

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN TECNOLOXÍAS DA INFORMACIÓN



Práctica 2: IPv6 en INET

Estudiante 1: Óscar Olveira Miniño

Estudiante 2: Alejandro Javier Herrero Arango

A Coruña, noviembre de 2024.

Índice general

| | | |
|----------|-------------------------|----------|
| 1 | IPv4 | 1 |
| 1.1 | Ejercicio 1.1 | 1 |
| 1.2 | Ejercicio 1.2 | 2 |
| 1.3 | Ejercicio 1.3 | 2 |
| 1.4 | Ejercicio 1.4 | 3 |
| 1.5 | Ejercicio 1.5 | 3 |

Índice de figuras

| | | |
|-----|--|---|
| 1.1 | Tráfico DHCP entre host[0] y los servidores | 1 |
| 1.2 | Tráfico DHCP entre host[0] y los servidores cuando host[2] recibe la ip servidor local | 4 |

1.1 Ejercicio 1.1

- 1.1.1 Muestra una captura del tráfico de paquetes DHCP intercambiados entre el nodo host[0] y los servidores DHCP durante el proceso de obtención de su IP, obtenida en Wireshark (Nota: para que los tiempos mostrados en Wireshark coincidan con los tiempos de simulación, activa Visualización → Formato de visualización de fecha → Segundos desde 1970-01-01). Explica lo que ocurre y para qué sirve cada paquete. Para facilitar la captura, configura el startTime del cliente DHCP para que se inicie antes en host[0] que el resto de equipos

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------------|-------------------|----------|--------|---|
| 1 | 0.000002 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0xc7f0aac |
| 2 | 0.000010 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0xc7f0aac |
| 3 | 0.000013 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0xc17c4aa2f |
| 4 | 0.000013 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0xc7f0aac |
| 5 | 0.000021 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0xc17c4aa2f |
| 6 | 4.000005 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0x3716a675 |
| 7 | 4.000007 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0x5821ccc0 |
| 8 | 4.000010 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x3716a675 |
| 9 | 4.000013 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x3716a675 |
| 10 | 4.000016 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x5821ccc0 |
| 11 | 4.000019 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x5821ccc0 |
| 12 | 4.000021 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0x1a4eb343 |
| 13 | 4.000024 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0x1a4eb343 |
| 14 | 4.000027 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0x5ba252fb |
| 15 | 4.000030 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0x5ba252fb |
| 16 | 6.000000 | 0a:aa:00:00:00:02 | Broadcast | ARP | 64 | Who has 192.168.0.10? Tell 192.168.0.11 |
| 17 | 6.000002 | 0a:aa:00:00:00:05 | 0a:aa:00:00:00:02 | ARP | 64 | 192.168.0.10 is at 0a:aa:00:00:00:05 |
| 18 | 6.000003 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [SYN] Seq=0 Win=7504 Len=0 MSS=536 |
| 19 | 6.000005 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1025 [SYN, ACK] Seq=0 Ack=1 Win=7504 Len=0 MSS=536 |
| 20 | 6.000005 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=0 |
| 21 | 6.000007 | 192.168.0.11 | 192.168.0.10 | TCP | 258 | 1025 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=200 |
| 22 | 6.000011 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1025 [ACK] Seq=1 Ack=201 Win=7504 Len=0 |
| 23 | 6.000020 | 192.168.0.10 | 192.168.0.11 | TCP | 594 | 80 → 1025 [ACK] Seq=1 Ack=201 Win=7504 Len=536 |
| 24 | 6.000021 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [ACK] Seq=201 Ack=537 Win=7504 Len=0 |
| 25 | 6.000021 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [FIN, ACK] Seq=201 Ack=537 Win=7504 Len=0 |
| 26 | 6.000030 | 192.168.0.10 | 192.168.0.11 | TCP | 522 | 80 → 1025 [ACK] Seq=537 Ack=201 Win=7504 Len=464 |
| 27 | 6.000030 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [ACK] Seq=202 Ack=1001 Win=7504 Len=0 |
| 28 | 6.000030 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1025 [ACK] Seq=1001 Ack=202 Win=7504 Len=0 |
| 29 | 6.000031 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1025 [FIN, ACK] Seq=1001 Ack=202 Win=7504 Len=0 |
| 30 | 6.000032 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [ACK] Seq=202 Ack=1002 Win=7504 Len=0 |
| 31 | 10.000032 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [SYN] Seq=0 Win=7504 Len=0 MSS=536 |
| 32 | 10.000034 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1026 [SYN, ACK] Seq=0 Ack=1 Win=7504 Len=0 MSS=536 |
| 33 | 10.000034 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=0 |
| 34 | 10.000036 | 192.168.0.11 | 192.168.0.10 | TCP | 258 | 1026 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=200 |
| 35 | 10.000040 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1026 [ACK] Seq=1 Ack=201 Win=7504 Len=0 |
| 36 | 10.000049 | 192.168.0.10 | 192.168.0.11 | TCP | 594 | 80 → 1026 [ACK] Seq=1 Ack=201 Win=7504 Len=536 |

