



Creación de escenarios de red virtuales distribuidos

David Fernández

Dpto. Ingeniería de Sistemas Telemáticos

Universidad Politécnica de Madrid

david@dit.upm.es

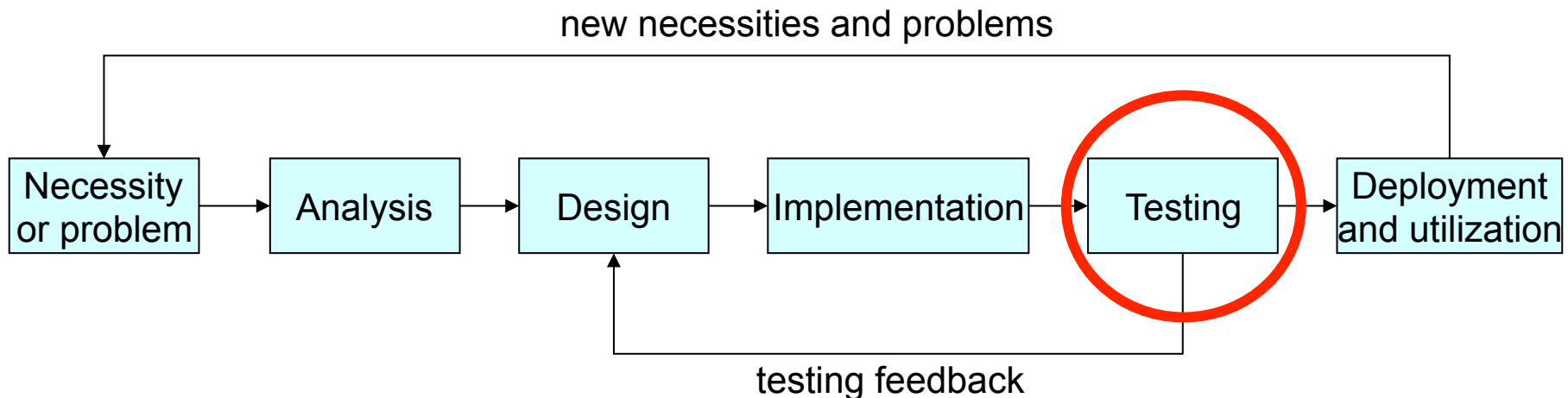
<http://www.dit.upm.es/vnx>

Contenido

- ◆ Objetivo
- ◆ Escenarios virtuales: VNX
- ◆ Despliegue distribuido: EDIV
- ◆ Conclusiones y evolución futura

Contexto

◆ Ciclo de vida de desarrollo de un sistema:

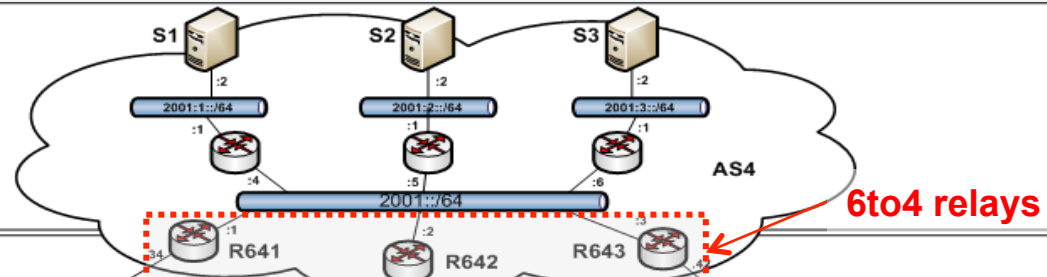


◆ Testbed de redes y servicios (definición informal)

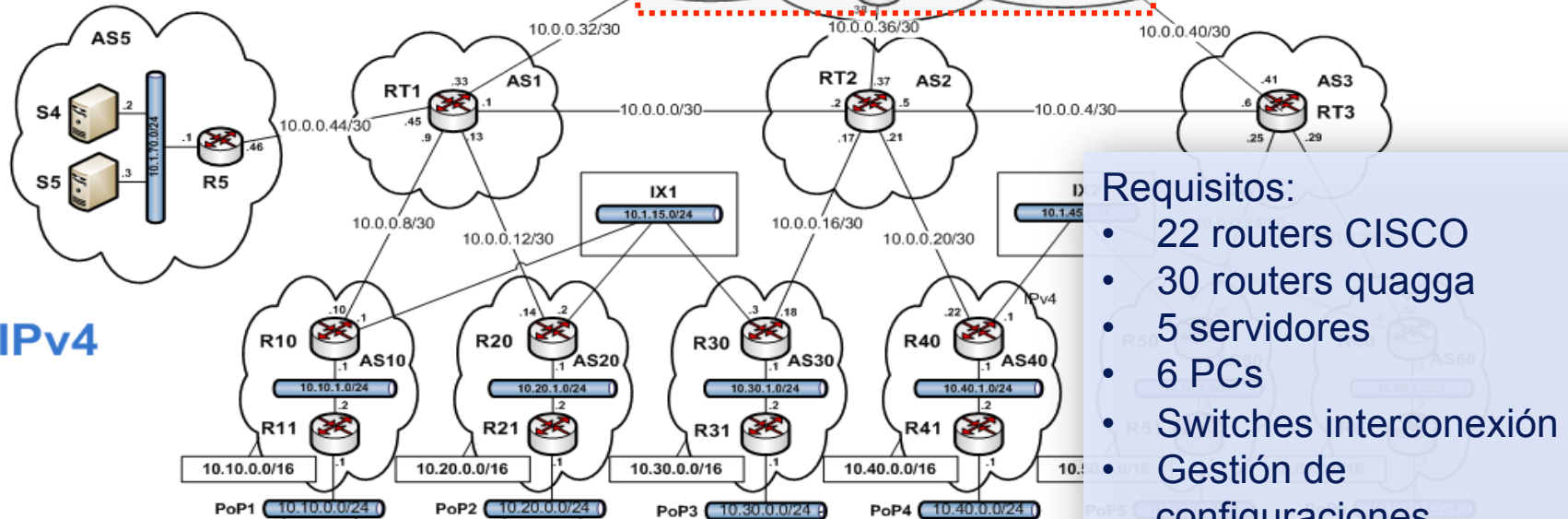
- Herramienta o sistema consistente en un entorno controlado de redes y sistemas, con semejanza a entornos del “mundo real”, donde los ingenieros prueban y evalúan sus implementaciones

Ejemplo escenario I: 6to4

IPv6 (no 6to4)



