Práctica 1: IPv6

Redes de Ordenadores para Robots y Máquinas Inteligentes

GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Febrero de 2024

1. Funcionamiento básico de IPv6

Para la realización de los siguientes ejercicios es necesario descomprimir el fichero `IPv6-lab.tgz` que descargarás de la siguiente página:

`https://mobiquo.gsyc.urjc.es/practicas/ror/p1.html`

Al descomprimir este fichero se generará un directorio `IPv6-lab` con los archivos de configuración de esta práctica necesarios para NetGUI.

Al arrancar NetGUI, debes abrir el escenario definido dentro del directorio `IPv6-lab`. Este escenario es el que se muestra en la figura 1.

1.1. Autoconfiguración de direcciones IPv6 (*link-local*)

Para empezar arranca únicamente `pc1`.

1. Indica cuál es la dirección IPv6 link-local que se ha configurado en `pc1`, y su relación con su dirección Ethernet.
2. Indica a qué dirección IPv6 multicast de nodo solicitado pertenece `pc1`.

Arranca `tcpdump` en `pc1` para que capture paquetes y guarda la captura en el fichero `ipv6-01.cap`. Arranca `pc2`.

3. Indica cuál es la dirección IPv6 link-local que se ha configurado en `pc2`, y su relación con su dirección Ethernet.
4. Indica a qué dirección IPv6 multicast de nodo solicitado pertenece `pc2`.
5. Interrumpe la captura que estabas realizando en `pc1`. Carga la captura en `wireshark` y localiza el mensaje enviado por `pc2` que indica que `pc2` está detectando si existen direcciones IPv6 duplicadas con su dirección link-local.
6. Fíjate en las direcciones IPv6 y en las direcciones Ethernet que lleva este mensaje. Explica si la máquina `pc1` recibe y procesa ese mensaje (aunque no responda).

