Network Simulator With Step Execution and Resume Functions for Understanding Mechanism of Communication: A Case Study for Layer 2 Access

Hnin Cherry
Faculty of Computer Systems and Technologies
University of Computer Studies
Yangon, Myanmar
hnincherry@ucsy.edu.mm

Yuichiro Tateiwa
Department of Computer Science
Nagoya Institute of Technology
Nagoya, Aichi, Japan
tateiwa@nitech.ac.jp

Zin May Aye
Faculty of Computer Systems and Technologies
University of Computer Studies
Yangon, Myanmar
zinmayaye@ucsy.edu.mm

***Abstract*— There are several students who cannot understand the mechanism of communication, which is a part of TCP/IP, by taking lectures and reading textbooks. Our system helps students to understand it by observing network behavior in trial and error. Our system takes network configuration, customized packets and step numbers as input and then calculates packet transmission step by step. The students can observe the behavior of networks simulated by our system. They can also edit the network states at any step and the simulation is restarted from the edited network state.**

Keywords— step execution, resume function, network simulator, TCP/IP

I. INTRODUCTION

In today's information society, developing networking skill is important for people. But one needs to know the underlying communication network to become a network engineer. To know how the network communication work, what action the devices takes when it sends and receives packets, which configuration values and field values in the device are responsible for those actions, and so on. When people who has not skillful in the network communication becomes network engineer, they create network application that seem to work well on the surface. However, due to the lack of understanding of how communication works, it may contain potential bugs.

Traditionally, students learn the mechanism of communication through lectures and textbooks mainly and by using physical devices supplementally. Traditional learning methods can't allow for the following easily:

- In building network by the learners, the communication is performed based on the content and timing of the packets that the learner wants to send.
- The learner checks the packet delivery and the internal state of the devices such as ARP table, MAC address table for one by one after receiving it; and
- Check the differences in the results for different inputs.

In order to solve the problems, we propose a network simulator with the following features:

- It takes network configurations, customized packets, and their transmission timings as inputs.
- It calculates sending, receiving and delivering packets step by step; and
- The learners can edit the network states at any step and the simulator restarts simulation from the edited states.

The rest of the paper is organized in sections as follows: Section 2 describes the related works in the studies of exercises with simulator and understanding TCP/IP with simulator. Section 3 describes how the students exercise with our system. Then, Section 4 presents how the simulator was implemented and the experimental result is described in Section 5. Section 6 includes our future works.

II. RELATED WORKS

George F. Riley proposed a network simulator used in classroom education [1]. It was especially used for graduate level, "CAD for Computer Networks" class. The class has two-pronged focus. Firstly, simply learning the capabilities and use one or more network simulation tools and secondly, these tools used to demonstrate network behavior of a number of different network topologies and under a variety of conditions, both wired networks and wireless networks. This study helps students to have learning experience in network course. In our study, students have to learn the communication mechanism in delivering packets.

In the study [2], they demonstrated how Packet Tracer can improve physical networking exercises to teach computer networking skills. They show that while Packet Tracer does help students perform better when using physical hardware, instructors must work with learners to bridge several key gaps. Learner perceptions on simulation were highly correlated with perceived skill transference. The difference between this study and our system is in the learning area. This study helps students to construct the networks but not to understand the TCP/IP. In our study, learners study the network behavior of

the communication mechanism as a virtual environment and how to send packets from one host to another host without using the physical devices.

In teaching various networking course, each learner has necessary accessed to build and test networks with different structures and components. Muhammad Wannous [3] used open source Virtualization and Virtual Network Computing technologies in a networking virtual web-based laboratory. This system gives the learners to draw, configure, test and troubleshoot in network running in the virtual mode on a host machine. In the experiment, a case study has been exercised to a group of learners as part of networking course. The evaluation was done in the comparison of their answers for a level test prior after the exercise has been completed. This showed that the learners achieved better results in this test and the system makes better efficiency. This study helps learners to exercise and practice computer networks. But it can't help the students to understand the communication mechanism for delivering packets.

Qian Liu[4] reports the use of network simulator to encourage personalized learning in networking courses. The aim was to suggest a learner-centered 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. In this study, they can help students to examine the network behavior and compare performance in various scenarios. In our system, the learners can study the network communication by resuming from any step with customized states and it provides students for efficient work. Our system also displays packet information with each step and it helps students to understand the communication mechanism.

