TERENA NETWORKING CONFERENCE 2012
Reykjavík, Iceland

Distributed Virtual Scenarios over Multi-host Linux Environments:
Virtual Networks over LinuX (VNX)

David Fernández, Jorge Somavilla,
Verónica Mateos, Omar Walid,
Jorge Rodríguez, Francisco J. Martín
Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid. Madrid, Spain.
david@dit.upm.es

Francisco J. Monserrat,
Miguel Ferrer
RedIRIS

Fermín Galán
Telefónica I+D

More info available at: http://www.dit.upm.es/vnx

Introduction

Virtualization based testbeds widely used for the creation of network environments needed to test protocols and applications.

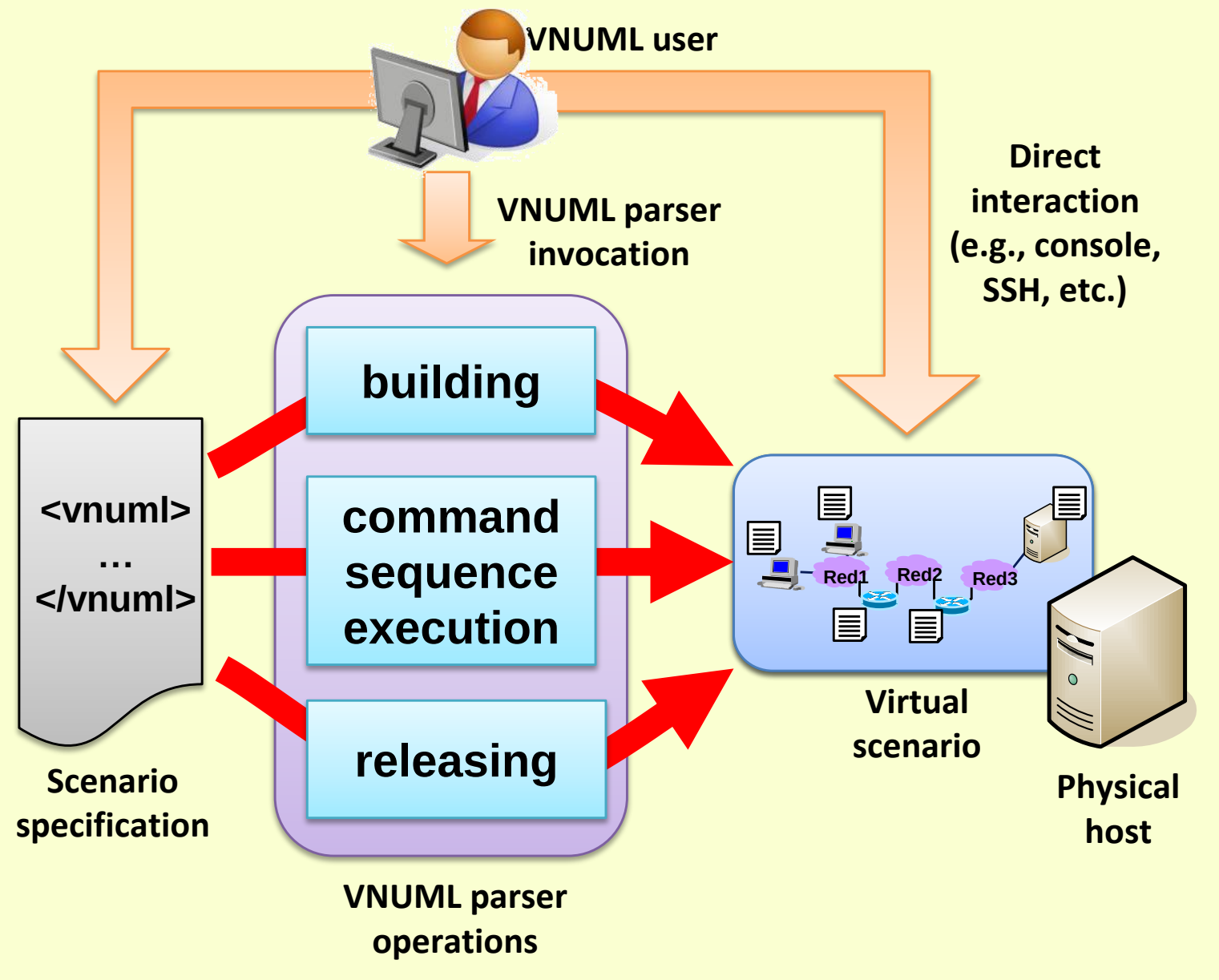
However, complexity of present networks and protocols arises the need of very complex network testbeds, made of tenths or hundreds of virtual machines.

