

# Administración de sistemas II

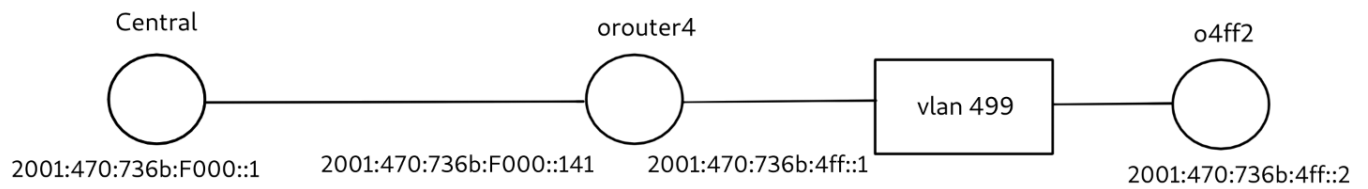
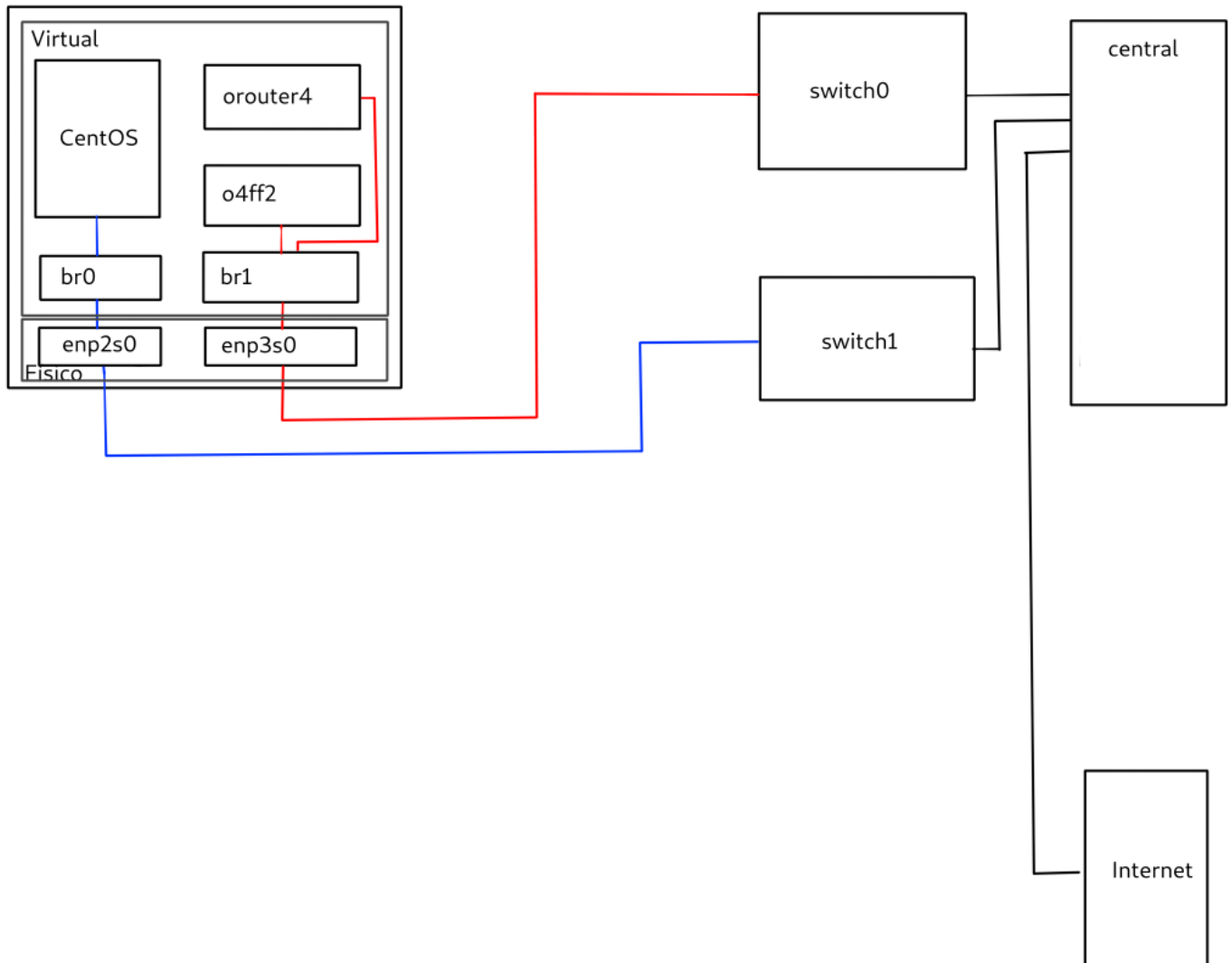


Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

## Práctica 1

# Mapa de red

PC Lab1.02



# Comandos útiles y pruebas de vida realizadas a lo largo de la práctica

## Comandos útiles

Para acceder desde la MV al lab:

```
ssh -6 a821259@fe80::e23c:19ee:bc23%vio0
```

```
scp -6 a821259@[fe80::e23c:19ee:bc23%vio0]:~/p.txt .
```

Para acceder desde lab a MV:

```
ssh -6 a821259@fe80::5054:ff:fe04:1101%br1
```

```
ssh -6 a821259@2001:470:736b:f000::141
```

Para realizar copias de seguridad de las qcow2 y de los xml desde local:

```
scp "a821259@central.cps.unizar.es:/misc/alumnos/as2/as22022/a821259/*{.qcow2,.xml}" .
```

## Pruebas de vida realizadas

- ☒ ping de router a central
- ☒ ping de router a mv interna
- ☒ ping de router a lab102
- ☒ ping de mv interna a router
- ☒ ping de mv interna a central
- ☐ ping de mv interna a lab102
- ☐ ping de lab102 a mv interna
- ☒ ping de central a mv interna

Los ping que no se han podido realizar han sido causados por el problema de la red del laboratorio y estudiados mediante el uso de tcpdump, aquí se adjunta una de las trazas. El error concreto se explica en el apartado de conclusiones.

```
orouter4$ doas tcpdump -r /root/traza.txt icmp6

15:52:37.770949 2001:470:736b:4ff:5054:ff:fe04:ff02 > 2001:470:736b:f000::1: icmp6: echo request
15:52:38.764180 2001:470:736b:f000::141 > 2001:470:736b:f000::1: icmp6: neighbor sol: who has
2001:470:736b:f000::1
15:52:38.764683 2001:470:736b:f000::1 > 2001:470:736b:f000::141: icmp6: neighbor adv: tgt is
2001:470:736b:f000::1

//Aquí se pierde el paquete :(
```

## Conclusiones de la práctica

El sistema operativo openbsd esta pensando para desempeñar labores de red, esto es evidente dadas las herramientas nativas para el manejo de redes como pueden ser el servicio rad, slaacd o la herramienta tcpdump entre otros.

Es cierto que debido a un error de configuración del laboratorio ha habido pruebas de vida que no se han podido realizar con éxito, concretamente las que salían de la máquina virtual interna de pruebas y sale de la red vlan privada a través de la máquina virtual router, de igual forma las pruebas inversas a las anteriores que van desde la red exterior a través del router hacía la máquina virtual interna a través de la vlan privada también fallan, ambos fallos se producen por la misma causa.

La causa de los errores mencionados es que en br1 no deja pasar el tráfico de forma completamente transparente como si fuese una conexión física a un switch, sino que a veces desvía paquetes, concretamente se aprecia en la siguiente secuencia el paquete desviado.

MV Interna -> Router

Router -> Central

Central -> Router

Router -/> MV Interna

En las distintas pruebas realizadas se puede visualizar como el paquete sale de la ip6 de la vlan privada del router y en cuenta de dirigirse a la ip6 de la vlan privada de la máquina virtual interna, se dirige a la ip6 del br1 de la máquina física del laboratorio en la que se encuentran hosteadas las máquinas virtuales.

También cabe resaltar la utilidad de las VLAN para aislar el tráfico distintas redes privadas dentro de un mismo segmento ethernet.

Otro detalle que cabe resaltar del desarrollo de la práctica es lo útil que resulta tener una imagen base con unas configuraciones estandar tales como la cuenta de usuario o la ssh key, además al trabajar con imagenes diferenciales ahorramos mucho espacio respecto al uso de imagenes completas.

Otro de los aprendizajes realizados a lo largo de la práctica es la importancia del uso tanto de copias de seguridad como de una guía de autodocumentación para en caso de ser necesario conocer los pasos realizados para rehacerlos o en su defecto deshacerlos. La guía de pasos realizados para autodocumentación se incluye en este archivo.

Otro aspecto que cabe la pena mencionar es que al emplear IPv6 es que resulta más tedioso buscar trazas de red concretas dado que este protocolo incluye mucho "ruido" en las comunicaciones.

