

Hybrid Testbed for Network Scenarios (Poster Abstract)

José M^a Saldaña, Eduardo Viruete, Julián Fernández-Navajas, José Ruiz-Mas, José I. Aznar
Communication Technologies Group (GTC) – Aragon Institute of Engineering Research (I3A)
Dpt. IEC. Ada Byron Building. CPS Univ. Zaragoza
50018 Zaragoza (Spain)
Tel: +34 976 762 698

{jsaldana, eviruete, navajas, jruiz, jiaznar}@unizar.es

ABSTRACT

In this paper we explain the deployment of a testbed that emulates a network by means of virtualization, allowing us to implement a set of virtual nodes in one physical machine or a small LAN. Nodes participating in the communication are virtual machines adequately connected. The system is a hybrid testbed, as it includes simulation and emulation. But simulations are done in a previous offline stage, in order to avoid computer load caused by the need of real-time simulations, which could limit the maximum number of hosts to be included into the scenario. Finally, two uses of the testbed are presented.

Categories and Subject Descriptors

C.2.4 [Computer Communication Networks]: Distributed Systems – *distributed applications*

General Terms

Measurement, Performance, Design.

Keywords

call admission control, emulation, hybrid testbed, IP mobility, virtualization, Xen

1. CONTENT

Many universities and R&D departments need tools to verify protocols' behaviour in realistic conditions. Sometimes, tests are very difficult to carry out because of the number of machines that are required. One of the possibilities to solve this problem is the use of simulation tools, which allow making controlled and repeatable measurements with low costs. They have two drawbacks: computational load and lack of accuracy, because they have specific implementations of protocols, making a simplification of the system under test. Another option is the use of real hardware. Large scale experiment facilities where these tests can be deployed [2, 7] have been started around the globe in recent years. This will achieve a great accuracy, but this kind

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTools 2010 March 15–19, Torremolinos, Malaga, Spain.

Copyright 2010 ICST, ISBN 978-963-9799-87-5.

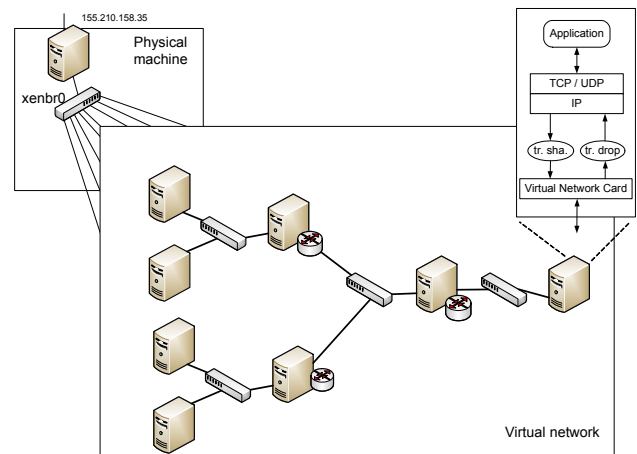


Figure 1. Control network and test network

of measurements could be very expensive and complicated. Some experiment facilities integrate emulators to imitate the behaviour of real elements. But some groups may prefer small dedicated and well controlled testbeds to carry out their experiments, as they allow accuracy, repeatability, dynamicity, flexibility and scalability. A large number of testbeds and emulators have been deployed to verify and perform in a laboratory the proposed solutions [3]. Some of them are hybrid testbeds that combine the advantages of simulation, emulation and real equipment tests, joining the accuracy and realism of emulated and physical networks, and the scalability and repeatability of simulation.

In the present work we explain the deployment of a testbed that emulates a network by means of virtualization, allowing us to implement a set of virtual nodes in one physical machine or a small LAN (Fig.1). Nodes participating in the communication will be virtual machines adequately connected. This technique improves the scalability of the system, and does not require a lot of hardware resources due to its simplicity. Virtual machines use applications and protocols without any modification, obtaining accuracy, as the behaviour is the same as in an actual system. We have used Xen *paravirtualization*.

At link level, different tools can be used in order to emulate the link behaviour: some of them are traffic shapers, as the Linux tool *Traffic Control* (*tc*), or NIST NET [1], and other ones that make packet dropping, like *iptables* and *Mackill* (Fig. 2), which is a part of APE testbed (Ad hoc Protocol Evaluation testbed) [5]. It consists of a kernel module that adds a MAC filter to the