IPv4

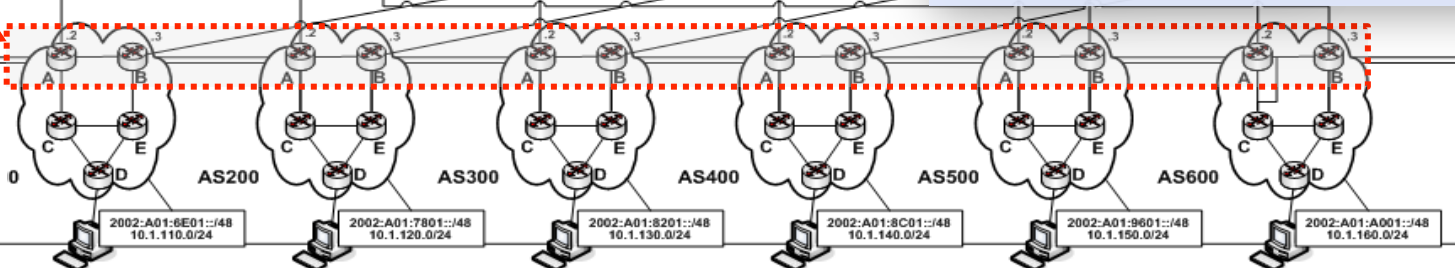


Requisitos:

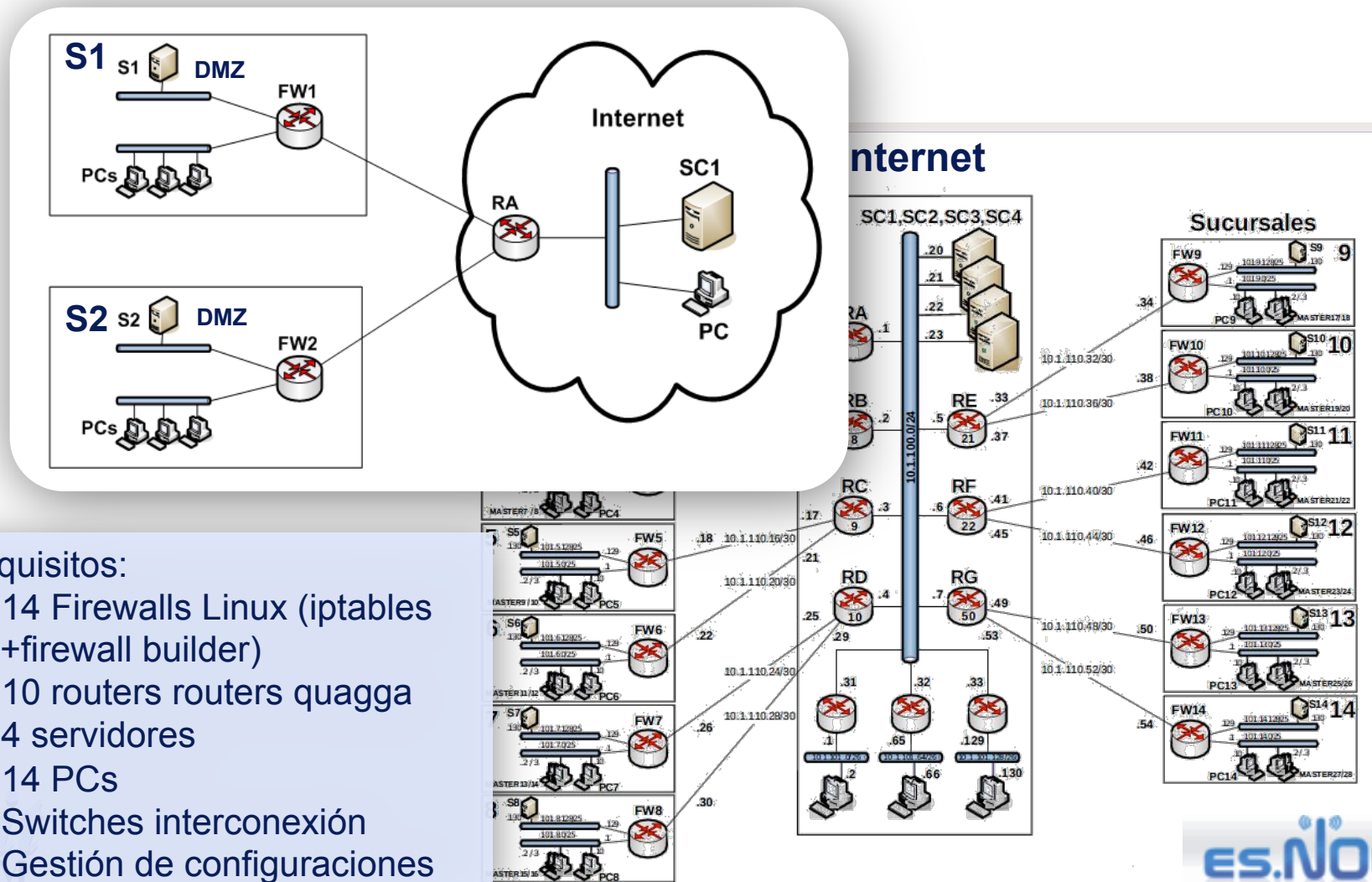
- 22 routers CISCO
- 30 routers quagga
- 5 servidores
- 6 PCs
- Switches interconexión
- Gestión de configuraciones

6to4 routers

IPv6 (6to4)
IPv4



Ejemplo escenario II: Cortafuegos

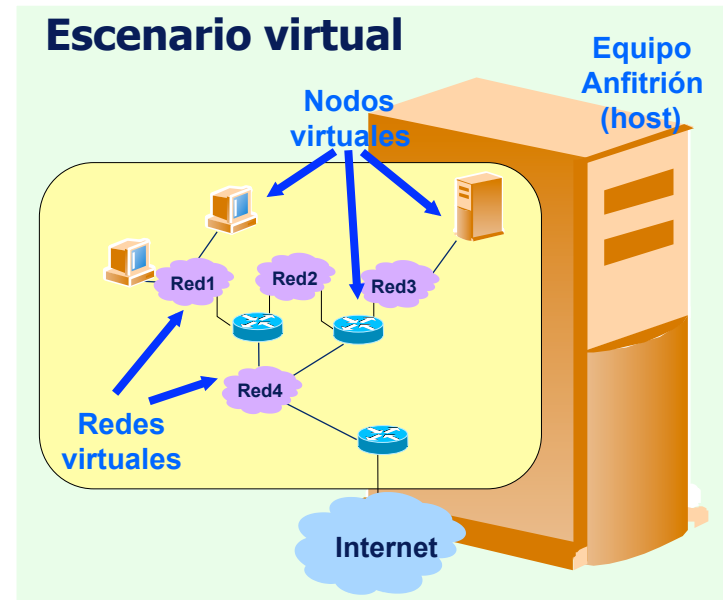
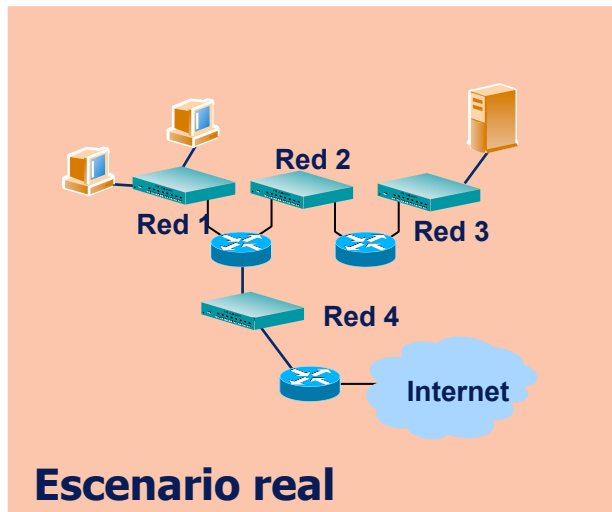


