The simulation and emulation verification that was based on NS-2

Ju-Young Shin, Jong-Wook Jang, Jin-Man Kim

Department of Computer Engineering Dong-Eui University, Busan, Korea
Department of Computer Engineering Dong-Eui University, Busan, Korea
Korea Institute of Science and Technology Information, Yuseong-gu Daejeon, Korea

Summary
At the present time, several Network Simulators are in use. However, due to economical and experimental limitations, the fields of network and computer engineering mostly utilize NS-2.This is where the necessity of emulation whose operation environment is built on hardware comes in. The comparative analysis between emulation node which is realtime scheduler including Ping of NS-2 Tool's external Host and the simulator of non-realtime scheduler of the existing simulation has been performed in this paper. The most significant difference is that emulation can interlock with the packet of external real network. The objectives of this article are as follows: the identical virtual network is constructed on NS-2, its performance is evaluated by simulation and emulation and emulation method is described.

Key words:

NS-2, emulator, simulator, ping packet, live packet

1. Introduction

Various tools are in use today to design models equivalent or similar to actual Network environments.

NS2(Network Simulator version 2), the representative of simulation tools, can simulate TCP/IP Protocol Group including TCP, UDP, FTP, and HTTP and Internet Protocols such as Multicasting Protocol, RTP(Real Time Protocol), and SRN(Scalable Reliable Network). As it supports ad-hoc Network, Base Station model, WLAN, Mobile-IP related Protocol, UMTS, and wireless Network such as Satellite Network, it is Network Simulator that has wide range of application.

Most of the current tools including simulation by NS-2 have certain limitations. For instance, the existing simulation tools to execute performance analysis for hand-over algorithm in network environments having many real-time elements such as wireless networks mostly do not fully reflect the actual network environment. This is where the necessity of emulation whose operation environment is built on hardware comes in.

In other words, simulation technology that creates virtual environments on software by using computer program is needed due to the situation that the technology creating the actual network environment on hardware and the designing of the actual network environment are not feasible.

The emulation technology exploited here replicates one system from another. The second system repeats the activity of the first system. Emulation technology which is shown above is different from simulation technology in the sense that it replicates external activity. The reason for this is that simulation technology is pertinent to abstract model of the system to be represented. The designing technology that realizes a network environment, most similar to the actual network environment is required. In particular, the necessity of an integrated tool of emulation and simulation technology that realizes network environment by using simulation technology and efficiently tests the realized network environment by utilizing emulation technology has become significant.

2. Structure Comparison

2.1 The Communication among the Existing Simulation Nodes

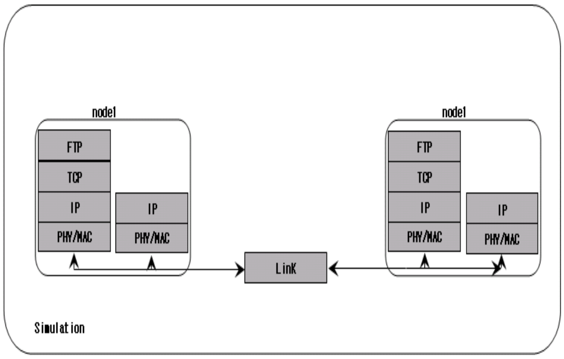


Figure1. The Communication among Simulation Nodes

The existing simulation nodes, whose Phy/Mac, IP/ICMP (Network), TCP/UDP (Transport), and FTP/CBR (Application) are set by node-config. They communicate with nearby nodes through link.

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

2.2 Emulation Structure

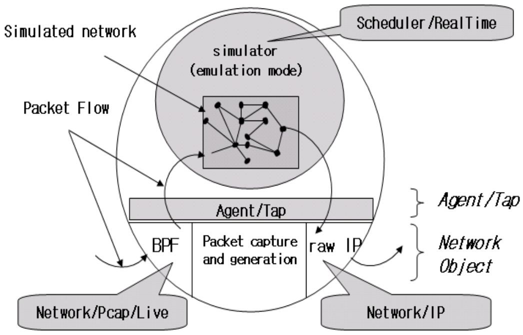


Figure 2. Emulation Structure

The figure above is the emulation structure. Network Objects as real-time scheduler are entry points that send and receive the actual network data (Live date hereafter). It also connects with network interface. TAP Agents takes a role to insert or extract Live Date in or from, respectively, Simulator Packet. They also send or receive Packet to or from, respectively, Network Objects.

