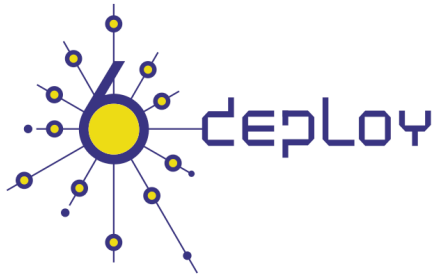
Lab BGP en IPv6

Simulación de un punto de intercambio de tráfico (IXP)



Workshop IPv6 – 8-10 de agosto 2011

Santiago de Chile

Carlos Martínez (carlos @ lacnic.net)

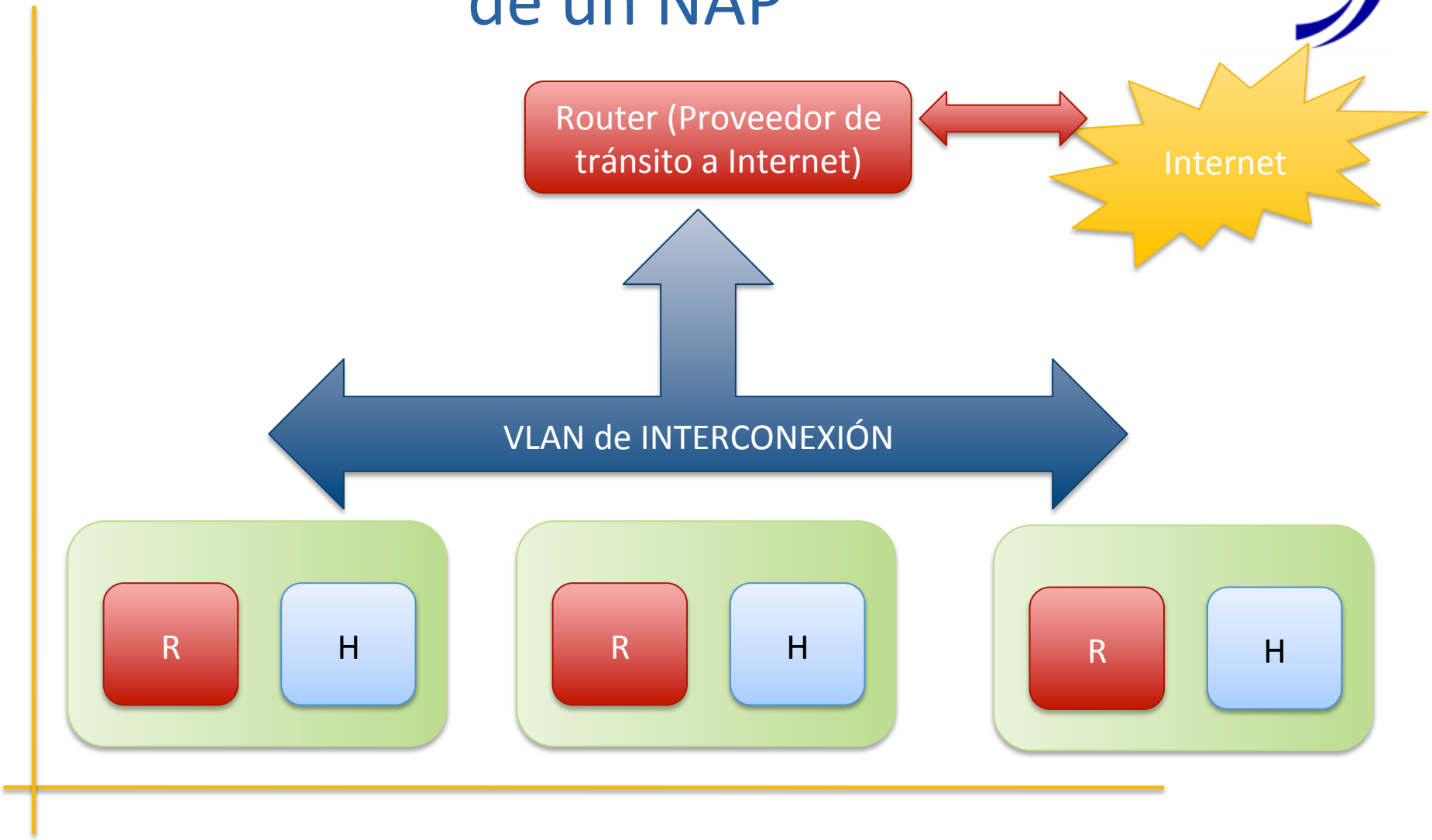
IXPs: Puntos de intercambio de tráfico

- Un punto de intercambio de tráfico (IXP) es un lugar donde de común acuerdo diferentes organizaciones deciden instalar infraestructura y establecer relaciones de intercambio de tráfico (*peering*)
- En el mundo
  - AMS-IX, LINX, NAP of the Americas
