A High Accurate and Component Based Network Emulator for Simulation of Complex Heterogeneous Network Topology.

K W Lien, Jeff Reeve
School of Electronic and Computer Science, University of Southampton, Southampton

Key words to describe this work: Network Simulator/Emulator, Heterogeneous Network

Key Results: This paper proposes a novel framework for a wireless network simulator/emulator. A complex wired network is simulated at individual Linux PC by reusing the real Linux TCP/IP stack. Then, traffic from simulated host is redirected to a wireless proxy for heterogeneous networks emulation. Links features, router setting, TCP parameters and MTU are readily adjusted. Modules of different queuing disciplines are also involved inside the emulator.

How does the work advance the state-of-the-art?: An network emulator with a simple architecture but designed for complex network topologies, can enhance the accuracy of simulated future wireless and heterogeneous researches.

Motivation (Problems addressed): Wireless features like quick handover, radio jamming and dynamic data rates cannot be truthful presented inside the simulator, but concentrates on all layers above the physical layer. Previous network emulators were designed for wired network or router relative emulations. A complex network topology is not easy to be constructed and integrated. Developing new modules are also a challenging job based on previous emulators' framework.

Introduction

Several different types of emulator have been presented. NIST Net [1] has been implemented as a Linux module and hooked inside Linux kernel that works as a router. It is used to emulate performance dynamics on live IP packets passing through a Linux-based router. Seawind [2] focusing on the wireless emulation has been used on wire-less protocols, such as GPRS and UMTS. Dummynet [3], Hitbox pseudo-devices [4] and x-SIM [5] are different kernel extensions intercepting packets via real network stack and emulate transmission performance in a live network. However, all earlier works are routing-base and hosts are built require separate machines, and setting up the nodes to emulate transport layer is difficult. Mahrenholz[6] has presented a new emulator combining NS-2 simulator [7] with live network for wireless emulation, but the restriction is the wireless protocols uses ns-2 wireless modules. Real wireless features, such as quick handover and external jamming, might be difficult to estimate; in addition, each wired node must be supported by individual Linux PC, so it is difficult to maintain PC clusters to emulate complex network topologies. IMUNES [8] presents a framework in which multiple virtual nodes are simulated in one FreeBSD PC. However, it is designed for wired network emulation and its queuing disciplines modules must cooperate with other emulators.

Architecture and Results

Our framework references but enhances the methodology of Harvard network simulator [9]. Main architecture can be divided into four parts as presented in figure 1.

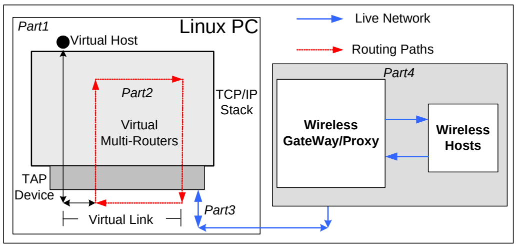

Fig 1. The emulator framework.

Part1: Complex network topologies are simulated on a Linux PC. Topologies are constructed with multiple network nodes by reusing the real Linux TCP/IP stack of the simulated PC multiple times, relying on features of the TAP device, virtual link, private IP address and routing mechanisms. Virtual links are simulated at a user-level program when packets are received by the TAP device.

Part2: The routing architectures are maintained inside the Linux kernel. Therefore, router relative mechanisms supported by Linux modules are suitable for our framework.

Part3: IP address has been changed to the wireless host IP address before a packet is injected into a live network; or is changed to our private IP address after a packet is received by the simulated PC. In this phase, correct paths must be also set on routing tables on both the simulated PC and the wireless gateway.

Part4: This is the live wireless networks. Because the real IP address and routing path are used, no additional tasks should be done.

The concept of using virtual nodes inside a kernel for network emulation is not novel, but our framework differs from other projects in the following points:

- Large numbers of virtual nodes are simulated on only one Linux machine. This is efficient to simulate large, complex network topologies.
- The simulator can be connected with different live wireless gateway for the emulation of heterogeneous networks.
- The traffic between virtual host and router are using the real Linux network stack; hence, emulation results are realistic and accurate compared with that combination of conceptual network simulator.
- The emulated architecture is not only routing emulation. A virtual edge node (client or server) can be created and connected by multiple virtual router nodes. It means both routing and end-to-end emulation can be accomplished at our simulator.
- The simulator is component based, which means Linux routing modules can be involved within minuscule changes

Tested results of four nodes routing path at Part1 have been presented as following:

```
traceroute 192.168.6.1
1  192.168.2.1 (192.168.2.1)  15.000 ms  15.000
ms  14.000 ms
2  192.168.4.1 (192.168.4.1)  28.000 ms  30.000
ms  28.000 ms
3  192.168.6.1 (192.168.6.1)  42.000 ms  42.000
ms  42.000 ms
```

"ftp" application is used for traffic generation. "ping" and "tcpdump" utilities are useful for traffic monitor also.

Conclusion

We have presented an accurate, component based, and versatile network emulator framework. The differences between variant projects have been addressed. Individual PC emulation is fully working. Live network connection and TCP performance over heterogeneous networks in the future will be improved in the revise version.

Reference:

[1] M. Carson and D. Santay, "NIST Net: a Linux-based Network Emulation Tool," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 111--126, 2003.

[2] M. Kojo *et al.*, "Seawind: A Wireless Network Emulator," *in Proc. of 11th GI/ITG Conferenc*, Aachen, Germany, Sept. 2001.

[3] L. Rizzo, "Dummynet: A Simple Approach to the Evaluation of Network Protocols," *ACM Computer Commun. Review*, vol. 27, no. 1, pp. 31--41, 1997.

[4] P. Danzig, Z. Liu, and L. Yan, "An Evaluation of TCP Vegas by Live Emulation," Department of Computer Science, USC," Technical Report, 1994.

[5] L. Brakmo and L. Peterson, "Experiences with Network Simulator," *in Proc. of SIGMETRICS*, Philadelphia, USA, May 1996, pp. 80--90.

[6] D. Mahrenholz and S. Ivanov, "Real-Time Network Emulation with ns-2," *in Proc. of 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications*, Budapest Hungary, Oct. 2004.

[7] S. McCanne and S. Floyd. ns-LBNL Network Simulator. [Online] http://www.isi.edu/nsnam/ns/.

[8] M. Zec, "Operating System Support for Integrated Network Emulation in IMUNES," *in Proc. of 1st Workshop on Operating System and Architectural Support for the on demand IT Infrastructure*, Boston, USA, Oct. 2004.

[9] S. Wang and H. Kung, "A Simple Methodology for Constructing Extensible and High-Fidelity TCP/IP Network Simulators," in Proc. IEEE INFOCOM'99, New York, USA, Mar. 1999, pp. 1134--1143.