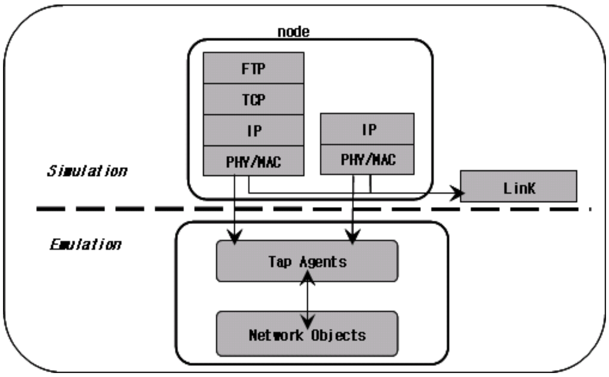


Figure 3. Emulation Node

Fig3. Emulation node is a node that Simulation Node's node-config is set and Tap Agents and Network Nodes are attached for Network Emulation.

2.3 The Communication Utilizing Emulation Node

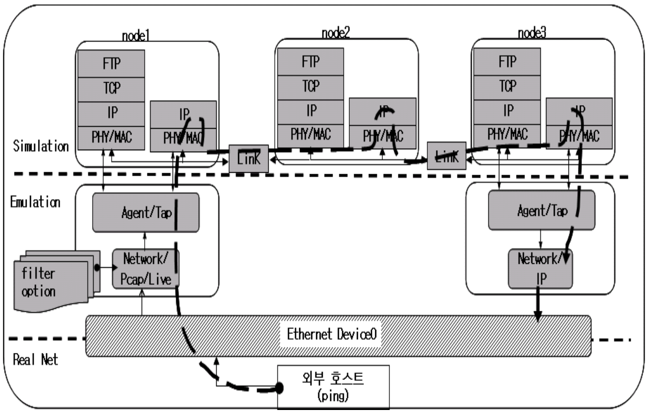


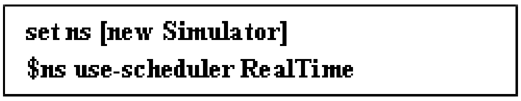
Figure 4. Communication among Emulation Nodes

Structure shown above is a simulation using the ping. In this structure, packet is transported from exterior to simulation interior through agent and network and the communications among simulation interior nodes take place. (For limitations, packet transport from exterior to interior is possible; however, packet transport from interior to exterior host is not.)

Exterior host performs ping with IP address of the host where simulation takes place. Simulation host that receives ping packet sends back ping response to exterior host that has sent ping immediately upon receiving. Ping terminates regardless of simulation. Ping received back is used as trigger. In this scenario, when ping packet arrives at simulation host, bpf agent (having pcap/live network object as network) of start_node captures icmppacket that has come into its own ip in accordance with filtering rule. Through this, start_node forms and delivers live packet to pass_node. The pass_node sends the packet to exit_node. (Now, live packet is not ping packet.)

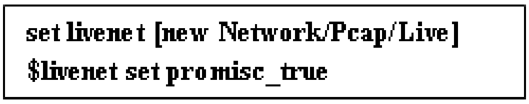
The following experiments compare the difference between simulator and emulator by drawing respective network results from simulator which is simulation environment through interior execution on NS-2 and emulator that includes emulation node using input signal from exterior host. That is to say that the objectives of the experiments are to show the difference between emulator environment and simulator environment and the similarity with the actual network by using time verification calculation dependent on whether exterior emulation including ping exists or not.

2.4 Emulation Core Tcl Code



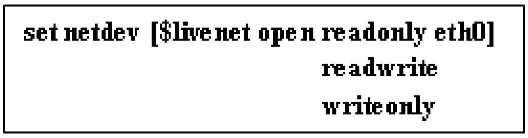
```
set ns [new Simulator]
$ns use-scheduler RealTime
```

Figure 5. Ns Scheduler Creation



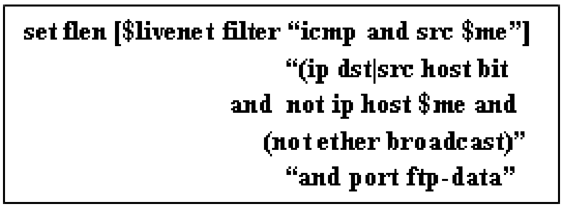
```
set livenet [new Network/Pcap/Live]
$livenet set promisc_true
```

Figure 6. Network Object Creation



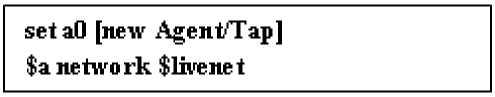
```
set netdev [$livenet open readonly eth0]
                          readwrite
                          writeonly
```