The education based new computer network simulator was designed and implemented by [5] after comparing several kinds of mainstream computer network simulator. In this system, students can make experiment on computers like real network devices. This simulator was implemented to achieve for combination teaching with learning and especially used for certification test (e.g. Cisco Certified Network Associate (CCNA), Cisco Certified Network Professional (CCNP) and Cisco Certified Internetwork Expert (CCIE)). In this study, students can learn the relative network course in network experiment. But they can't learn how the packets are transmitting step by step between devices. Our system can help the learners to understand the packets information on each step along the communication.

III. EXERCISE WITH THE SYSTEM

In this section, we describe how students can learn with our system. They can observe packet transition between devices with animation and watch the position of packets each step using the visualizer. Our system visualizes how to process packets in devices and helps learners to understand the communication mechanism in transferring packet across a network.

A. Communication Mechanism

Our system helps learners to understand how to transport packets with basic network parameters such as communications protocols to organize network traffic, the network size, the topology, traffic control mechanism, IP fragmentation and error checking. For example, when a device will send ICMP echo requests, first, it tries to find the MAC address corresponding to the destination IP address from the ARP cache. Observing the actions, learners may understand the role of ARP cache. If learners understand the role of such parameters, learners can set up network configurations under considering the usage of the parameters in communication. The network size is being used depends on the network topology and network traffic control mechanism to capture packets transferring between devices. IP fragmentation occurs when packets are broken up into smaller pieces (fragments). While sending the data from the sender to the receiver there is a high possibility that the data may get lost or corrupted. In this case, error control is used to retransmit the data.

B. Learning Flow

The communication mechanism can be learned in the following step:

Step 1: Students design network configuration with IP addresses, MAC addresses, default gateway, subnet masks, and links between devices. They can also design the scenario that defines what packets are sent on what step numbers. The scenario contains the steps at which packets are sent, the devices that sends the packets, the packet types which the devices send and packet parameters consisting of a source IP address and a destination IP address when the packet type is ICMP-Echo and consisting of a source IP address and a source MAC address, a destination IP address, a destination MAC address when the packet type is ARP.

Step 2: Our simulator simulates the network communication (ICMP and ARP) for delivering packets using the network configuration and network scenario.

Step 3: Students observe the network behavior with visualization. In the visualizer as shown in figure 1, four main buttons are included for visualizing the network: start, stop, next and previous. Students can use the start button to begin the network visualization. They also can see the network visualization may be continued from the start number in the scenario until the step number has been reached at the end number. Students can see the current step number that is shown in the top-left corner on the user interface. Students can see the device information such as the packet creation and protocol state at that device on each step by clicking the red square button. The visualizer provides animation by displaying network states, including packet position, frame by frame. In the animation, the blue square is moved across the network. For example, if a device sends the packet with ICMP, the information may contain destination hosts and source hosts with Ethernet header, IP addresses and ICMP information. They can push stop button to pause the animation in a step by pressing the stop button; they can visualize a network state for the next step using the next button and previous step using previous button.

Step 4: If students want to change the packet transmission at a step, they can edit the network state at the step by changing IP addresses on devices and so on. After that, our system resumes the simulation from the step with the changed information.

Students can save time to compare the simulation result by resuming networks than by initializing networks. This help leaners to do try and error efficiently.

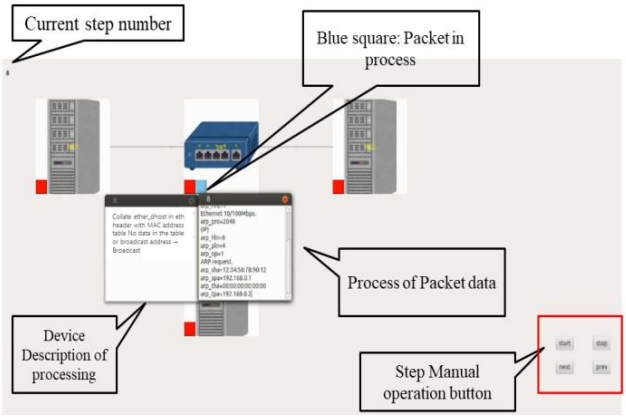

Fig. 1. Simulation Visualizer

IV. Implementation

A. System Structure

Figure 2 shows the system structure. Learners write a network configuration, a scenario, device states such as sending ICMP packet, finding MAC address, generate ARP request packet, and a step number under suspension of Simulator. Simulator loads network configuration, device states and scenario based on the input file. And then it writes out device states which contains of packet information, and the step number on a log file. Visualizer displays animation in network simulation from the first step to the end step. Learners can observe how the packet sends between devices with visualizer. The simulator can resume network simulation from any step on the customized statement using the edited log file.

