

NEX: A Network Emulator boX to support the design of networked systems

Stefano Cailotto, Franco Fummi, Davide Quaglia, Walter Vendraminetto

{cailottofummilquaglia}@sci.univr.it

ESD Group - University of Verona – Italy

<http://esd.sci.univr.it>

Abstract

The Network Emulation boX (NEX) merges live and simulated networks to provide a complete testing tool for the design of networked systems and the planning of the network infrastructure.

1. Introduction

Networks are becoming more and more important in everyday life and, therefore, computers and embedded systems are required to interact each other in a complex communication environment.

Manufacturers, both HW and SW, have to test the network capabilities of their products while network administrators should plan link capacity and Quality of Service policies.

Until now, to test a network appliance or an application we can either connect it to an actual network or describe its behavior into a network simulator. The former option has a high fidelity but limits the reproducibility of tests and may be expensive if different types of networks have to be supported; the latter option is less expensive and more deterministic but it requires an extra effort to re-implement the functionalities under test in the network simulator. A possible solution to merge advantages from both options is the so-called *network emulation* which consists in connecting the systems under test to a server which reproduces the behavior of the network. Since this server can be seen as a router, current solutions are based on the network functionalities of the Linux kernel [1-5]. These solutions reproduce link behavior such as queue overflow, delay, and reordering but do not allow to model multilink networks, physical and data link protocols, routing protocols, and concurrent traffic.

The Network Simulator 2 (NS-2) is a well-known simulation tool with network emulation capabilities [6] even if they are not yet completely stable. Many issues have to be faced to provide a mature tool for network emulation based on NS-2 and important improvements have been introduced by Mahrenholz and Ivanov [7].

Our work consisted in the further improvement of this solution and in the development of a complete and easy-to-use HW/SW appliance named Network Emulation boX (NEX).

2. Architecture

Figure 1 shows the basic architecture of the NEX. The NEX is a Linux box with a high performance CPU (typically 3GHz) and a large amount of RAM (typically 4GB).

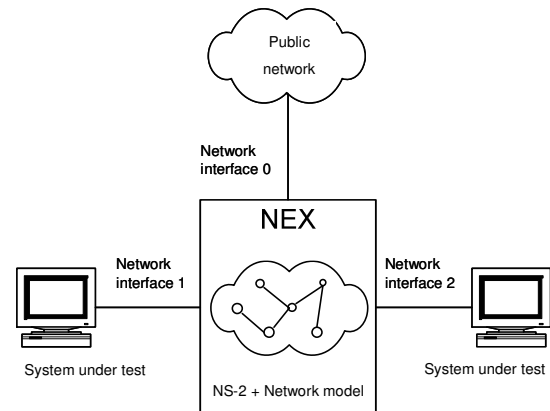


Figure 1: Basic architecture of the NEX.

The NEX executes a patched version of NS-2 which simulates a model of the network. This model consists of nodes connected to links with given capacity and delay; packets circulate in this network according to models of MAC and routing protocols; UDP and TCP connections can be created between nodes and traffic sources can be associated to these connections.

Some nodes can be associated to the actual network interfaces of the NEX (e.g., interfaces 1 and 2 in Figure 1). Actual packets arrived at these interfaces are translated into virtual packets of the simulated network; viceversa, NS-2 packets received by these special nodes are translated into actual packets leaving the corresponding interface. It is worth noting that packet forwarding between the systems under test is performed by NS-2, not by Linux forwarding capabilities.

It is important to keep separated the systems under test from the public LAN to avoid that extraneous packets interfere with the experiment and reduce the available bandwidth. For this reason dedicated network interfaces are used while a further interface (i.e., interface 0 in Figure 1) can be used to access NEX for remote management.

NEX simulates the network behavior in real time and it accepts packets at a typical LAN rate (about 100 Mb/s), even if this rate may vary according to the model complexity.

Figure 2 shows a more complex version of the NEX which provides more than two network interfaces to connect systems under test. Since the number of network card that can be installed into a PC is limited, a switch has been embedded into the NEX. In this architecture, virtual LANs may be used to separate the systems under test from the public LAN.

