

Applying simulators in computer networks education to encourage personalised learning

Qian Liu

Rhode Island College
Providence, United States of America

ABSTRACT: Simulators can play an important role in computer networks education. They can help students to examine network behaviour and compare performance in various scenarios without requiring dedicated hardware. However, the integration of simulation into networking courses usually depends on course topics or instructor preference, which may not be learner-centred. Investigations in this research were carried out widely and included the use of simulators, as well as reconsideration of the integration of simulation into networking education to develop a personalised learning environment in which learners can use the most appropriate technologies to drive learning based on individual progress and needs.

Keywords: Network simulators, computer networks, personalised learning

INTRODUCTION

Simulation typically is applied in computer networks courses to illustrate network internals and protocols. It is also used to emulate various network scenarios without requiring dedicated hardware. Simulation is introduced and applied in different ways according to different course requirements or instructor preference at colleges and universities. Some use simulation as an animation in order to explain network concepts visually to students [1][2]. Some use simulation for specific courses, with emphasis on wireless communication [3][4]. In these activities, the integration of simulation into education is initiated by instructors to improve teaching effectiveness through visualisation. In this teacher-centred procedure, instructors determine the concepts needing illustration to accommodate the learning requirements.

The need for a personalised learning environment is increasing, because *one-size-fits-all* instruction does not meet the needs of today's learners and it fails to engage students with specific needs [5]. The traditional approach does not motivate students' interest or help them to apply knowledge effectively [6]. For example, simulation tools, such as the applets discussed by Kurose and Ross [7], and Holiday [8], illustrate network fundamentals in animation, but they do not enable students to analyse abstract concepts [9]. Besides, they illustrate fundamentals in an ideal network scenario only. In an actual environment network events are not well-organised and do not occur in the same order as in the animations. Therefore, a student's understanding of fundamentals may stay at a theoretical level.

Table 1: Topics in a typical networking course.

Internet stack	Details
Layer 5	Application protocols, such as HTTP, FTP, DNS, etc
Layer 4	TCP error control, flow control, congestion control TCP/UDP (user datagram protocol) headers and their usage in data transfer
Layer 3 and Layer 2	IP addressing, subnet management Routing and switching, Ethernet, ARP, etc

To encourage personalised learning in networking education, simulators should play an important role because they provide necessary interactions between learners and network abstractions. Although there are often differences in the teaching order or depth [10], a typical networking course in undergraduate education usually focuses on the topics listed

in Table 1. Licence fee, ease-of-use and a good coverage of network core elements are important factors to consider when integrating simulators into classes to encourage personalised learning. On the other hand, graduate level/research courses usually focus on open-source simulators with a good support for all network layers, so that new functionalities and models can be added easily.

Reported in this article is a detailed survey of the most common network simulators, an analysis of their pros and cons, and the use of simulators to encourage personalised learning, so that learners can learn at their own pace and explore networking topics that reflect their learning needs.

DESIGN AND DEPLOYMENT

Although one concern about network simulators is that it may take time to learn how to use them, this is minimised in simulators with a GUI (graphic user interface) that allows users to drag and drop network elements to build a virtualised topology. To create a personalised learning environment, teachers guide the process first and then students take ownership of their learning and explore topics based on their needs. However, a single simulator does not meet the needs of all learners, because each simulator has limitations on specific areas and cannot cover all aspects of networks. Multiple simulators must be combined to provide for learner differences, learning needs and learning preferences. Because of this, the focus here was on the evaluation of simulators based on the following questions:

- whether simulators could provide learners with hands-on practice and allow them to design personalised experiments during their learning to fulfil the course requirements;
- whether simulators could provide an environment in which students could explore networking, and how to encourage personalised learning by using simulators to help learners achieve their learning goals.

Simulators - OMNeT++ and NS-3

The OMNeT++ (Objective Modular Network Testbed in C++) [11] and NS-3 (Network Simulator-3) are open source network simulators that organise basic network elements (end nodes, switches, routers, etc) into simulator-private data structures and classes [12]. They provide several build-in simulation examples of TCP/UDP (User Datagram Protocol) data transfer. Each example specifies various network topology attributes (e.g. link delay) and data traffic parameters (e.g. packet size, sending interval) in plain text, so that students can easily configure a network, run the simulation, and then analyse simulation output to observe differences between various configurations and to explain how the changes in parameters impact network behaviour. For example, learners can modify the *bit error rate* parameter in OMNeT++ and then compare the changes in *cwnd* (congestion window). Illustrated in Figure 1 are the changes in *cwnd* when the *bit error rate* is changed. Students thus are able to delve into the behaviour of TCP congestion control algorithms.