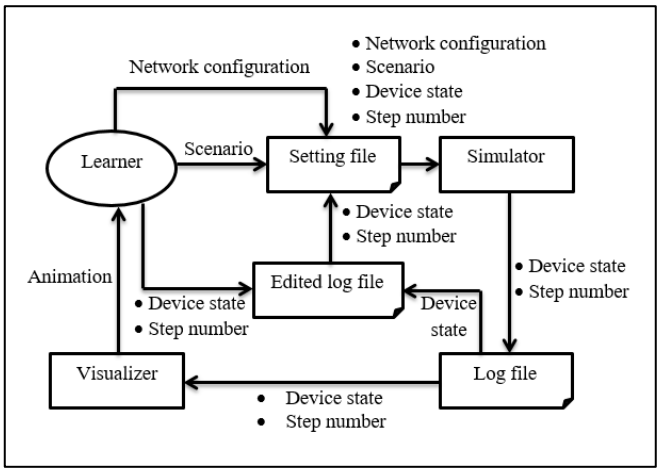

Fig. 2. System structure

B. Simulator

The simulator consists of device controller, scenario interpreter, and step counter in figure 3.

It has mainly used two inputs to simulate network states: network configuration and a scenario for simulating networks. Network configuration consists of device information such as the number of devices, device ID, IP address, MAC address, Default Gateway, Subnet Mask and cable connection between devices. Scenario consists of the step at which packet are sent, the devices that send the packets, the packet types , a source device IP address and a destination IP address and an end step number to end the simulation.

Scenario interpreter requests packet transmission to Device controller according to the scenario and the received step number from Step counter. If a step number in the scenario is matched with the step number on Step counter, scenario interpreter requests packet transmission to Device controller. Scenario interpreter interprets for the next step which packets will be sent and step counter also increases step number until the end step number in the scenario.

Device controller supports device parameters such as device ID, IP address, MAC address, Default Gateway, Subnet Mask and cable connection between devices to control packet forwarding. After a network state at a step number has been finished, the network state and packet information are saved as log file.

Step counter increases the step number one by one until the scenario interpreter have interpreted to the end. After having finished a packet transmission at a step according to the scenario, step number is increased by one. When network state needs to be resume, it set step number for next step.

Log file is used to gather the required information that can be visualized. Log file includes device information such as name, IP addresses, cable connections, step number and packet information on that state. Network states for each step that packets being sent and received through communication are recorded as log file. It can also be used in resuming network states from any step with reconfigured network settings. All device objects are written out to the log file with serialization technique and the file is saved as ".dat" format.

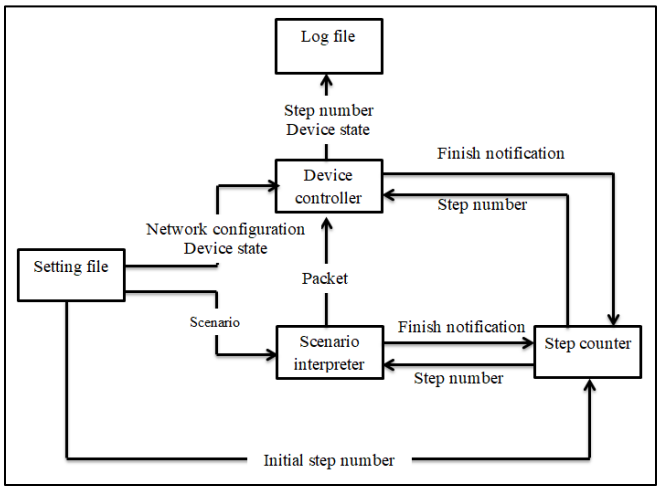

Fig.3. Data flow diagram for simulator

C. Step Execution

We divided the procedures for processing packets in devices and cables and call such a divided part step. We also appended the procedure for transiting to next step on the tail of each step.

Device controller executes the current step and transits to next step in each device after receiving a step number from the step counter. The order of the execution is hosts, switches and cables. Each step execution changes the network state by one step.

Figure 4 is a sample model of step division. When a step number is executed, the process of network communication can be changed depends on the step number. After the process at a step number has been finished, the step number is increased for the next step number. In executing step() function, either Statement 1 or Statement 2 or Statement 3 corresponding to ID is executed and then ID is updated.

