



Creación de escenarios de red virtuales distribuidos

David Fernández

Dpto. Ingeniería de Sistemas Telemáticos

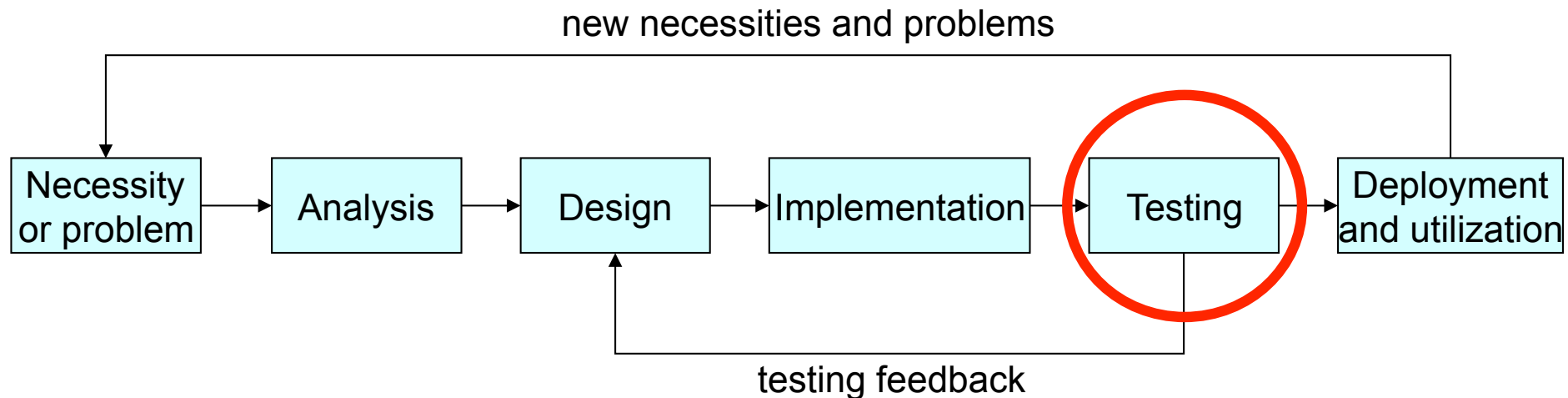
Universidad Politécnica de Madrid

david@dit.upm.es

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

Contexto

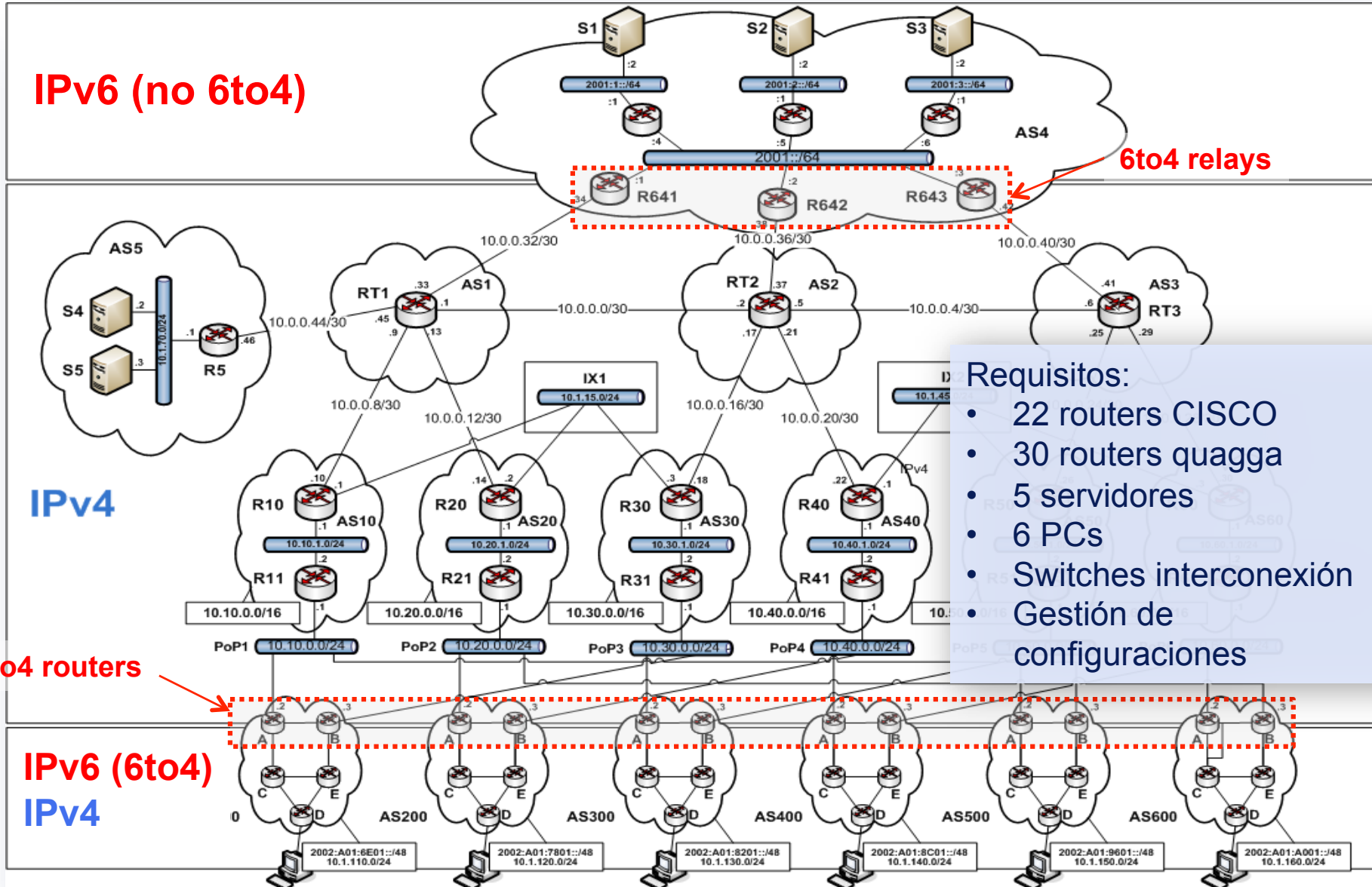
◆ Ciclo de vida de desarrollo de un sistema:



◆ Testbed de redes y servicios (definición informal)

