Distributed Virtual Scenarios over Multi-host Linux Environments

David Fernández, Alejandro Cordero, Jorge Somavilla, Jorge Rodríguez
Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Madrid, Spain
david@dit.upm.es

Aitor Corchero, Luis Tarrafeta
S21Sec
Pamplona, Spain
ltarrafeta@s21sec.com

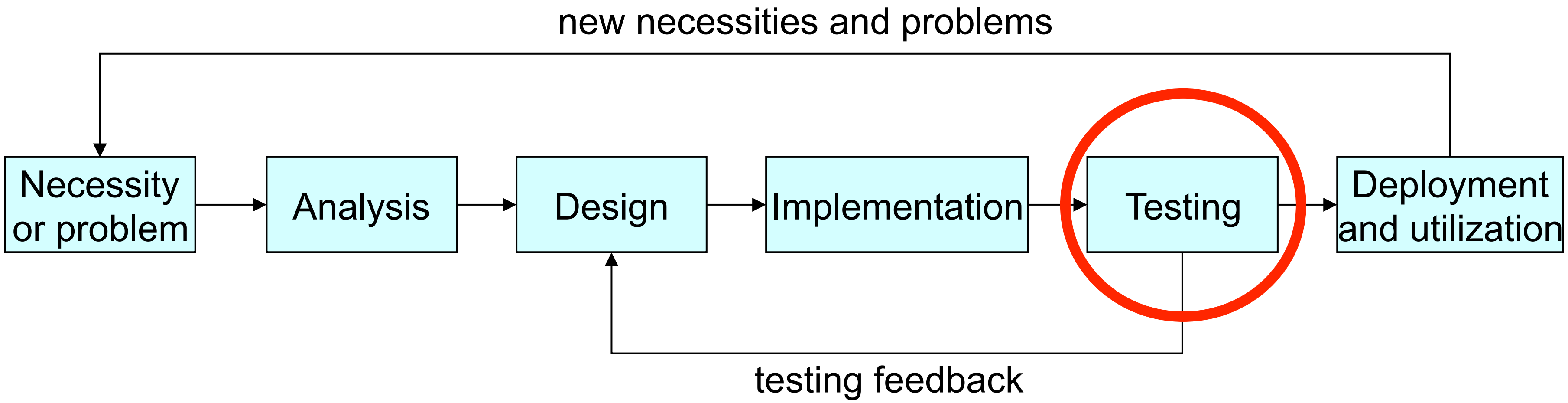
Fermín Galán
Telefónica I+D
Madrid, Spain
fermin@tid.es

Contents

- Motivation
- Previous work:
  - Virtual Networks User Mode Linux (VNUML)
  - Distributed deployment: EDIV
  - VNUML/EDIV Limitations
- Virtual Networks over LinuX (VNX)
  - Goals and architecture
  - Implementation
  - Validation and tests
- Conclusions and future work

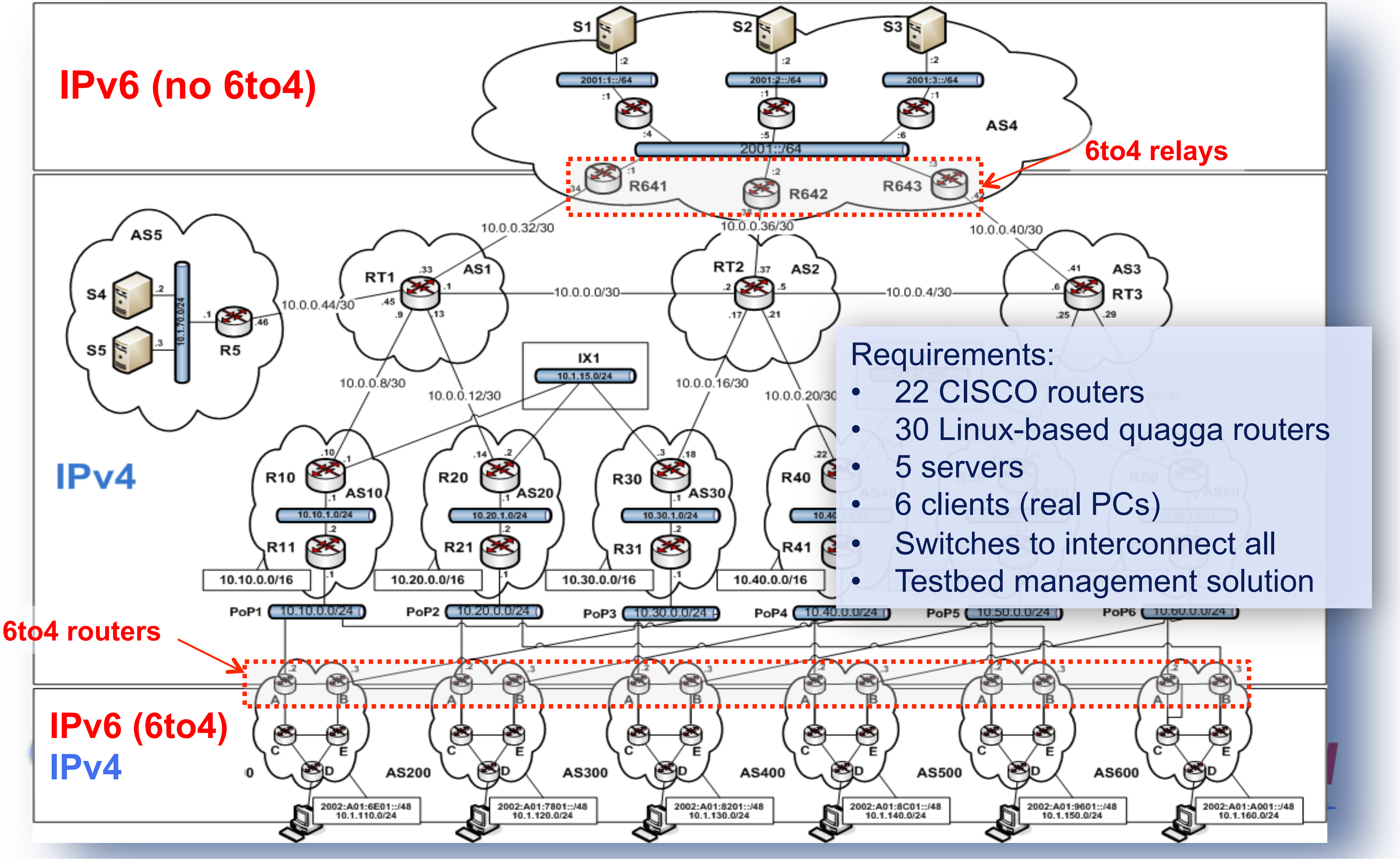
Context

- Systems development life cycle:



- Network and services testbeds (informal definition):
  - infrastructure platform used to experiment with networking systems and technologies under controlled conditions that often resemble those found in production networks

Testbed example: 6to4 network laboratory

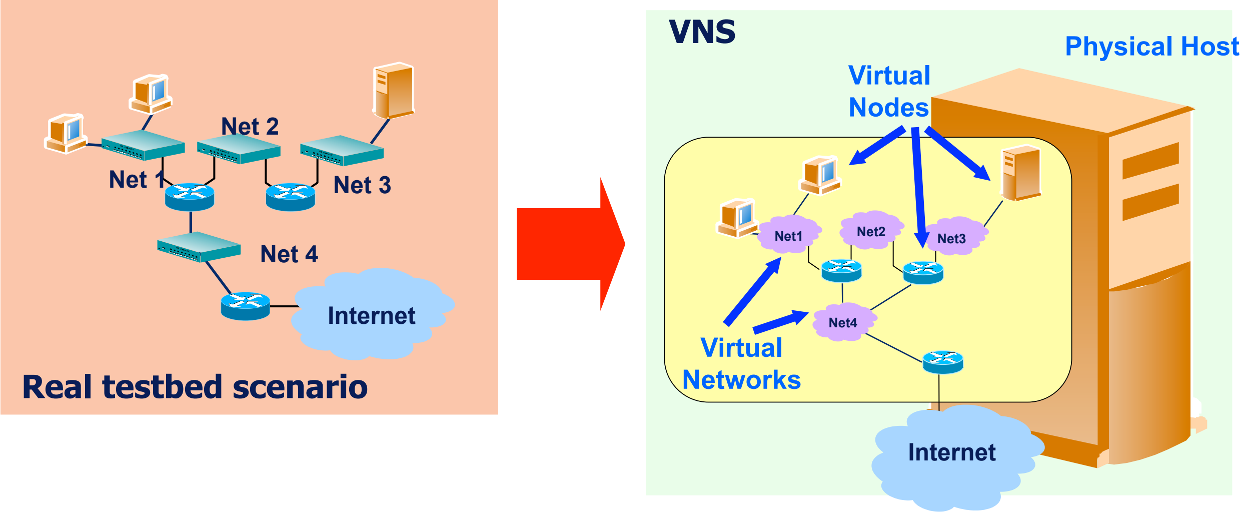


Requirements:

- 22 CISCO routers
- 30 Linux-based quagga routers
- 5 servers
- 6 clients (real PCs)
- Switches to interconnect all
- Testbed management solution