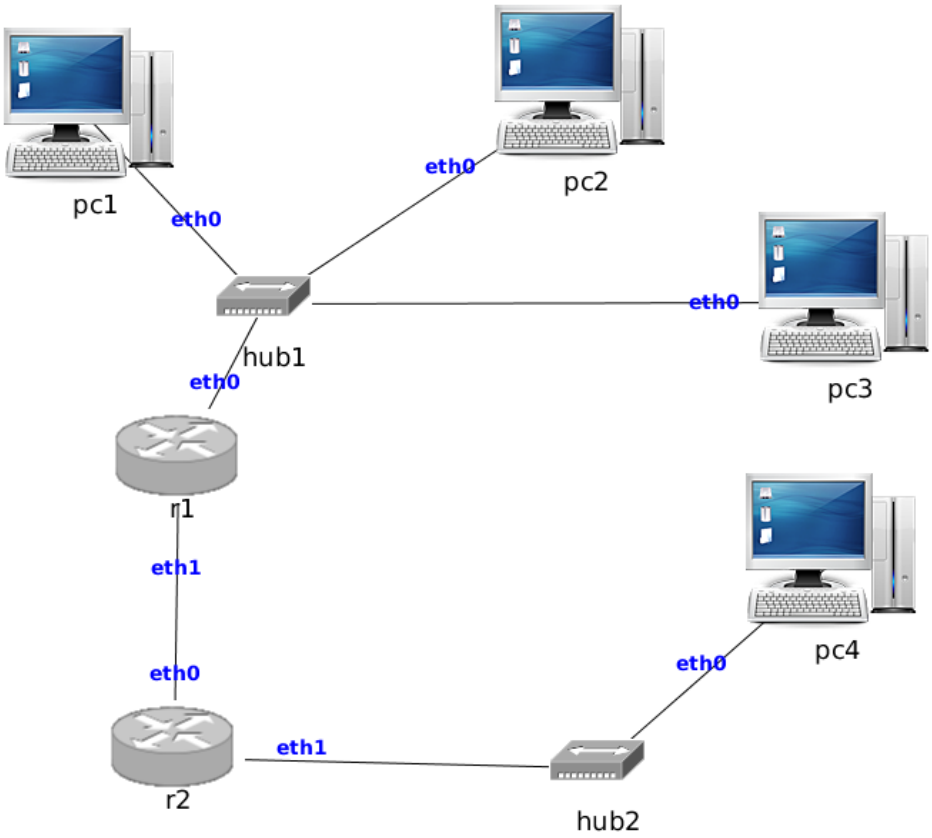

Figura 1: Escenario de IPv6

7. Localiza los mensajes ICMPv6 Multicast Listener Report e indica cuál crees que es su propósito.

8. Explica los mensajes ICMPv6 Router Solicitation que observas en la captura y explica su contenido y su propósito.

Arranca `tcpdump` en `pc1` para que capture paquetes y guarda la captura en el fichero `ipv6-02.cap`. Arranca `pc3`.

8. Indica cuál es la dirección IPv6 link-local que se ha configurado en `pc3`, y su relación con su dirección Ethernet.

9. Indica a qué dirección IPv6 multicast de nodo solicitado pertenece `pc3`.

10. Interrumpe la captura que estabas realizando en `pc1`. Carga la captura en `wireshark` y localiza el mensaje enviado por `pc3` que indica que `pc3` está detectando si existen direcciones IPv6 duplicadas con su dirección link-local.

11. Fíjate en las direcciones IPv6 y en las direcciones Ethernet que lleva este mensaje. Indica si las máquinas `pc1` y/o `pc2` reciben y procesan este mensaje (respondan o no).

12. Observa en la captura si `pc1` o `pc2` responden al mensaje enviado por `pc3`, y explica por qué.

1.2. Tráfico IPv6 entre 2 máquinas directamente conectadas

1. Comprueba con el comando `route` las rutas IPv6 que tiene configuradas las máquinas `pc1`, `pc2` y `pc3` y explica el significado de las mismas.

2. Ejecuta `tcpdump` en `pc3` (guardando los paquetes en un fichero `ipv6-03.cap`) y realiza un `ping6` desde `pc1` a la dirección link-local de `pc2`.

3. Comprueba que funciona desde `pc2` el `ping6` a la dirección IPv6 multicast de nodo solicitado de `pc1`. Explica la respuesta que obtienes.

4. Comprueba que funciona desde `pc1` el `ping6` a la dirección IPv6 multicast de nodo solicitado de `pc2`. Explica la respuesta que obtienes.

5. Comprueba que funciona desde `pc1` el `ping6` a la dirección IPv6 multicast de todos los nodos del enlace. Explica la respuesta que obtienes.

6. Interrumpe la captura.

7. Localiza en la captura todos los mensajes de *Neighbor Solicitation*. Identifica en ellos qué máquina los envía, explica la causa por la que los envía. Fíjate en la dirección Ethernet de destino de dichos mensajes y explica su valor. ¿Qué máquinas recibirán cada uno de esos mensajes de *Neighbor Solicitation*?

8. Comprueba que tras la realización del `ping6`, las direcciones Ethernet de máquinas vecinas que han aprendido `pc1` y `pc2`, mostrando la información de su caché de vecinos. Observa cuándo la información contenida cambia de estado y/o desaparece.

   NOTA: Ten en cuenta que la caché de vecinos de IPv6 en Linux tiene menor tiempo por defecto que la caché de ARP en IPv4. Prueba a utilizar `watch -n 1` para repetir automáticamente el comando de consulta de la caché de vecinos, y repite el `ping6` entre máquinas para ver mejor las transiciones entre estados.

9. Comprueba qué direcciones aprende `pc3` en su caché de vecinos tras todo el tráfico anterior.

1.3. Autoconfiguración de direcciones IPv6 globales

Arranca la máquina `pc4`, pero todavía no arranques los *routers* `r1` y `r2`.

Los *routers* `r1` y `r2` tienen configurado en el protocolo ICMPv6 el envío de mensajes *Router Advertisement*. Nada más arrancar, estos *routers* mandan mensajes *ICMPv6 Router Advertisement* que contienen anuncios de los prefijos de subred a los que pertenecen sus interfaces. De esta forma, las máquinas que estén directamente conectadas a dichas interfaces podrán configurar su dirección IPv6 en función de los anuncios que reciban.