Figure 7. Network Device Acquisition of Network Object



```
set flen [$livenet filter "icmp and src $me"]
                          "(ip dst|src host bit
                     and  not ip host $me and
                       (not ether broadcast)"
                         "and port ftp-data"
```

Figure 8. Filter Setting



```
set a0 [new Agent/Tap]
$a network $livenet
```

Figure9. Network Object Installation at Tap Agent

3. Experiment and Analysis

3.1 Simulation Environment

Classification	Type	Detail	Note
System	CPU	Pentium-3 933MHz	
	RAM	512MB	
	OS	Redhat Linux-9	
Tool	Ns-2	Ns-2 2.29	Network simulator
	NAM	NAM 1.11	Animation Tool
	Gnuplot	Gunplot 3.7	Graph Tool
	Xgraph		
Experiment Environment	Cueing Model	Drop-Tail Queue	

Table 1. Simulation Environment

3.2 Experiment 1

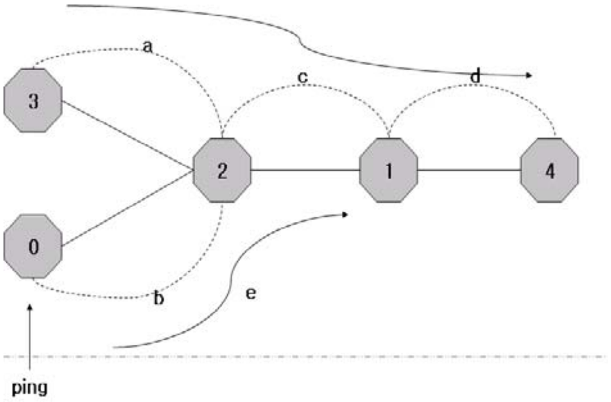


Figure 9. UDP Structure

UDP does not perform extra error controls but it is checked with checksum. When error occurs, thing that's created is dependent on the countermeasure. ICMP (Internet Control Message Protocol) is used to announce the occurrence of an error. UDP Agent is an agent to realize connectionless transport protocol Agent/UDP and Agent/NULL. In general, the node that creates UDP packet is designated as source node, the node that produces NULL packet as destination node.

Fig. 10 is Node 3 and 4 inserted in Fig. 4's movement mechanism. As common insertions for simulator and emulator, UDP Agent and CBR Traffic are attached to node 3 and NULL is attached to node 4.

Analysis elements are packet size, packet interval time, link type, bandwidth, delay time, etc. In this verification, for packet transport delay in particular, c in the above operation structure is only varied as 1Mb, 8Mb, and 15Mb.

- a,b,d : bandwidth=30Mb
link delay=10msec
- c : bandwidth=1Mb, 8Mb, 15Mb
link delay=10msec
- e : packet interval=1
packet size=1000byte
- f : packet interval=0.001
packet size=1000byte