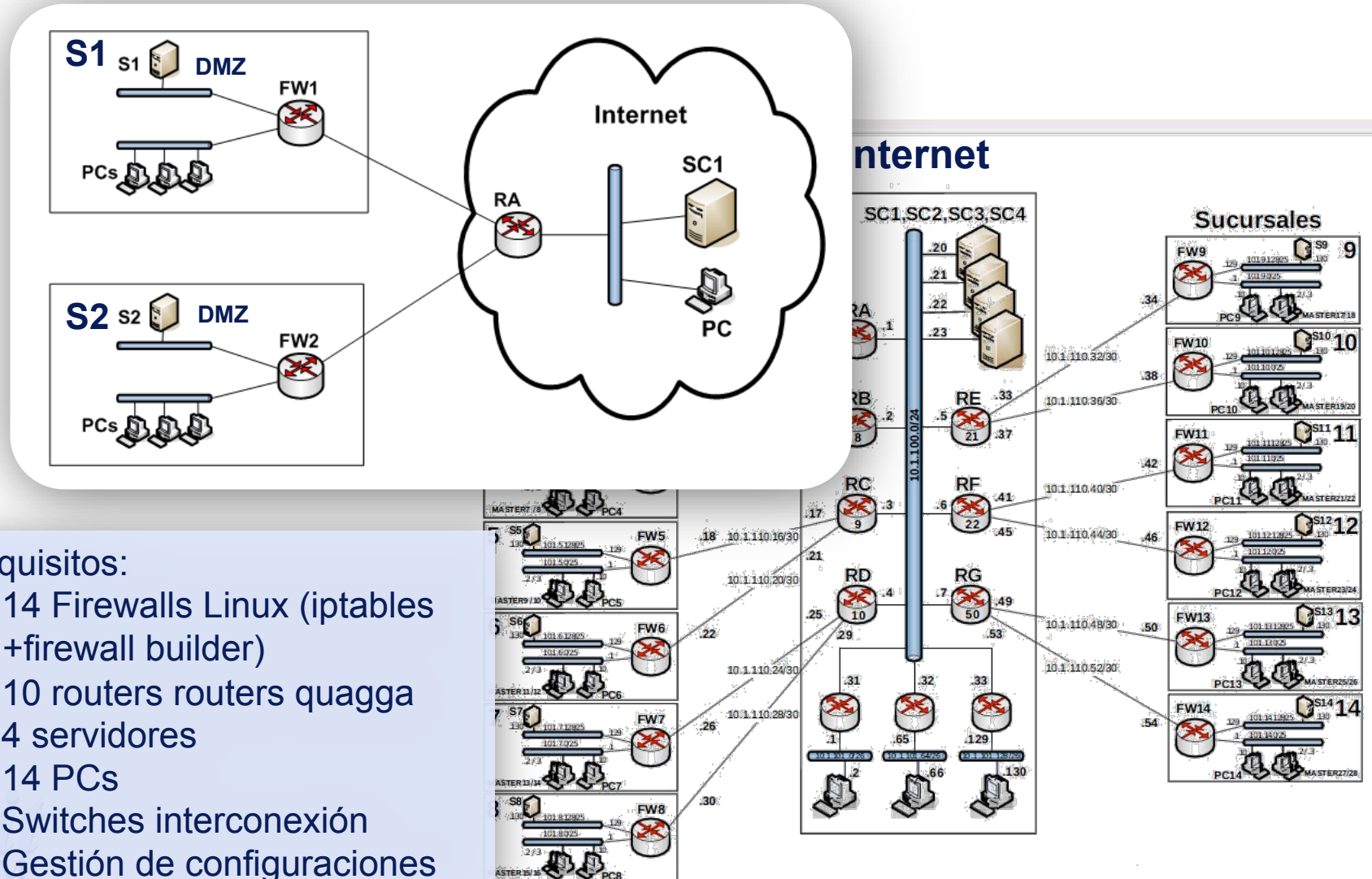
- Herramienta o sistema consistente en un entorno controlado de redes y sistemas, con semejanza a entornos del “mundo real”, donde los ingenieros prueban y evalúan sus implementaciones

Ejemplo escenario I: 6to4



- Requisitos:
- 22 routers CISCO
 - 30 routers quagga
 - 5 servidores
 - 6 PCs
 - Switches interconexión
 - Gestión de configuraciones

Ejemplo escenario II: Cortafuegos

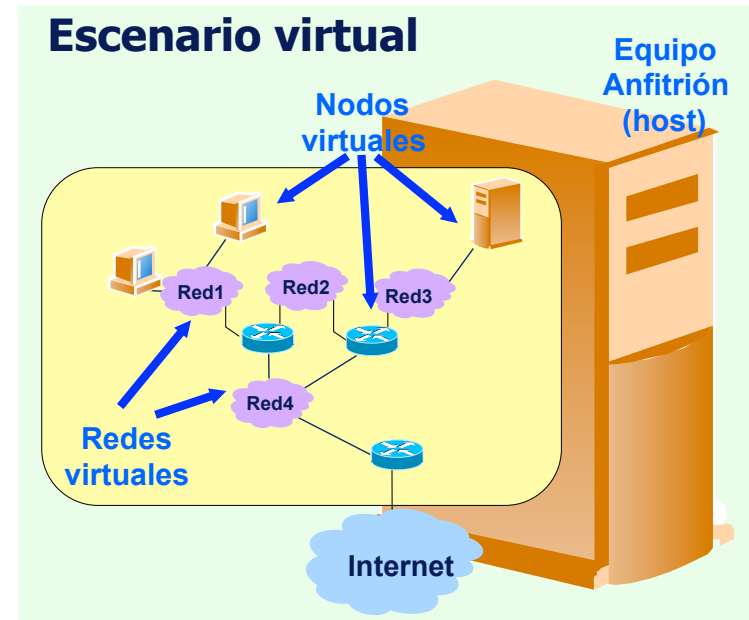
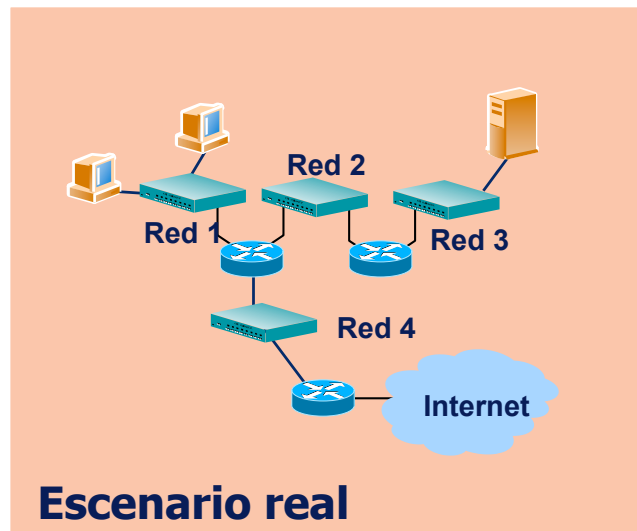