Need for tools to support the design, deployment and management of large virtual network scenarios over clusters of servers

Apart from large scale initiatives, none of the available tools (Netkit, MLN, Marionnet, CORE) support distributed deployment or the diversity of virtual machine operating systems needed for complex testbeds.

Previous work : VNUML

Virtual Networks User Mode Linux (VNUML) is a general purpose open-source scenario-based management tool designed to help building virtual network testbeds automatically.



VNUML Operation Workflow

VNUML made of two main components:

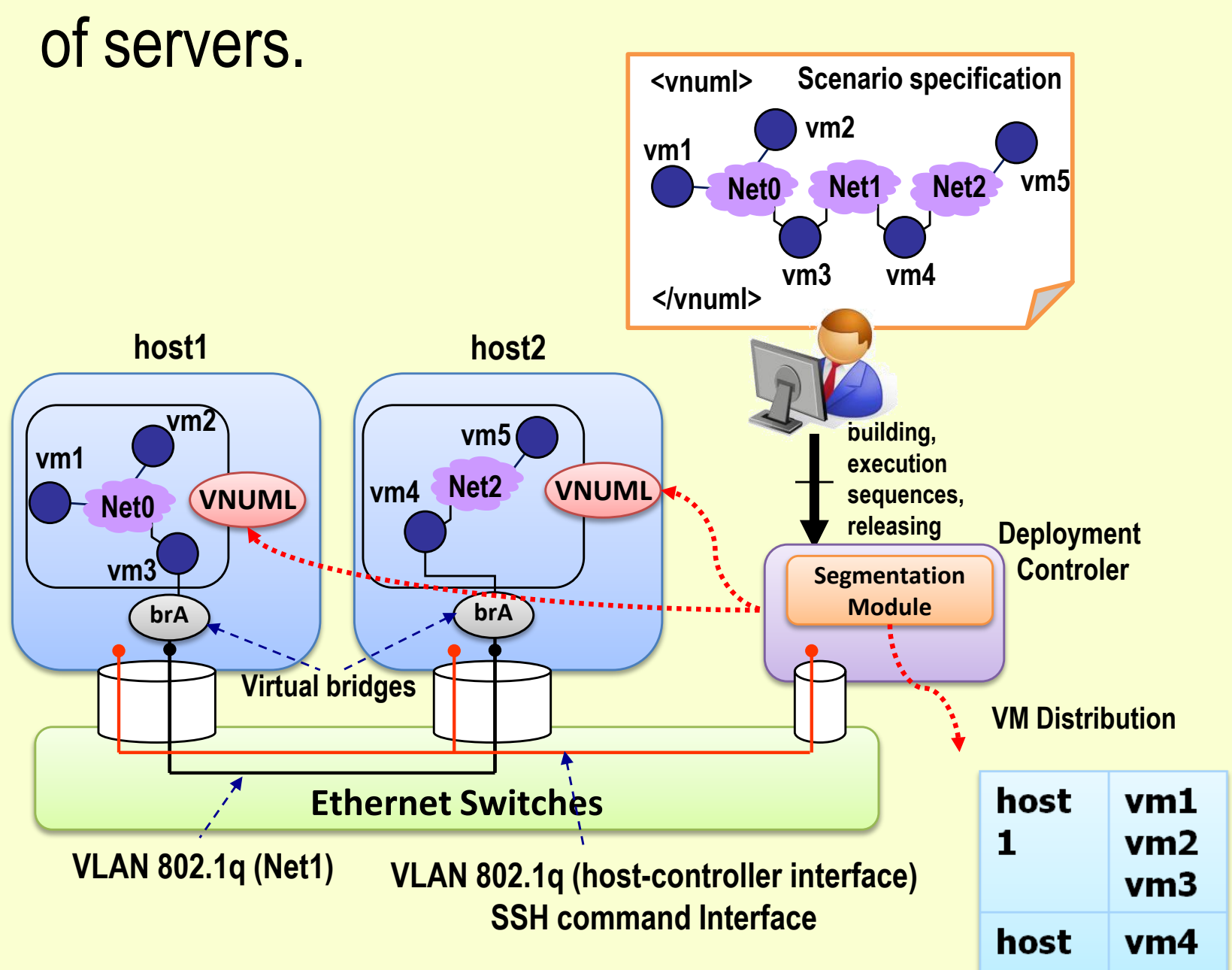
1. XML based scenario specification language
2. Language interpreter