Arranca (en *background*) una captura en `pc4` y guárdala en un fichero `ipv6-04.cap`.

1. Indica qué direcciones y rutas ha configurado `pc4`.

Arranca `r2`.

2. Indica qué direcciones y rutas tiene ahora configuradas `pc4`.

3. Interrumpe la captura en `pc4` y explica los mensajes que observas en dicha captura. Fíjate en las direcciones IPv6 origen y destino de cada paquete. Explica el sentido del último mensaje que aparece en la captura que NO es un *Router Advertisement*.

4. Muestra las direcciones de vecinos aprendidas por `r2` y `pc4` y justifica tu respuesta.

5. Indica los valores *Valid Lifetime (valid_lft)* y *Preferred Lifetime (preferred_lft)* de la dirección IPv6 global que se ha configurado en `pc4`. ¿De dónde los ha tomado `pc4`?. Relaciona estos valores con los intervalos entre mensajes *Router Advertisement* que se ven en la captura.

6. Interrumpe la ejecución del demonio `radvd` en `r2`.

   Indica qué ocurre con los valores *valid_lft* y *preferred_lft.* en `pc4`. Indica también qué ocurre con la dirección IPv6 global que se había configurado en `pc4`, y en cuánto tiempo. Muestra las direcciones de vecinos aprendidas por `pc4` y justifica tu respuesta.

Inicia en `r2` el protocolo *Router Advertisement* y arranca `r1`.

7. Indica qué direcciones IPv6 globales se han configurado en `pc1`, `pc2` y `pc3`.

8. Indica qué rutas IPv6 se han configurado en `pc1`, `pc2` y `pc3`. Ejecuta repetidas veces en uno de los pcs el comando que visualiza las rutas y fíjate en lo que ocurre con el campo *expires* y trata de explicarlo. ¿De donde toma el pc ese valor?

9. Explica qué ocurre si haces un `ping6` entre dos máquinas que no están directamente conectadas, por ejemplo, `pc1` y `pc4`, o entre `pc1` y `r2`. Para entenderlo consulta las rutas en las máquinas, y haz capturas en las interfaces que necesites.

10. Haz un `ping` desde `pc1` a la IPv6 destino `ff02::2`. ¿Quién responde? Justifica la respuesta.

1.4. IPv6 entre 2 máquinas de subredes diferentes

Los pcs tienen configuradas rutas por defecto, pero los *routers* sólo tienen configurada ruta hacia máquinas vecinas. Como habrás comprobado en el apartado anterior, para que las máquinas de diferentes subredes puedan intercambiar tráfico es necesario añadir rutas en los *routers*.

1. Añade las rutas que consideres necesarias para que todas las máquinas de la figura puedan intercambiar tráfico entre ellas. Indica qué rutas has configurado.

2. Arranca un captura en alguna de las máquinas conectadas al `hub1` y guárdala en un fichero `ipv6-05.cap`. Realiza un `ping6` de `pc1` a `pc4` y otro de `pc1` a la dirección global de la interfaz `eth0` de `r2`. Interrumpe la captura y comprueba el fichero de captura. Observa las direcciones Ethernet e IP de los mensajes capturados, y el valor del *hop limit*.

2. Fragmentación en IPv6

Para analizar la cabecera de extensión para la fragmentación en IPv6 vamos a provocar que sea necesario fragmentar los datagramas IPv6.

En IPv6 sólo puede fragmentar la máquina que crea un datagrama y por tanto, no pueden fragmentar los routers intermedios que hay entre el origen y el destino (en IPv4 los routers intermedios sí pueden fragmentar). NOTA: Ten en cuenta que los tamaños de los fragmentos de IPv6 deben ser un múltiplo de 8 bytes, salvo el último (igual que en IPv4).

El valor de MTU por defecto en Ethernet es 1500 bytes (puedes comprobarlo con el comando `ip -6 addr`).