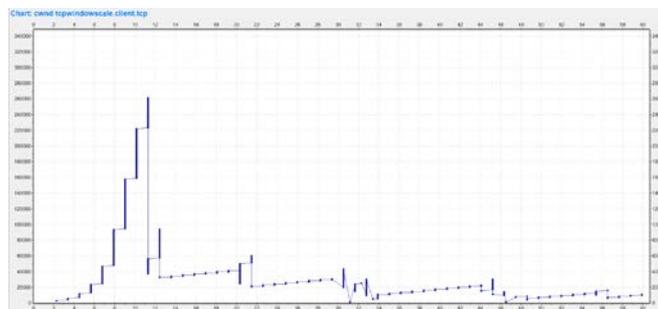


Figure 1: *Cwnd* variations in OMNeT++.

Although it is easy for learners to investigate network basics without dedicated hardware in OMNeT++ and NS-3, if learners need to design experiments with specific traffic patterns or data communication logic or to collect statistics that at present are not recorded by existing experiments, they must use internal libraries in OMNeT++ or NS-3. This requires a thorough understanding of the simulator internals; for instance, those simulator-private structures and interfaces that are not directly associated with network fundamentals. This would create a steep learning curve for students that goes beyond the course requirements and puts an extra burden on students in an introductory networking course.

In general, these simulators are widely used in new model research and development. For undergraduate education, it is a good resource for students to observe network behaviour easily through applying the simulator built-in examples, but learners may experience a difficulty if they use them to explore topics that are not covered by the built-in examples.

Simulator - GNS3

The GNS3 (Graphical Network Simulator-3) virtualises commercial network devices and allows users to set up device configurations and to manage networks as needed without dedicated hardware [13]. There are some advantages for students in using GNS3. For example, a traditional method in teaching IP addressing focuses on the conceptual

understanding only; it usually lets students calculate addresses, such as subnet ID and mask manually or implement them in programs. If students could apply this in practice in GNS3 to simulate communication among different hosts and subnets, they could analyse and better understand abstract concepts. For example, the topology in Figure 2 is used to introduce GNS3 to students. Each virtual PC's IP address must be set up manually via the *ip* command, and the switch will learn each PC's MAC address automatically (gratuitous ARP, shown in Figure 2). Then, the router in Figure 2 could be configured by using actual hardware *config* commands in order to activate subnet communication. The built-in command *traceroute* could be used on each virtual PC to examine network connectivity.

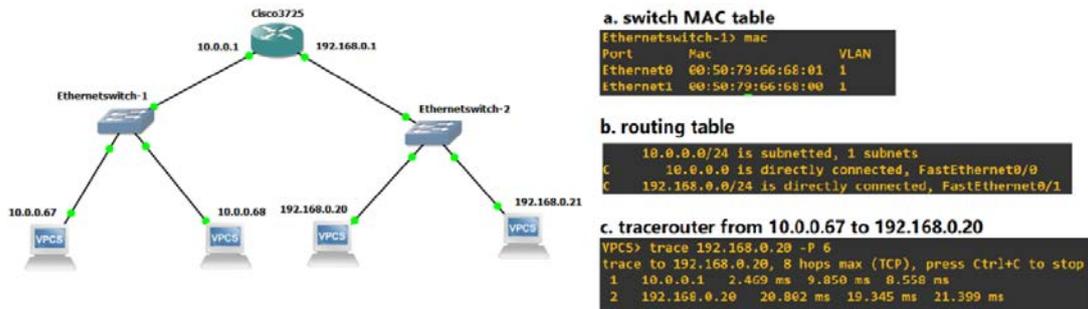


Figure 2: GNS3 topology and device configuration example.