- En la región
  - NAP Colombia, NAP Ecuador, NAP CABASE (Buenos Aires), PTT Metro (Brasil)
  - Y varios muchos!

Ventajas de los NAPs

- Abaratan los costos de interconexión
  - La infraestructura ya esta instalada, solamente se negocian acuerdos de peering y cada actor va gestionando sus upgrades de ancho de banda
- Acercan el contenido a los usuarios
  - Evitando saltos de tránsito innecesarios
- Ademas, son el lugar ideal para
  - Instalar copias de los servidores raíz de DNS
  - Instalar caches de contenido (Google, Akamai, Limelight)

Diseño de nuestra implementación de un NAP



Conexión a Internet

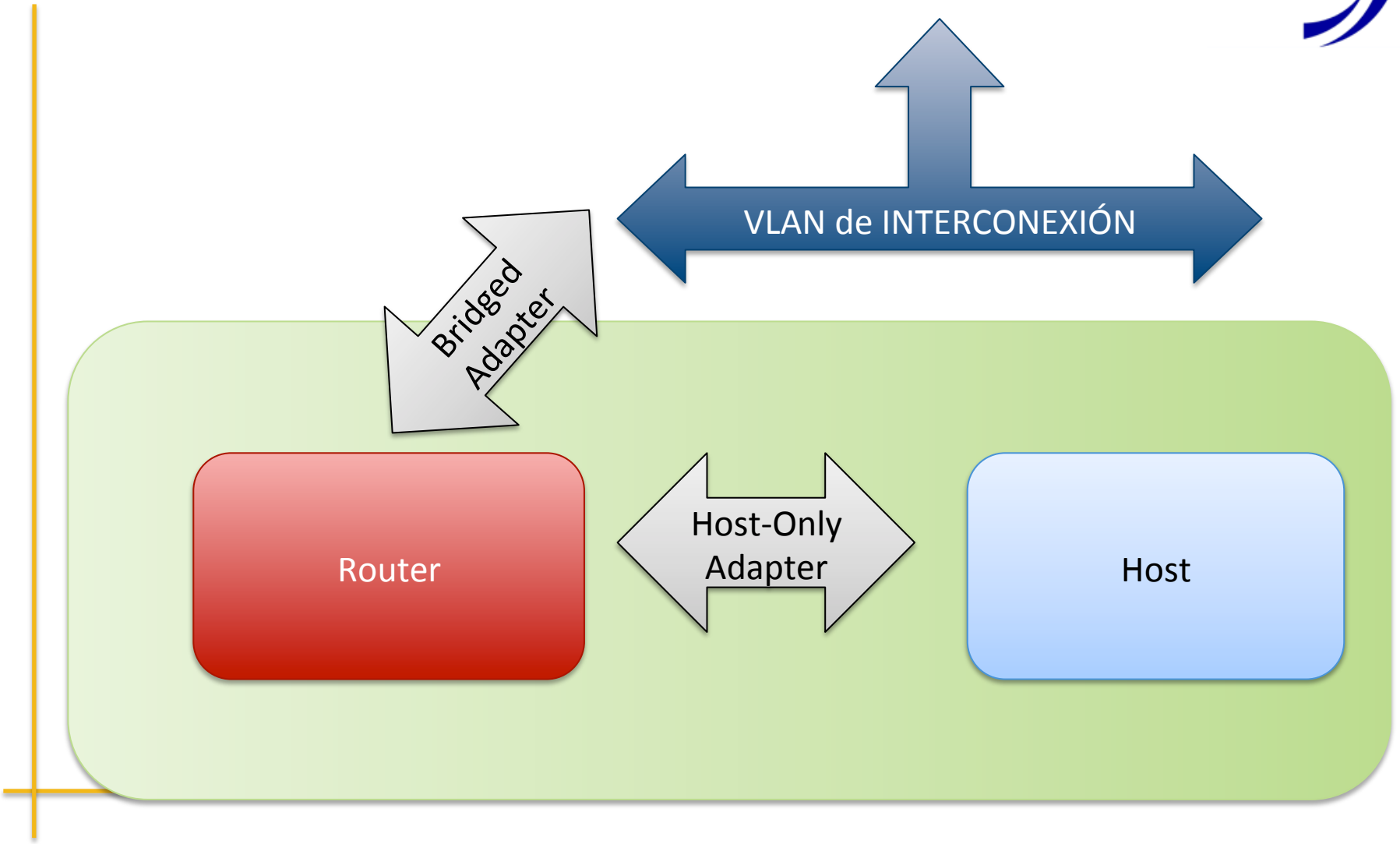
- El “Router de tránsito” implementa
  - Salida a Internet IPv4 a través de la red inalámbria de la universidad
  - Salida a Internet IPv6 vía un tunel six-in-four levantado contra Hurricane Electric (http://tunnelbroker.net)
- Hurricane Electric le asigna a los usuarios un prefijo /48
  - El prefijo es 2001:470:8aeb::/48
  - Cualquiera de ustedes puede utilizar este servicio!

