Journal of Online Engineering*, **7**(2), 2011, pp. 15–23.
20. MIMIC simulator, http://www.gambitcomm.com., Accessed 15 July 2015.
21. CNET simulator, http://www.csse.uwa.edu.au/cnet/introduction.html, Accessed 25 July 2015.
22. L. Tamie Veith, E. John Kobza and C. Patrick Koelling, Netsim: Java^{TM}-based simulation for the World Wide Web, *Computers & Operations Research*, **26**(6), 1999, pp. 607–621.
23. Jérôme Vernez, Jürgen Ehrensberger, and Stephan Robert, Nessi: a python network simulator for fast protocol development, *Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, 11th International Workshop on. IEEE*, 2006, pp.67–71.
24. R. Bye, S. Schmidt, K. Luther and S. Albayrak, Application-level simulation for network security, *In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST*, 2008, p. 33
25. S. Schmidt, R. Bye, J. Chinnow, K. Bsufka, A. Camtepe and S. Albayrak, Application-level simulation for network security, *Simulation*, **86**(5–6), 2010, pp. 311–330.
26. N. Jovanovic, R. Popovic and Z. Jovanovic, WNetSim: a web-based computer network simulator, *International Journal of Electrical Engineering Education*, **46**(4), 2009, pp. 383–396.
27. Computer Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, http://www.acm.org/education/curric_vols/, Accessed 05 July 2015.

Nenad Jovanović is a professor at the Department of Computing and Informatics at the Faculty of Technical Sciences, University of Pristina in Serbia. His research interests include computer networks, simulation and educational systems. He received a BSc in computer engineering at the University of Pristina, a MSc degree at the University of Belgrade and a PhD in computer science at the University of Pristina.

Zoran Jovanović does research at the Computer and Informatics Department at the Business School of Applied Studies—Blace, Serbia. He received MSc in computer science at Singidunum University, Serbia and a PhD in computer science from the Singidunum University Belgrade, Serbia. His research interests include Virtualization, Network Protocols.

Aleksandar Jevremovic is a professor at the Department of Computing and Informatics, Singidunum University, Belgrade, Serbia. His principal research interests include information and communication technologies, artificial intelligence, and education. He is the author of many publications and systems in these areas.