Realiza una captura de tráfico en `r1(eth0)` (fichero ipv6-06.cap) y en `r2(eth0)` (fichero ipv6-07.cap).

Ejecuta un `ping6` desde `pc1` a la dirección global de `pc4` con la opción `-s 2000` obligando a que los paquetes de `ICMPv6 echo request` tengan 2000 bytes de datos, provocando un tamaño de datagrama IPv6 mayor que la MTU de Ethernet.

Interrumpe las capturas y estúdialas.

1. Explica qué máquina ha fragmentado los datagramas y cómo sabe a qué tamaño máximo debe hacerlo.

2. Estudia los valores de las cabeceras `Next Header` cuando un datagrama se fragmenta, y trata de comprobar los tamaños de los fragmentos y el tamaño del datagrama original sin fragmentar que se quería enviar.

Ahora vamos a modificar el valor de la MTU entre `r1` y `r2` para que sea 1304 bytes (en vez de los 1500 típos de Ethernet). Para ello ejecuta el siguiente comando en `r1` para modificar el valor de MTU en su interfaz `eth1`:

```
r1:~# ip link set eth1 mtu 1304
```

Y ejecuta el siguiente comando en `r2` para modificar el valor de MTU en su interfaz `eth0`:

```
r2:~# ip link set eth0 mtu 1304
```

Realiza una captura de tráfico en `r1(eth0)` (fichero ipv6-08.cap) y en `r2(eth0)` (fichero ipv6-09.cap). Ejecuta un `ping6` desde `pc1` a la dirección global de `pc4` con la opción `-s 1400`.

Interrumpe las capturas y estúdialas.

3. Explica qué máquina ha fragmentado los datagramas y cómo sabe a qué tamaño debe hacerlo. Trata de comprobar los tamaños de los fragmentos y el tamaño del datagrama original sin fragmentar que se quería enviar.

4. Explica la diferencia que ves entre los 2 ficheros de capturas.

3. Túnel *IPv6 in IPv4*

Descomprime el laboratorio `IPv6-tun-lab.tgz` y carga el escenario dentro de NetGUI. Arranca de una en una todas las máquinas del escenario.

Observa en la figura 2 que hay 3 zonas diferenciadas en el escenario:

- Zona A - Zona IPv6: `pc1`, `pc2` y `r1`.
- Zona B - Zona IPv4: `r3`, `r4` y `r5`
- Zona C - Zona IPv6: `r7`, `pc3` y `pc4`.

Los *routers* `r2` y `r6` son *routers* que interconectan zonas diferentes. Estos *routers* se comunican por IPv4 en una de sus interfaces y por IPv6 en la otra. Son *routers* frontera que tienen la doble pila (IPv4 e IPv6) instalada. Las máquinas `r4` y `r5` sólo se comunican por IPv4, y el resto de máquinas (`pc1`, `pc2`, `r1`, `r2`, `r7`, `pc3` y `pc4`) sólo se comunican por IPv6.

Todos los *routers* y máquinas tienen configuradas direcciones IP y rutas válidas para comunicarse con los nodos de su misma zona.