# Pasos seguidos para crear y configurar las máquinas

## Configuración de la imagen base

\*Se realizan los pasos del 1 al 4 sin lugar a perdida o error\*

### Paso 5

Editamos el fichero /etc/hostname.vri0 para que su contenido sea:

```
up
inet6 ie64
```

Reiniciamos el servicio de red con el siguiente comando:

```
sh /etc/netstart
```

Para comprobar el correcto funcionamiento de la red ejecutamos el siguiente comando:

```
ping6 ${direcciónIp6DelBr1DelPCDelLab}%vio0
```

### Paso 6

Creamos el usuario con el comando:

```
adduser
```

\*(tampoco tiene lugar a perdida o error dejamos todos los ajustes en default, es bastante intuitivo)

A la hora de elegir el grupo del usaurio elegimos el grupo wheel

Para configurar correctamente los permisos del comando doas editamos el fichero /etc/doas.conf escribiendo en el lo siguiente:

```
permit nopass :wheel
```

### Paso 7

Ejecutamos los siguientes comandos

```
scp -6 a821259@[fe80::e23c:19ee:bc23%vio0]:~/ssh/id_ed25519.pub .
cat id_ed25519.pub >> /home/a821259/.ssh/authorized_keys
rm id_ed25519.pub
```

### Paso 8

Apagamos la máquina, ejecutamos un scp para descargar los archivos o4.xml y o4.qcow2 para guardar una copia de seguridad de los mismos, adicionalmente ejecutamos el comando

```
chmod g-w o4.qcow2
```

Para evitar modificar la imagen base por accidente

# Configuración del router

## Paso 1

Para crear la imagen diferencial empleamos el comando

```
qemu-img create -f qcow2 -o backing_file=o4.qcow2 orouter4.qcow2
```

*Se realizan los pasos 2 y 3 sin posible pérdida ni error*

## Paso 4

Definimos la MV con el siguiente comando:

```
virsh -c qemu:///system define orouter4.xml
```

## Paso 5

Escribimos en el fichero /etc/hostname.vio0 el siguiente contenido

```
up
net6 alias 2001:470:736b:f000::141 64
inet6 -temporary
```

Escribimos en el fichero /etc/mygate el siguiente contenido:

```
2001:470:736b:f000::1
```

## Paso 6

Copiamos fichero /etc/hostname.vio0 a otro con nombre /etc/hostname.vlan499 en su interior escribimos el siguiente contenido:

```
vlan 499 vlandev vio0
inet6 2001:470:736b:04ff::1 64
inet6 -temporary
```

## Paso 7

Escribimos en el fichero /etc/sysctl.conf el siguiente contenido:

```
net.inet6.ip6.forwarding=1
```

Para chequear si se han aplicado los cambios ejecutamos el comando:

```
sysctl net.inet6.ip6.forwarding
```

Es posible que para que los cambios efectuados hasta ahora surjan efecto haya que ejecutar el siguiente comando:

```
sh /etc/netstart
```

## Paso 8

Modificamos el fichero /etc/rad.conf con el siguiente contenido:

```
interface vlan499
```

Guardamos el fichero y ejecutamos el siguiente comando para activar el servicio rad:

```
rcctl enable rad
```

## Paso 9

Modificamos el nombre de la máquina editando el fichero /etc/myname con el siguiente contenido:

```
orouter4
```

\*los pasos 10,11,12 no tienen perdidad(además el paso 11 en caso de haber hecho bien la imagen base prácticamente no es necesario)

# Configuración de la máquina de pruebas

## Paso 13

Para crear la imagen diferencial empleamos el comando:

```
qemu-img create -f qcow2 -o backing_file=o4.qcow2 o4ff2.qcow2
```

El resto de acciones de este paso tampoco tienen pérdida, al igual que el paso 14.

## Paso 15

Editamos el contenido del fichero `/etc/hostname.vio0` y escribimos el siguiente:

```
up
-inet6
```

## Paso 16

Creamos el fichero `/etc/hostname.vlan499` y su contenido es el siguiente:

```
vlan 499 vlandev vio0 up
inet6 autoconf
inet6 -temporary
inet6 -soii
```

## Paso 18

En el fichero `/etc/myname` escribimos lo siguiente:

```
o4ff2
```

## Paso 19

Habilitamos el servicio `slaacd` mediante el comando:

```
rcctl enable slaacd
```

El servicio `slaacd` realiza la función de escuchar en busca de los avisos enviados por el servicio `rad` del router para asignar `ip6` automáticas a las interfaces que tienen el flag `autoconf`.

## Paso extra

Escribimos lo siguiente en `/etc/mygate`

```
2001:470:736b:4ff::1
```

## Paso 20

Ejecutamos el siguiente comando para reconfigurar todos los aspectos de red:

```
sh /etc/netstart
```