Requisitos:

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

Técnicas de Virtualización

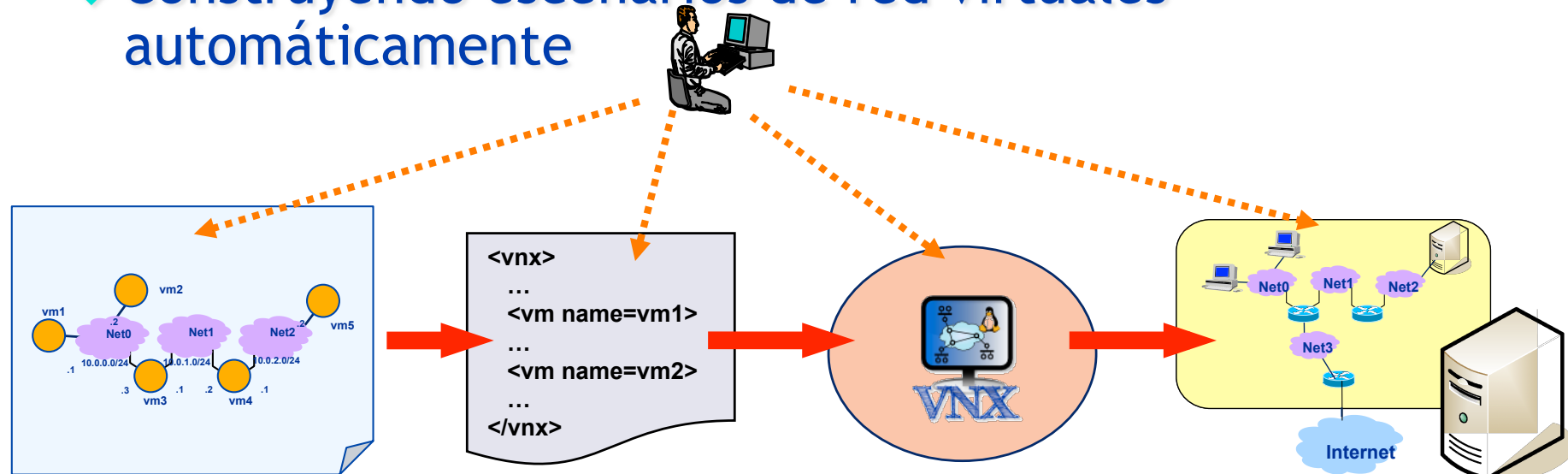
- ◆ Permiten ejecutar varias máquinas virtuales sobre un equipo anfitrión
 - Ej: Xen, VMware, User Mode Linux, etc
- ◆ Combinadas con el uso de redes virtuales emuladas en el host permiten crear **Escenarios de Red Virtuales** incluso con conexiones externas



- ◆ Herramientas gestión de escenarios virtuales:
 - GNS3, Netkit, MNL, Marionnet, VNX/VNUML, etc

VNX: Virtual Networks over Linux

◆ Construyendo escenarios de red virtuales automáticamente



Diseño

El usuario diseña el escenario (off-line)

Especificación

El usuario especifica el escenario en el lenguaje VNUML usando:

- Editor Gráfico (VNUMLGUI)
- Editor XML

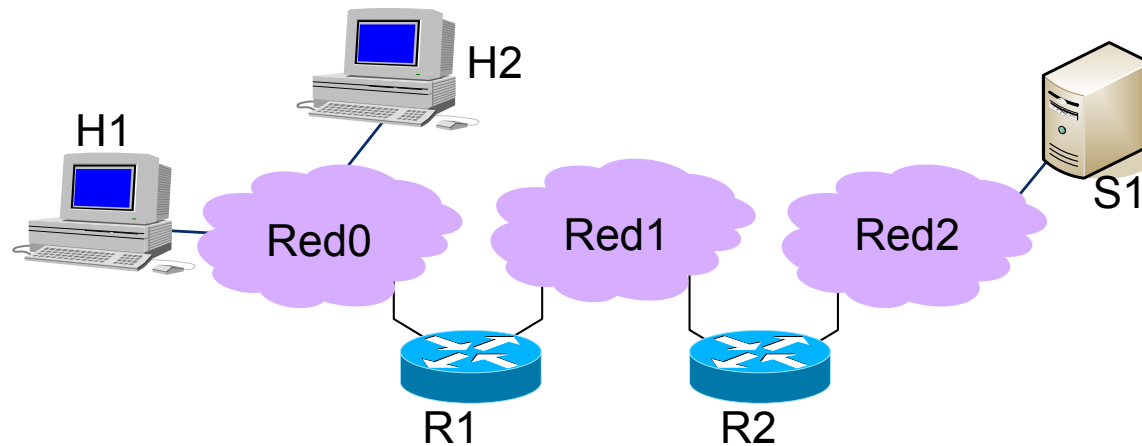
Creación

VNUML procesa la especificación y crea el escenario virtual

Interacción con el escenario

El usuario ejecuta comandos en las máquinas virtuales a través de VNUML o directamente

Lenguaje VNX (I)



```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<vnx>
```

(definiciones globales: **<global>**)