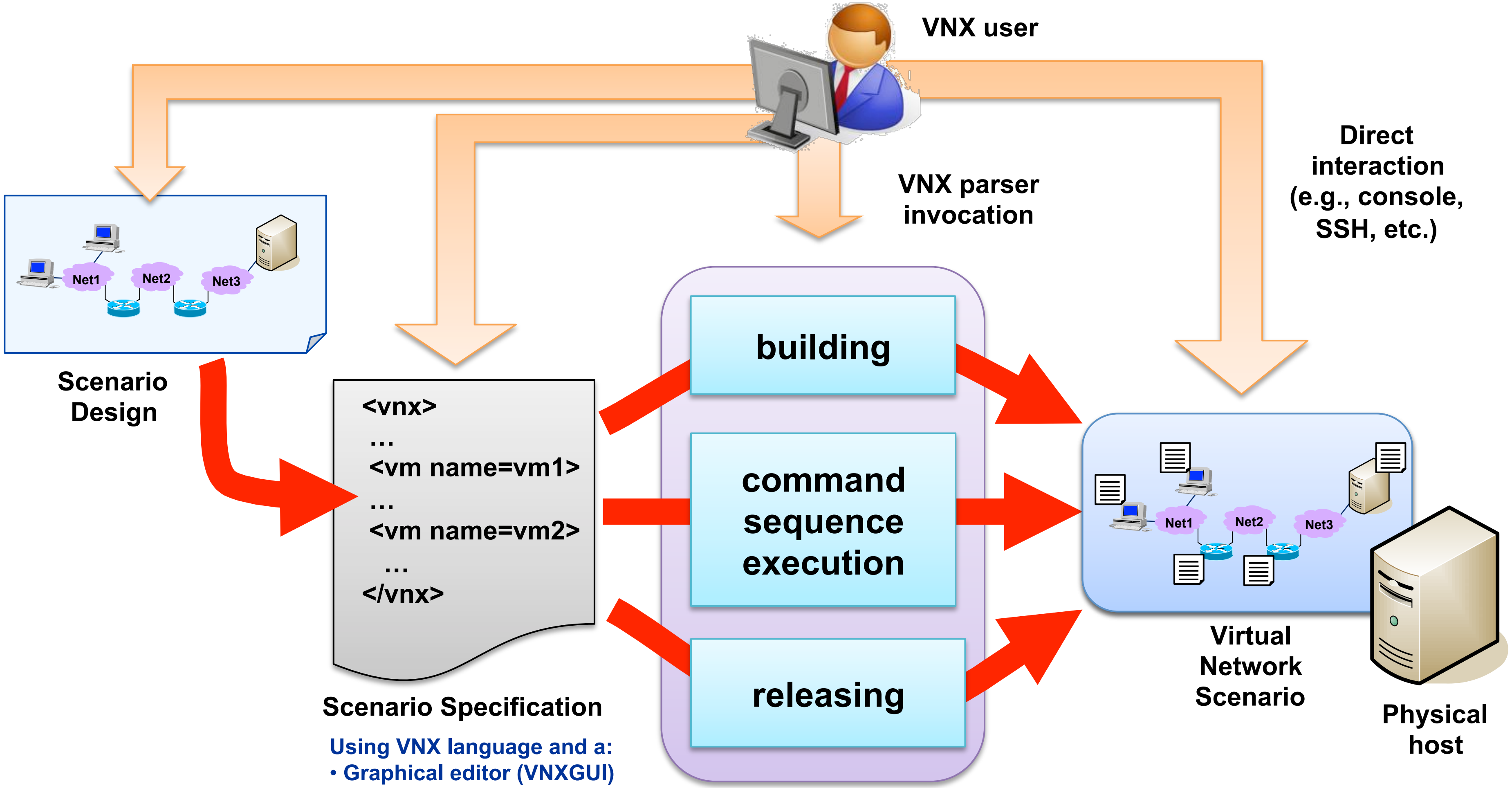
Virtualization in testbeds

- Virtualization techniques allow to execute multiple virtual machines over a physical host (Ex.: KVM, Xen, VMware, User Mode Linux, etc.)
- Combined with the use of virtual emulated networks allow the creation of **Virtual Network Scenarios (VNS)**
  - Following a user-defined topology
  - Possibly including connection to external equipment and networks



- Several tools available to manage VNS:
  - GNS3, Netkit, MNL, Marionnet, VNUML, etc.

VNUML Operation Workflow



VNX user

VNX parser invocation

Direct interaction (e.g., console, SSH, etc.)

Scenario Design

```
<vnx>
 …
 <vm name=vm1>
 …
 <vm name=vm2>
  …
</vnx>
```

building

command sequence execution

releasing

Virtual Network Scenario

Physical host

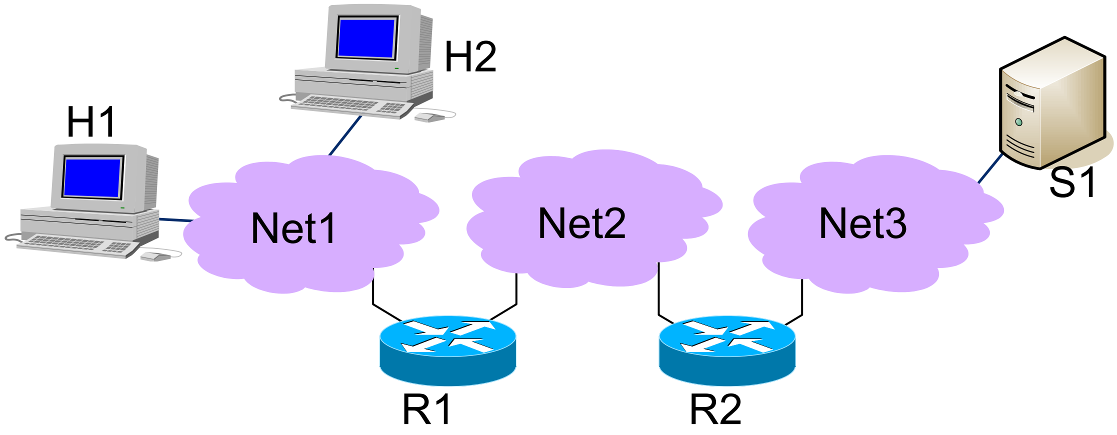
Scenario Specification

Using VNX language and a:
- Graphical editor (VNXGUI)
- XML editor

Scenario Management

VNX processes scenario specification and creates/releases it over the hosts and helps on command execution

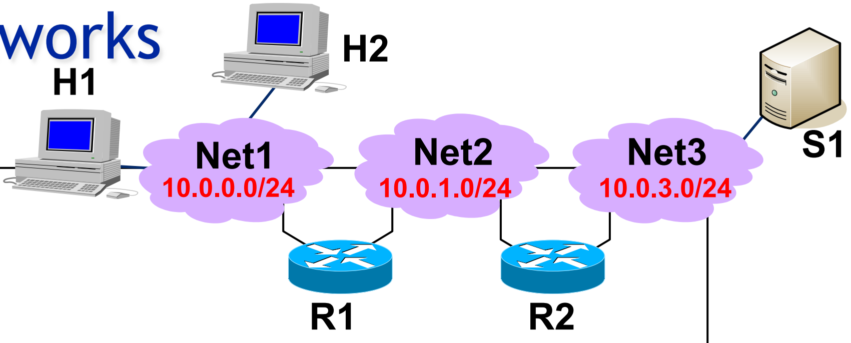
VNS Specification Language (I)



```
<?xml version="1.0" encoding="UTF-8"?>
<vnx>
      (global definitions: <global>)
      (virtual networks definitions: <net>)
      (virtual machine definitions: <vm>)
</vnx>
```

VNS Specification Language (II)

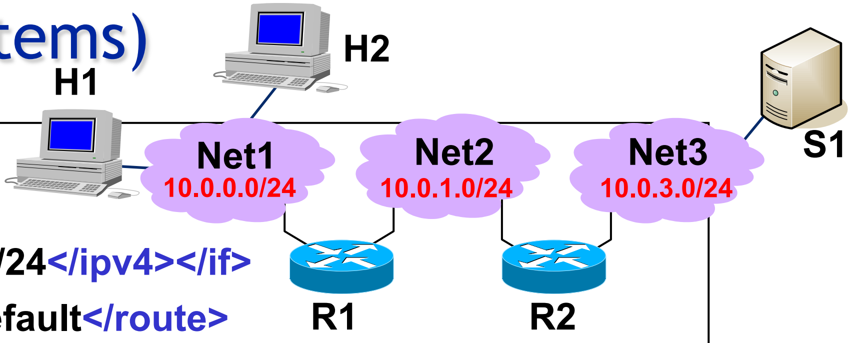
- Global definitions and networks



```
<vnx>
     <global>
              <simulation_name>cinco_vms</simulation_name>
              <vm_defaults>
                          …
             </vum_defaults>
     </global>

     <net name="Net1" mode="virtual_bridge" />
     <net name="Net2" mode="virtual_bridge" />
     <net name="Net3" mode="virtual_bridge" />
     …
```

VNS Specification Language (III)

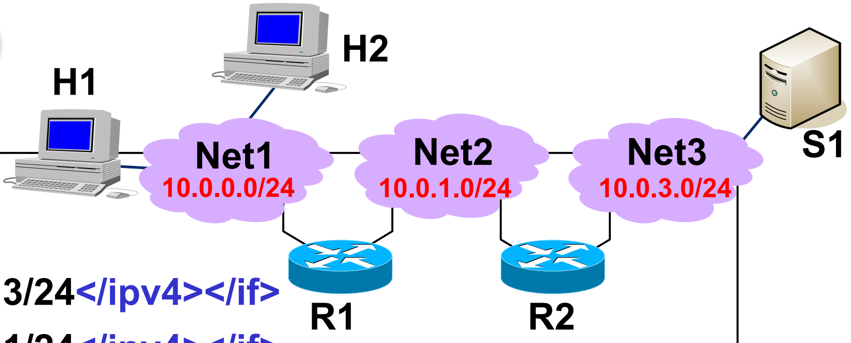
- Virtual machines (end-systems)



```
…
<vm name="H1">
      <if id="1" net="Net1"><ipv4>10.0.0.1/24</ipv4></if>
      <route type="ipv4" gw="10.0.0.3">default</route>
</vm>
<vm name="H2">
      <if id="1" net="Net2"><ipv4>10.0.0.2/24</ipv4></if>
      <route type="ipv4" gw="10.0.0.3">default</route>
</vm>
<vm name="S1">
      <if id="1" net="Net3"><ipv4>10.0.2.2/24</ipv4></if>
      <route type="ipv4" gw="10.0.2.1">default</route>
</vm>
…
```

VNS Specification Language (IV)

- Virtual machines (routers)

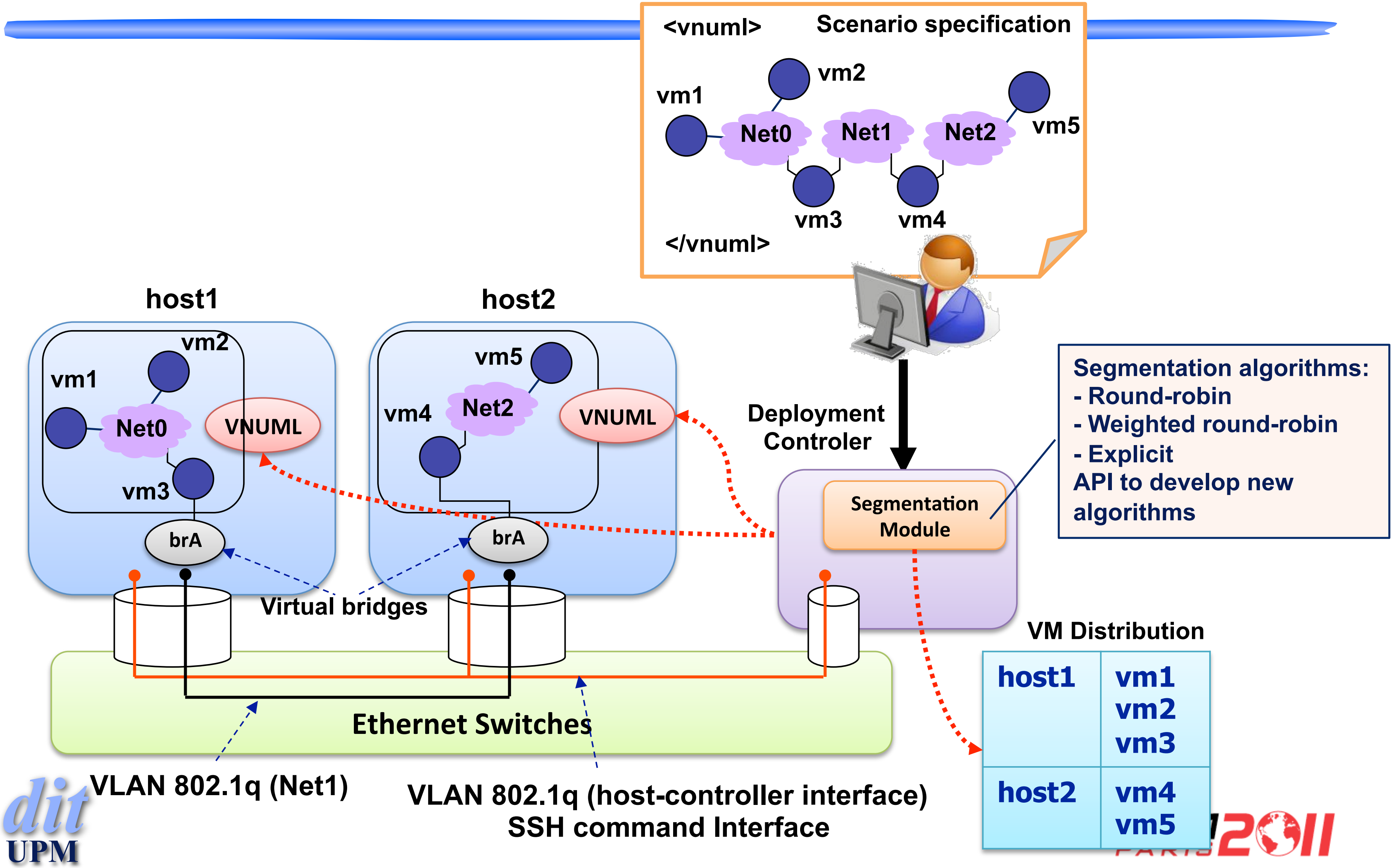


```
…
<vm name=“R1”>
      <if id=“1” net=“Net1”><ipv4>10.0.0.3/24</ipv4></if>
      <if id=“2” net=“Net2”><ipv4>10.0.1.1/24</ipv4></if>
      <route type=“ipv4” gw=“10.0.1.2”>10.0.2.0/24</route>
      <forwarding type=“ip” />
</vm>
<vm name=“R2”>
      <if id=“1” net=“Net2”><ipv4>10.0.1.2/24</ipv4></if>
      <if id=“2” net=“Net3”><ipv4>10.0.2.1/24</ipv4></if>
      <route type=“ipv4” gw=“10.0.1.1”>default</route>
      <forwarding type="ip" />
</vm>
…
```

