NEPTUNE
Network Emulation for Protocol Tuning and Evaluation

[Poster Abstract]

Pasquale Di Gennaro, Roberto Canonico, Giorgio Ventre
University of Napoli Federico II
Via Claudio 21, 80125, Napoli, Italy
{pasquale.digennaro, roberto.canonico, giorgio.ventre}@unina.it

ABSTRACT

In the last few years many emulation systems have been developed to evaluate the effectiveness of new protocols and applications in realistic and controllable network scenarios. NEPTUNE is a flexible architecture for cluster-based network emulation systems designed and implemented at University of Napoli Federico II. In this poster we present the key components of the NEPTUNE architecture, an in particular the NEPTUNE manager, an open-source software managing the lifecycle of emulation experiments.

Categories and Subject Descriptors

C.2.1 [Computer Communication Networks]: Network Architecture and Design

General Terms

Design, Experimentation

Keywords

Network emulation, Testbed, Xen, Virtual routing

1. NETWORK EMULATION SYSTEMS

In the last few years, network emulation has gained interest in the community of network researchers, being considered an important technique to evaluate the effectiveness of new protocols and applications in heterogeneous, controllable and realistic network scenarios. Today's most complex network emulation systems are cluster-based. These systems are made of a large number of hardware components arranged in a common facility that can be remotely accessed by users through a web interface. In a typical cluster-based network emulation system, users submit to the system an experiment request. An experiment request contains a "virtual" network description to be reproduced with the available cluster resources. An experiment description usually contains at least the following information:

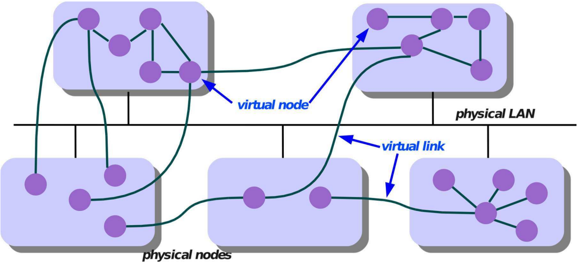


Figure 1: Virtual network topology on a physical cluster.

- the "virtual" network topology, including a list of relevant parameters for each of the nodes and links;
- operating systems and software packages to be loaded in each of the emulated nodes.

When the experiment is activated (swap-in), a system module maps the logical topology of the experiment onto actual testbed hardware, loads the requested operating system and software onto the allocated devices and establishes the network links.

2. NEPTUNE ARCHITECTURE

NEPTUNE is a cluster-based emulation system developed by the COMICS Research Group at University of Napoli Federico II. A NEPTUNE system consists of a collection of physical machines running a hypervisor software layer, and a centralized management entity, the NEPTUNE manager, which acts as a web frontend for end users, and takes responsibility of instantiating the emulated virtual topologies onto the cluster physical nodes.

As of today, the NEPTUNE emulation system runs on a cluster of workstations consisting of 28 HP ProLiant DL380 servers, each equipped with two Intel Pentium IV Xeon 2.8 GHz CPUs, 5 GB of PC-2100 RAM, one 100 Mbps Ethernet NIC, one Gigabit Ethernet NIC. Each node is equipped with a 34.6 GB SCSI disk. A 700GB centralized disk array is also available to the whole cluster. The cluster nodes are connected each other through a set of 100/1000 Ethernet switches.

SIMUTOOLS '09, Rome, Italy

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5653
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5653

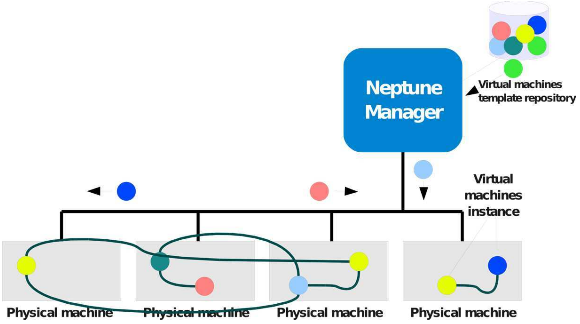


Figure 2: VM instantiation.

2.1 Node and link multiplexing

A conservative resource allocation policy consists in mapping the emulated "virtual" nodes onto dedicated PCs and emulated links onto switched ethernet links. Nowadays, with increasing computational power made available at low-cost, it is possible to exploit virtualization techniques to map multiple "virtual" nodes on a single CPU.

The problem of emulating more than a network node on just one physical cluster node is called node multiplexing. The node multiplexing technique we chose for NEPTUNE is Xen, due to its many advantages, and in particular because Xen (i) potentially supports different kinds of Operating Systems; (ii) provides good isolation among different virtual machines running concurrently; (iii) supports virtual machine migration, allowing dynamic re-allocation of experiments on the cluster nodes; (iv) exhibits good communication performance among virtual machines implemented within the same physical node.