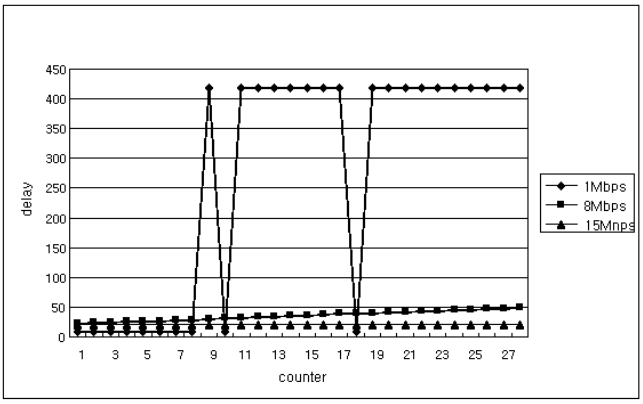


Figure 10. UDP-simulator

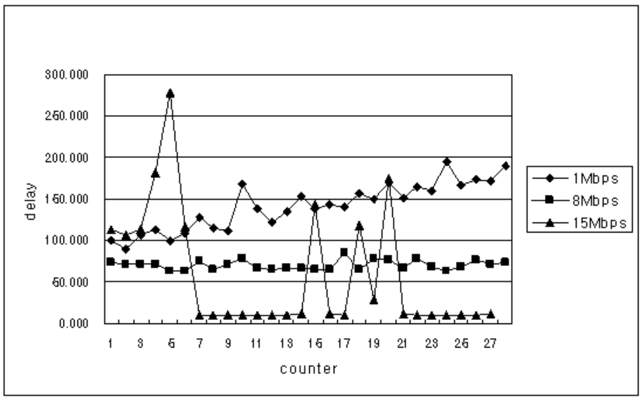


Figure 11. UDP-emulator

The above graphs in Fig. 10 and 11 show link delay from n_0 to n_2. For simulator, within 1Mbps environment, the sections of 1~8 seconds, 10 seconds, and 18 seconds are the arrival times from node 0 to 2. From the rest, it can be seen that packet drops. The other sections include delay time to node 1. For 8Mpbs, a small amount of delay increase equivalent to the bandwidth of cbr0 packet's size can be identified. For 15Mbps, the same amount of delay took place.

For emulator, in 1Mpbs situation, apart from simulator's constant delay pattern, a more varying delay and increasing delay time have been recognized. In 8Mpbs situation, as simulator, it shows incremental increase.

In 15Mpbs situation, different from other 7~14 seconds sections, a large amount of delay error happened. Several sections have been identified for delay time to significant extents by repeating the experiment several dozen times.

3.3 Experiment 2

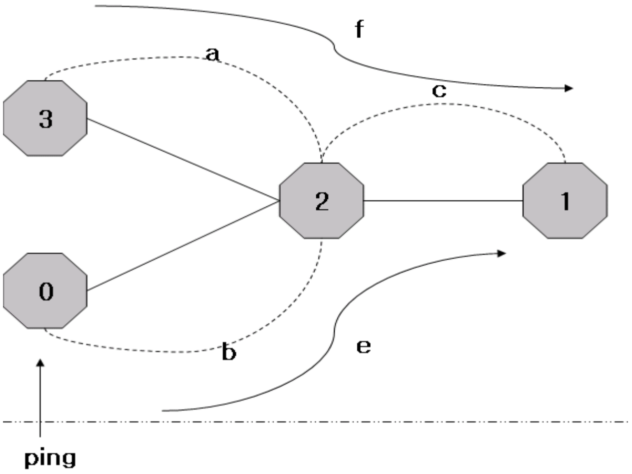


Figure12. TCP Structure

To simulate TCP transport, transport layer protocol, Agent/TCP and Agent/TCPSink must be set in pairs. In general, the agent that creates and sends TCP Segment is designated as source host and the agent that receives TCP Segment as destination host. TCPSink Agent that acts as destination TCP Agent forms and sends ACK packet which is the response to TCP Segment sent from TCP Agent of source host. In short, in simulator, TCP class substitutes for TCP Agent and sends data to TCPSink.

Only node 4 is omitted from Fig. 10 to make Fig. 13. As common insertions for simulator and emulator, node 3 uploads TCP agent and attaches FTP agent. In a similar way, simulator uploads cbr traffic of UDP agent to node 0 to make a structure that does not contain ping from emulator structure. This experiment is centered on packet transport delay. Due to TCP operation structure, each section is subject to a single simulation with bandwidths of 5Mbps.

- a,b,d : bandwidth=5Mb
  link delay=10msec
- e : packet interval=1
  packet size=1000byte
- f : packet interval=0.004
  packet size=1000byte

Due to characteristic of TCP, as no arbitrary interval can be given to e, the interval to e was set by calculating the number of packet per second by checking packet data in tr file.

