NEPTUNE for fast and easy deployment of OMF virtual network testbeds

[Poster Abstract]

Roberto Bifulco, Giovanni Di Stasi, Roberto Canonico
University of Napoli Federico II
Via Claudio 21, 80125, Napoli, Italy
r.bifulco@consorzio-cini.it, {giovanni.distasi, roberto.canonico}@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 first introduce the key components of NEPTUNE, then we show how, by leveraging the features of the OMF open-source platform (cOntrol and Management Framework) our platform is able to completely automatize the experimental procedure, from the definition of an emulated network topology to the collection of experimental results. Our system allows the fast creation of "virtual OMF testbeds", a feature that can be useful for teaching OMF and demoing its capabilities.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation

Keywords

Virtual Testbeds, Network emulation, OMF, Xen

1. INTRODUCTION

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
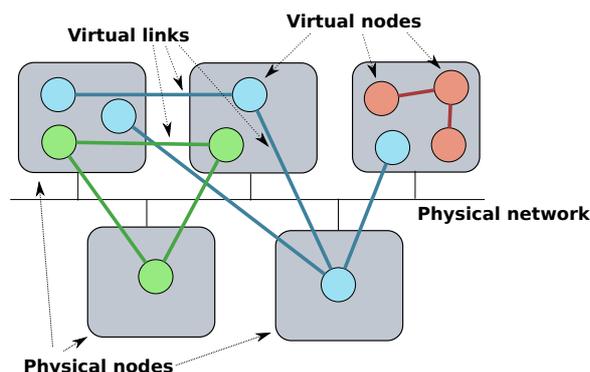
Figure 1: Virtual network topology on a cluster of PCs.

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.

Most of existing emulation systems concentrate on the provisioning of resources but lack of procedures which would automatize the execution of experiments and the collection of results. These features are some of the main characteristics of OMF, a testbed *cOntrol and Management Framework* originally developed for the ORBIT testbed. This is why we decided to leverage OMF to add these important capabilities to our NEPTUNE emulation system. What we obtained is a system that allows the fast and automatic creation of "virtual OMF testbeds", a feature that can be useful for teaching OMF and demoing its capabilities.

2. OMF ARCHITECTURE

OMF is a platform supporting the management and the automatic execution of experiments on a networking testbed. Currently OMF is being developed by NICTA [3].

OMF supports all the phases of an experiment lifecycle, from the provisioning of resources to the collection of experimental data. The most important component is the *Experiment Controller* (EC), which is also the interface to the user. It accepts as input an *experiment description* and takes care of orchestrating the testbed resources in order to accomplish the required experiment steps. It interacts with the *AggregateManager*, the entity responsible of the resources of the
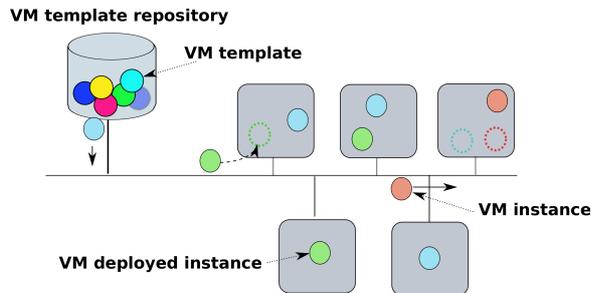
Figure 2: VM instantiation in NEPTUNE.

testbed as a whole, and provides some basic services to the EC, such as checking the status of a node, rebooting a node, etc. The EC also interacts with the *Resource Controllers* (RCs) installed on the testbed nodes, that are responsible of performing local configuration steps and of controlling the applications, e.g. a traffic generator.

3. NEPTUNE ARCHITECTURE

NEPTUNE is a cluster-based emulation system developed by the COMICS Research Group at University of Napoli Federico II [2]. 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, NEPTUNE runs on a cluster of 28 HP Pro-Liant 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.

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

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

4. AUTOMATIC DEPLOYMENT OF A OMF VIRTUAL TESTBED

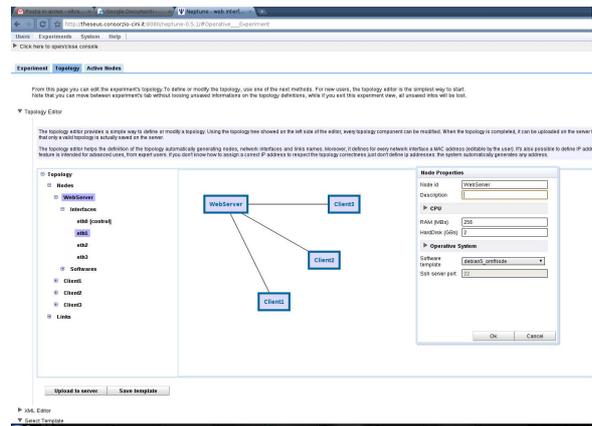To allow the fast and automatic execution of experiments

Figure 3: Neptune Manager screenshot.

on user-defined emulated virtual networks, we decided to extend our NEPTUNE platform by giving to it the ability to instantiate OMF-ready virtual testbeds.

The integration of the two platforms comprised two steps:

1. Configuration of two VM templates: i) one preconfigured with OMF EC and AG instances and ii) another one preconfigured with a OMF RC.

2. Creation of a set of scripts for automatically configuring OMF components using NEPTUNE's virtual topology description.

To automatically deploy an OMF virtual testbed, a user describes the testbed's topology he/she needs to perform experiments, by means of the NEPTUNE GUI. The defined virtual testbed must use virtual nodes templates that include OMF components. A typical virtual testbed setup could be composed of a single virtual node hosting OMF EC and OMF AG, and several virtual nodes hosting OMF RCs. Once the virtual topology has been defined in NEPTUNE, our scripts automatically configure the OMF components according to the provided virtual topology. Once this last step is completed, the virtual testbed is ready to be used. User can then access the nodes through secure ssh connection or supply an experiment description to the OMF EC, which will execute the experiment. At the end of the experiment execution, the experimental results will be available on a MySql database.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] P. Di Gennaro, R. Canonico, and G. Ventre. NEPTUNE network emulation for protocol tuning and evaluation. In *Proceedings of Simutools '09*, ICST, Brussels, Belgium, Belgium, March 2009.

[2] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar. OMF: A control and management framework for networking testbeds. *ACM SIGOPS Operating Systems Review*, 43(4):54–59, 2009.

7. DEMO DESCRIPTION

During the demo we will show how it is possible with NEP-TUNE to create a virtual testbed and execute a networking experiment on it by using the facilities offered by OMF.

The first stage will be the definition of the network topology, which will be performed with a handy graphical tool available on NEPTUNE. It is also possible, but it will not be shown, to define the topology by means of other tools, being the network definition an xml file with a well defined xml-schema.

After the definition of the virtual testbed, the platform will be asked to deploy the network and to enable access to the nodes and the OMF EC. We will then log into the machine hosting the EC and ask it to perform a simple routing experiment. At the end of the experiment the database hosting the results will be queried to get the some experimental results.