Requisitos:

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

Técnicas de Virtualización

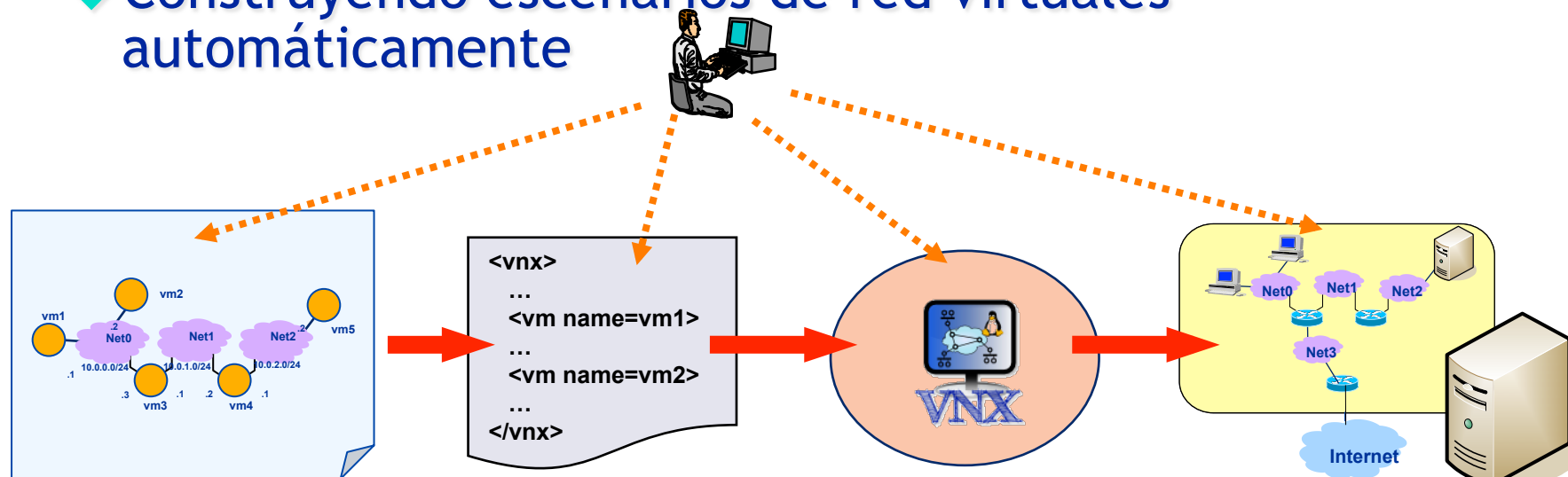
- ◆ Permiten ejecutar varias máquinas virtuales sobre un equipo anfitrión
 - Ej: Xen, VMware, User Mode Linux, etc
- ◆ Combinadas con el uso de redes virtuales emuladas en el host permiten crear **Escenarios de Red Virtuales** incluso con conexiones externas



- ◆ 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

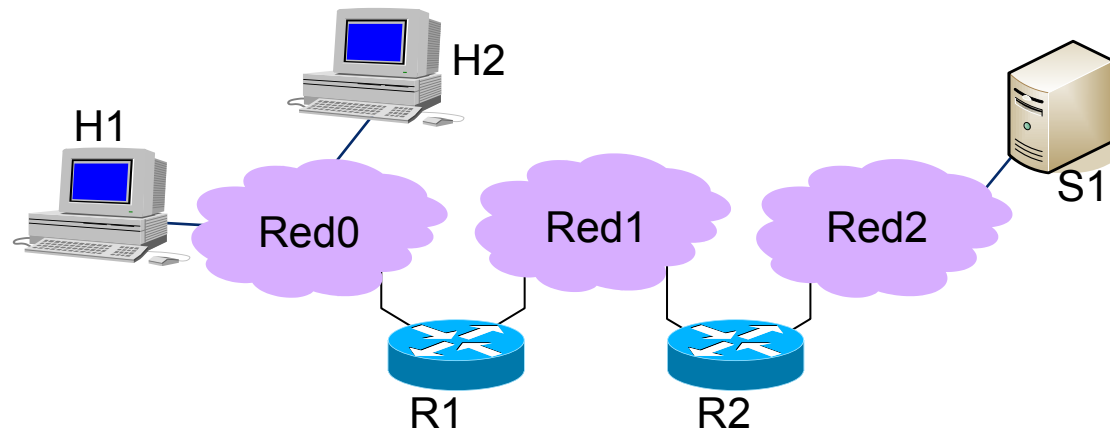
Creación

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>**)