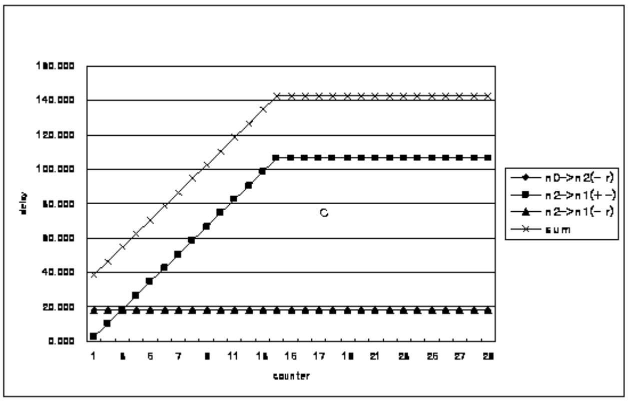


Figure 13. Tcp-simulator

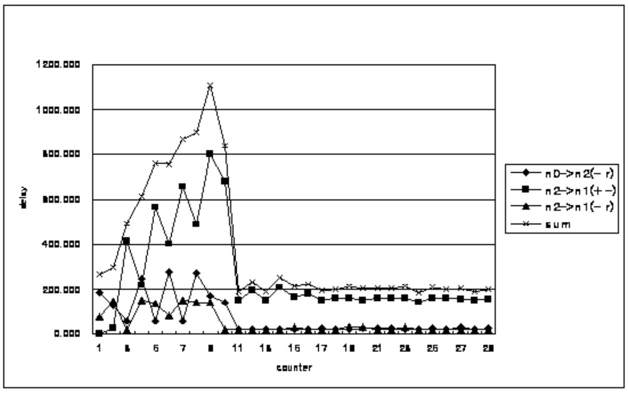


Figure 14. Tcp-emulator

Fig. 14 and 15 reveal experiment results in TCP environment. (In legend for above graph, the 1st item means the delay time from node 0 to node 2. The 2nd item time stands for cueing delay time prior to transport from node 2 to node 1. The last item shows the sum of them.) For simulator, the delay times from node 0 to 2 and packet transport time after cueing among delay times from node 2 to node 1 are constant; however, the cueing time prior to the transport from node 2 to node 1 gradually increases for some time and becomes constant.

Tr file analysis result shows that the number of packet from node 0 to 2 is constant for the previous 30 seconds. The number of TCP transport packet gradually increases from node 3 to 1 during the section of node 2 and node 1 and becomes saturated after 14 seconds. For emulator, the amount of delay is large at 1~11 second section. According to tr file analysis, the section up to 11 seconds shows highly reduced TCP packet transport number different from the sections afterwards. The sections afterwards are characterized by more delay than that of simulator.

3.4 Result Analysis

In UDP environment for simulator, cbr packet is delivered from node 3 to node 4 at the interval of 0.001. For emulator, however, the packet delivered from node 3 to node 4 in simulation part is not regular. If TCP shows the same result, it can be considered as the characteristics of emulator rather than simulator. Having this as the hypothesis, tr file of either UDP or TCP was compared. Similar to UDP, for TCP and different from TCP of simulator, the irregular delivery of ftp packet was revealed on Nam. Therefore, we reached a temporary result that "Regardless of agent insertion in emulator structure, apart from 'pure' simulator, it is variable bearing similar characteristics of emulator." As previous UDP and TCP showed the same pattern, packet delay time from node 0 to node 1 was comparatively analyzed. Question arising from the result is the fact that in emulator environment, only slight difference in transport time delay between nodes was shown but there were no other delays.

As real-time scheduler part of emulator showed similar result from simulator when applied to annotation, it is reasoned that real-time scheduler part that includes ping agent was involved to bring the result.

As the result, either TCP or UDP, arbitrary agent insertion in emulator structure derives change in cueing time. For other delay times, the change during specific section is not constant but variable.

4. Conclusion

As emulation technology performs replication on a hardware basis, ideal performance test can be executed. However, the actual realization of network environment using hardware is difficult and it faces problems such as high cost and time consuming nature of the performance test. On the contrary, simulation technology can build up network environment with less money and less time.

Also, maintenance work and performance test are easier for simulation technology. Yet, simulation technology has the problem of not actually reflecting obstacles and limitations in the real network environment as it operates in ideal network environment which is different from the real network environment. Considering this problem, this thesis provides simulation method by utilizing emulator to overcome the limitation of simulation by simulator and to approach the more realistic value of the real network. Due to limited resources for experiment, various analysis elements were not used. If following experiments take place and verification and analysis in multiple network environments are executed, the reliability for the organization of actual network can be enhanced and quality network can be secured. I hope that our new trial can have a significant impact on the field of simulation.

Acknowledgements

This paper was supported in part by MKE (Ministry of Knowledge Economy) & IITA (Institute for Information Technology Advancement). (08-Infrastructure-13, Convergence of IT Devices Institute) and Ministry of Commerce, Industry and Energy (MOCIE) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation

References

[1] The Network Simulator ns-2: Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html

[2] NAM: Network Animator(NAM), http://www.isi.edu/nsnam/nam/

[3] NS by Example, Jae Chung, Mark Claypool, http://perform.wpi.edu/NS/index.html

[4] Park, Joo-won, Tutorial for the Network Simulator "ns", K-JIST

[5] Ns Simulator Course for Beginners, Uni. De Los Andes, Merida, Venezuela, http://www-sop.inria.fr/mistral/personnel/Eitan.Altman/ns.htm

[6] Hye Hwan Ahn, Jea gi Son, A research about Adding a New Network Component in network Simulator 2, SungKyunkwan University