Starting from this example, students could make changes or design new experiments based on their needs to learn other routing and switching topics. For example, students could set up various network topologies to delve into hardware configuration, especially commercial routers' configurations in GNS3 without requiring actual devices. In the real world, this cannot be done without affecting external networks. In addition, students could capture traffic flows on every link in GNS3. In the real world, it is not possible to see how data are transferred in external networks. This feature of GNS3 gives the opportunity to study the details of data transfer. As an example, the complete TCP traffic (from connection setup to connection close) on every network link (e.g. all links in Figure 2) can be captured, and students can examine how packets are transferred among different subnets and how packet headers are altered from the source to the destination, thus acquiring a deep understanding of addressing in the link layer and network layer. Typically, these topics and hands-on practices are not required in undergraduate-level courses, but this practical experience will help students to better comprehend network communications and to prepare them for advanced study.

A single simulator could not meet the demands of all learners. The GNS3 focuses on IP addressing, subnet management and protocols, such as ICMP (Internet Control Message), ARP, and DHCP (Dynamic Host Configuration). It does not provide ways to adjust protocol behaviour or to specify traffic pattern, so that learners could not use it to explore application layer protocols such as HTTP, FTP, and transport layer basics listed in Table 1, such as TCP error control techniques.

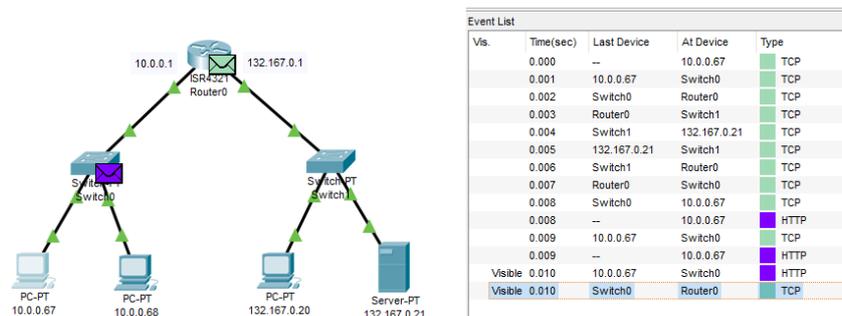


Figure 3: Data transfer animation with packet details from Packet Tracer.

Simulator - Packet Tracer

Packet Tracer is designed by Cisco and provides similar functionality to GNS3 [14][15]. One big difference between GNS3 and Packet Tracer is that Packet Tracer supports more application layer protocols, such as FTP, SMTP, SSH (Secure Shell). Besides, it visualises all traffic flows on every link in animation, so that users could follow data communication procedure step-by-step to learn protocol details. Therefore, Packet Tracer is an ideal tool to observe and learn network internals. Its integration into networking courses has been evaluated many times, but there has been little discussion on how to encourage personalised learning by using Packet Tracer.

Take the topology shown in Figure 3 as an example, to build a personalised learning environment. Instructors could first guide students through a simple HTTP transfer experiment in Packet Tracer: one host (the virtual PC with IP 10.0.0.67 in Figure 3) uses its built-in Web browser to load an image from another host (the virtual server with IP 132.167.0.21). Packet Tracer will then animate the complete traffic flows, from the initial TCP setup to the data transfer, and will record packet details, as shown in Figure 3. To encourage personalised learning, students could ask themselves the following two questions in each step in following traffic traces:

- 1) Why it is necessary to send the current packet in the current step?
- 2) What is the relationship between the previous *request* packet and the current *reply* packet?

To answer these questions, protocol details and the whole communication procedure should be considered, especially the relationship between the application layer protocol and the transport layer protocol. For instance, an HTTP request would trigger a TCP connection request, which means HTTP relies on TCP to transport data. Learners will delve into protocol essentials by answering these questions. For example, from the recorded packet details, students can investigate the connections between data packets and ACKs (acknowledgements) to learn how reliability is guaranteed. Furthermore, students can set up different experiments to explore how a protocol reacts to various situations and learn at their own pace. They can set up an error-free network first, and then introduce error models into it; for example, using a virtual PC's browser to load an image that does not exist on a HTTP server to examine how HTTP reports errors, thus exploring and learning protocol details step-by-step.

Simulator - OPNET Modeller

The OPNET Modeller is a similar simulator to Packet Tracer that allows students to drag and drop network elements in its GUI and adjust traffic attributes, such as traffic type (FTP, HTTP), repeatability, start/stop time, and data size [16]. Although it does not illustrate traffic flows, OPNET makes it easier for students to collect a great deal of network statistics, ranging from global statistics, such as throughput and CPU usage, to protocol-level statistics, such as *cwnd* and the number of retransmissions.