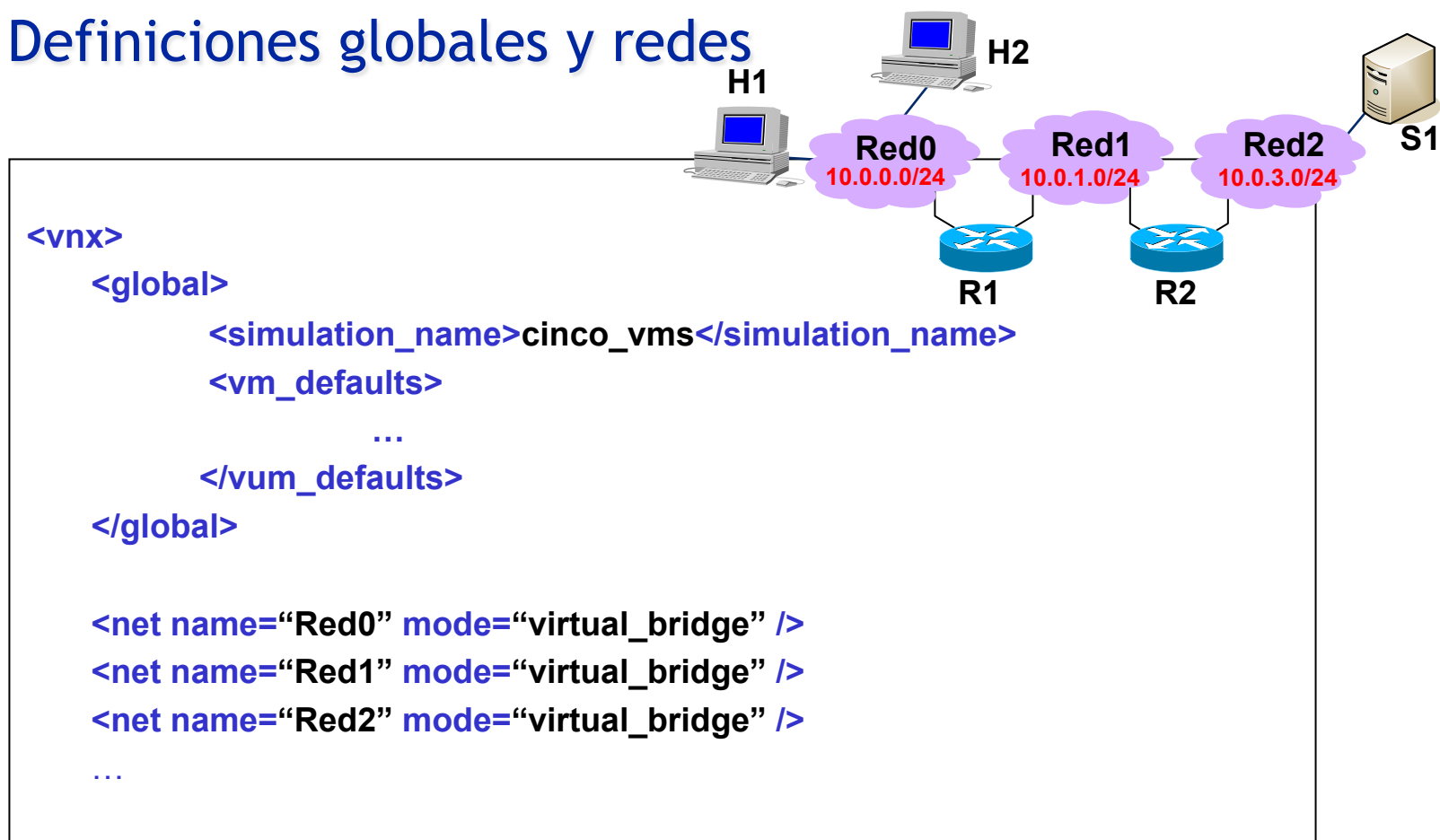
(definiciones de redes virtuales: **<net>**)

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

```
</vnx>
```

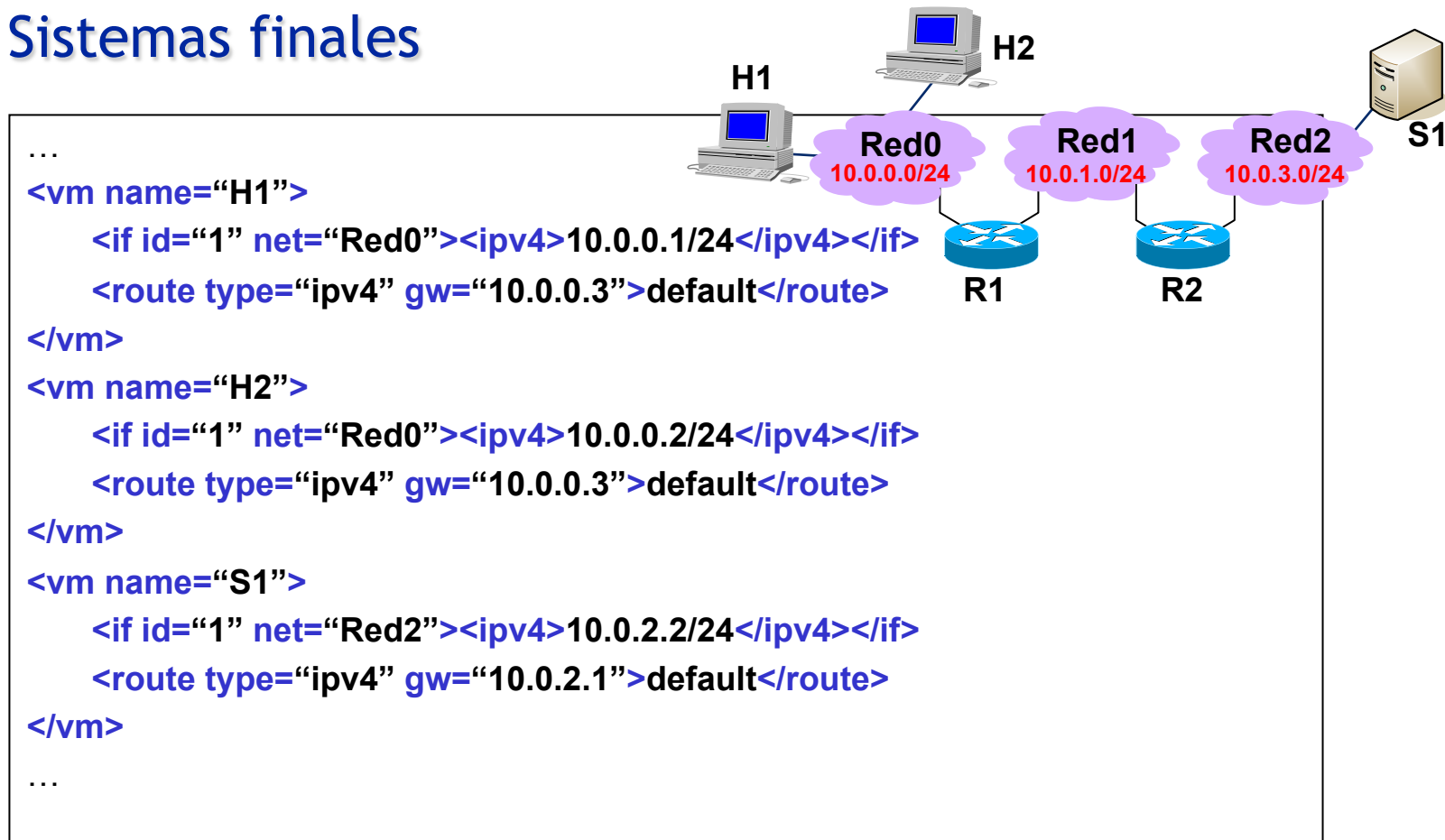

Lenguaje VNX (II)