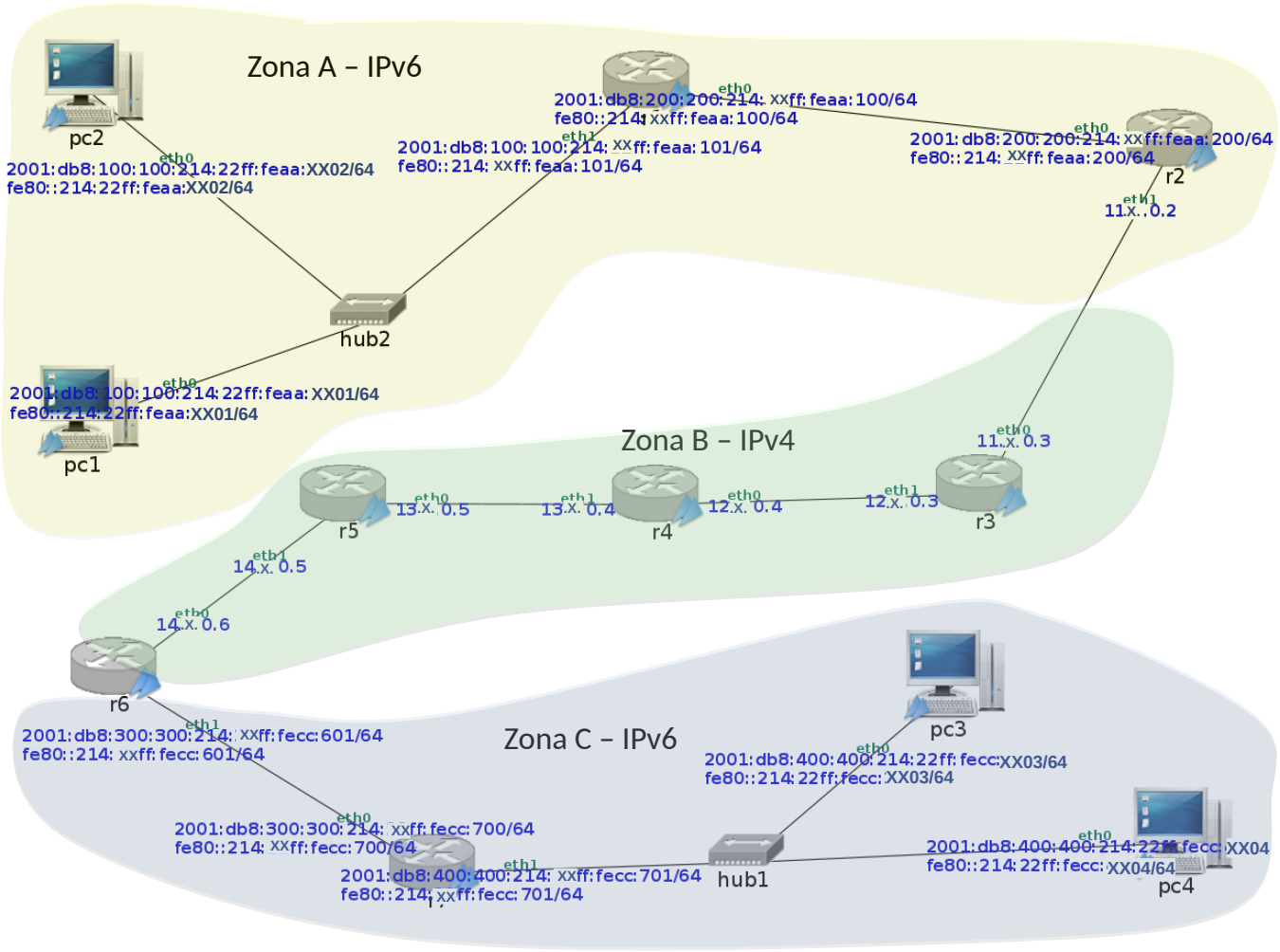

Figura 2: Zonas IPv6 a través de una zona IPv4

Si haces `ping6` desde `pc1` a `pc3` observarás que no funciona. Ambas máquinas están utilizando IPv6, sin embargo, tienen que atravesar una zona que sólo está utilizando IPv4.

Para solucionar este problema vamos a configurar un Túnel IP punto a punto, metiendo los paquetes IPv6 que se generen en ambas zonas IPv6 dentro de paquetes IPv4. De esta forma, las máquinas IPv6 de diferentes zonas podrán comunicarse.

1. Indica qué *routers* crees que deberían ser los extremos del túnel IPv6 dentro de IPv4.
2. Configura en `r2` un extremo del túnel, con ttl 32, y añade la/s ruta/s necesaria/s en `r2` para que los paquetes IPv6 generados en la zona A puedan llegar a la Zona C.
3. Arranca 3 `tcpdump`:
   - `tcpdump` en la interfaz `eth1` de `r1` (captura `ipv6-tun-01.cap`).
   - `tcpdump` en la interfaz `eth1` de `r4` (captura `ipv6-tun-02.cap`).
   - `tcpdump` en la interfaz `eth1` de `r7` (captura `ipv6-tun-03.cap`).