BGP: Border Gateway Protocol

- Cada participante en un IXP utiliza un protocolo llamado BGP para intercambiar rutas con otros miembros del IXP

- Cada extremo de una sesion BGP se conoce como *peer* y por ello a las relaciones de intercambio de rutas se las llama *peerings*

Dentro de cada nodo del NAP



Software en cada nodo

- Vamos a utilizar (y a configurar)
  - Quagga
    - Implementación open source de BGP
    - Intercambio de rutas con el proveedor de transito
  - RADVD o DHCPv6 (opcion de cada nodo)
    - Servicios de autoconfiguración para el host

Procedimiento

- Paso 1
  - Importar la *virtual appliance* del router
- Paso 2
  - Configurar el networking del VirtualBox de acuerdo al diagrama anterior
    - Paso 2.1: configurarlo para el “Host”
    - Paso 2.2: configurarlo para el “Router”
- Paso 3
  - Configurar las direcciones estáticas y el radvd / DHCPv6 en cada nodo

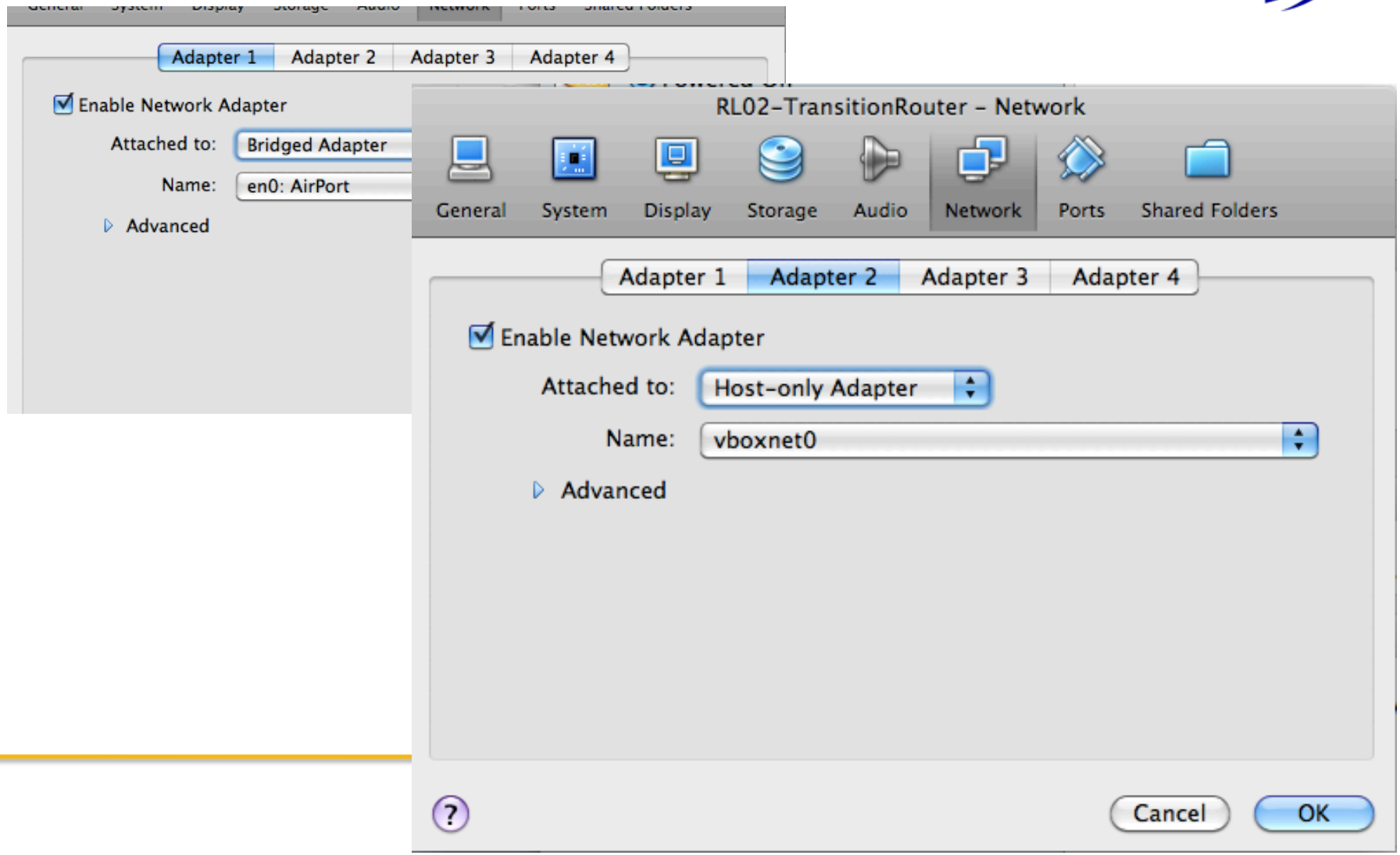
Procedimiento (2)

- Paso 4:
  - Configurar túnel punto-a-punto con el router central o de tránsito
- Paso 5:
  - Configurar el peering BGP con el router de tránsito a Internet