◆ Definiciones globales y redes



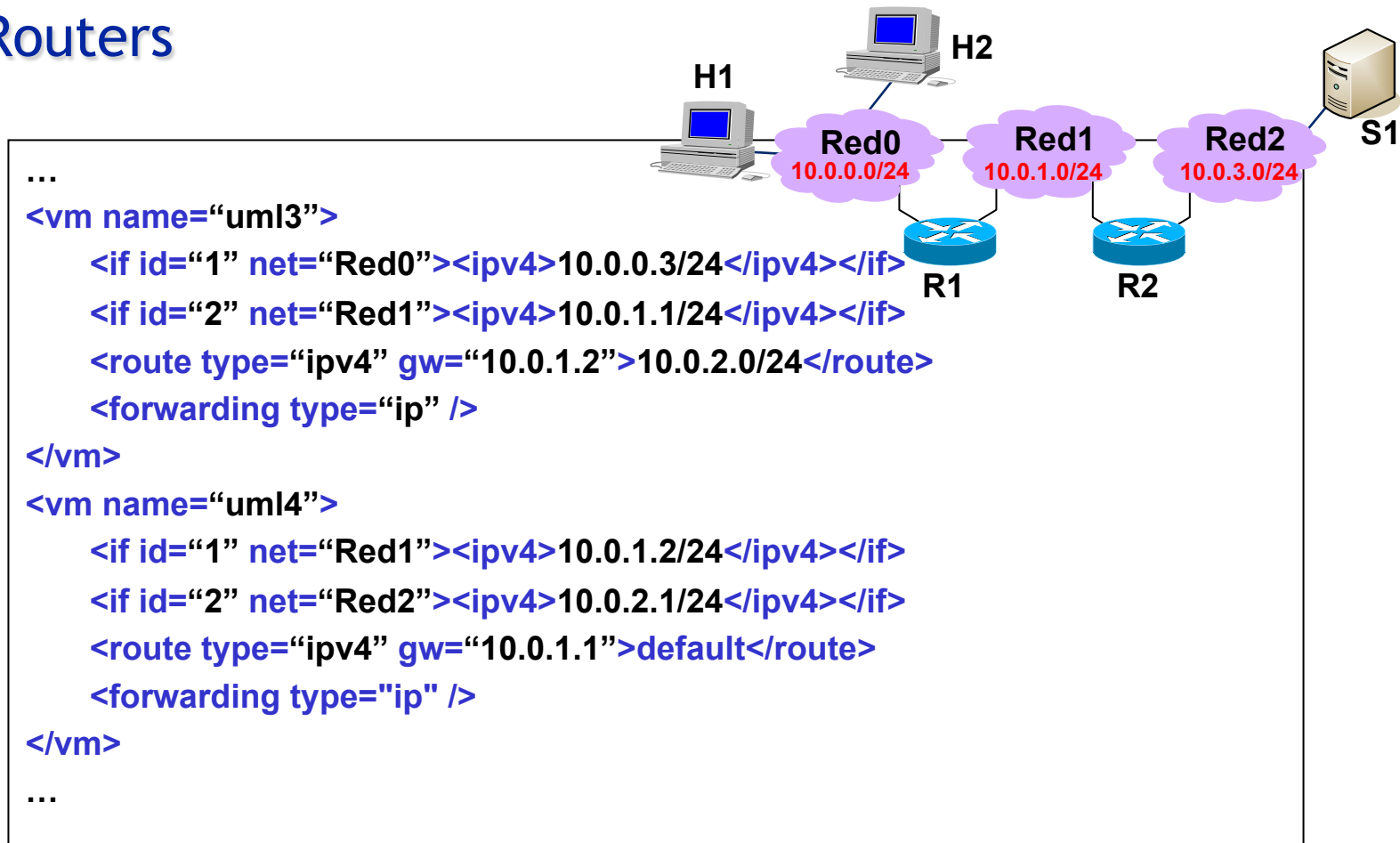
Lenguaje VNX (III)

◆ Sistemas finales



Lenguaje VNX (IV)

◆ Routers



Lenguaje VNX (V)

◆ Router Dynamips

```
<vm name="RT2" type="dynamips" subtype="3600" os="">
  <filesystem type="cow">/usr/share/vnx/filesystems/c3640</filesystem>
  <mem>96M</mem>
  <if id="1" net="rt1-rt2" name="e0/0">
    <ipv4>10.0.0.2/30</ipv4>
  </if>
  <if id="2" net="rt2-rt3" name="e0/2">
    <ipv4>10.0.0.5/30</ipv4>
  </if>
  ...
  <if id="5" net="rt2-r40" name="e1/1">
    <ipv4>10.0.0.21/30</ipv4>
  </if>
  <exec seq="brief" type="verbatim">show ip interface brief</exec>
  <exec seq="loadcfg7" type="verbatim" mode="telnet" ostype="load">merge
    conf/rt2-config-bgp4</exec>
</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), FreeBSD (8.1), Fedora 14
- 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

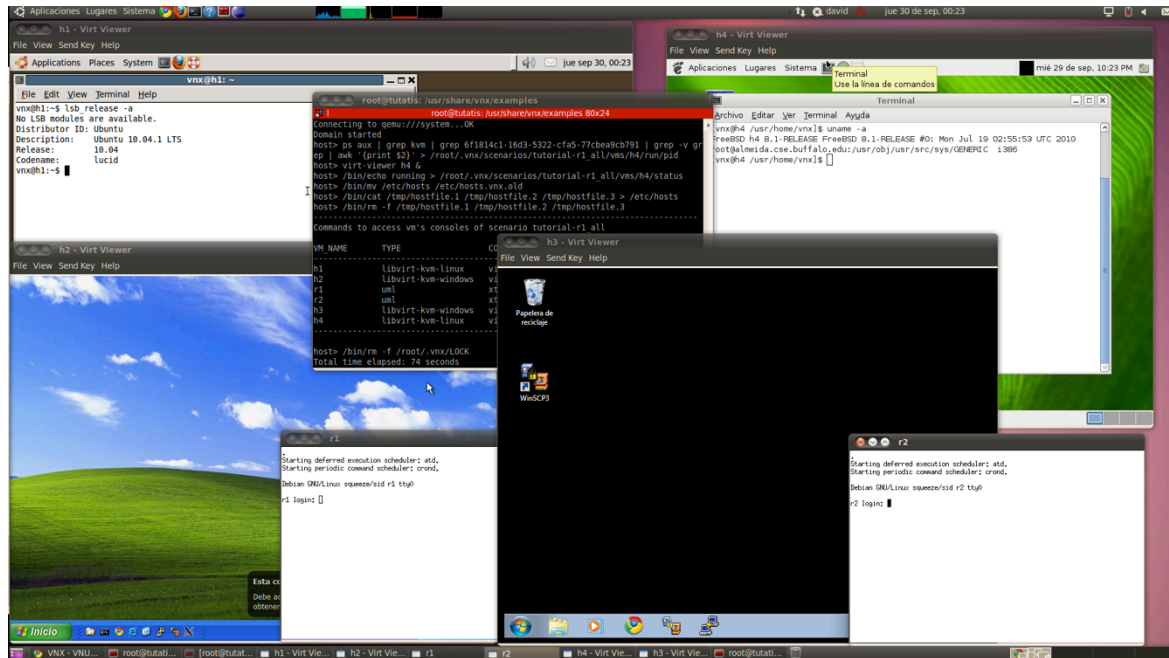
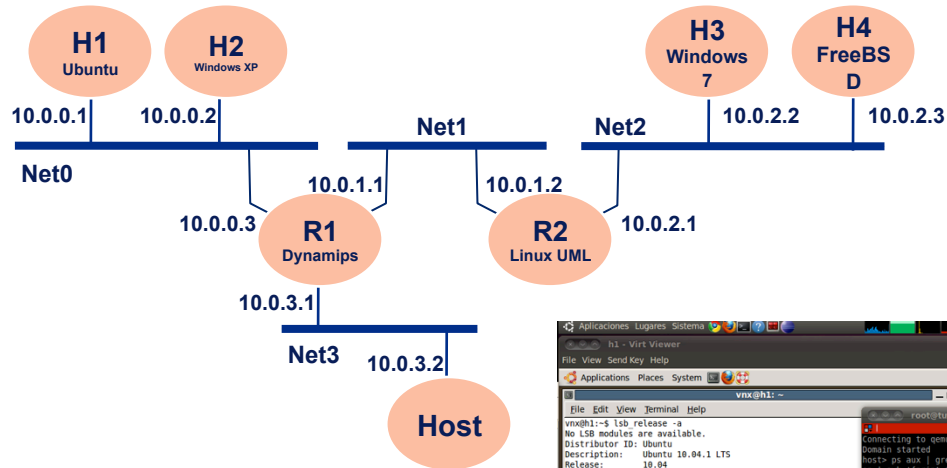
- ◆ Basada en discos (CDROM) creados dinámicamente y ofrecidos a las máquinas virtuales
 - Similar propuesta OVF
- ◆ Démonio de autoconfiguración y ejecución de comandos (ACED) corriendo dentro de las VMs
 - Lee ficheros xml y los traduce a comandos