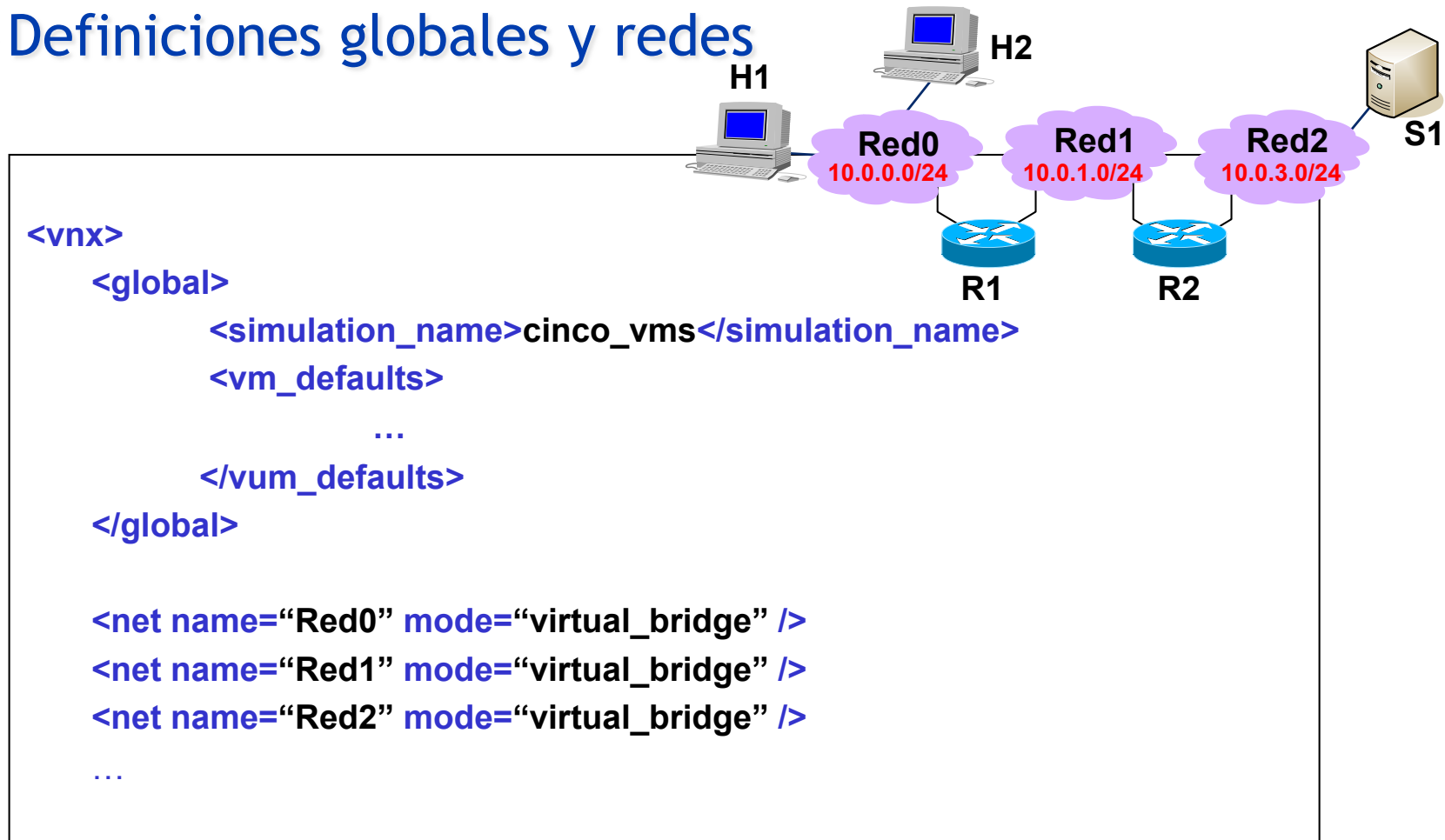
(definiciones de redes virtuales: **<net>**)

(definiciones de máquinas virtuales: **<vm>**)

```
</vnx>
```

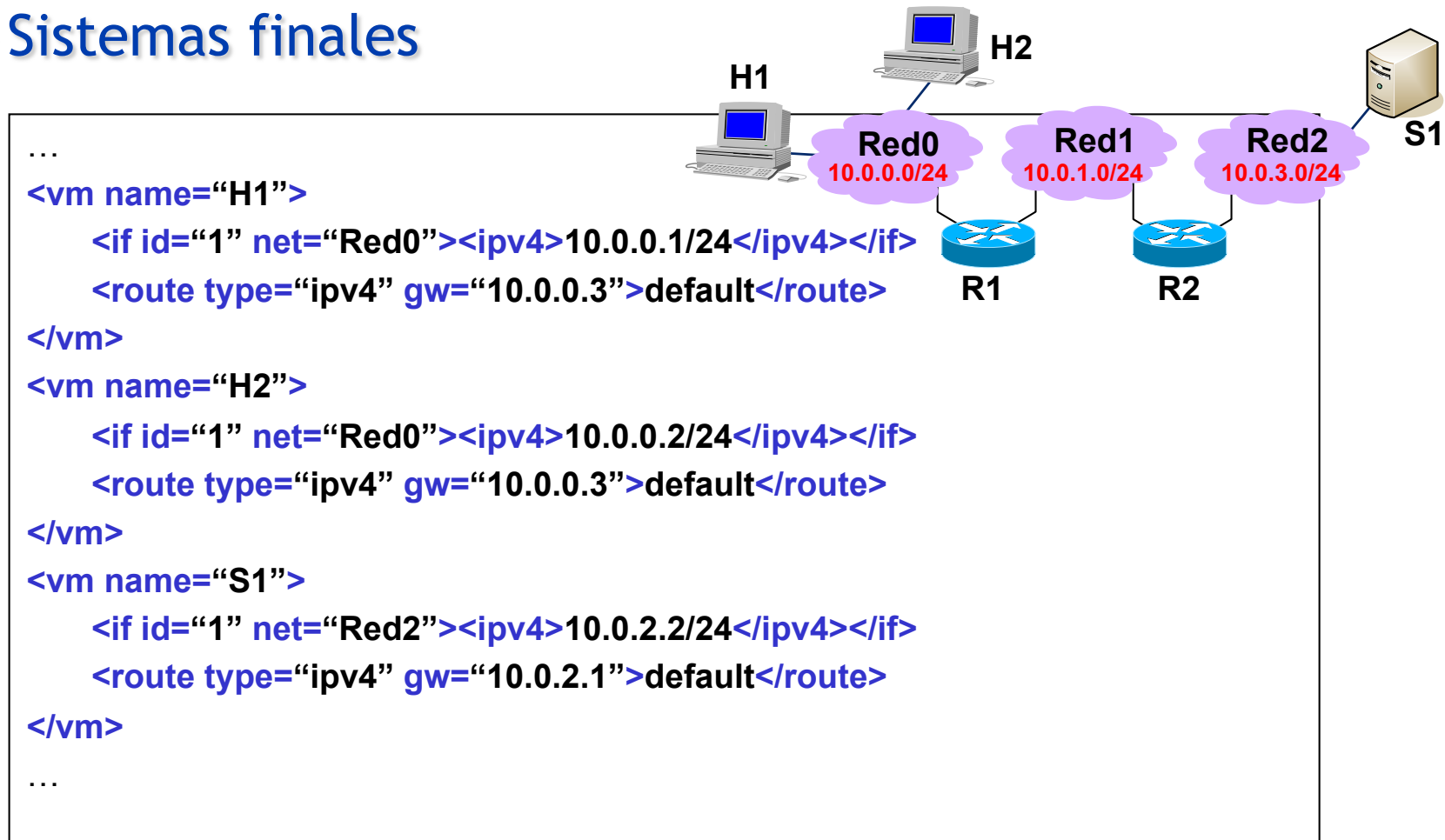
Lenguaje VNX (II)