Scalability

- Complexity of virtual scenarios supported dependant on:
  - host resources (cpu, memory, disk, etc) available
  - VMs resources demanded
- Need to distribute virtual machines over multiple hosts to deploy bigger scenarios:
  - Clusters of virtualization servers
  - Need for mechanisms to interconnect virtual machines in different hosts
  - Requirements: transparency, efficiency, etc.

EDIV: Distributed Architecture



Segmentation algorithms:
- Round-robin
- Weighted round-robin
- Explicit

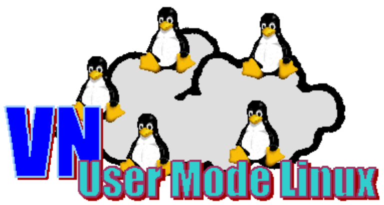
API to develop new algorithms

VM Distribution

host1	vm1 vm2 vm3
host2	vm4 vm5

VNUML/EDIV Limitations

- Limitations of VNUML and EDIV tools:
  - Only Linux virtual machines (User Mode Linux limitation)
  - Performance problems
  - Inability to manage virtual machines individually
  - Autoconfiguration and command execution limited
  - Distributed version (EDIV) limitations: manual network configuration for disperse clusters, lack of monitoring tools, etc

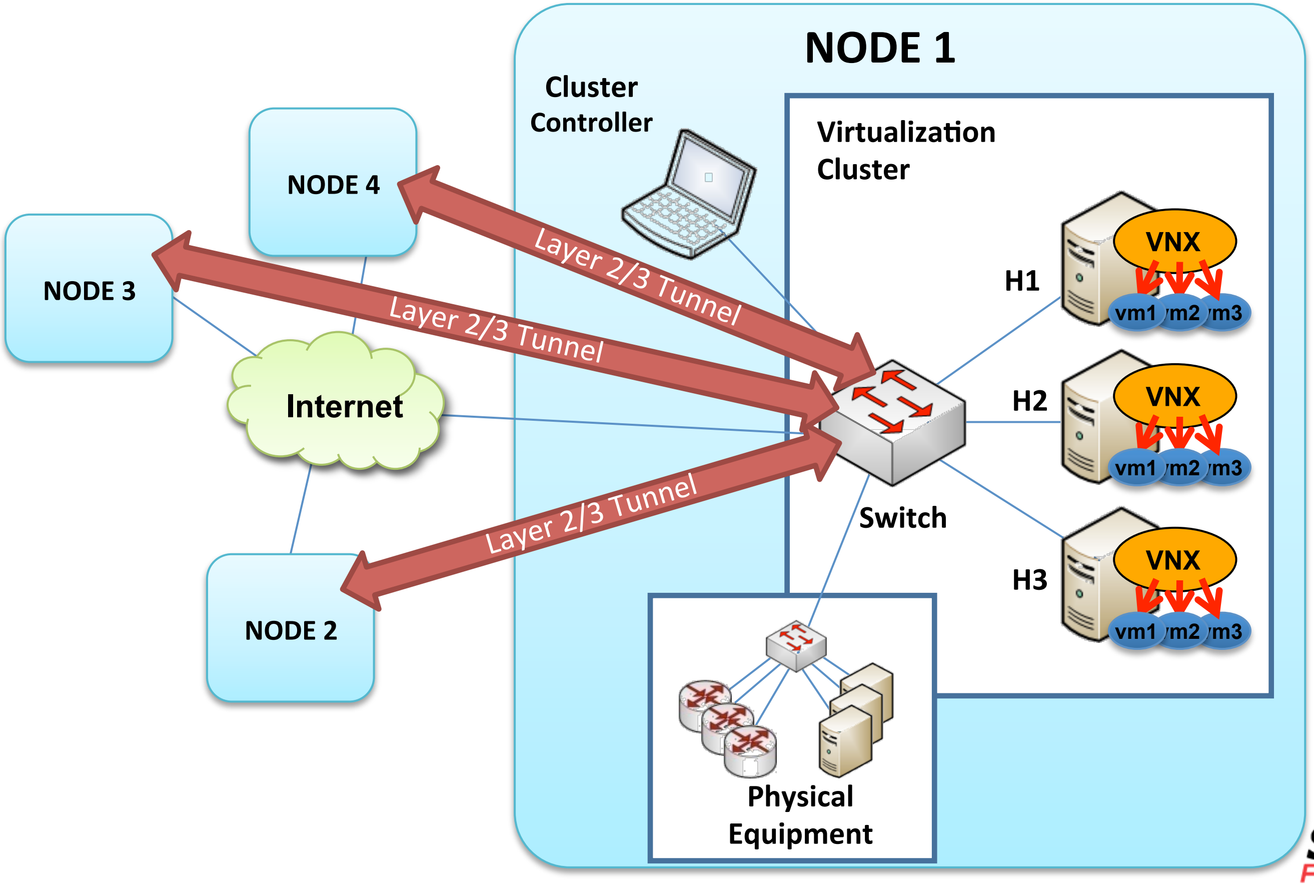


http://www.dit.upm.es/vnuml

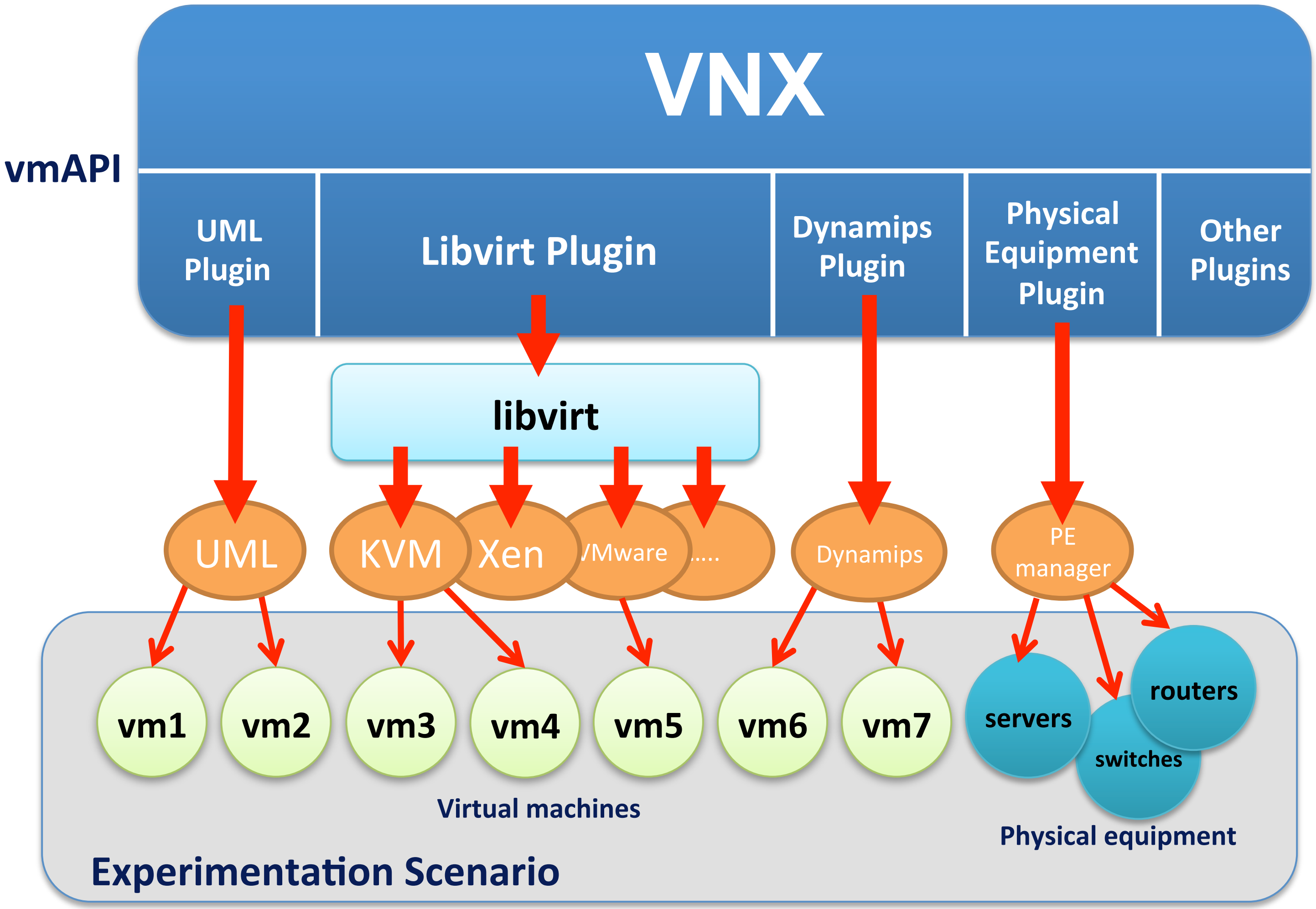
- All these limitations led us redesign and rewrite VNUM create:

VNX Objective

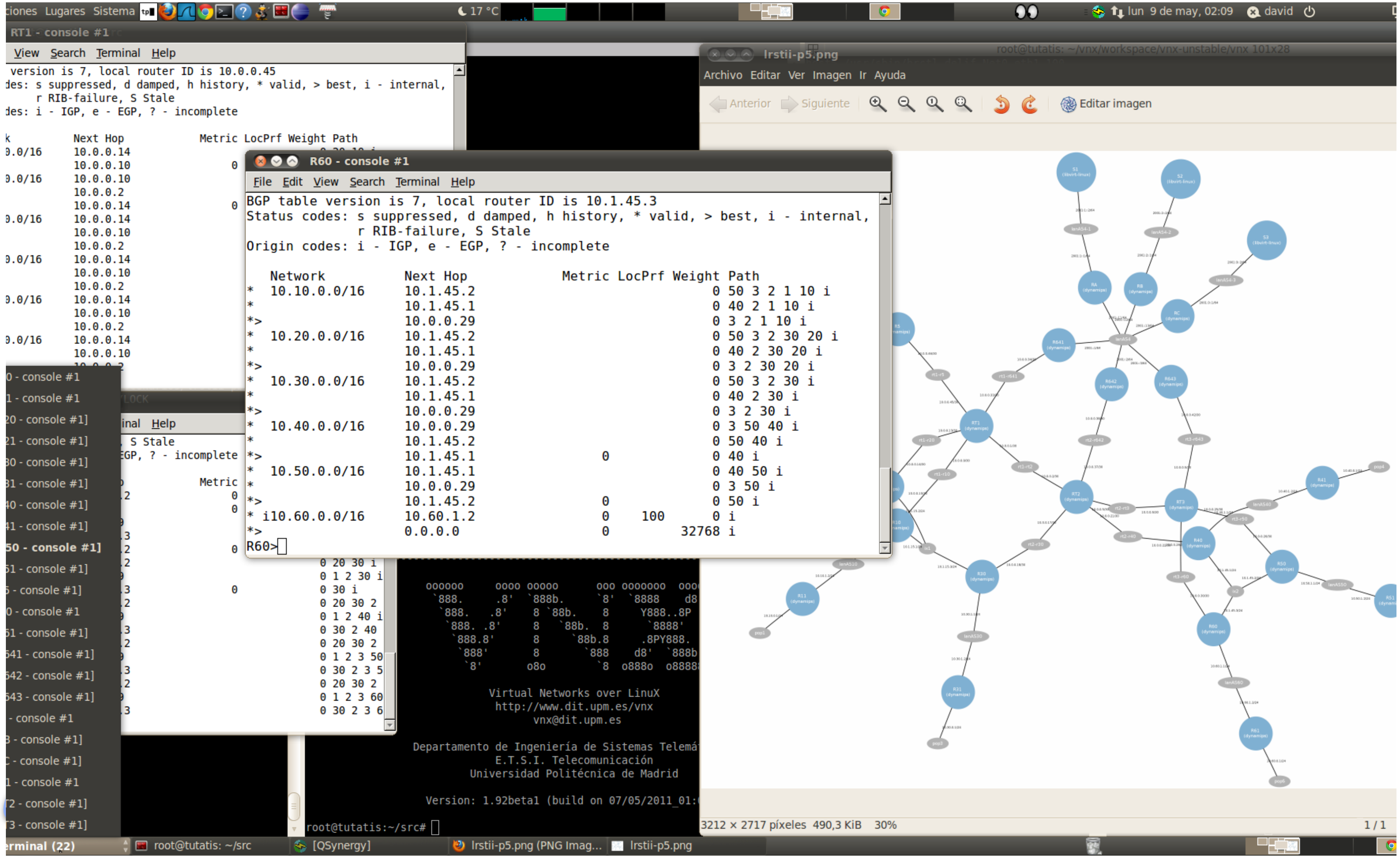
- Deployment of large VNS over distributed clusters made of virtualization hosts and physical equipment



VNX Internal Architecture



Example scenario screenshot: 6to4



```
 version is 7, local router ID is 10.0.0.45
des: s suppressed, d damped, h history, * valid, > best, i - internal,
     r RIB-failure, S Stale
des: i - IGP, e - EGP, ? - incomplete

k           Next Hop            Metric LocPrf Weight Path
0.0/16      10.0.0.14
            10.0.0.10                0
0.0/16      10.0.0.10
            10.0.0.2
            10.0.0.14                0
0.0/16      10.0.0.14
            10.0.0.10
            10.0.0.2
0.0/16      10.0.0.14
            10.0.0.10
            10.0.0.2
0.0/16      10.0.0.14
            10.0.0.10
            10.0.0.2
0.0/16      10.0.0.14
            10.0.0.10
```

```
BGP table version is 7, local router ID is 10.1.45.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  10.10.0.0/16     10.1.45.2                              0 50 3 2 1 10 i
*                   10.1.45.1                              0 40 2 1 10 i
*>                  10.0.0.29                              0 3 2 1 10 i
*  10.20.0.0/16     10.1.45.2                              0 50 3 2 30 20 i
*                   10.1.45.1                              0 40 2 30 20 i
*>                  10.0.0.29                              0 3 2 30 20 i
*  10.30.0.0/16     10.1.45.2                              0 50 3 2 30 i
*                   10.1.45.1                              0 40 2 30 i
*>                  10.0.0.29                              0 3 2 30 i
*  10.40.0.0/16     10.0.0.29                              0 3 50 40 i
*                   10.1.45.2                              0 50 40 i
*>                  10.1.45.1                0             0 40 i
*  10.50.0.0/16     10.1.45.1                              0 40 50 i
*                   10.0.0.29                              0 3 50 i
*>                  10.1.45.2                0             0 50 i
* i10.60.0.0/16     10.60.1.2                0    100      0 i
*>                  0.0.0.0                  0         32768 i
R60>
```

```
 0 20 30 i
 0 1 2 30 i
 0 30 i
 0 20 30 2
 0 1 2 40 i
 0 30 2 40
 0 20 30 2
 0 1 2 3 50
 0 30 2 3 5
 0 20 30 2
 0 1 2 3 60
 0 30 2 3 6
```

```
 oooooo     oooo ooooo      ooo ooooooo  oooo
  `888.     .8'  `888b.     `8'  `8888    d8
   `888.   .8'    8 `88b.    8     Y888..8P
    `888. .8'     8   `88b.  8      `8888'
     `888.8'      8     `88b.8     .8PY888.
      `888'       8       `888    d8'  `888b
       `8'       o8o        `8  o888o  o8888

           Virtual Networks over LinuX
           http://www.dit.upm.es/vnx
                 vnx@dit.upm.es

   Departamento de Ingeniería de Sistemas Telemá
             E.T.S.I. Telecomunicación
         Universidad Politécnica de Madrid

    Version: 1.92beta1 (build on 07/05/2011_01:

root@tutatis:~/src#
```

Implementation Details

- **VNX** is an open source tool (GPL) for the management of VNS based on Virtual Networks User Mode Linux (VNUML)
- First beta version available at http://www.dit.upm.es/vnx with:
  - **libvirt** support, tested with Linux (Ubuntu, Fedora, CentOS), FreeBSD and Windows (XP and 7).
  - **Dynamips and Olive** router emulation support
  - Individual management of virtual machines
  - General OVF-Environment-like autoconfiguration and command execution mechanism for Windows, Linux and FreeBSD
  - Plug-in architecture to allow extensions to VNX
  - Improved distributed deployment support (EDIV)
  - Library of root filesystems available
- VNX written in Perl (around 25000 lines of code); Windows autoconf daemon in C++.
  - ~40% of VNUML code reused with minor modifications

Autoconfiguration and Command Execution

- Based on OVF Environment approach:
    - A dynamically created CDROM is offered to virtual machines with:
        - Initial configuration values
        - Commands to execute and files to copy
- Virtual machines run an Autoconfiguration and Command Execution Daemon (ACED) that:
    - Waits for CDROMs and,
    - Read XML files and process them
- ACED include auto-update functionality

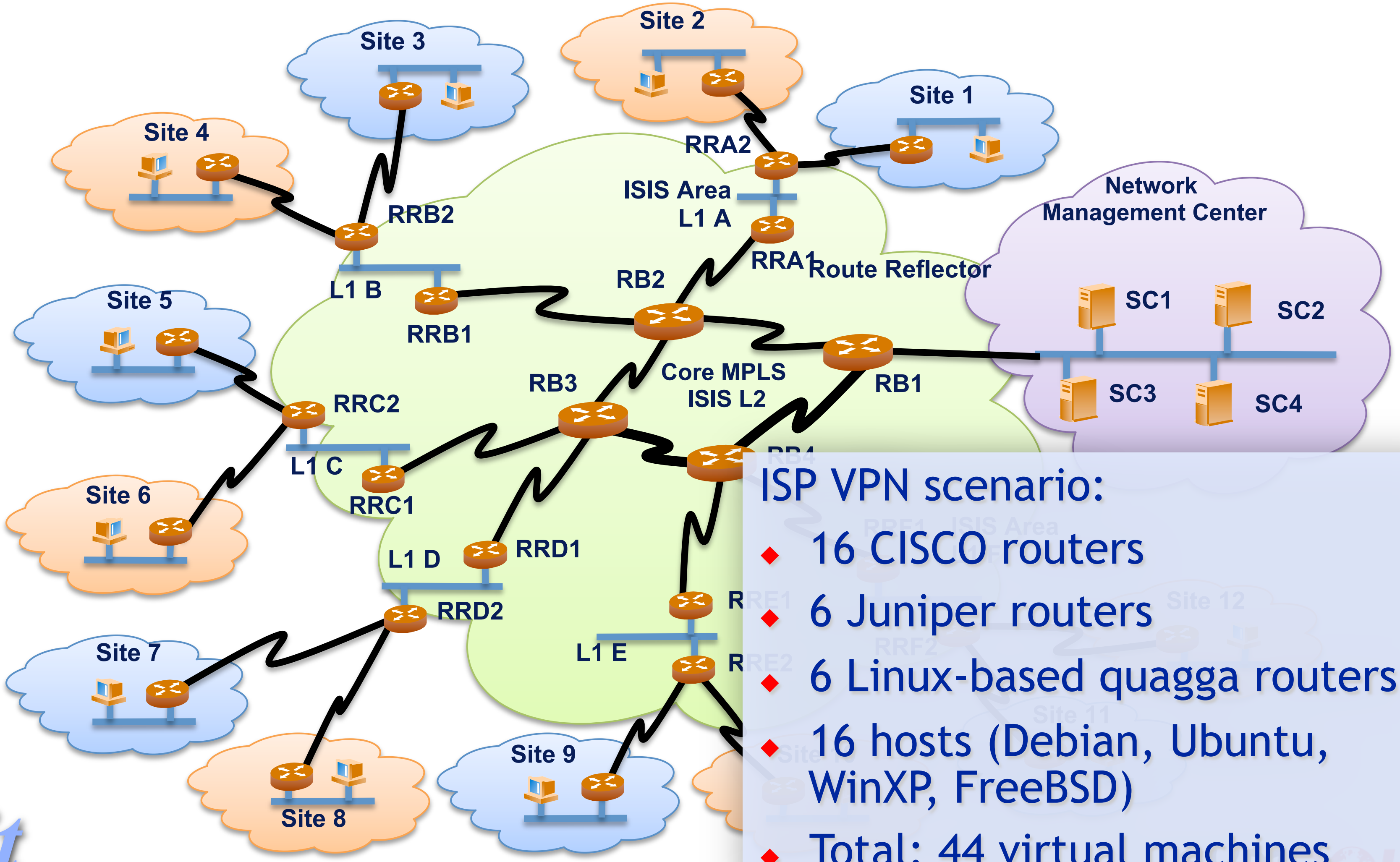
```
<?xml version="1.0" encoding="UTF-8"?>
<create_conf>
 <vm name=”vm4” >
   <filesystem type="cow”>rootfs_ubuntu</filesystem>
   <mem>128M</mem>
   <if id="1" net="Net1" mac=",fe:fd:00:00:04:01">
     <ipv4 mask="255.255.255.0">10.0.1.2</ipv4>
    </if>
   <if id="2" net="Net2" mac=",fe:fd:00:00:04:02">
     <ipv4 mask="255.255.255.0">10.0.2.1</ipv4>
   </if>
   <route type="ipv4" gw="10.0.1.1">default</route>
   <forwarding type="ip"/>
 </vm>
</create_conf>
```

```
<command>
 <id>ubuntu-fYH3pA</id>
 <filetree seq="start-www" root="/var/www/"
      user="www-data" group="www-data"
      perms="644">conf/txtfile</filetree>
 <exec seq="start-www" type="verbatim"
      ostype="system">service apache2 start</exec>
</command>
```