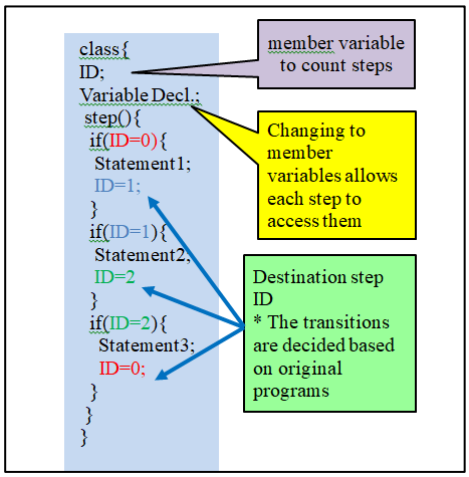

Fig. 4. Model of step division

D. Simulation Visualizer

We implement visualizer with current step number, device description of the process, packet processing information and step manual operation as shown in figure 1. Visualizer provides animation by displaying sequential network states from the first step to the final step. At first, visualizer loads the log file and then increase step number at a regular interval. Simultaneously, it displays the network state corresponding to the step. When the increment is stopped with the stop button, the next button increases the step number and the previous button decreases the step number one by one. Also, the visualization is updated depending on the step number.

E. Resume Function

When the simulation run firstly, each device state is empty and the step number is set to 0. The step number is increased and packet information are calculated based on the scenario. Network states are saved on log files after having the network simulated at each step. When resuming network simulation, each device state is set based on the edited log file and the step number is set based on the edited step number.

V. Experiment

The experiments are performed by configuring two network topologies through the simulation and analyzed the result depends on step numbers. The simulator is implemented by C++ and the visualizer is implemented by Qt. The experiments were conducted on Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz processor and 12GB RAM and VMware workstation player 16 as a virtual machine.

In our simulator, two network configurations are supported. The first configuration includes three hosts and one switch as shown in figure 5 and its configuration file and scenario file as shown in figure 7 and figure 8. In the figure 7, the first line consists of the number of host, switching hub, and cable. Each line in lines 2-4 consists of host ID, IP address, MAC address, default gateway, and subnet mask. The fifth line defines switching hub ID. Each line in lines 6-8 consists of source device ID, source port number, destination device ID, destination port number. In figure 8, each line in lines 1-3 consists of a step number for sending a packet, a source host ID, a protocol name, and a field data array. The array consists of a source IP address and a destination IP address when the protocol name is ICMP. The array consists of a source IP address and a source MAC address, a destination IP address, a destination MAC address when the protocol name is ARP. The fourth line defines an end step number. The second configuration consists of four hosts and two switches as shown in figure 6 and its configuration file and scenario file as shown in figure 9 and figure 10.

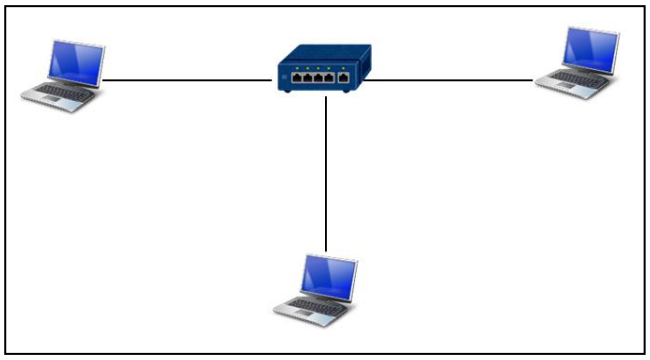

Fig. 5. A network that three hosts connect with one host

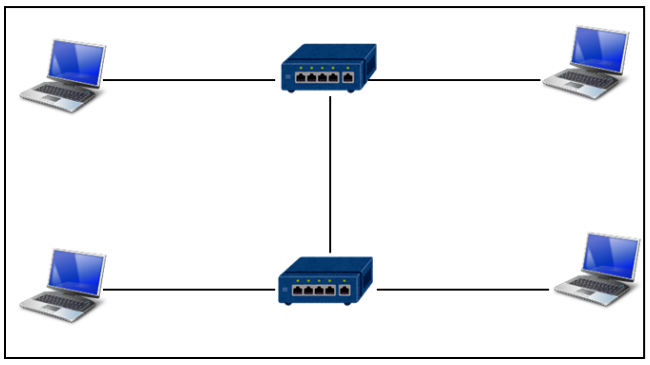

Fig. 6. A network that connects two switches with each other and each switch attached with two hosts

```
3,1,3,
hst1,192.168.0.1,12:34:56:78:90:12,192.168.0.254,255.255.255.0,
hst2,192.168.0.2,34:56:78:90:12:34,192.168.0.254,255.255.255.0,
hst3,192.168.0.3,56:78:90:12:34:56,192.168.0.254,255.255.255.0,
sw1,
hst1,0,sw1,0,
hst2,0,sw1,1,
hst3,0,sw1,2,
```