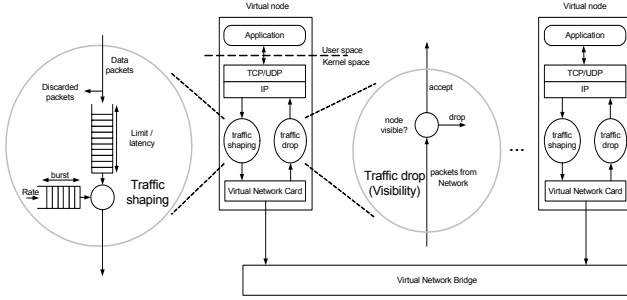


Figure 2. Network model with link emulation

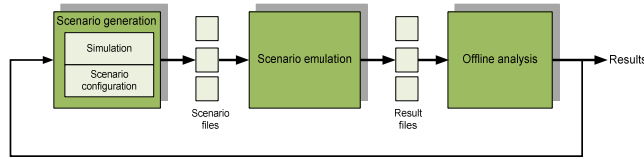


Figure 3. Testbed usage stages scheme

Table 1. Call establishment delay in ms

$Delay_{no\ CAC}$		$Delay_{CAC}$	
mean	std. dev.	mean	std. dev.
2.36	0.52	7.87	0.83

protocol stack, and drops packets coming from non-visible nodes. *Mackill* reads the information of node's visibility from a previously generated file in order to emulate mobility. We have used the *NetEm* [4] tool to emulate the network. It allows introducing controlled delays, packet losses, duplication and reordering with different distributions and statistics.

The presented system is a hybrid testbed, as it includes simulation and emulation tools. But simulations are done in a previous offline stage, in order to avoid computer load caused by the need of real-time simulations, which could limit the maximum number of hosts to be included into the scenario. The testbed is used in three stages (Fig. 3): The first one includes two sub-stages that have to be carried out simultaneously, because the scenarios have to be simulated and also established in the testbed with virtual machines. Simulation and network configuration sub-stages will affect each other. Anyway, the use of virtualization will require the monitor of processor load, as one physical machine will include a number of virtual machines. This drawback has led us to use applications with low computer load and to measure CPU usage (Fig. 4).

We have used the testbed to carry out two different kinds of measurements, one for wireless networks in which some handoff measures of MIPv4 have been deployed (Fig. 5), and other in which the preliminary tests of a Call Access Control for an IP Telephony platform [6] are done (Table 1). They have been useful to validate the testbed, comparing our results with the ones obtained by other groups.

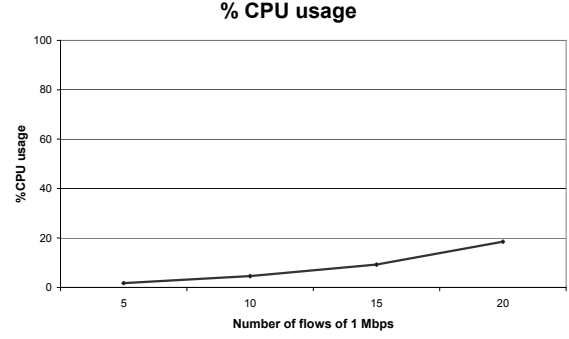


Figure 4. CPU usage with 1500 bytes packets

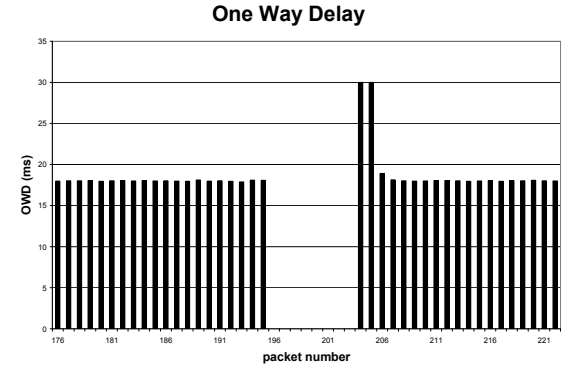


Figure 5. OWD during a MIPv4 handover

REFERENCES

- [1] Carson, M. and Santay, D., NIST Net: a Linux-based network emulation tool. In SIGCOMM Comput. Commun. Rev. 33, 3, pp. 111-126 (2003)
- [2] Gavras, A., Karila, A., Fdida, S., May, M., Potts, M., Future internet research and experimentation: the FIRE initiative, ACM SIGCOMM Computer Communication Review, v.37 n.3 (2007)
- [3] Göktürk, E., A stance on emulation and testbeds, in Proceedings of the 21st European Conference on Modelling and Simulation ECMS 2007. SCS (2007)
- [4] Hemminger, S., Network Emulation with NetEm. In Proceedings of Linux Conference AU, Canberra (2005)
- [5] Lundgren, H., Lundberg, D., Nielsen, J., Nordström, E., Tschudin, C., A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations. Proc. IEEE Wireless Communications and Networking Conference WCNC'02 (2002)
- [6] Saldaña, J. M., Aznar, J. I., Viruete, E. A., Fernández-Navajas, J., Ruiz Mas, J., QoS Measurement-Based CAC for an IP Telephony system, In .QShine, The Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Las Palmas de Gran Canaria, Spain (2009)
- [7] Turner, J. S., A proposed architecture for the GENI backbone platform, In Proc. Architecture for Networking and Communications Systems (2006)