   Realiza un `ping6` desde `pc1` a `pc3`. Verás que no funciona, pues aún no está configurado el otro extremo del túnel. Interrumpe las capturas y estúdialas.

Con la configuración actual ¿llegan a viajar los *ICMPv6 echo request* por el túnel? Estudia las cabeceras exactas que llevan y explica sus valores.

4. Configura en `r6` el otro extremo del túnel, con ttl 32 y añade la/s ruta/s necesaria/s en `r6` para que los paquetes IPv6 generados en la zona C puedan llegar a la Zona A.

5. Arranca 3 `tcpdump`:

   - `tcpdump` en la interfaz `eth1` de `r1` (captura `ipv6-tun-04.cap`).
   - `tcpdump` en la interfaz `eth1` de `r4` (captura `ipv6-tun-05.cap`).
   - `tcpdump` en la interfaz `eth1` de `r7` (captura `ipv6-tun-06.cap`).

   Realiza de un `ping6` desde `pc1` a `pc3`. Interrumpe las capturas y analízalas. Para los paquetes de cada una de las capturas, observa los siguientes campos y explica sus valores:

   a) Versión del protocolo IP que hay en la cabecera IP que va justo detrás de la cabecera Ethernet.

   b) direcciones IP origen y destino de esa cabecera

   c) `TTL` (IPv4) o `Hop limit` (IPv6)

   d) `Protocol` (IPv4) o `Next Header` (IPv6)

   e) Contenido del datagrama IPv4 o IPv6.

Utiliza la herramienta `traceroute6` para conocer el número de saltos IPv6 que se dan desde `pc2` a `pc4`. Esta herramienta tiene un comportamiento similar al `traceroute` en IPv4. Si no recuerdas su funcionamiento, por favor, revísalo antes de comenzar este apartado.

IMPORTANTE: Para usar `traceroute6` en este escenario, utiliza siempre la opción `-z 200` para que `traceroute6` espere 200ms entre cada paquete que envía.

6. Piensa en qué paquetes se van a capturar en las interfaces `r4(eth1)`, `r1(eth1)` y `r7(eth1)` cuando ejecutes `traceroute6` desde `pc2` a `pc4`.

7. Inicia 3 capturas de tráfico:

   - en la interfaz `eth1` de `r1` (captura `ipv6-tun-07.cap`).
   - en la interfaz `eth1` de `r4` (captura `ipv6-tun-08.cap`).
   - en la interfaz `eth1` de `r7` (captura `ipv6-tun-09.cap`).

   y realiza `traceroute6` desde `pc2` a `pc4`.

8. Interrumpe las capturas y analiza el contenido de los paquetes capturados, observando especialmente los campos `TTL` (IPv4) o `Hop limit` (IPv6) de los paquetes.

9. Tras lo analizado en las capturas, explica con detalle cómo cambian los valores de `Hop limit` y `TTL` según los *ICMPv6 echo request* van avanzando por la zona A, después por la zona B, y por último por la zona C.

10. Indica si ante el tráfico recibido por `pc2`, es posible que `pc2` conozca que se ha atravesado un túnel para llegar a `pc4`.

Entrega de la práctica

Guarda los ficheros de captura en una carpeta que se llame `p1` que contenga todas las capturas de tráfico: `ipv6-1.cap` hasta `ipv6-9.cap` y `ipv6-tun-1.cap` hasta `ipv6-tun-9.cap`. Sube al enlace que encontrarás en Aula Virtual, y antes de que termine el plazo de entrega, un fichero de nombre `p1.zip` o `p1.tgz` resultado de comprimir la carpeta `p1` y otro fichero diferente con la memoria en formato pdf.

Así pues debes entregar en el aula virtual 2 ficheros:

- Memoria en formato PDF.
- `p1.zip`: Fichero comprimido con carpeta p1 y dentro los ficheros de captura