Fig. 7. Network Configuration Setting for figure 5

```
1,hst1,icmp,192.168.0.1,192.168.0.2,
50,hst1,icmp,192.168.0.1,192.168.0.2,
70,hst1,icmp,192.168.0.1,192.168.0.2,
100,
```

Fig. 8. Network Scenario for figure 5

```
4,2,5,
hst1,192.168.0.1,12:34:56:78:90:12,192.168.0.254,255.255.255.0,
hst2,192.168.0.2,34:56:78:90:12:34,192.168.0.254,255.255.255.0,
hst3,192.168.0.3,56:78:90:12:34:56,192.168.0.254,255.255.255.0,
hst4,192.168.0.4,78:90:12:34:56:78,192.168.0.254,255.255.255.0,
sw1,sw2,
hst1,0,sw1,0,
hst2,0,sw1,1,
hst3,0,sw2,0,
hst4,0,sw2,1,
sw1,2,sw2,2,
```

Fig. 9. Network Configuration Setting for figure 6

```
1,hst1,arp,192.168.0.1,12:34:56:78:90:12,192.168.0.4,FF:FF:FF:FF:FF:FF,
60,hst1,icmp,192.168.0.1,192.168.0.4,
61,hst1,icmp,192.168.0.1,192.168.0.4,
100,
```

Fig. 10. Network Scenario for figure 6

A. Result

The performance measurement was done based on the following parameters: Execution time (s), Resident Set Size (KB), CPU utilization (MB), and Log File Size (MB) based on the total number of step. We experimented the two types of network topologies while varying the number of steps from 100 to 1000. As shown in the following table I, as the number of steps increases there is a linear growth in execution time and the log file size. Resident set size (RSS) is not much difference along changing the number of steps. CPU utilization was measured while varying the number of steps. For our simulator, there is little effect on CPU utilization as the number of steps increases. Number of devices use in topology does not depend on step numbers. Step number depends on time taken for packet delivered from each device.

TABLE I. ANALYSIS RESULT

Total number of step	Execution time (s)	RSS (KB)	CPU Usage(MB)	Log Size (MB)
100	1.6	11076	113	12.1
300	3.2	11044	113	36.3
700	7.4	11004	113	84.4
1000	11.1	11052	113	120.0

VI. FUTURE WORK

We have implemented on integration with protocols such as TCP and UDP and several network topologies into the simulator. Additionally, existing implementations of protocols have to be modifying with different devices such as routers. Although the evaluation in performance showed that the simulator is quite scalable, we further want to improve resource usage and faster execution time.

VII. CONCLUSION

Network simulation is the procedure in which we simulate and implement our basic ideas in it. The network simulator is designed for step execution and resumes function in communicating network. By resuming the network state, the network can restart from an arbitrary point. It generate log file to resume the network state with step number and is easy to use and flexible in simulation. The simulator is implemented by C++ and Qt Library is used for visualizer. It can help us to achieve a good result. In the future work, other network protocols functions such as Transmission Control Protocol TCP and UDP will be implemented.

REFERENCES

[1] George F. Riley, “Using Network Simulation in Classroom Education”, Proceedings of the 2012 Winter Simulation Conference.

[2] Jim Marquardson, David L. Gomillion, “Simulation for Network Education: Transferring Networking Skills Between Simulated to Physical Environments”, Information Systems Education Journal (ISEDJ), February 2019.

[3] Muhammad Wannous, Student Member, IEEE, and Hiroshi Nakano . NVLab, a Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologies. IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 3, NO. 2, APRIL-JUNE 2010.

[4] Qian Liu, “Applying simulators in computer networks education to encourage personalised learning”, Global Journal of Engineering Education Volume 21, Number 2, 2019.

[5] Wei-Xin Cai, Gui-Sen Li, Xu-Hui Chen, Chao-Qun Hong, Shun-Zhi Zhu, Qin-Hong Wu, Ren Chen. Education Based New Computer Network Simulator Design and Implementation .The 11th International Conference on Computer Science & Education (ICCSE 2016) August 23-25, 2016. Nagoya University, Japan.