```
<?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>
```

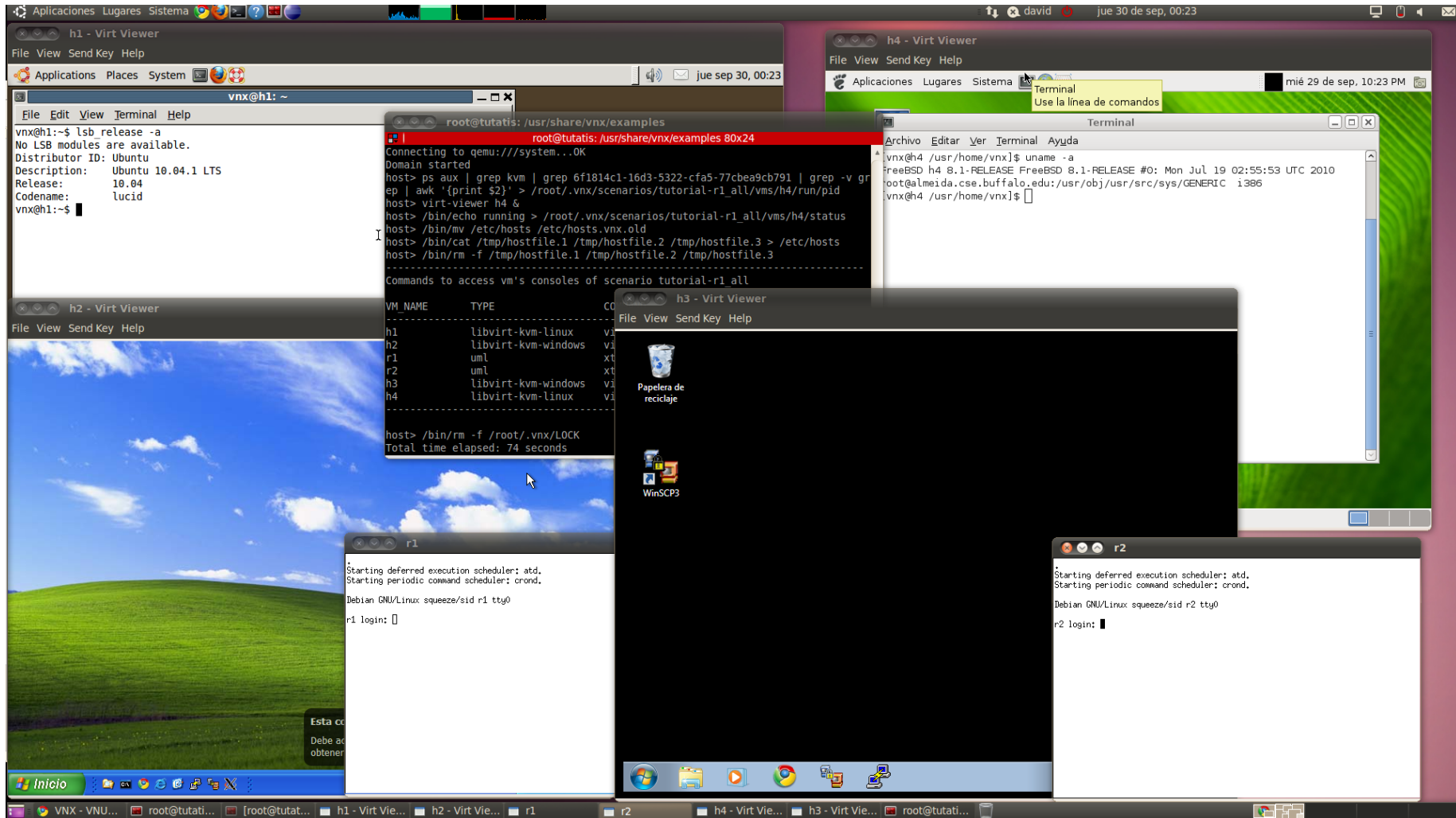
Ejemplos de uso

- ◆ 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



Captura escenario I: 6to4

The screenshot displays a Linux desktop with several open windows. The primary focus is on network configuration and a network diagram.

RT1 - console #1 (Terminal):
BGP table version is 7, local router ID is 10.0.0.45
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 | 10.0.0.14 | 0 | 0 | 50 | 3 2 1 10 i |
| * 10.20.0.0/16 | 10.0.0.10 | 0 | 0 | 40 | 2 1 10 i |
| * 10.30.0.0/16 | 10.0.0.2 | 0 | 0 | 3 | 2 1 10 i |
| * 10.40.0.0/16 | 10.0.0.14 | 0 | 0 | 50 | 3 2 30 20 i |
| * 10.50.0.0/16 | 10.0.0.14 | 0 | 0 | 40 | 2 30 20 i |
| * 10.60.0.0/16 | 10.0.0.2 | 0 | 0 | 3 | 2 30 i |

R60 - console #1 (Terminal):
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 | 10.1.45.2 | 0 | 50 | 3 2 30 20 i | |
| * 10.40.0.0/16 | 10.1.45.1 | 0 | 40 | 2 30 20 i | |
| * 10.50.0.0/16 | 10.0.0.29 | 0 | 3 | 2 30 i | |
| * 10.60.0.0/16 | 10.1.45.2 | 0 | 50 | 40 i | |
| * 110.60.0.0/16 | 10.1.45.1 | 0 | 40 | 50 i | |