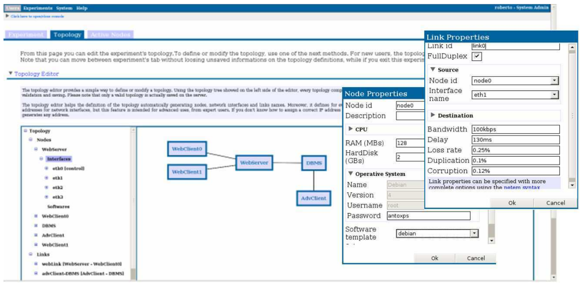
The problem of emulating multiple point-to-point connections by means of a single NIC is called link multiplexing. In NEPTUNE multiple virtual links are multiplexed onto the same shared physical link by associating each virtual link endpoint to a different Virtual NIC.

2.2 NEPTUNE manager

The main component of the architecture is the NEPTUNE Manager (NM in short). On one hand, NM acts as a web frontend for end users. By adopting the GWT technology, it provides a highly interactive User Interface to design network topologies and let users manage their experiments lifecycle. During the experiment design phase, NM is also able to automatically define an IP address allocation scheme on behalf of the user. When it comes to activate an experiment, NM decides how to allocate emulated virtual nodes onto available cluster physical machines, using a Lin-Kernighan derived optimization algorithm. Once the allocation mapping has been defined, NM instantiates virtual machines whose image is taken from a repository of predefined VM templates. The NM also enforces the emulated network topology by automatically assigning private IP addresses to the VM's virtual NICs.

The topology of emulated networks is internally saved by NM in an XML format that can be later submitted to the system to create new experiments based on the same networking scenario.

NEPTUNE manager is an open-source project, available at http://code.google.com/p/neptune-network-emulator/.



From this page you can edit the experiment's topology. To define or modify the topology, use one of the next methods. For new users, the topolog
Note that you can move between experiment's tab without loosing unsaved informations on the topology definitions, while if you exit this experim

The topology editor provides a simple way to define or modify a topology. Using the topology tree showed on the left side of the editor, every topology comp
validation and saving. Please note that only a valid topology is actually saved on the server.

The topology editor helps the definition of the topology automatically generating nodes, network interfaces and links names. Moreover, it defines for e
addresses for network interfaces, but this feature is intended for advanced uses, from expert users. If you don't know how to assign a correct IP address
generates any address.

```
⊟ Topology
  ⊟ Nodes
    ⊟ WebServer
      ⊟ Interfaces
        ⊞ eth0 [control]
        ⊞ eth1
        ⊞ eth2
        ⊞ eth3
        Softwares
    ⊞ WebClient0
    ⊞ DBMS
    ⊞ AdvClient
    ⊞ WebClient1
  ⊟ Links
    ⊞ webLink [WebServer - WebClient0]
    ⊞ advClient-DBMS [AdvClient - DBMS]
```

Node Properties

Node id	node0
Description	
▶ CPU	
RAM (MBs)	128
HardDisk (GBs)	2
▼ Operative System	
Name	Debian
Version	4
Username	root
Password	antoxps
Software template	debian

Link Properties

Link id	link0
FullDuplex	☑
▼ Source	
Node id	node0
Interface name	eth1
▶ Destination	
Bandwidth	100kbps
Delay	130ms
Loss rate	0.25%
Duplication	0.1%
Corruption	0.12%

Link properties can be specified with more complete options using the netem syntax

Figure 3: Neptune Manager screenshot.

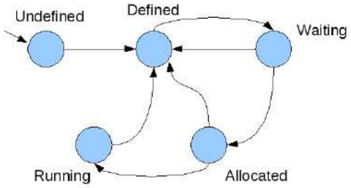


Figure 4: Experiment lifecycle.

3. RELATED WORK

Maybe the most complex cluster-based emulation system developed so far is EMULAB, which is developed by the Flux Group at the University of Utah. EMULAB is a free-for-use, Web-accessible, time- and space-shared, reconfigurable network testbed, providing integrated access to a wide range of experimental environments. In EMULAB an enhanced version of FreeBSD jail has been used, which allows the creation of isolated environments (vnodes) on the same physical node. The jail virtualization mechanism does not offer fully isolated execution environments, potentially creating some security issues. Even the network is not completely virtualized, since much of the network stack is shared between physical host and vnodes. Even though many design assumptions made for NEPTUNE were borrowed from EMULAB, NEPTUNE has assumed Xen para-virtualization as a key technology.

The NET testbed at University of Stuttgart also provides a configurable network environment for the performance analysis of distributed applications and protocols. It consists of a cluster of PCs running Linux and connected by means of a Gigabit Ethernet switch on which an arbitrary number of VLANs can be configured. Through the use of VLANs, nodes can be connected according to any possible virtual topology, allowing the emulation of different kinds of networks. Network traffic on emulated links is controlled by a special traffic shaper module, called NETshaper, implemented as a Linux kernel module and providing a link-layer emulation that is completely transparent to upper layers.

4. ACKNOWLEDGMENTS

This work has been partially supported by the CONTENT EU Network of Excellence (IST-FP6-038423) and partly funded by the FP7-ICT-224263 "OneLab2" project and by Regione Campania in the framework of the Project ANGELO (POR 3.17).

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5653
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5653