Three basic operations:

- **Scenario building**: virtual machines and networks created following the scenario topology in the user specification.
- **Command execution**: users can directly interact with virtual machines or automatize the execution of commands.
- **Scenario releasing**: virtual machines and networks are released.

EDIV: VNUML Distributed Deployment

EDIV is a wrapper application to VNUML developed to allow the distributed deployment of virtual scenarios over clusters of servers.



Distributed Deployment Architecture

EDIV segments virtual scenarios into sub-scenarios deployed to the different servers, interconnecting them by means of VLANs.

EDIV includes basic segmentation algorithms (R, RR, WRR) as well as an API to allow the development of new ones.

Why VNX?

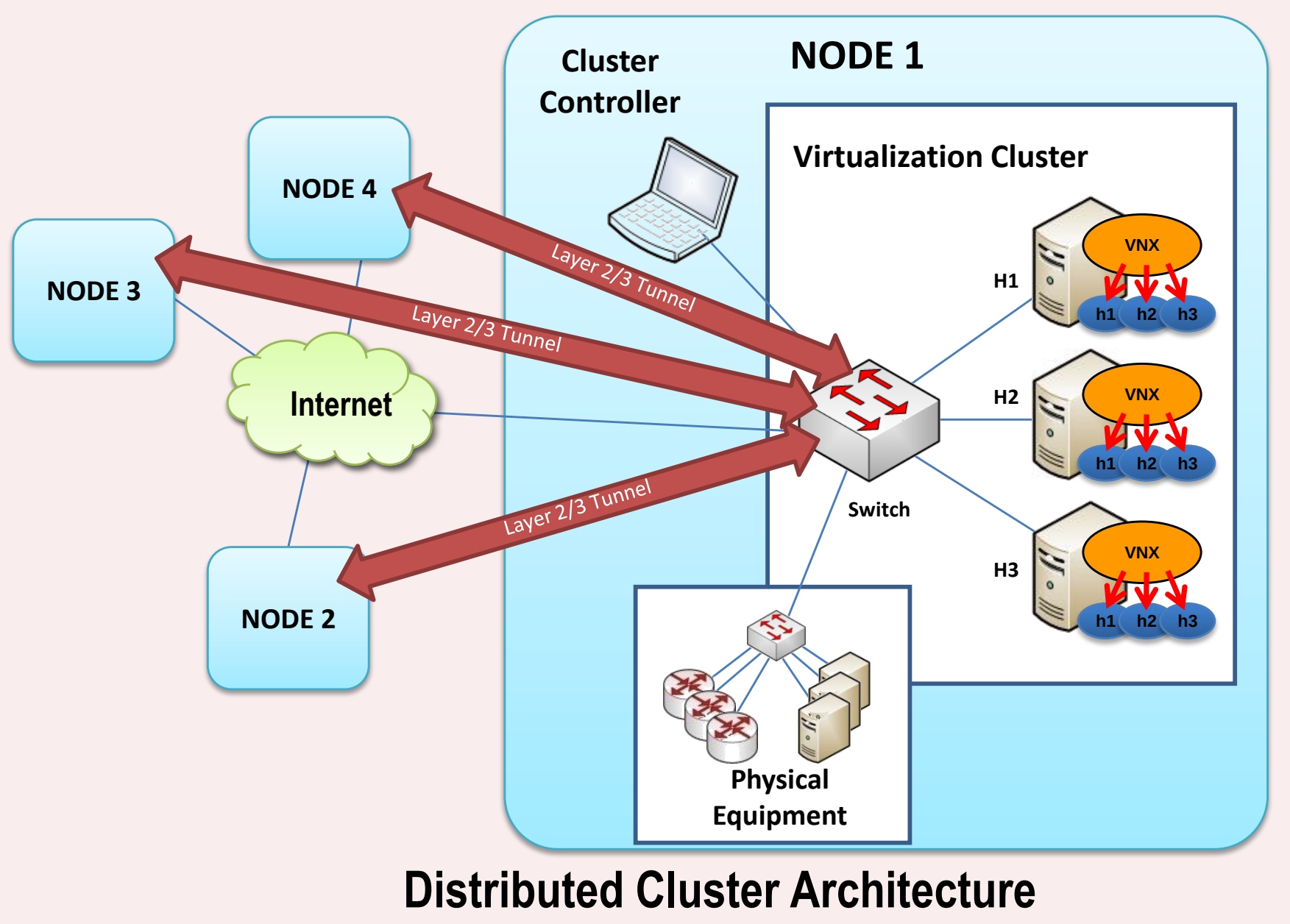
Limitations of VNUML and EDIV tools:

- Only Linux virtual machines (User Mode Linux limitation). Performance problems and need of additional plattforms and operating systems.
- Inability to manage virtual machines individually
- Limited autoconfiguration and command execution
- Distributed version (EDIV) limitations: manual network configuration for disperse clusters, lack of monitoring tools, etc

VNX project overall goal is the creation of a tool to allow the deployment of large virtual network scenarios over a federated cluster environment made of disperse nodes interconnected by means of layer 2/3 tunnels over Internet.

Each node is composed of:

- Virtualization servers running different types of hypervisors
- Physical (non-virtual) equipment to allow the creation of hybrid virtual scenarios

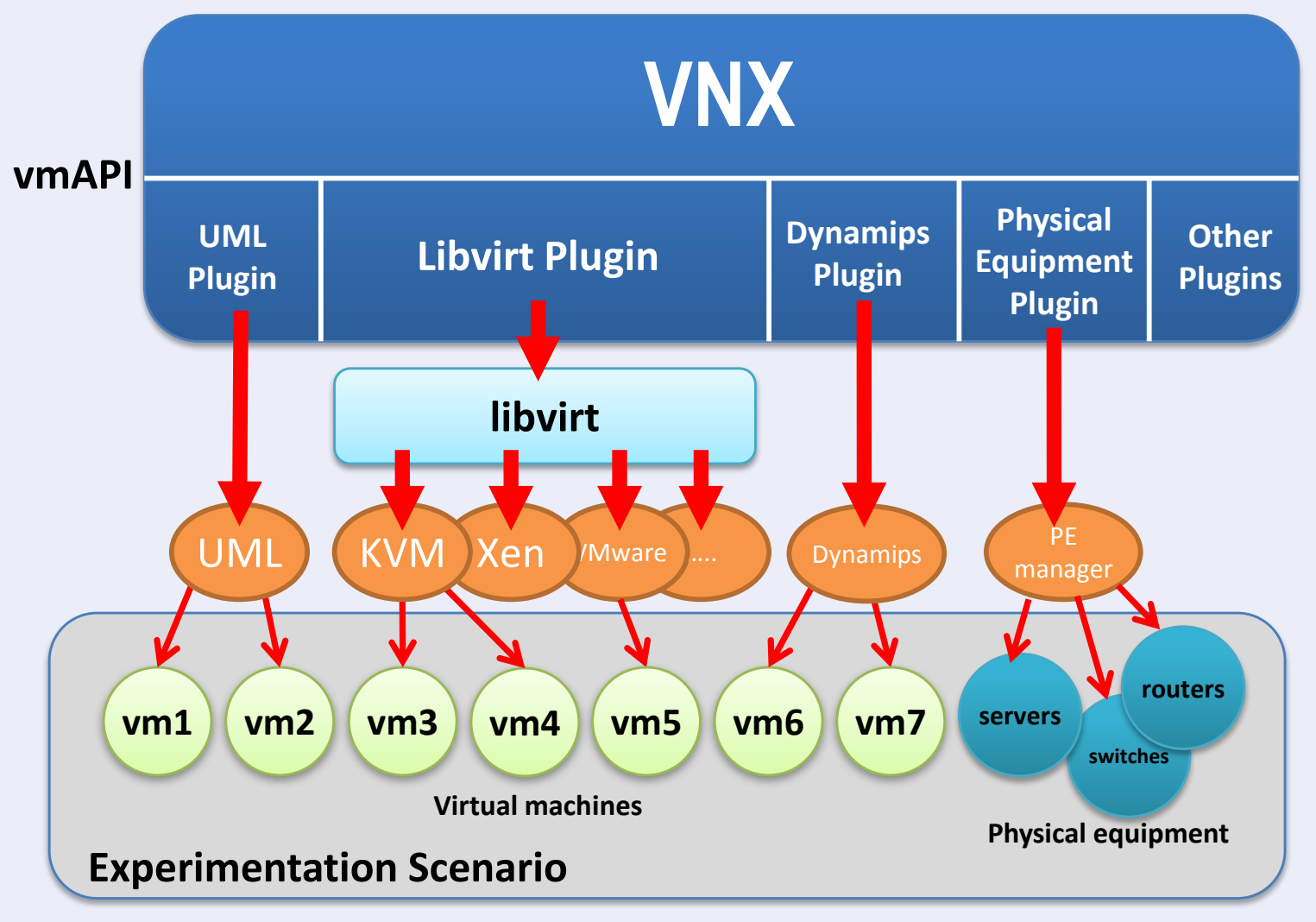


Distributed Cluster Architecture

VNX
Virtual Networks over linuX

VNUML redesign: VNX

VNX (Virtual Networks over LinuX) is a major rewrite of VNUML. A modular architecture based on a virtual machine control API has been defined to accommodate new virtualization platforms:



VNX Architecture

Plug-ins developed:

- **libvirt**, the Linux standard API for virtualization (libvirt.org), provides access to most of virtualization platforms supported in Linux (KVM, Xen, UML, etc)
- **Dynamips**: emulated CISCO routers
- **UML**: includes old VNUML code

The internal API is a simplified version of libvirt API, with the addition of a primitive to execute commands inside virtual machines.

Primitive	Description
defineVM	Defines a new virtual machine
undefineVM	Undefines an existent virtual machine
startVM	Starts a virtual machine
shutdownVM	Shutdowns a virtual machine in an ordered way.
destroyVM	Kills (switches off) a virtual machine
saveVM	Hibernates a virtual machine (saves state to disk)
restoreVM	Restores a virtual machine previously hibernated
suspendVM	Suspends a virtual machine (saves state to memory)
resumeVM	Resumes a previously suspended virtual machine
rebootVM	Reboots a virtual machine (=shutdown+define+start)
resetVM	Resets a virtual machine (=destroy+define+start)
executeCMD	Executes a command inside the virtual machine

VNX Internal API

New autoconfiguration and command execution mechanism created. Based on the OVF Environment approach: a dynamic cdrom offered to virtual machines with an XML file with autoconfiguration parameters and commands to execute.