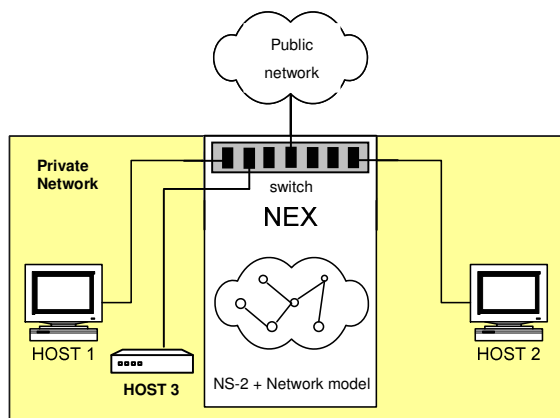


Figure 2: NEX with multiple interfaces.

3. Some real-life applications

The NEX has been successfully used to simulate network behaviors in real life cases.

One of the initial usage of NEX has been the estimation of video quality degradation of IPTV. The transmitter and the receiver exchange data at a constant bitrate. The network model exhibits slowing or breaking links according to predefined testing patterns; on the receiver side, data arrives with delayed/lost packets, so the receiver will experience a low image quality. NEX was effectively used to test adaptive transmission algorithms which try to mitigate network congestion problems. In this case, the traditional approach of implementing such strategies in a network simulator was impossible because of the presence of mixed HW/SW multimedia codecs. On the other hand, the use of an actual network as testbed would have introduced non-deterministic events.

NEX has also been used by a railway telephony manufacturer to model dependability of redundant telephone lines in case of multiple failures, and to test the introduction of VoIP solutions into an existing network. In this case, NEX was used not to test new HW but to support network planning. The setup of an actual network as testbed would be too expensive since the reference scenario consisted of geographical links and several nodes.

In both cases, a former knowledge of the actual network has been used to build the models, and then simulation results have been used to identify and solve weakness in the systems under test and in the network infrastructure, respectively.

In both cases, the presence of real-time multimedia transmissions shows the effectiveness of the solution to guarantee a prompt fulfillment of time constraints.

4. Conclusions

This paper shows a complete and easy-to-use tool based on network emulation to test networked systems and communication architectures. The tool is based on a well-known network simulator –NS-2– which allows to reproduce many aspect of the communication process. The simulator is hosted by a fine-tuned HW architecture consisting of a high performance processing unit and a switch to separate test network from public network.

The use of NEX for network simulations eliminates the need to re-describe the behavior of the system under test in the network simulator; devices can be added by just plugging them into the NEX. Furthermore, support for new protocols can be added as soon as they are available for NS-2. Network models can be re-used thus accelerating the development steps.

The application of NEX to the test of multimedia transmissions shows its effectiveness in performing real-time network simulation.

Acknowledgment

This research activity was partially supported by the European project FP6-2005-IST-5-033506 - ANGEL

5. References

- [1] M. Carson and D. Santay, *NIST Net: a Linux-based network emulation tool*, in ACM SIGCOMM Computer Communication Review, Vol. 33, No. 3, Jul. 2003, pp. 111-126.
- [2] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan. *Evaluation of TCP Vegas: Emulation and experiment*, In SIGCOMM 95, Cambridge, MA, August 1995.
- [3] L. Rizzo, *An Embedded Network Simulator to Support Network Protocols' Development*, in Proc. of Tools '97, St.Malo, June 1997, LNCS 1245, Springer Verlag.
- [4] L. Rizzo, *DummyNet: a simple approach to the evaluation of network protocols*, in ACM Computer Communication Review, Vol. 27, No. 1, pp. 31-41, Jan. 1997.
- [5] S. Hemminger, *Netem - emulating real networks in the lab*, <http://linux-net.osdl.org/index.php/Netem>.
- [6] S. McCanne and S. Floyd, *NS Network Simulator – version 2*, URL: <http://www.isi.edu/nsnam/ns>.
- [7] D. Mahrenholz, S. Ivanov, *Real-Time Network Emulation with ns-2*, in Proc. of Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications, 21-23 Oct. 2004, pp 29-36.