Autoconfiguration and Command Execution (II)

- Important effort dedicated to virtual machine autoconfiguration and command execution
- OVF-Environment like CDROM based autoconfiguration is a general approach, however:
  - It is slow
  - Interferes with CDROM OS autoexecute mechanisms
  - Differences among operating systems and releases make the development of ACED costly
    - Tune and test for every OS and release
  - Unidirectional: no feedback from the virtual machine
- Alternative mechanisms implemented based on a shared filesystems and a serial line:
  - Files copied to shared filesystem
  - Simple signalling protocol over the serial line
  - Implemented for Olive routers; being extended to other VMs

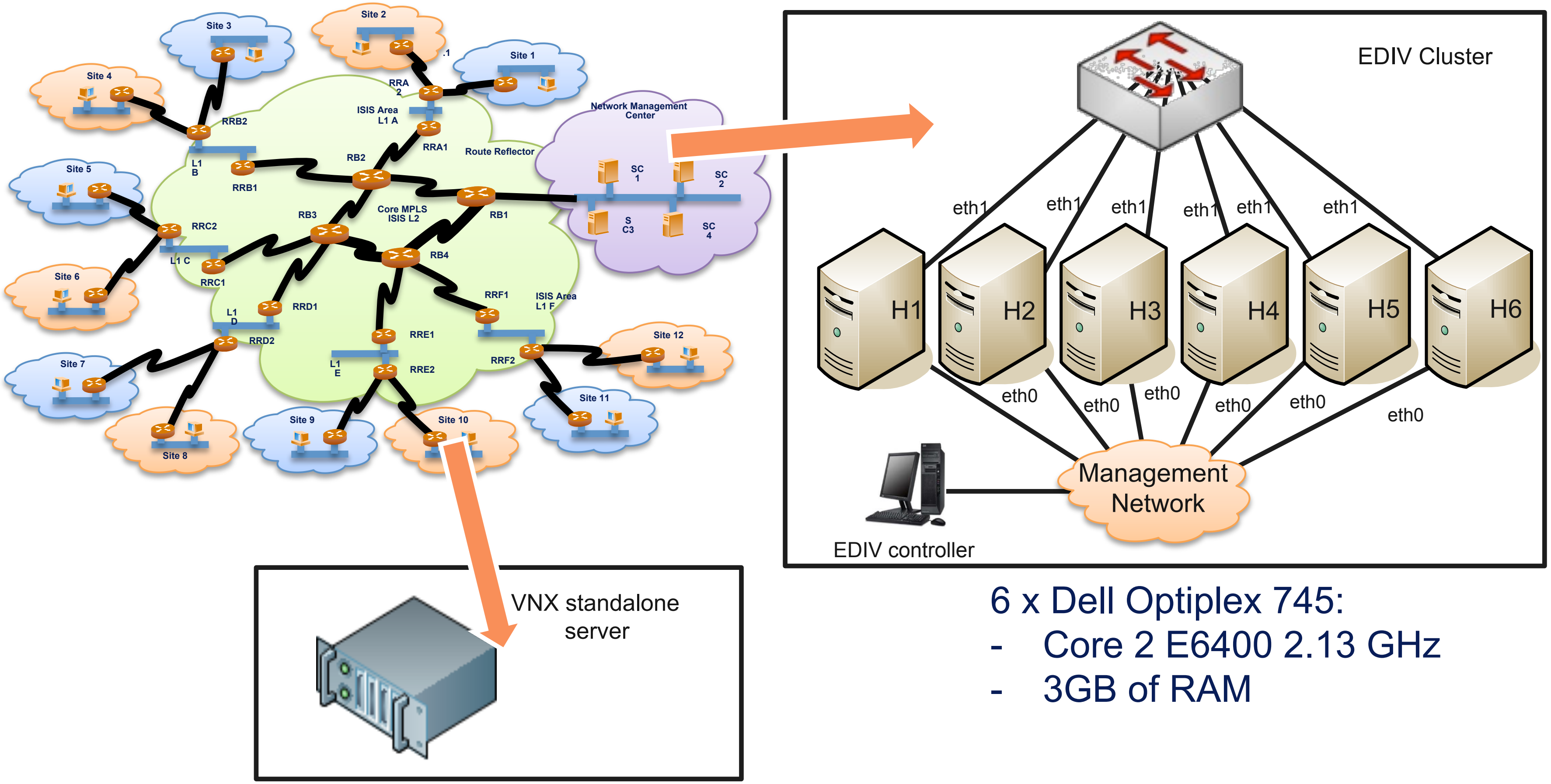
EDIV Validation Scenario: MPLS VPN



ISP VPN scenario:

- 16 CISCO routers
- 6 Juniper routers
- 6 Linux-based quagga routers
- 16 hosts (Debian, Ubuntu, WinXP, FreeBSD)
- Total: 44 virtual machines

EDIV Validation Scenario: MPLS VPN



6 x Dell Optiplex 745:
- Core 2 E6400 2.13 GHz
- 3GB of RAM

Sun Fire X4150:
- dual Xeon E5440 (2.83GHz)
- 8 GB of RAM

Conclusions

- VNX helps the management of testbed scenarios, saving on equipment investment and management resources
  - Reusability of testbeds
  - Facilitates sharing testbed infrastructures
- First version of VNX/EDIV successfully used in university computer network laboratories
- Distributed version needs improvements to cope with VMs heterogeneity

Future work

- Finish VNX GUI
- Complete and improve distributed cluster support:
  - Improve cluster interconnection mechanisms (OpenvSwitch,TRILL?)
  - Improve management of server heterogeneity
- Support dynamic scenarios: adding/releasing VMs and networks, VM mobility (libvirt+Sheepdog?)
- Improve network emulation capabilities
- Integrate and test new virtualization platforms (i.e. VMware)
- New types of virtual machines (i.e. Android)
- Support the integration of physical equipment into testbeds (plug-in)
- Testbeds over the Cloud
- Full OVF Environment support
- New applications:
  - Security: dynamic creation of (honeynets)

Thanks for your attention!

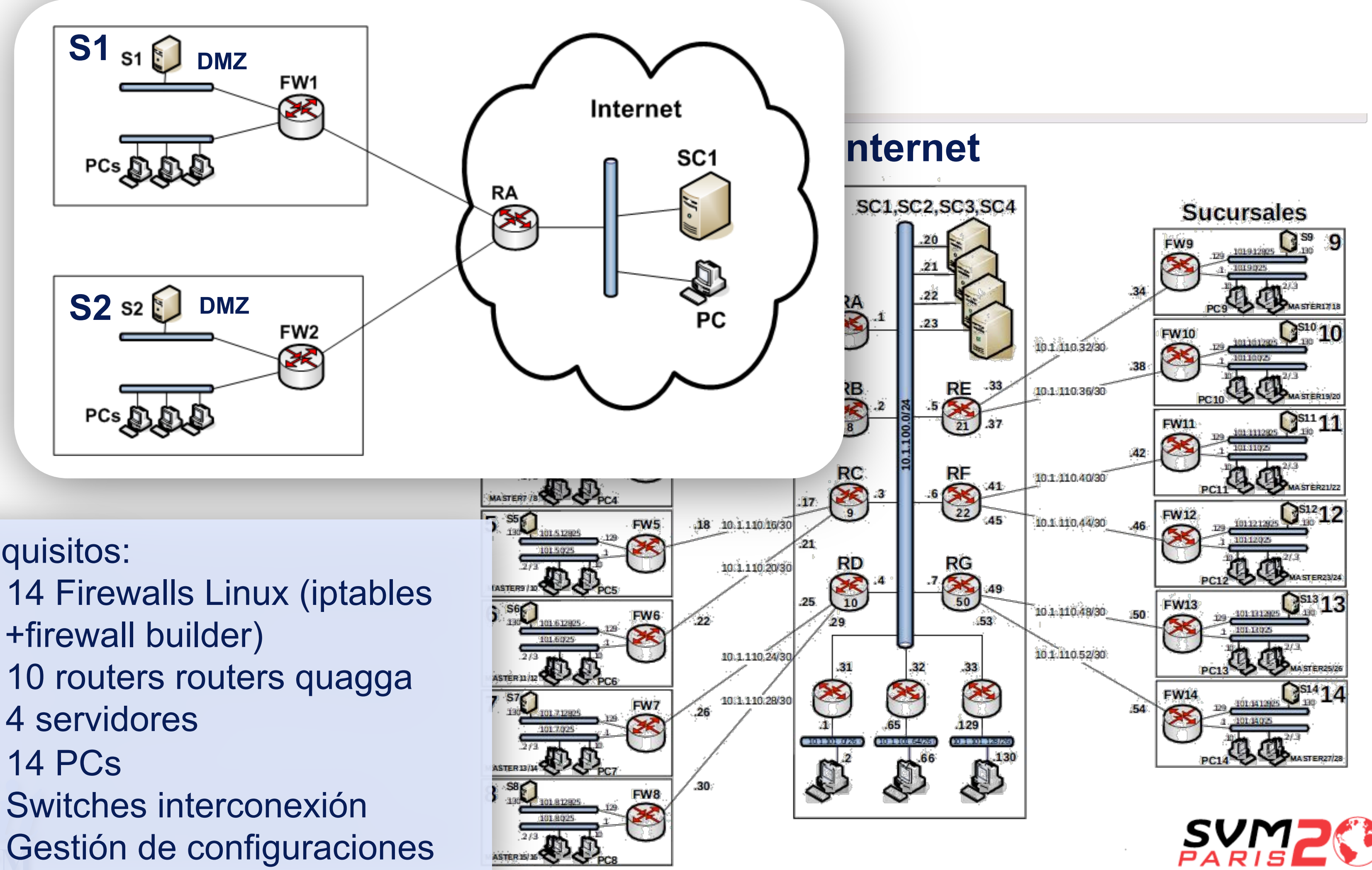
VNX

Virtual Networks over linuX

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

Additional Slides

Testbed Example: Firewalls



Requisitos:

- 14 Firewalls Linux (iptables +firewall builder)
- 10 routers routers quagga
- 4 servidores
- 14 PCs
- Switches interconexión
- Gestión de configuraciones