| * 10.0.0.0/0 | 10.1.45.2 | 0 | 50 | 50 i | |
| * 10.0.0.0/0 | 10.60.1.2 | 0 | 100 | 0 i | |
| * 10.0.0.0/0 | 0.0.0.0 | 0 | 0 | 1 | |
| * 10.0.0.0/0 | 0.0.0.0 | 0 | 0 | 32768 i | |

Network Diagram (Irstii-p5.png):
A complex network diagram showing multiple routers (R1 through R10) interconnected in a mesh-like structure. The routers are represented by blue circles with labels. The diagram illustrates a multi-tier network topology.

Other Windows:
- **RT1 - console #1**: Shows a list of console windows for routers RT1 through RT3.
- **RA - console #1**: Shows a list of console windows for routers RA through RB.
- **Virtual Networks over Linux**: A window displaying technical information and a logo for the Universidad Politécnica de Madrid.

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

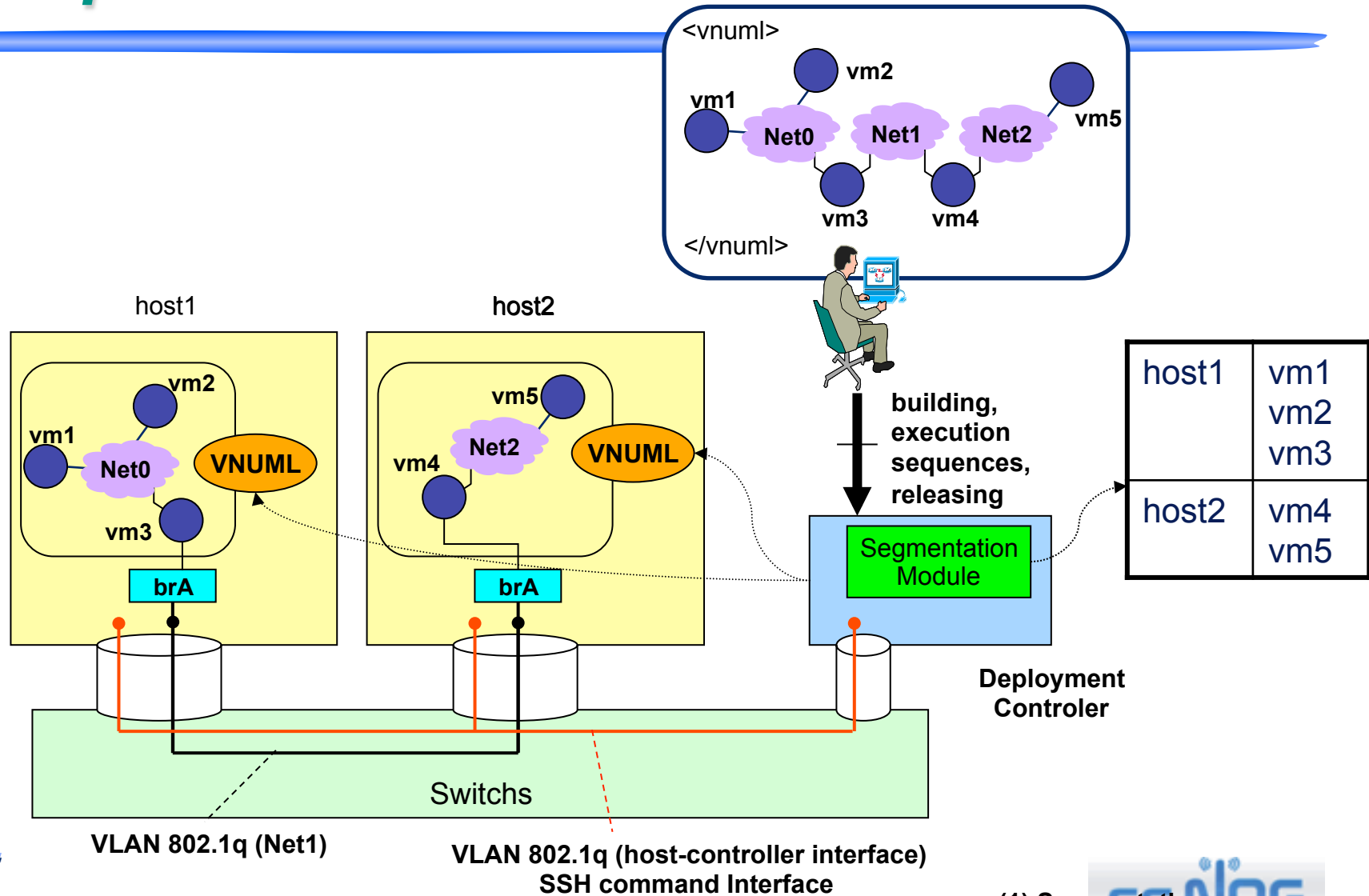
- ◆ Escenarios Virtuales con VNUML (EDIV)
- ◆ Colaboración entre TID y UPM en el contexto de la línea BOI (Business Oriented Infrastructure) dentro de la dirección de Sistemas de Apoyo a Negocio de TID.
- ◆ Objetivos:
 - Desarrollo de un prototipo distribuido de VNUML
 - ✦ Fermín Galán, David Fernández, "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)



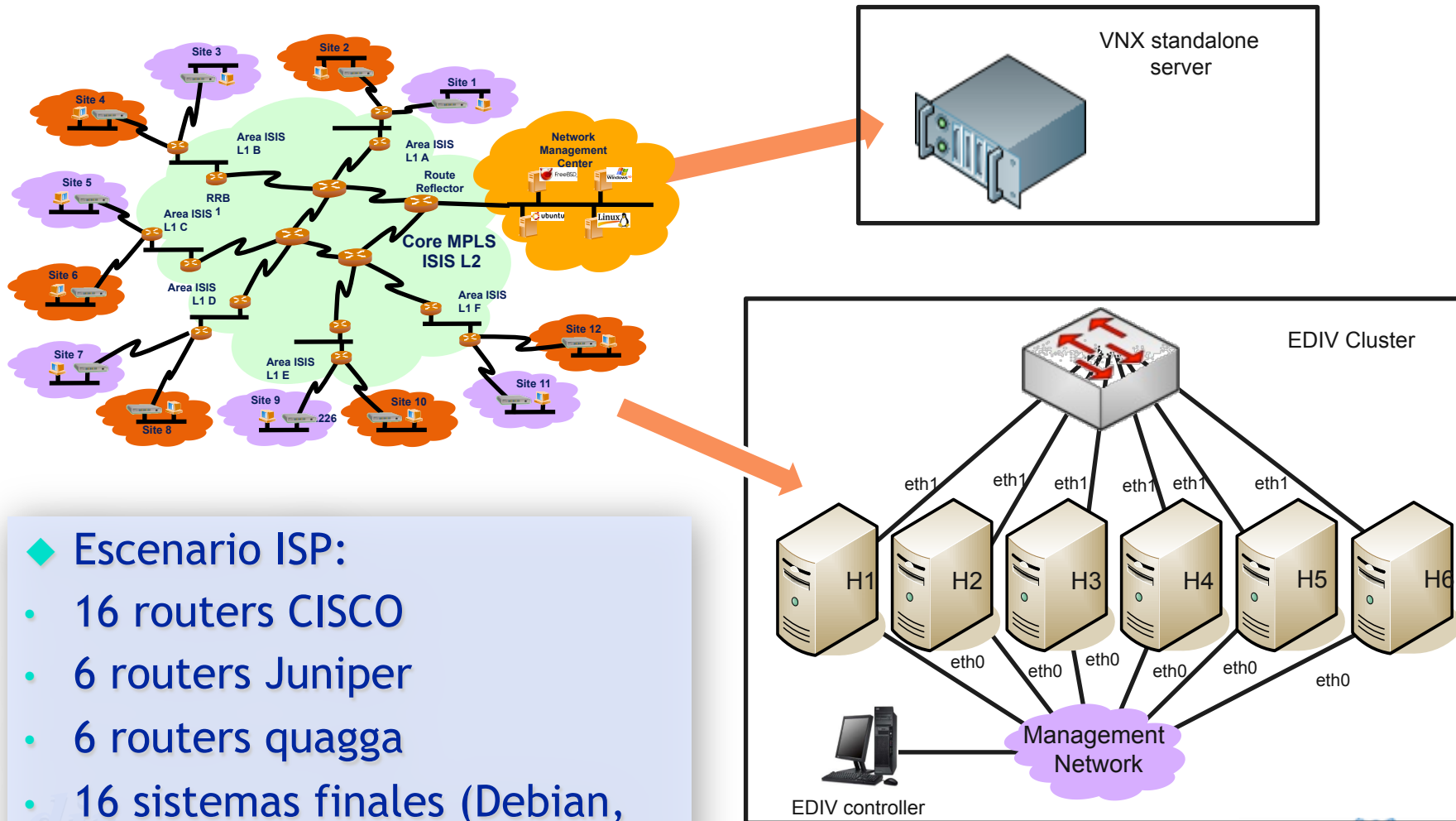
Arquitectura VNX distribuido: EDIV



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

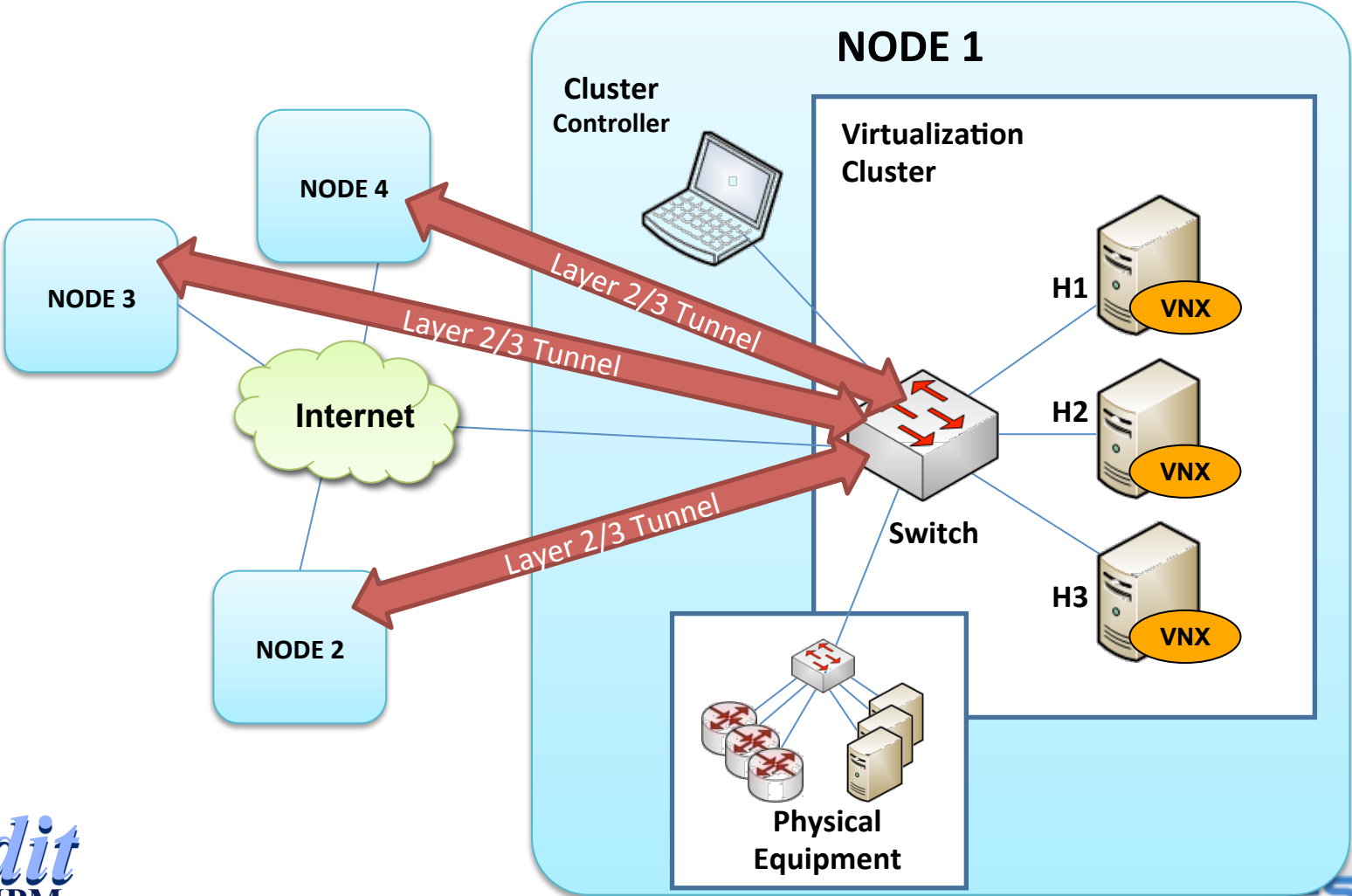
Conclusiones

- ◆ La utilización de técnicas de virtualización y, en particular, herramientas de virtualización de escenarios como VNX en infraestructuras de experimentación permite:
 - un ahorro de costes, tanto de equipamiento como esfuerzo invertido en su gestión y configuración
 - mejor compartición de infraestructuras de experimentación
 - crear y reutilizar escenarios complejos mediante un esfuerzo razonable
 - concentrar el esfuerzo en el sistema o servicio a probar y no en la infraestructura de pruebas

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



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 Segur@ por posibilitar el desarrollo de VNX
- ◆ A todos por vuestra atención



Virtual Networks over linux

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