The OPNET is an appropriate tool by which to explore performance differences caused by various network topologies or traffic patterns, and to engage students in answering *what-if* questions based on their needs. Students are instructed to adjust one parameter at a time to investigate the reason why network behaviour changes. That is, students could set up a network to find out the impact of adding one more FTP connection. Figure 4a shows the FTP server CPU usage and server traffic when there is only one active FTP client that downloads a big file periodically. Figure 4b shows the impact on the server CPU and traffic when one more FTP client starts downloading from the same server. The traffic is randomly generated using a uniform distribution and, in this case, the server CPU is busy all the time, because it needs to deal with different clients alternatively. In Figure 4c, a comparison is shown of the server traffic in the previous two scenarios, while both FTP clients in the second scenario start downloading from the server at the same time and, in this case, the server behaviour is totally different from that in Figure 4b.

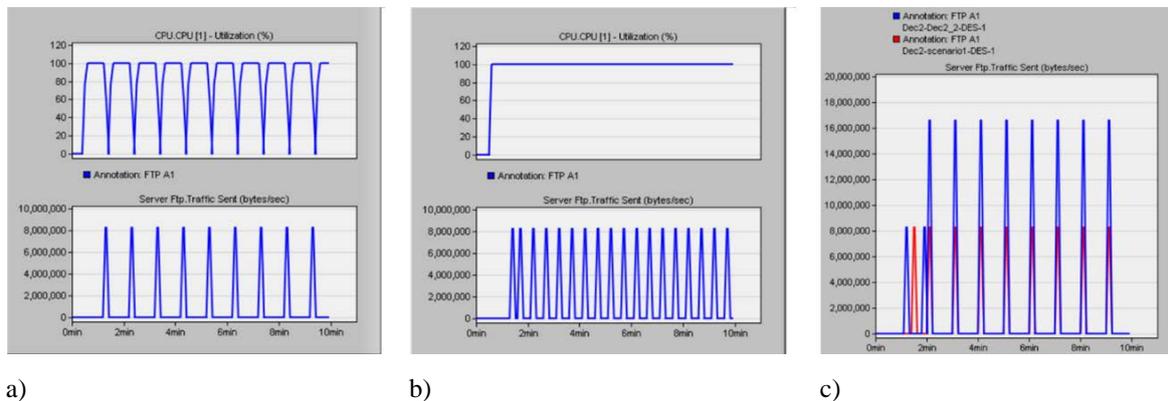


Figure 4: CPU usage and server traffic comparison between different traffic patterns in OPNET.

RESULTS

Using simulators in an undergraduate introductory networking course enables coverage of lower layer details, such as physical device configuration, which generally is not covered in a typical networking course. This also allows coverage of routing and switching topics, such as IP addressing and subnet management, which are usually discussed without hands-on practice in traditional teaching. Also, higher layer topics can be covered, such as mechanisms used in TCP and application protocols. More importantly, students can use these simulators to learn at their own pace and to explore new topics based on their interests. This is especially true for those students who are *hard-to-reach*, since they may have more flexible timetables and study plans. The author encourages students to submit weekly reports of their progress and concerns, so as to direct their learning and help them achieve study goals effectively.

Presented in Table 2 is an evaluation of simulators according to a set of criteria. One important criterion is the ability to drive personalised learning. Various simulators provide different functionalities and focus on different areas, and a single simulator is not adequate to cover all aspects or to meet all needs. The author evaluated those simulators from the perspective of all network layers, to demonstrate the potential learning outcomes and skills students would achieve by using them. For OMNeT++ and NS-3, only their built-in examples were evaluated, because it is easy to observe and investigate network fundamentals in those examples, but, as stated before, it is difficult and time-consuming for undergraduate students to introduce specific traffic patterns or behaviours into them. In a personalised learning environment, students choose the

most appropriate methods, work at their own pace, and design personalised experiments based on individual strengths and interests. Their learning activities should enhance and expand their comprehension of computer networks.

Table 2: Evaluation of simulators.