SVM2011 PARIS

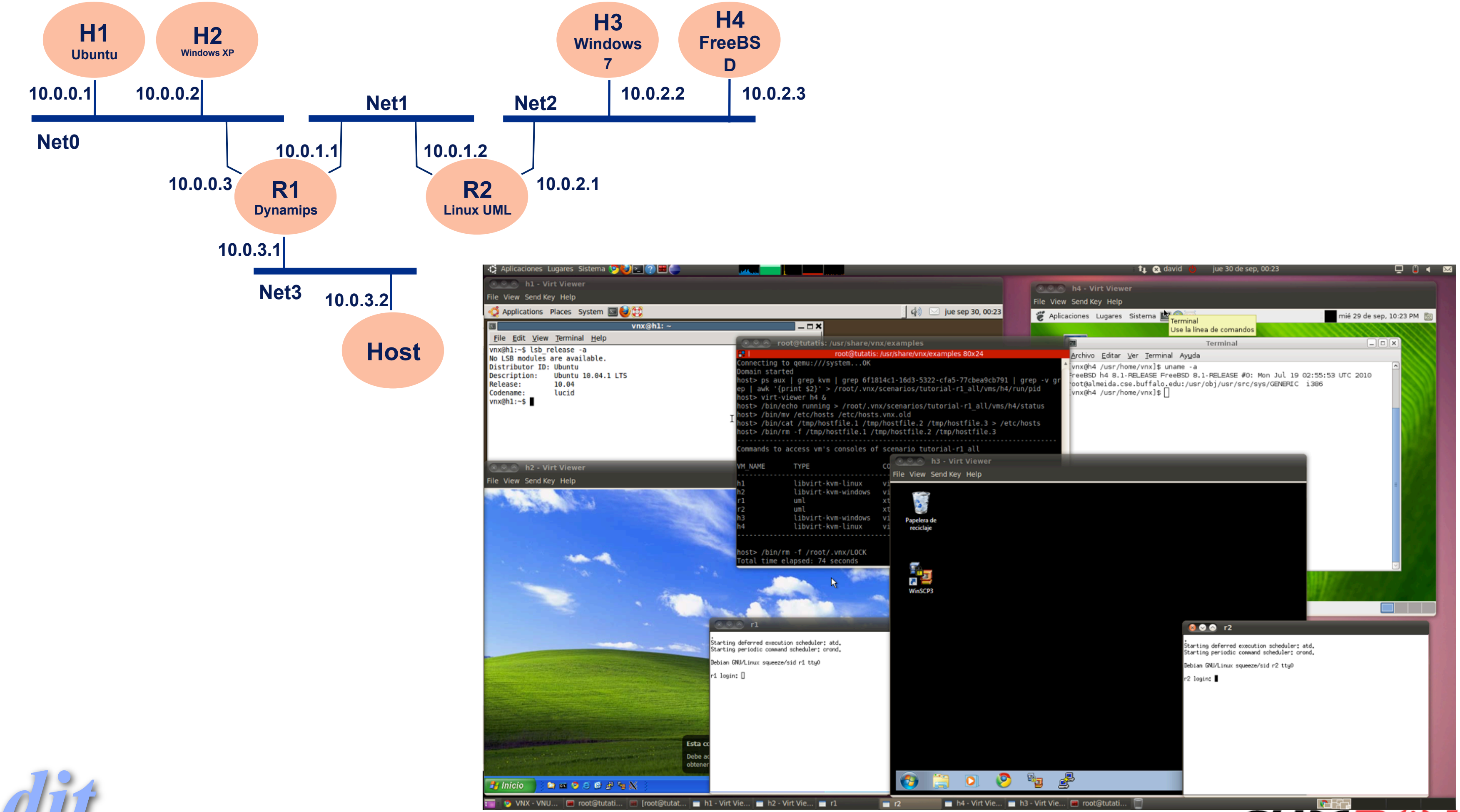
VNX Internal API

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

Use Examples

- Starting a VNS:
  - vnx –f escenario.xml --create
- Accessing consoles:
  - vnx –f escenario.xml --console –M vm1
- Executing commands:
  - vnx –f escenario.xml --execute start
- Restarting a VM:
  - vnx –f escenario.xml --reboot –M vm1
- Stopping/releasing the VNS:
  - vnx –f escenario.xml --shutdown
  - vnx –f escenario.xml --destroy

Example: tutorial_root1_all



```
vnx@h1:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 10.04.1 LTS
Release:        10.04
Codename:       lucid
vnx@h1:~$
```

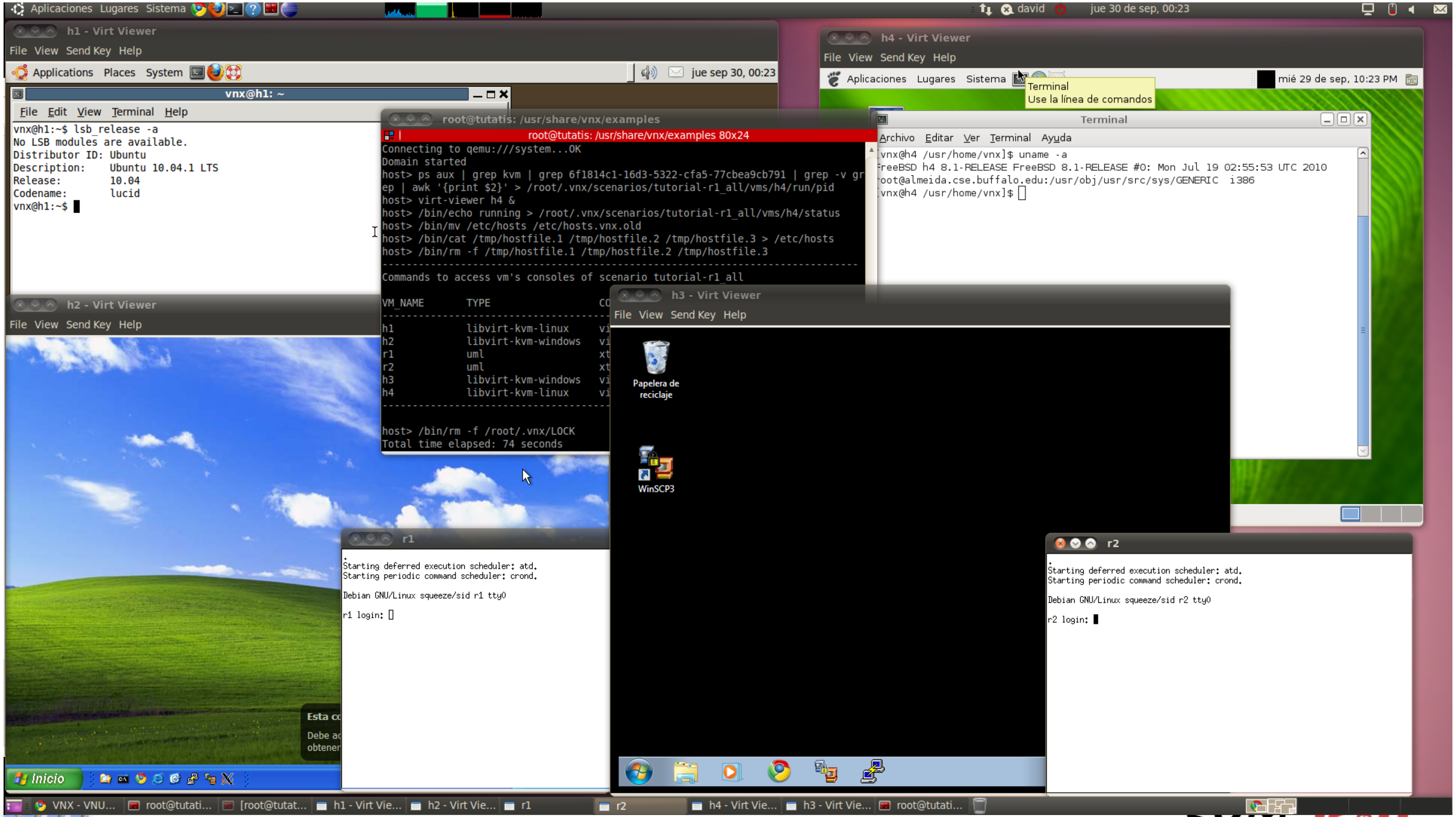
```
Connecting to qemu:///system...OK
Domain started
host> ps aux | grep kvm | grep 6f1814c1-16d3-5322-cfa5-77cbea9cb791 | grep -v gr
ep | awk '{print $2}' > /root/.vnx/scenarios/tutorial-r1_all/vms/h4/run/pid
host> virt-viewer h4 &
host> /bin/echo running > /root/.vnx/scenarios/tutorial-r1_all/vms/h4/status
host> /bin/mv /etc/hosts /etc/hosts.vnx.old
host> /bin/cat /tmp/hostfile.1 /tmp/hostfile.2 /tmp/hostfile.3 > /etc/hosts
host> /bin/rm -f /tmp/hostfile.1 /tmp/hostfile.2 /tmp/hostfile.3
--------------------------------------------------------------------------------
Commands to access vm's consoles of scenario tutorial-r1_all

VM_NAME       TYPE                CO
------------------------------------
h1            libvirt-kvm-linux   vi
h2            libvirt-kvm-windows vi
r1            uml                 xt
r2            uml                 xt
h3            libvirt-kvm-windows vi
h4            libvirt-kvm-linux   vi
------------------------------------

host> /bin/rm -f /root/.vnx/LOCK
Total time elapsed: 74 seconds
```

```
[vnx@h4 /usr/home/vnx]$ uname -a
FreeBSD h4 8.1-RELEASE FreeBSD 8.1-RELEASE #0: Mon Jul 19 02:55:53 UTC 2010
root@almeida.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  i386
[vnx@h4 /usr/home/vnx]$
```

```
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: crond.

Debian GNU/Linux squeeze/sid r1 tty0

r1 login:
```

```
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: crond.

Debian GNU/Linux squeeze/sid r2 tty0

r2 login:
```

Example: tutorial_root1_all



```
vnx@h1:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 10.04.1 LTS
Release:        10.04
Codename:       lucid
vnx@h1:~$
```