◆ Definiciones globales y redes



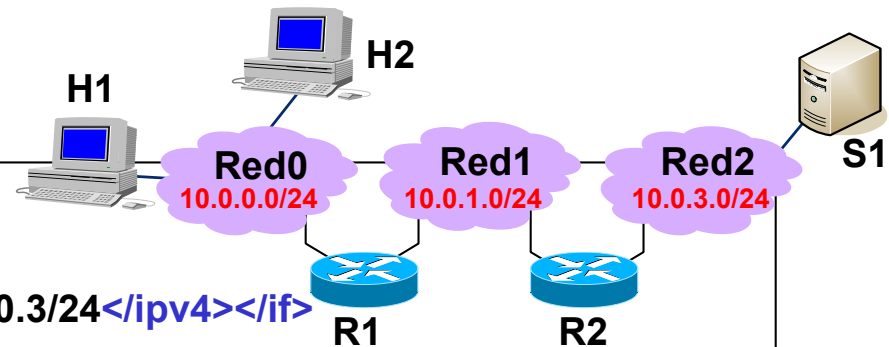
Lenguaje VNX (III)

◆ Sistemas finales



Lenguaje VNX (IV)

◆ Routers



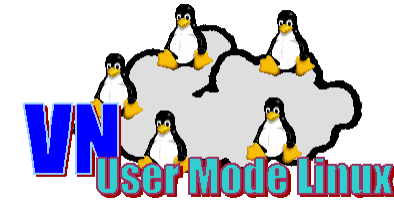
```
...  
<vm name="uml3">  
  <if id="1" net="Red0"><ipv4>10.0.0.3/24</ipv4></if>  
  <if id="2" net="Red1"><ipv4>10.0.1.1/24</ipv4></if>  
  <route type="ipv4" gw="10.0.1.2">10.0.2.0/24</route>  
  <forwarding type="ip" />  
</vm>  
<vm name="uml4">  
  <if id="1" net="Red1"><ipv4>10.0.1.2/24</ipv4></if>  
  <if id="2" net="Red2"><ipv4>10.0.2.1/24</ipv4></if>  
  <route type="ipv4" gw="10.0.1.1">default</route>  
  <forwarding type="ip" />  
</vm>  
...
```

VNX: Virtual Networks over Linux

- ◆ VNX se distribuye con licencia libre GPL y con vocación de herramienta de uso público para investigadores y docentes

- ◆ Basada en:

- Virtual Networks User Mode Linux (VNUML)
- Limitada a máquinas virtuales Linux



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

- ◆ Nuevas funcionalidades incorporadas:

- Integración de libvirt (estándar acceso a virtualización de Linux)
- Autoconfiguración para Windows XP, Windows 7, Linux (Ubuntu 9.10/10.04/11.04, Fedora 14, CentOS 5.6), FreeBSD (8.1)
- Integración Dynamips (CISCO)
- Integración Olive (Juniper)
- Funcionalidad de gestión individual de máquinas
- Versión distribuida: EDIV
- Versión beta disponible con documentación preliminar (recetas instalación):



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

✚ <http://www.dit.upm.es/vnx>

Autoconfiguración y Ejecución de Comandos

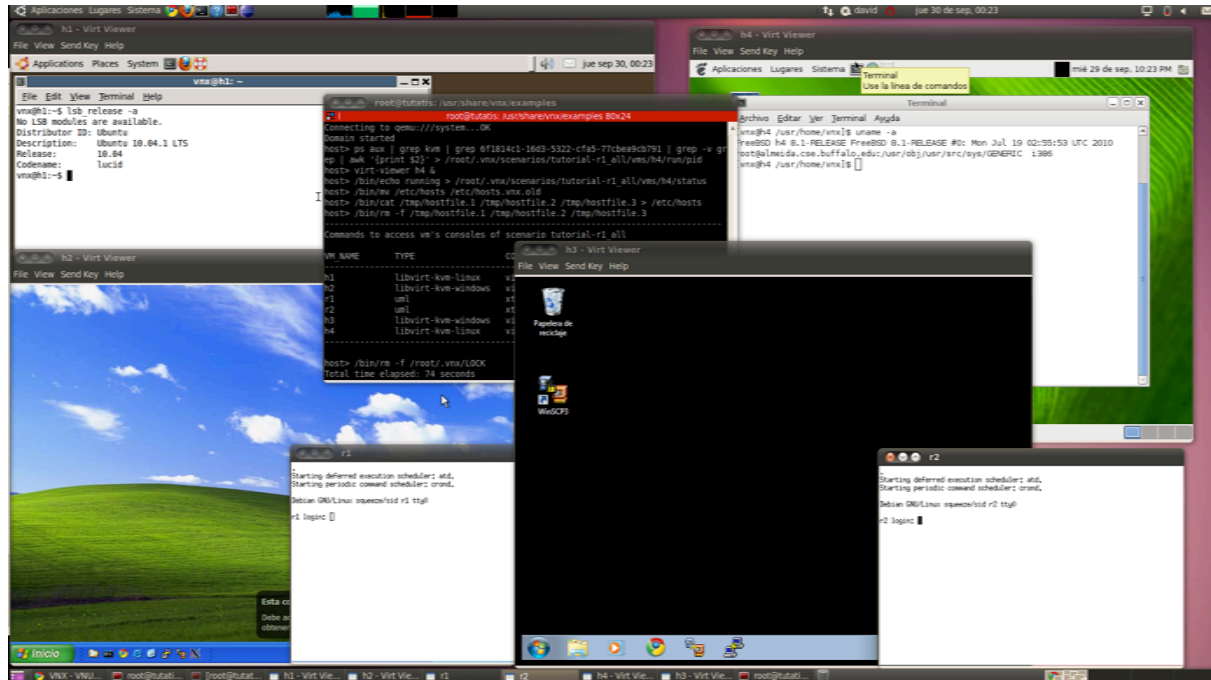
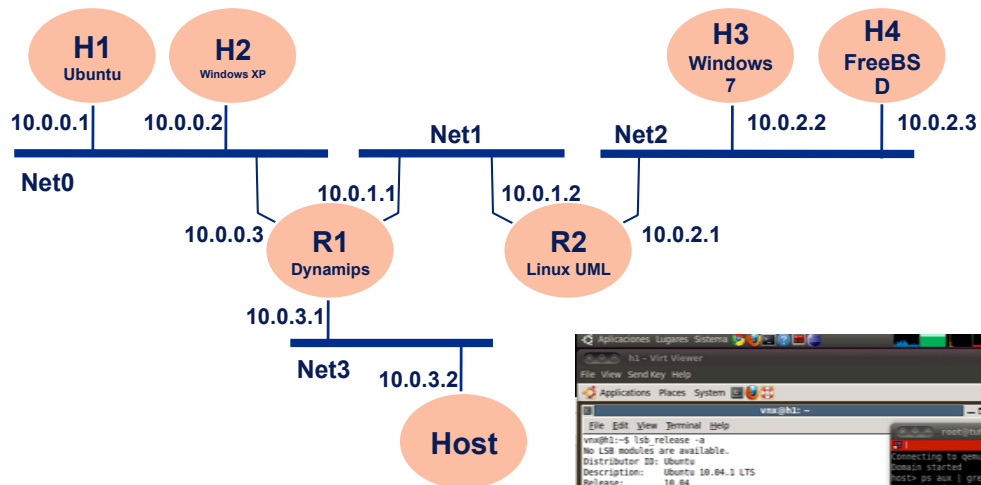
- ◆ Basada en discos (CDROM) creados dinámicamente y ofrecidos a las máquinas virtuales
 - Similar propuesta OVF
- ◆ Demonio de autoconfiguración y ejecución de comandos (ACED) corriendo dentro de las VMs
 - Lee ficheros xml y los traduce a comandos

```
<?xml version="1.0" encoding="UTF-8"?>
<create_conf>
  <vm name="uml4">
    <kernel>default</kernel>
    <filesystem type="cow">/usr/share/vnuml/
filesystems/root_fs_winxp</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>
```