VNX in Academic Education

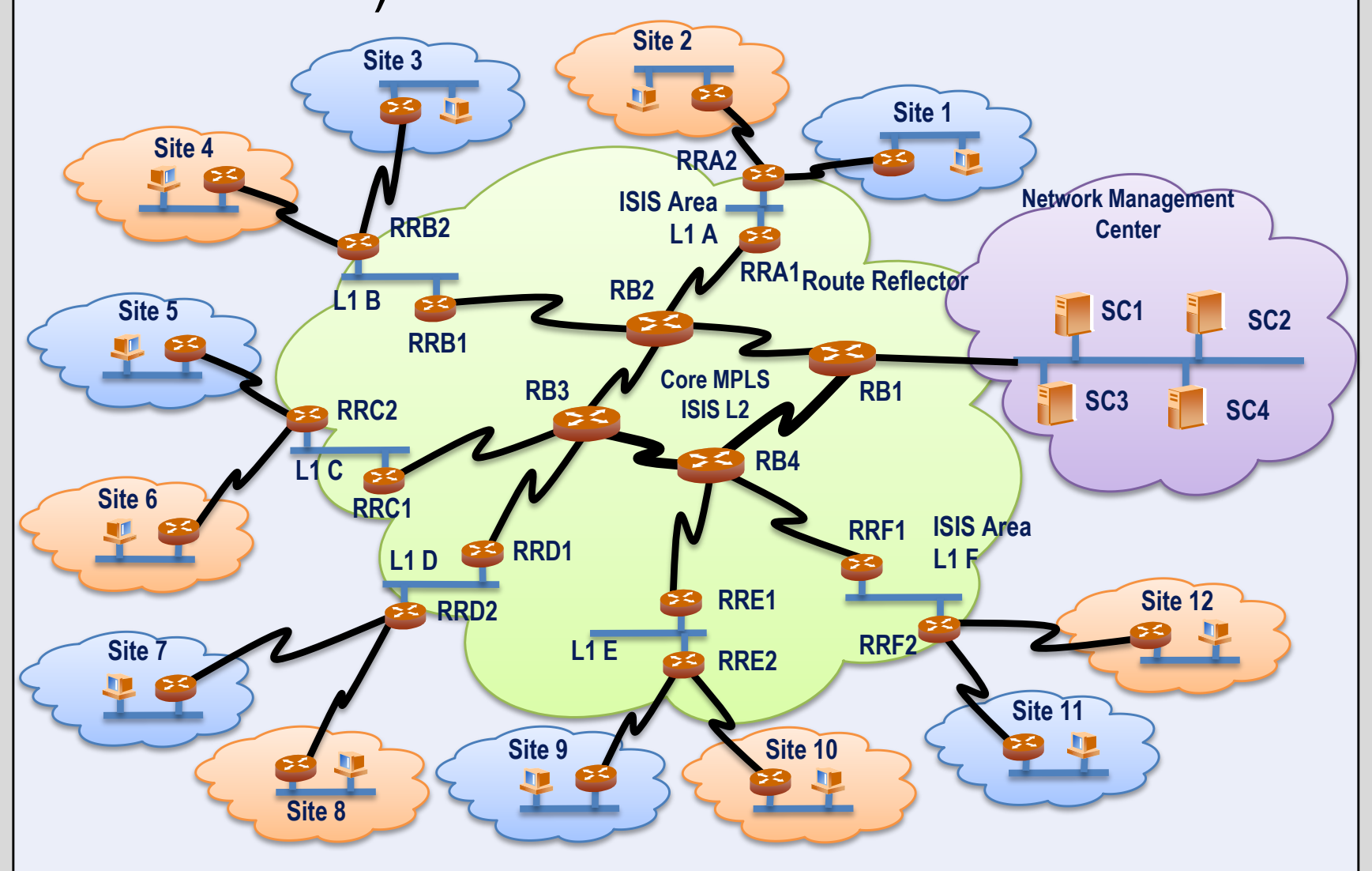
VNX is being extensively used in ETSIT-UPM computer network laboratories to create complex network scenarios to teach, for example, routing protocols like OSPF and BGP, IPv6 transition mechanisms, IPv6 mobility, security tools, etc.
For example, recently used in:

- A network security laboratory exercise to experiment with fwbuilder, nmap, nessus and Metasploitable Framework and Backtrack tools
- Designed to allow 14 student groups to simultaneously work on a common scenario to interact among them
- Based on a scenario composed by more than 65 linux virtual machines