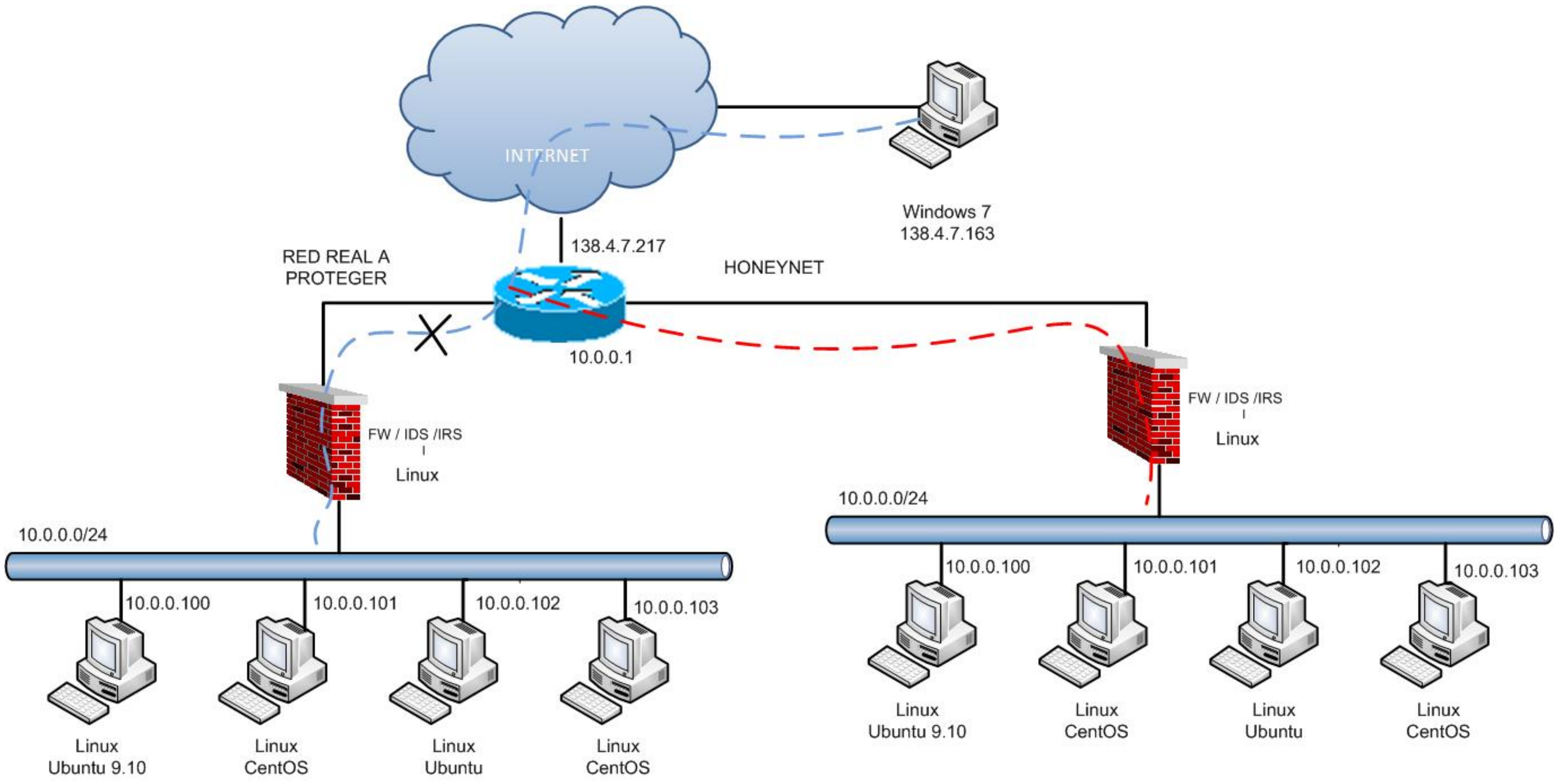
```
Connecting to qemu:///system...OK
Domain started
host> ps aux | grep kvm | grep 6f1814c1-16d3-5322-cfa5-77cbea9cb791 | grep -v gr
ep | awk '{print $2}' > /root/.vnx/scenarios/tutorial-r1_all/vms/h4/run/pid
host> virt-viewer h4 &
host> /bin/echo running > /root/.vnx/scenarios/tutorial-r1_all/vms/h4/status
host> /bin/mv /etc/hosts /etc/hosts.vnx.old
host> /bin/cat /tmp/hostfile.1 /tmp/hostfile.2 /tmp/hostfile.3 > /etc/hosts
host> /bin/rm -f /tmp/hostfile.1 /tmp/hostfile.2 /tmp/hostfile.3
--------------------------------------------------------------------------------
Commands to access vm's consoles of scenario tutorial-r1_all

VM_NAME       TYPE                  CO
--------------------------------------
h1            libvirt-kvm-linux     vi
h2            libvirt-kvm-windows   vi
r1            uml                   xt
r2            uml                   xt
h3            libvirt-kvm-windows   vi
h4            libvirt-kvm-linux     vi
--------------------------------------

host> /bin/rm -f /root/.vnx/LOCK
Total time elapsed: 74 seconds
```

```
[vnx@h4 /usr/home/vnx]$ uname -a
FreeBSD h4 8.1-RELEASE FreeBSD 8.1-RELEASE #0: Mon Jul 19 02:55:53 UTC 2010
root@almeida.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  i386
[vnx@h4 /usr/home/vnx]$
```

```
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: crond.

Debian GNU/Linux squeeze/sid r1 tty0

r1 login:
```

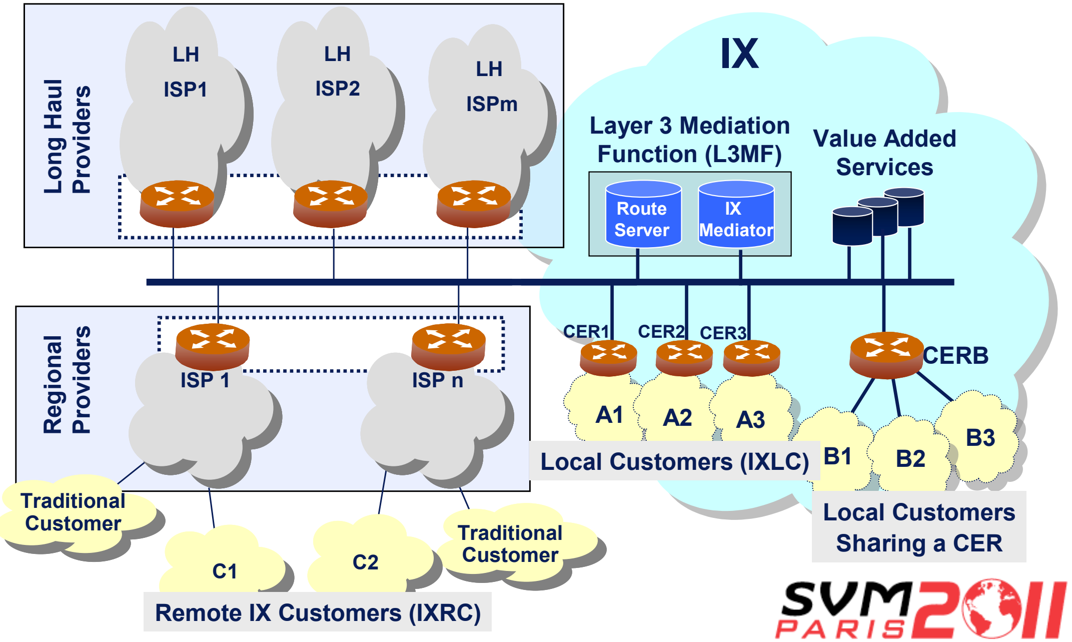
```
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: crond.

Debian GNU/Linux squeeze/sid r2 tty0

r2 login:
```

Segur@: Dynamic Deployment of Honeynets



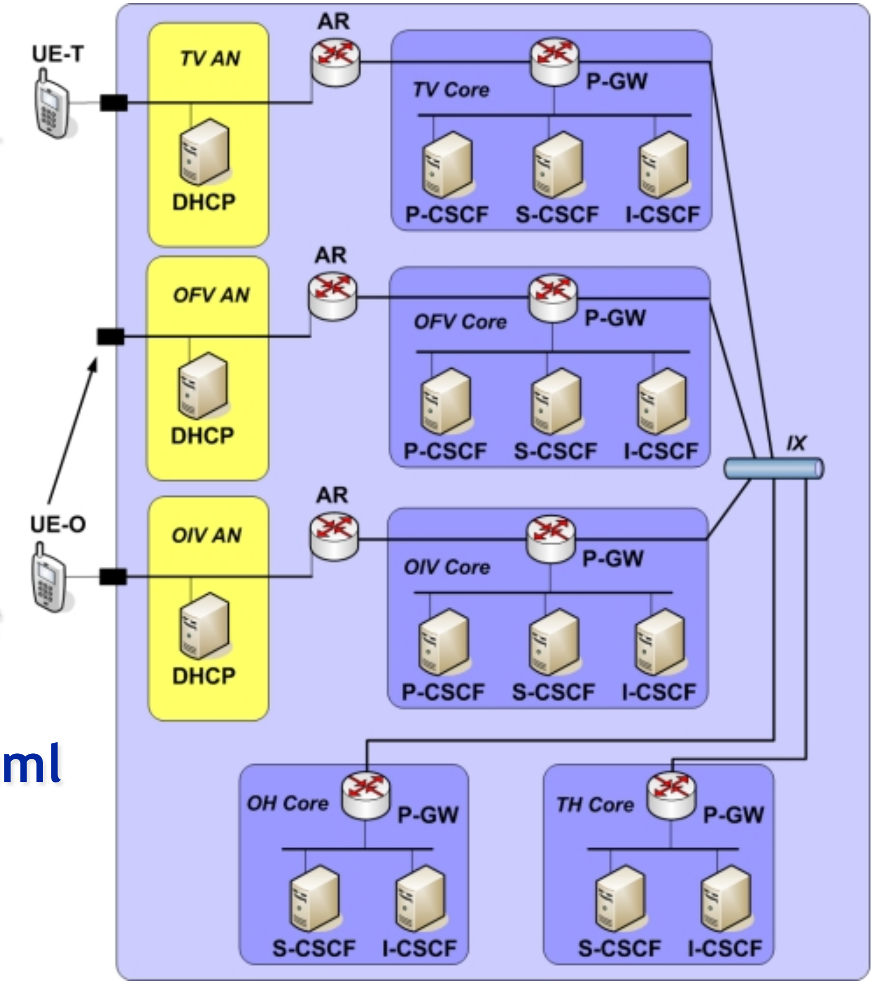
Proyecto Euro6IX

- VNUML se desarrollo inicialmente en el contexto del proyecto Euro6IX
  - Diseño de un nuevo modelo de punto de intercambio (IX) para IPv6 con asignación de direcciones basadas en IX
- Validación basada en VNUML:
  - Escenarios de hasta 20 vms
  - Quagga, BGP, RPSLng, etc



3GPP System Architecture Evolution (SAE)

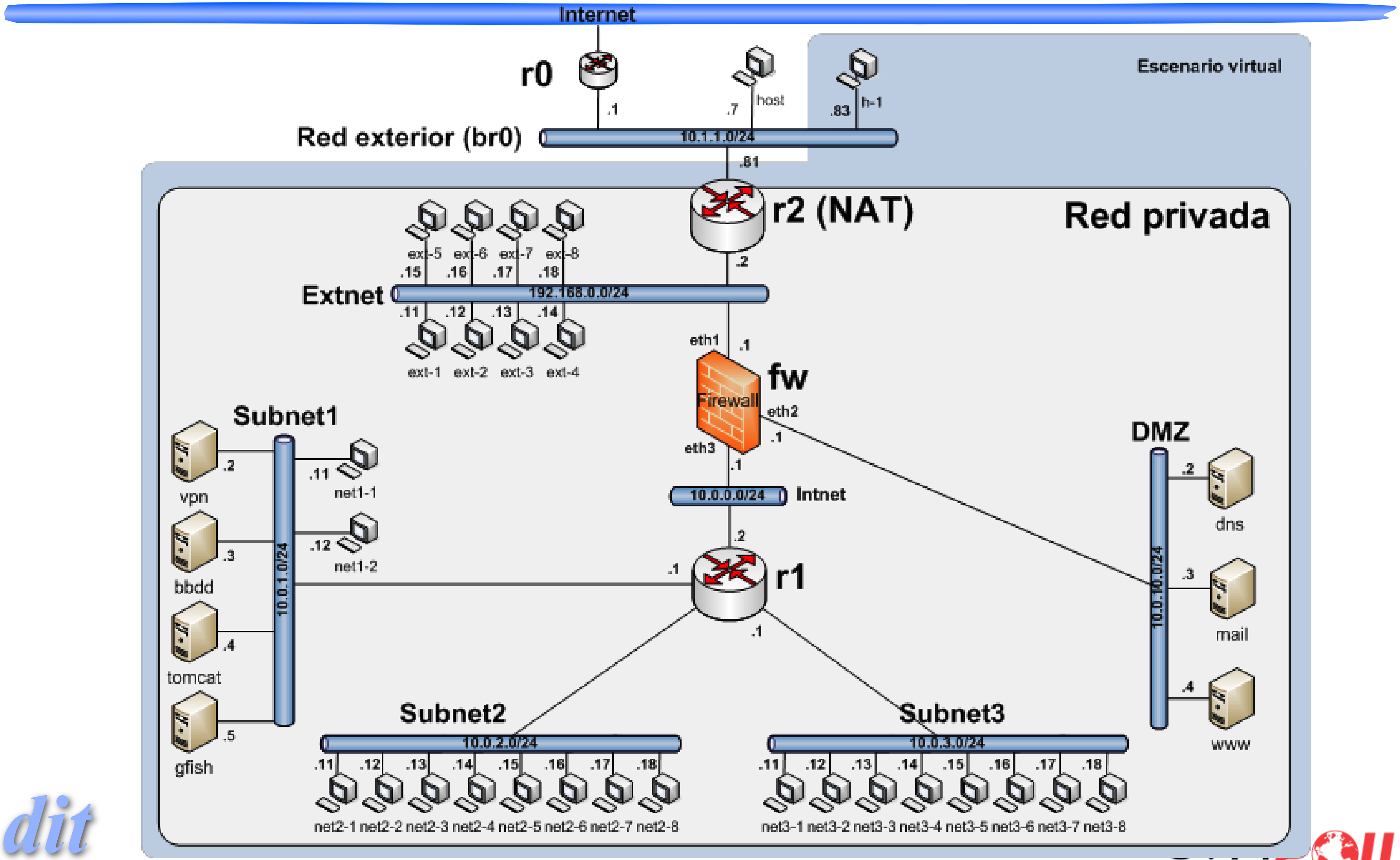
- Plataforma de pruebas de escenarios 3GPP SAE multi-dominio basados en movilidad e IMS
  - Open source IMS
  - Movilidad IPv6 intra e interdominio
- Disponible en la sección de ejemplos:
  - **http://www.dit.upm.es/vnuml**