Ejemplos de uso

- ◆ Arranque del escenario:
 - `vnx -f escenario.xml --create`
- ◆ Acceso a consolas:
 - `vnx -f escenario.xml --console -M vm1`
- ◆ Ejecución de comandos:
 - `vnx -f escenario.xml --execute start`
- ◆ Rearranque de una máquina virtual:
 - `vnx -f escenario.xml --reboot -M vm1`
- ◆ Parada del escenario:
 - `vnx -f escenario.xml --shutdown`
 - `vnx -f escenario.xml --destroy`

Ejemplos: tutorial_root1_all



Ejemplos: tutorial_root1_all

The screenshot displays a Linux desktop environment with several Virtual Machine (VM) windows open. The windows show terminal outputs for different VMs, including h1, h2, h3, h4, r1, and r2. A central terminal window shows commands and their results, including a table of VM names and types.

```
root@tutatis: /usr/share/vnx/examples
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
-----
h1            libvirt-kvm-linux
h2            libvirt-kvm-windows
r1            uml
r2            uml
h3            libvirt-kvm-windows
h4            libvirt-kvm-linux
-----

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

Captura escenario I: 6to4

RT1 - console #1

```
View Search Terminal Help
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
```

| Network | Next Hop | Metric | LocPrf | Weight | Path |
|-----------------|-----------|--------|--------|--------|-----------|
| * 10.10.0.0/16 | 10.1.45.2 | 0 | 50 | 3 | 2 1 10 i |
| * 10.20.0.0/16 | 10.1.45.1 | 0 | 40 | 2 | 1 10 i |
| * 10.30.0.0/16 | 10.0.0.29 | 0 | 3 | 2 | 1 10 i |
| * 10.40.0.0/16 | 10.1.45.2 | 0 | 50 | 3 | 2 30 20 i |
| * 10.50.0.0/16 | 10.1.45.1 | 0 | 40 | 2 | 30 20 i |
| * 110.60.0.0/16 | 10.60.1.2 | 0 | 3 | 2 | 30 20 i |

R60 - console #1

```
File Edit View Search Terminal Help
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 |
| * 10.20.0.0/16 | 10.1.45.1 | 0 | 40 | 2 | 1 10 i |
| * 10.30.0.0/16 | 10.0.0.29 | 0 | 3 | 2 | 1 10 i |
| * 10.40.0.0/16 | 10.1.45.2 | 0 | 50 | 3 | 2 30 20 i |
| * 10.50.0.0/16 | 10.1.45.1 | 0 | 40 | 2 | 30 20 i |
| * 110.60.0.0/16 | 10.60.1.2 | 0 | 3 | 2 | 30 20 i |

Virtual Networks over Linux
<http://www.dit.upm.es/vnx>
vnx@dit.upm.es

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

Version: 1.92beta1 (build on 07/05/2011 01:00:00)

Escenarios Virtuales: Escalabilidad

- ◆ El número de máquinas virtuales que puede gestionar un anfitrión está limitado por los recursos disponibles y por los consumidos
 - Tráfico de red
 - Memoria
 - Disco, etc.
- ◆ Si queremos emular escenarios mayores es necesario:
 - Dividir los escenarios virtuales en partes que se ejecuten en anfitriones distintos e interconectar las partes a través de una infraestructura de red común
 - ✚ Cluster de virtualización
 - Requisitos: transparencia, eficiencia.