	<i>OMNeT++ (INET model)</i>	<i>NS-3</i>	<i>GNS3</i>	<i>Packet Tracer</i>	<i>OPNET</i>
<i>GUI support (drag and drop)</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
<i>operating system</i>	<i>Windows Linux</i>	<i>Linux</i>	<i>Windows Linux</i>	<i>Windows Linux</i>	<i>Windows</i>
<i>easy-to-build network topology</i>	<i>needs coding</i>	<i>needs coding</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
<i>capture packets and their details</i>	<i>Yes, in animation</i>	<i>Yes, in trace files</i>	<i>Yes, in trace files</i>	<i>Yes, in animation</i>	<i>No</i>
<i>adjust traffic patterns</i>	<i>needs coding</i>	<i>needs coding</i>	<i>No</i>	<i>Yes, parameter controlled</i>	<i>Yes, parameter controlled</i>
<i>support statistics collection</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>
<i>new model/logic development</i>	<i>needs coding</i>	<i>needs coding</i>	<i>No</i>	<i>No</i>	<i>needs coding</i>
<i>actual hardware configurations</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>
<i>support of L2 switching trace</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>
<i>routing table configuration</i>	<i>automatic/manual</i>	<i>automatic/manual</i>	<i>manual</i>	<i>manual</i>	<i>automatic/manual</i>
<i>routing protocol configuration</i>	<i>Yes, needs coding</i>	<i>Yes, needs coding</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
<i>IP addressing/ subnet config</i>	<i>automatic / manual</i>	<i>manual</i>	<i>manual</i>	<i>manual</i>	<i>automatic/manual</i>
<i>demonstrate L4 protocol details</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>
<i>support L5 protocols</i>	<i>needs coding</i>	<i>needs coding</i>	<i>part of</i>	<i>Yes, and show traces</i>	<i>Yes, performance comparison</i>

One undergraduate student delved into the example displayed in Figure 2, studied network firewall fundamentals, and configured a router to deny specific traffic between two hosts, as shown in Figure 5. In this case, the router served as a firewall to block packets from the host with IP *10.0.0.67* to host *192.168.0.20* in another subnet.

```

d. router denies TCP packets
Cisco3725#show access-lists
Extended IP access list 110
 10 deny tcp any any

e. 10.0.0.67 cannot reach 192.168.0.20
VPCS> trace 192.168.0.20 -P 6
trace to 192.168.0.20, 8 hops max (TCP), press Ctrl+C to stop
 1 *10.0.0.1 0.933 ms (ICMP type:3, code:13, Communication administratively prohibited)

```

Figure 5: Student configuration of a firewall to deny TCP packets.

One student group designed an experiment focusing on TCP error control. They used Packet Tracer to keep track of every packet and make connections among the packet sequence numbers and ACK numbers (partly shown in Figure 6), to explore how TCP relies on an ACK mechanism to guarantee reliability. These activities not only help students to build connections between the abstractions of networking fundamentals and levels of thinking, they also engage them in active learning and help them to achieve learning goals.

CONCLUSIONS

Network simulation is not only a modelling and testing method that helps researchers to perform network studies in a reproducible environment, it is also an important approach that helps students to observe network performance without the need for dedicated hardware. However, the integration of simulators into networking education depends on course requirements and instructor preference. The traditional teacher-centred approach does not meet the need of today's learners.

The use of simulators was reported in this article, to encourage personalised learning in networking courses. Widely used simulators were investigated to determine their advantages and disadvantages for integration into classes. The aim was to suggest a learner-centred environment in which learners can choose and use the most appropriate technologies to drive their learning at their own pace. Based on observation, following this approach, students have

more opportunities to learn protocol and mechanism details, and to delve into topics in which they are interested. Furthermore, students achieve a deeper understanding of network internals and gain practical networking skills.