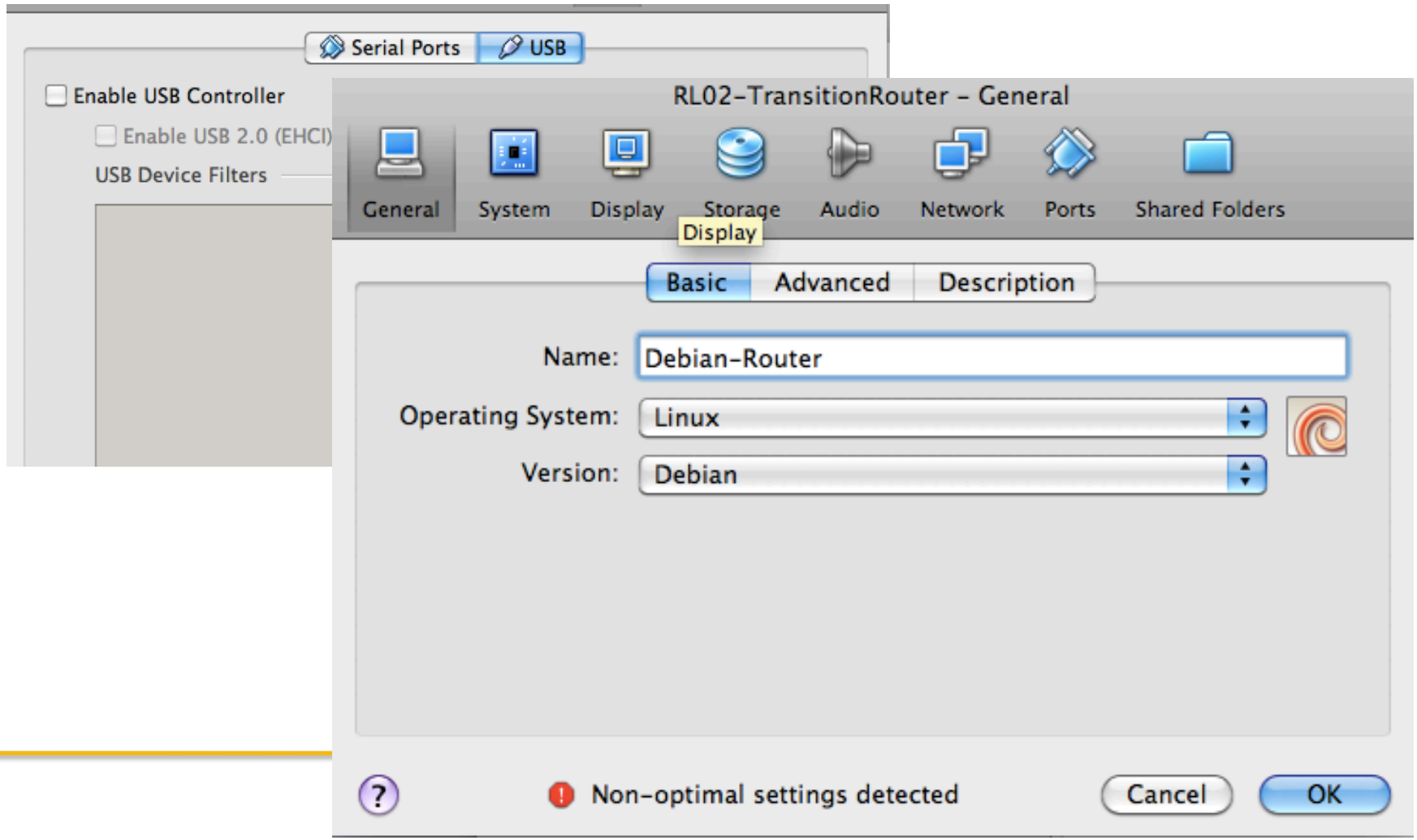
Configurar red en VB

- El router
  - Adaptador 1: “puente” con interfaz inalámbrica
  - Adaptador 2: “host-only” o “solo-anfitrión”
- El host
  - Adaptador 1: “host-only” o “solo-anfitrión”
    - Recordar que hasta hoy este adaptador estaba en modo puente

Configurar red en VB



Configurar red en VB



Recursos de numeración

- LAN de interconexión
  - 192.168.100.X/24
    - 192.168.100.1 es el router de tránsito
  - Cada grupo deriva la propia de acuerdo al número que tienen asignado
    - 3001::X/64 (X=101, 102,…)
- Prefijos públicos
  - 2001:470:8aeb:X::/64 (X=101, 102,…)
- Servidor DNS
  - 3001::1

Autoconfiguración para el host

- Se configurará el RADVD para autoconfigurar el host
  - Editar el /etc/radvd.conf
  - El prefijo a anunciar es el prefijo público de cada grupo
  - Incluir el anuncio del servidor DNS