Proyecto EDIV

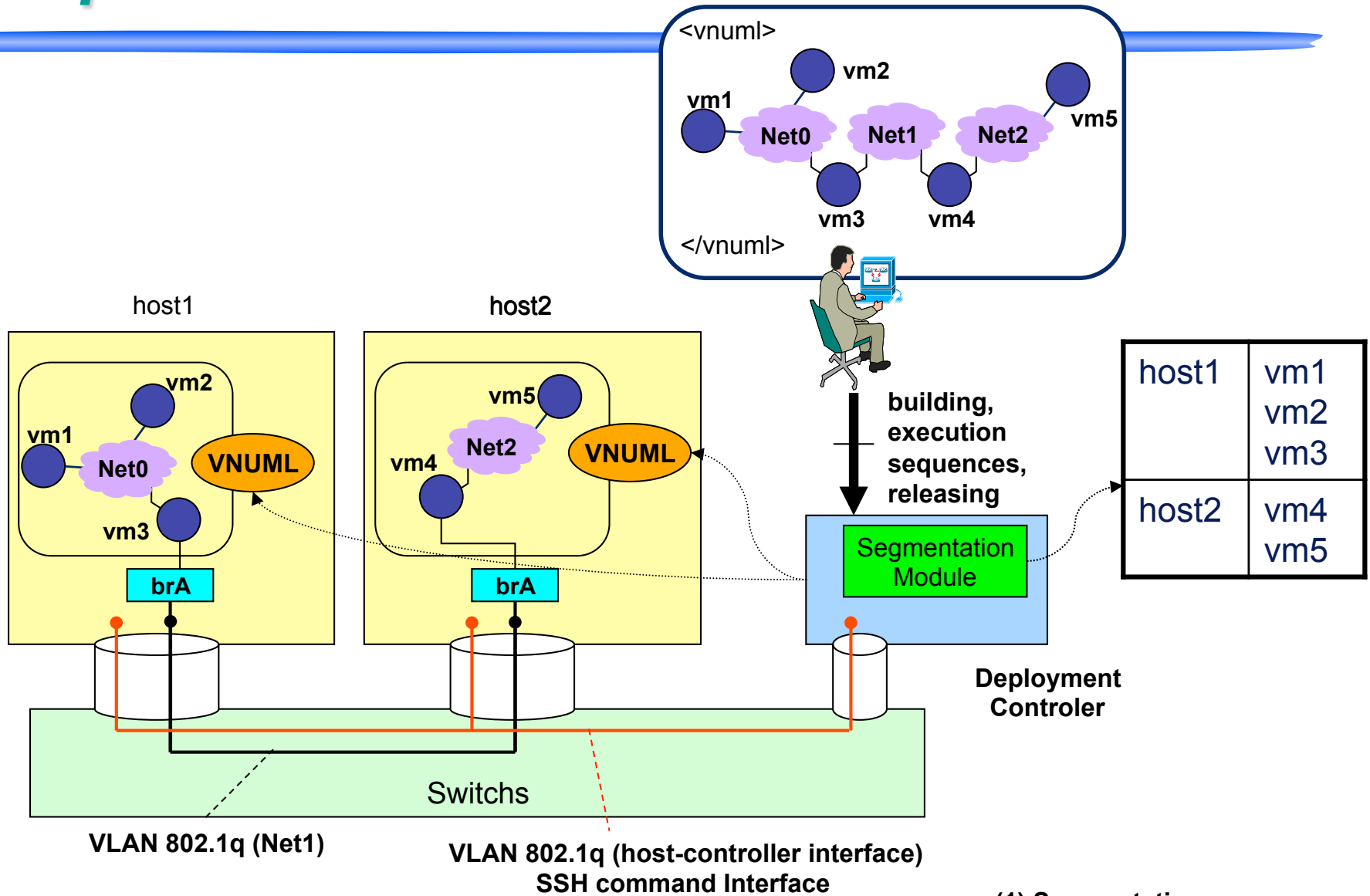
- ◆ Escenarios Virtuales con VNUML (EDIV)
- ◆ Colaboración entre TID y UPM en el contexto de la línea BOI (Business Oriented Infrastructure) dentro de la dirección de Sistemas de Apoyo a Negocio de TID.
- ◆ Objetivos:
 - Desarrollo de un prototipo distribuido de VNUML
 - ✚ Fermín Galán, David Fernández, "Distributed Virtualization Scenarios Using VNUML", System and Virtualization Management Workshop (SVM 2007), Toulouse (France), October 23-24 2007
 - Estudio sobre la generalización de VNUML a otras técnicas de virtualización



Duración: 6 meses (enero-junio 2008)



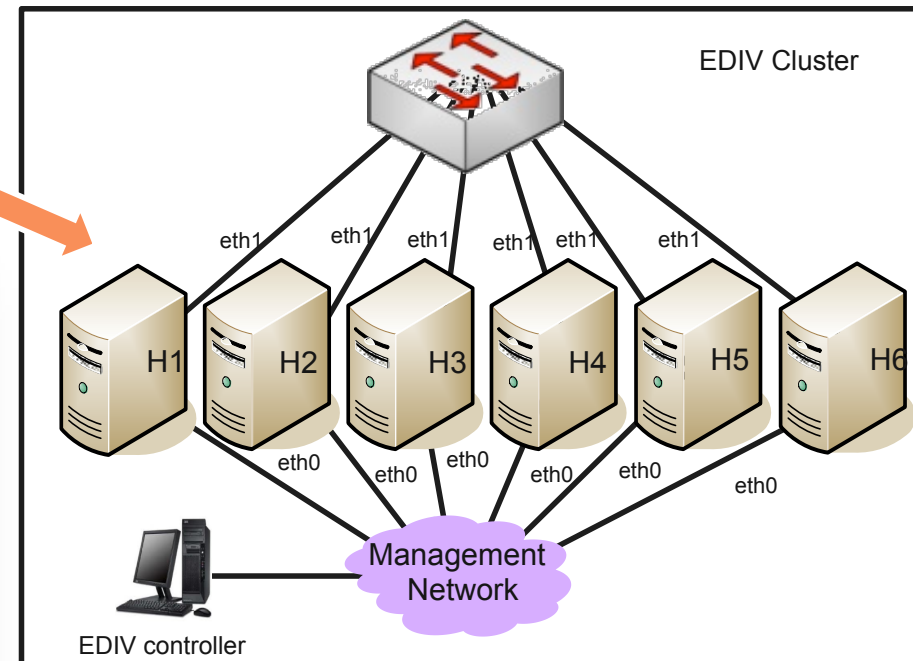
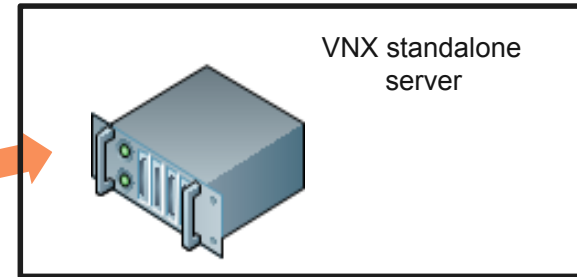
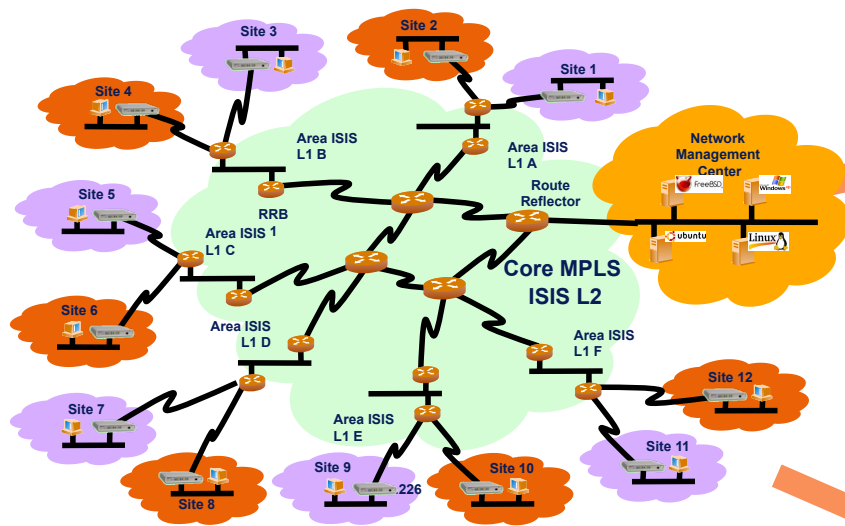
Arquitectura VNX distribuido: EDIV