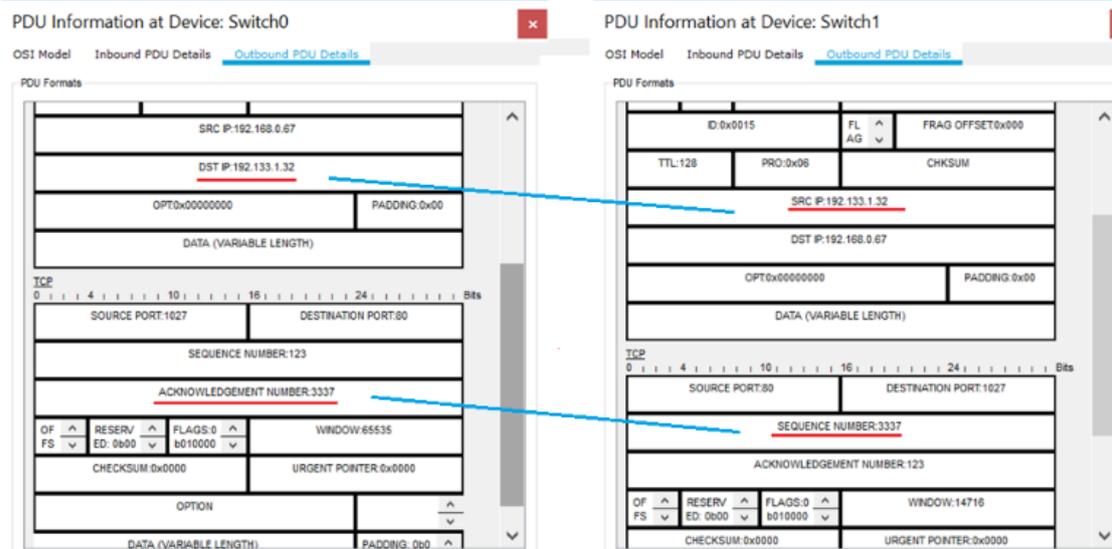


Figure 6: Connections between data packet header and ACK header in Packet Tracer.

REFERENCES

- Galip, E. and Bulut, H., Implementing the distributed breadth first search algorithm in OMNeT++ for teaching and learning purpose. *The Eurasia Proc. of Educational & Social Sciences*, 42-46 (2016).
- Fonseca, A., Camoes, A. and Vazao, T., Geographical routing implementation in ns3. *Proc. 5th Inter. ICST Conf. on Simulation Tools and Techniques*, 353-358 (2012).
- Valenzuela-Valdes, J., Pardo, P., Padilla, P. and Lozano-Guerrero, A., Low cost ubiquitous context-aware wireless communications laboratory for undergraduate students. *IEEE Trans. on Learning Technologies*, 9, 1, 31-36 (2016).
- Sanguino, T., Lopez, C. and Hernandez, F., WiFiSiM: an educational tool for study and design of wireless networks. *IEEE Trans. on Educ.*, 56, 2, 149-155 (2013).
- Tate, K. and Woody, T., Personalised learning, technology, and teacher education. In: Resta, P. (Ed), *Proc. SITE 2012-Society for Infor. Technology & Teacher Educ. Inter. Conf.*, 2406-2409 (2012).
- Robinson, C. and Sebba, J., Personalising learning through the use of technology. *Computers & Educ.*, 54, 3, 767-775 (2010).
- Kurose, J. and Ross, K., *Computer Networking: a Top-Down Approach*. (7th Edn), Pearson (2016).
- Holiday, M., Animation of computer networking concepts. *J. on Educational Resources in Computing*, 3, 2, article 2 (2003).
- Chang, R., Teaching computer networking with the help of personal computer networks. *Proc. 9th Annual SIGCSE Conf. on Innov. and Technol. in Computer Science Educ.* (2004).
- Schönwälder, J., Friedman, T. and Pras, A., Using Networks to Teach about Networks (Report on Dagstuhl Seminar #17112). *ACM SIGCOMM Computer Communication Review* (2017).
- OMNeT++. (2019), 1 April 2019, <https://omnetpp.org/>
- NS3 (2019), 1 April 2019, <https://www.nsnam.org/>
- GNS-3 (2019), 1 April 2019, <https://www.gns3.com/>
- Cisco Packet Tracer (2019). 1 April 2019, <https://www.netacad.com/courses/packet-tracer>
- Makasaranondh, W., Maj, S.P. and Veal, D., Pedagogical evaluation of simulation tools usage in Network Technology Education. *World Trans. on Engng. and Technol. Educ.*, 8, 3, 321-326 (2010).
- OPNET simulator (2019), 1 April 2019, <https://www.riverbed.com/products/steelcentral/opnet.html>

BIOGRAPHY



Dr Qian Liu is an assistant professor with the Mathematics and Computer Science Department at Rhode Island College, Providence, the United States of America. Dr Liu received his PhD degree in computer science from the University of New Hampshire in 2016. His research interests include computer networks, operating systems, simulation and learning technologies.