Validation and Tests

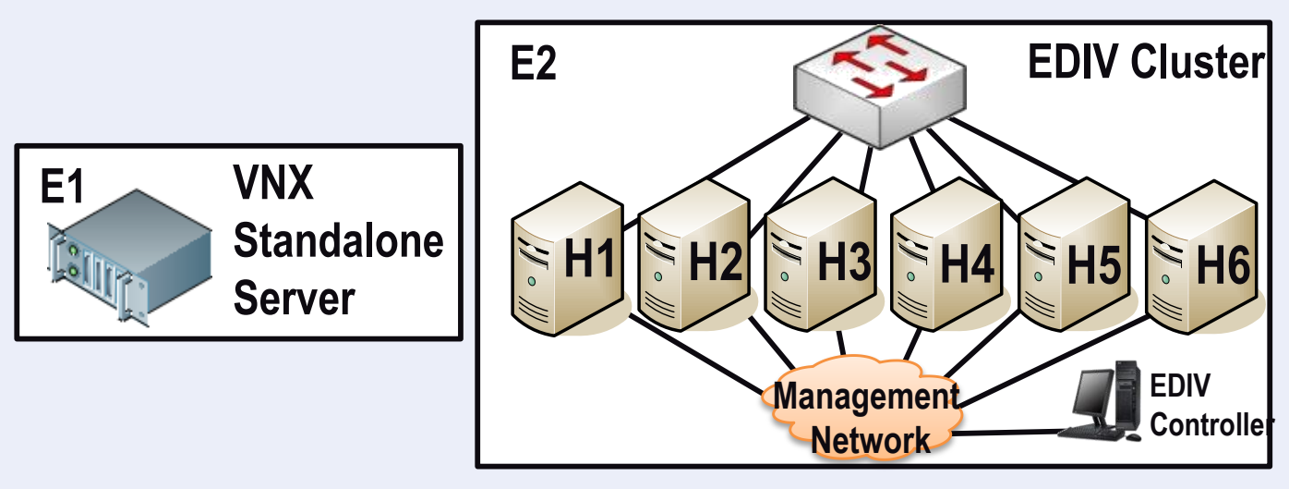
Networking laboratory for dynamic routing tests, resembling the topology of an ISP and involving 44 virtual devices as follows:

- 16 Cisco routers
- 6 Juniper routers
- 6 Linux/Quagga routers
- 12 end user computers
- 4 Servers (WinXP, FreeBSD, Ubuntu, Debian)



Example Validation Scenario

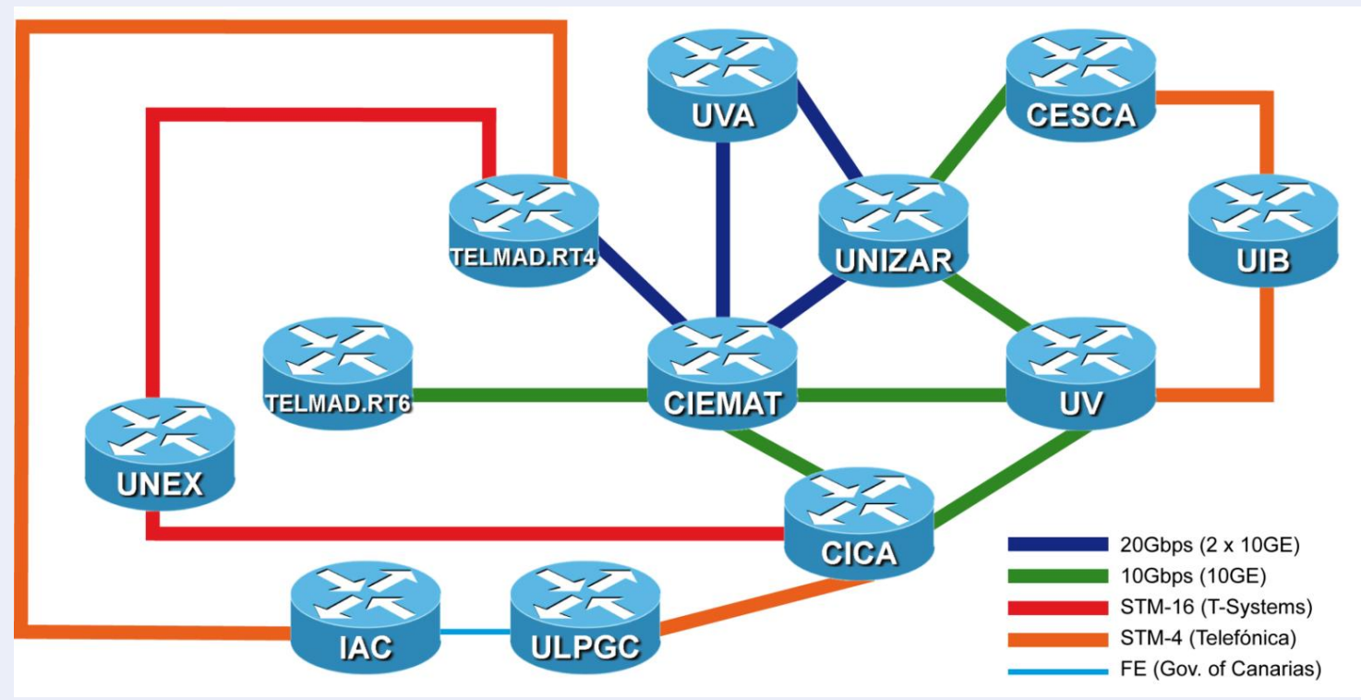
Deployed over two testing platforms:



Deployment scenarios

VNX Application: RedIRIS-NOVA

RedIRIS-NOVA is the high capacity photonic network for the Research and Education Comunity in Spain, that provides connectivity to international academic networks like the portuguese FCCN and the european reseach network GÉANT.



RedIRIS-NOVA IP infrastructure

VNX has been choosen to build a model of the RedIRIS-NOVA IP infrastructure (same topology, metrics and links) to be used to define, implement and test new network features.

Test case: building a route server reflector to provide BGP DDOS mitigation without accessing the router server.

Scenario deployed over a SunFire X4150 server (8Gb RAM, 2 CPU with 4 cores) in the infrastructure of PASITO project.

- 39 network links
- 17 virtual machines (15 JunOS Olive routers, core + route server + reflector, and 2 Linux servers for DNS and mitigation)
- IS-IS & iBGP convergence
- Time to boot the simulation: ~ 7 minutes

VNX implementation

Current version of VNX available:

- **libvirt** support. Tested with Linux (Ubuntu, Fedora, CentOS), FreeBSD and Windows (XP and 7)
- **Dynamips and Olive** router emulation support
- Virtual machine individual management (start, stop, restart, reboot, suspend, etc)
- OVF-Environment-like autoconfiguration and command execution support
- Plug-in architecture to add extensions to VNX
- Distributed deployment support (EDIV)
- Library of root filesystems available: Ubuntu, Fedora, CentOS, FreeBSD, etc

VNX is mostly written in Perl (around 25000 lines of code); Windows autoconf daemon writen in C++. Around 40% of VNUML code reused with minor modifications.

Dynamic honeynets deployment

In security research projects RECLAMO and Segur@, as part of the development of Automated Intrusion Response Systems (AIRS), VNX is being used as a tool to dynamically deploy virtual honeynets.

Conclusions and Future Work

Development continues, mainly focused on improving VNX functionalities, its robustness and completing the distributed version capabilities.

Future work includes:

- Complete and improve distributed support
- Dynamic scenarios (adding/deleting VMs and networks, machine migration, etc)
- Graphical user interface
- New virtual machine types (e.g. Android)
- Plug-in to control physical equipment
- Better network emulation capabilities

Acknowledgment

This research was partially supported by Ministry of Science and Innovation of the Spanish Government under the RECLAMO project (*Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management)* with code TIN2011-28287-C02-01, as well as S21Sec company and the Centre for the Development of Industrial Technology (CDTI) as part of the SEGUR@ project (https://www.cenitsegura.es/), within the CENIT program, with reference CENIT-2007 2004.

Contact information

Coordinator:
David Fernández
Dpto. Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Avda. Complutense, 30
28040 MADRID - SPAIN
E-mail: david@dit.upm.es
www: http://www.dit.upm.es

http://www.dit.upm.es/vnx