Algoritmos de Segmentación

- ◆ Algoritmos implementados en EDIV
 - *Round-robin*
 - *Weighted round-robin* en función de la carga de cada servidor y de su potencia
 - *Explicito* (definido por usuario mediante restricciones)
- ◆ API para incorporar nuevos algoritmos

Escenario pruebas EDIV: MPLS VPN



◆ Escenario ISP:

- 16 routers CISCO
- 6 routers Juniper
- 6 routers quagga
- 16 sistemas finales (Debian, Ubuntu, WinXP, FreeBSD)
- Total: 44 sistemas

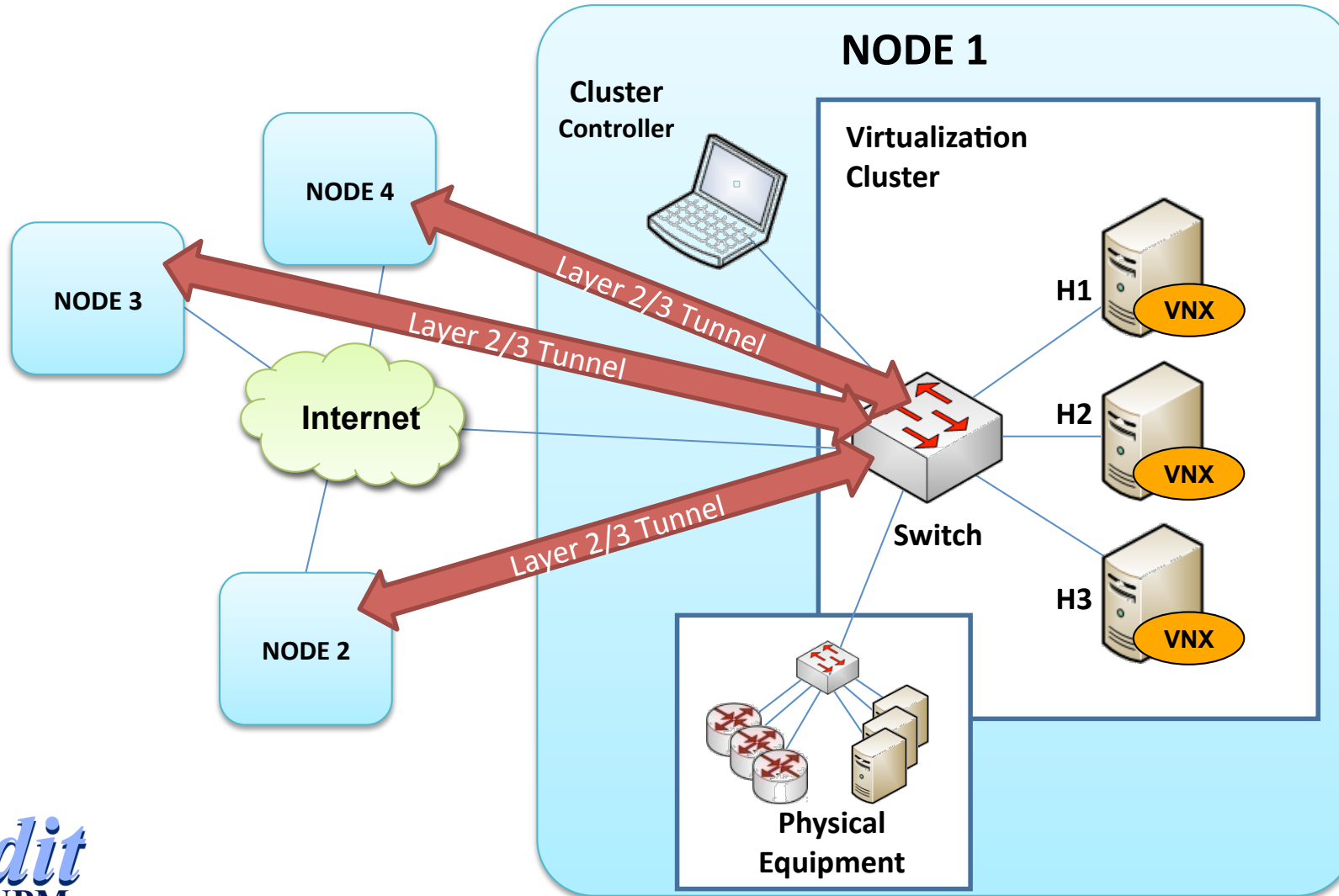
UPM

DIT-UPM, 2011

Trabajos futuros

- ◆ Seguridad: aplicaciones a la creación de redes señuelo (honeynets) dinámicas)
- ◆ Mejorar emulación de redes: Open vSwitch
- ◆ Clusters distribuidos y heterogéneos:
 - Mejorar la interconexión de clusters mediante bridges inteligentes (TRILL?)
 - Gestión de las capacidades de cada servidor
- ◆ Integración de nuevas plataformas de virtualización: VMware, Clouds
- ◆ Nuevos tipos de máquinas virtuales: Android
- ◆ Integración de equipos reales en escenarios
- ◆ Sistematización de pruebas sobre escenarios virtuales
- ◆ Mejorar la gestión de escenarios de pruebas (repositorios de escenarios, salvar estado, recuperar, etc)
- ◆ Interfaz de usuario gráfico

Escenario Futuro



Ideas PASITO

- ◆ Prueba de escenarios distribuidos sobre PASITO/ RedIRIS-Nova en contextos de investigación y/o docencia
 - Requisitos:
 - ✚ Linux instalado en los servidores
 - ✚ Acceso directo a las extensiones de virtualización; no es posible la doble virtualización
 - ✚ Conexión entre nodos a nivel 2 o mediante túneles

Agradecimientos

- ◆ A todos los que han participado directa o indirectamente en el desarrollo de VNUML/VNX/EDIV, en particular a Fermín Galán, principal desarrollador de VNUML
- ◆ A TID y la línea BOI por poner en marcha el proyecto EDIV
- ◆ A S21secs y el proyecto Secur@ por posibilitar el desarrollo de VNX
- ◆ A todos por vuestra atención



Virtual Networks over linux

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