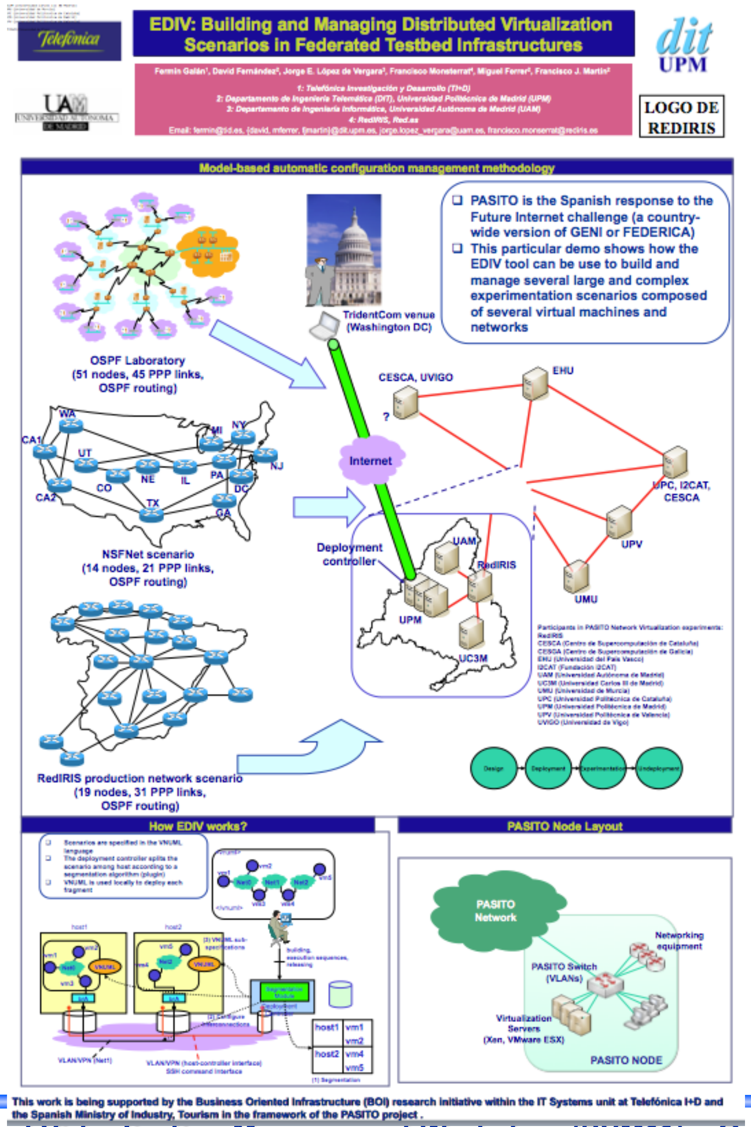
Movilidad interdominio

http://www.dit.upm.es/vnumlwiki/index.php/3gpp-emulator

Segur@: Escenario Red Corporativa



EDIV Demo in TridentCom 2009



EDIV: Building and Managing Distributed Virtualization Scenarios in Federated Testbed Infrastructures

Fermín Galán[1], David Fernández[2], Jorge E. López de Vergara[3], Francisco Monserrat[4], Miguel Ferrer[2], Francisco J. Martín[2]

1: Telefónica Investigación y Desarrollo (TI+D)
2: Departamento de Ingeniería Telemática (DIT), Universidad Politécnica de Madrid (UPM)
3: Departamento de Ingeniería Informática, Universidad Autónoma de Madrid (UAM)
4: RedIRIS, Red.es
Email: fermin@tid.es, {david, mferrer, fjmartin}@dit.upm.es, jorge.lopez_vergara@uam.es, francisco.monserrat@rediris.es

Model-based automatic configuration management methodology

- ❑ PASITO is the Spanish response to the Future Internet challenge (a country-wide version of GENI or FEDERICA)
- ❑ This particular demo shows how the EDIV tool can be use to build and manage several large and complex experimentation scenarios composed of several virtual machines and networks

Participants in PASITO Network Virtualization experiments:
RedIRIS
CESCA (Centro de Supercomputación de Cataluña)
CESGA (Centro de Supercomputación de Galicia)
EHU (Universidad del País Vasco)
I2CAT (Fundación I2CAT)
UAM (Universidad Autónoma de Madrid)
UC3M (Universidad Carlos III de Madrid)
UMU (Universidad de Murcia)
UPC (Universidad Politécnica de Cataluña)
UPM (Universidad Politécnica de Madrid)
UPV (Universidad Politécnica de Valencia)
UVIGO (Universidad de Vigo)

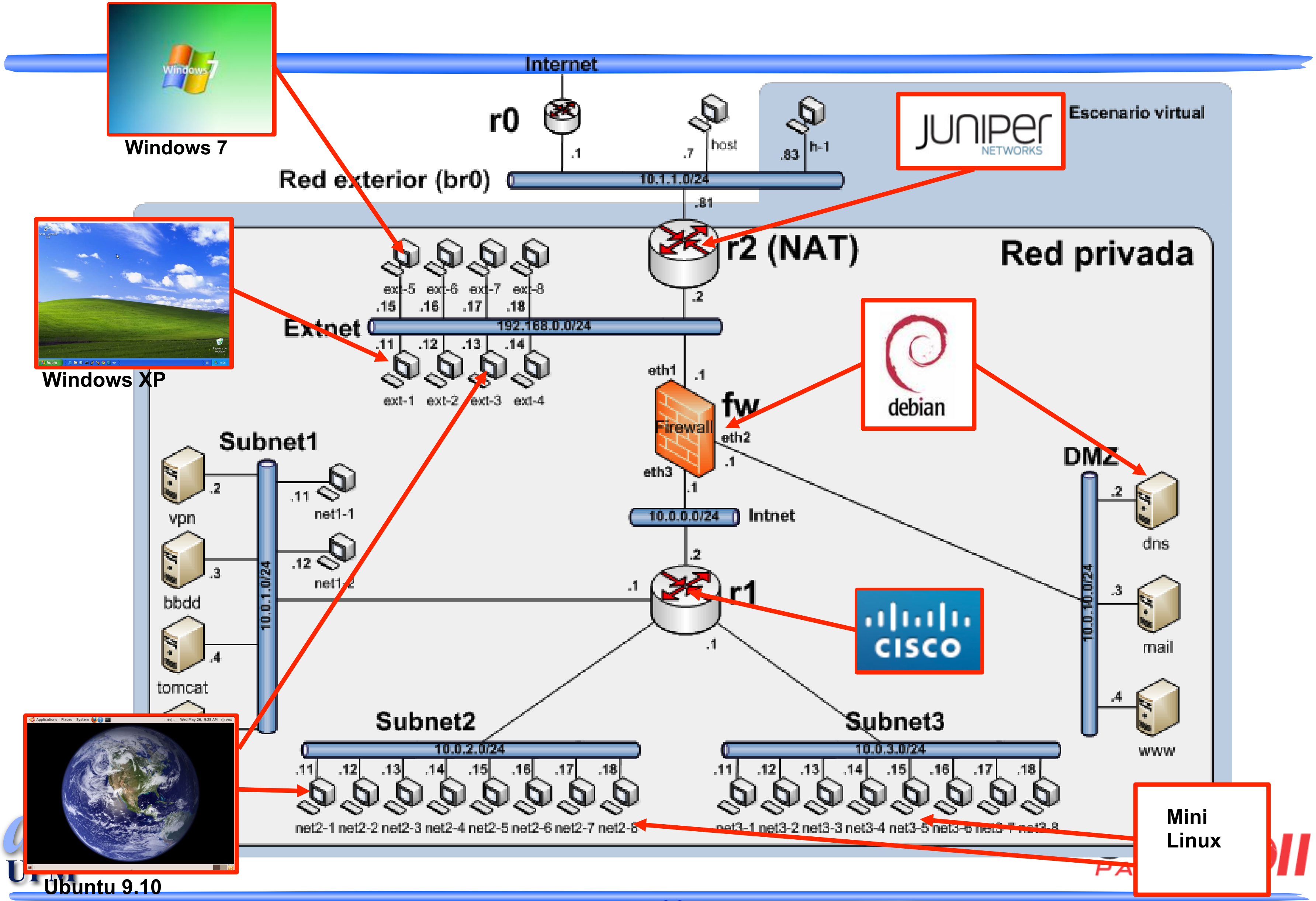
How EDIV works?

- ❑ Scenarios are specified in the VNUML language
- ❑ The deployment controller splits the scenario among host according to a segmentation algorithm (plugin)
- ❑ VNUML is used locally to deploy each fragment

PASITO Node Layout

This work is being supported by the Business Oriented Infrastructure (BOI) research initiative within the IT Systems unit at Telefónica I+D and the Spanish Ministry of Industry, Tourism in the framework of the PASITO project .

Escenario de la Demostración: Red Corporativa


Windows 7
Internet
r0
.1
.7
host
.83
h-1
Escenario virtual
Red exterior (br0)
10.1.1.0/24
.81
r2 (NAT)
Red privada
Windows XP
ext-5 ext-6 ext-7 ext-8
.15 .16 .17 .18
.2
Extnet
192.168.0.0/24
.11 .12 .13 .14
ext-1 ext-2 ext-3 ext-4
eth1
.1
fw
Firewall
eth2
debian
Subnet1
eth3
.1
DMZ
.2
.11
vpn
net1-1
.1
10.0.0.0/24
Intnet
.2
dns
.3
.12
net1-2
bbdd
10.0.1.0/24
.1
r1
.3
.4
mail
tomcat
.1
10.0.10.0/24
.4
Subnet2
Subnet3
www
10.0.2.0/24
10.0.3.0/24
.11 .12 .13 .14 .15 .16 .17 .18
.11 .12 .13 .14 .15 .16 .17 .18
net2-1 net2-2 net2-3 net2-4 net2-5 net2-6 net2-7 net2-8
net3-1 net3-2 net3-3 net3-4 net3-5 net3-6 net3-7 net3-8
Mini Linux
Ubuntu 9.10