Configuración de túnel

- Como nuestra hipotética LAN de interconexión no soporta IPv6, utilizaremos tuneles 6-in-4 para comunicar los routers de c/grupo con el central
- Parámetros:
  - Router central (IPv4): 192.168.100.1/24
  - Router grupo X (IPv4): 192.168.100.10+X/24

Configuración de BGP

- Editar
  - /etc/quagga/bgpd.conf
- Reiniciar
  - Ejecutar /etc/init.d/quagga restart

```
hostname router-lab
password zebra
enable password zebra
log stdout
!
router bgp 10i
 bgp router-id 10.0.1.1
 neighbor 3001::1 remote-as 1
 no neighbor 3001::1 activate
!
 address-family ipv6
 network 2001:470:8aeb:10i::/64
 neighbor 3001::1 activate
 exit-address-family
!
line vty
!
```

Verificación del funcionamiento de BGP (i)

- BGP funciona a través de una **sesión** que puede estar en diferentes estados
  - {up, idle, active}
- Lo primero que hay que verificar es que la sesión este levantada
- Ejecutar
  - vtysh –c “sh ipv6 bgp summary”
- Si la sesión no levanta, verificar la conectividad con el router de transición usando ping6
  - ping6 3001::1

Verificación del funcionamiento de BGP (ii)

- Si la sesión está levantada, entonces debemos verificar la tabla de enrutamiento de nuestro router
- Tabla tal como la ve Quagga:
  - vtysh –c “show ipv6 route”
- Tabla tal como la ve el kernel de Linux:
  - ip -6 route show

Verificación de la configuración de BGP (iii)

- Sesión caída o “idle”

```
root@stdrouter01:~# vtysh -c "show ipv6 bgp summary"
BGP router identifier 10.0.1.1, local AS number 5
RIB entries 1, using 64 bytes of memory
Peers 1, using 2520 bytes of memory

Neighbor        AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3001::1          4      1       0        0    0    0    0 never    Active
```

- Sesión levantada o “up”

```
root@stdrouter01:~# vtysh -c "show ipv6 bgp summary"
BGP router identifier 10.0.1.1, local AS number 5
RIB entries 1, using 64 bytes of memory
Peers 1, using 2520 bytes of memory

Neighbor        AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3001::1          4      1       0        0    0    0    0 never    00:03:35
```

Visualizando la tabla de enrutamiento

- Visualizando la tabla de Quagga

```
root@stdrouter01:~# vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
C>* 2001:470:8aeb:1::/64 is directly connected, eth1
C>* 3001::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
```

- Visualizando la tabla del kernel

```
root@stdrouter01:~# ip -6 route show
2001:470:8aeb:1::/64 dev eth1  proto kernel  metric 256  mtu 1500 advmss
1440 hoplimit 0
3001::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440 hoplimit
0
fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440 hoplimit
0
fe80::/64 dev eth1  proto kernel  metric 256  mtu 1500 advmss 1440 hoplimit
0
```

Verificación de la conectividad del host

- El host se debe auto-configurar por SLAAC
  - Incluyendo el DNS usando la opción de RDNSS
- Verificar
  - Que se haya autoconfigurado
    - Ifconfig eth0
  - Que el host tenga conectividad con el router local
    - ping6 2001:470:8aeb:X::1
  - Que el host tenga conectividad con el router de tránsito
    - ping6 3001::1

El objetivo final

- El objetivo final es poder conectarnos a Internet IPv6 desde la máquina virtual “host”
- En el entorno gráfico tienen un Firefox que pueden utilizar para navegar
- *NO* olviden, antes de correr hacia el navegador, verificar la conectividad con ping6
  - Primero lo primero ☺

Sitios accesibles y verificables

- Estos sitios se pueden acceder por IPv6 y también se pueden verificar con ping6
  - ipv6.google.com
  - www.v6.facebook.com
  - www.sixxs.net
  - www.consulintel.es
  - www.lacnic.net

¡A Trabajar!

carlos @ lacnic.net