Figura 1.1: Tráfico DHCP entre host[0] y los servidores

Como podemos observar, el host[0] empieza haciendo un DHCP Discover para descubrir un servidor DHCP disponible. A continuación, el servidor local es el primero en responder la solicitud con un paquete DHCP Offer con una dirección IP disponible. El cliente (host[0]), responde a su solicitud para confirmar la asignación ofrecida por el servidor local. El router también envía el paquete DHCP Offer, pero al enviarlo más tarde, el cliente lo ignora. Finalmente, el servidor local contesta con un ACK (estos procesos se repiten para todos los cliente, host[1] y host[2]).

Posteriormente, el cliente host[0] intenta hacer la conexión con el servidor local, para lo cual manda primero un broadcast ARP, para así saber cuál es la MAC de la máquina, con la IP que establece en la cabecera ARP. Después, el servidor contesta al broadcast ARP que mandó el cliente identificándose su mac, ya que la cabecera ARP incluye su IP.

Finalmente, la conexión sigue adelante con los mensajes de la capa Transporte correspondientes para la comunicación, restableciéndose cada 4 segundos esa conexión (esto ocurre ya que establecemos un idleInterval de 4 segundos).

Tipos de paquetes:

1. DHCP DISCOVER: Para ubicar servidores DHCP disponibles. El cliente (en este caso Host[0]) manda un paquete con IP destino la de broadcast global.
2. DHCP OFFER: Respuesta del servidor a un paquete DHCPDISCOVER. El servidor enviará un paquete donde la ip de origen será la del propio servidor (192.168.0.10) al destino 255.255.255.255
3. DHCP REQUEST: Solicitud del cliente. Este paquete será la respuesta del cliente al anterior paquete. Esta comunicación sigue siendo broadcast ya que no tiene una dirección IP privada válida

1.2 Ejercicio 1.2

1.2.1 ¿Cuál de los servidores proporciona la IP a host[0]? ¿Sabe el otro servidor que host[0] no cogió la IP ofrecida por él? ¿Cómo? (Muestra el contenido de los paquetes relevantes en Wireshark.)

1.3 Ejercicio 1.3

1.3.1 ¿De qué tipo son los primeros paquetes que se intercambian a partir de t = 6 segundos? ¿Cuál es su objetivo?

Los primeros paquetes que se intercambian a partir de 6s son paquetes ARP. Este paquete lo empieza mandando el cliente en modo broadcast para saber a qué MAC le corresponde la IP

que establece en la cabecera del paquete ARP. Posteriormente, el servidor local (el que tiene la IP de la cabecera ARP), responde enviando un paquete ARP de respuesta unicast al host con su dirección MAC. Sin este paso, no se podría establecer la conexión.

1.4 Ejercicio 1.4

1.4.1 ¿Cuáles son las direcciones origen y destino de estos paquetes (solicitud y respuesta)? ¿Tienen IP origen o destino? ¿Por qué?

1.5 Ejercicio 1.5

1.5.1 Configura el servidor DHCP en serverlocal con numReservedAddresses = 10 y el cliente DHCP en host[2] para que arranque antes que los otros clientes DHCP. Esto hará que host[2] reciba la IP fija asignada a serverlocal (192.168.0.10). ¿Ocurre algún error cuando el host[2] recibe la IP de serverlocal? Configura el cliente TCP en host[0] para que se conecte a serverlocal y describe paso a paso qué ocurre durante el establecimiento de conexión TCP a partir de $t = 6$ segundos debido a esta duplicidad.

Configurando la red para que el host[2] reciba de primero la IP con un numReservedAddresses = 10, efectivamente recibe la IP estática del servidor local tal y como lo menciona en el enunciado y no ocurre ningún error aunque tengamos dos dispositivos con la misma IP. No obstante, aunque no ocurra ningún error, esto puede desencadenar problemas en la red ya que habría un conflicto de IPs como por ejemplo problemas de rendimiento en la red (retrasos en el procesamiento de paquetería o aumento de tráfico ARP al no poder resolver la MAC del dispositivo), pérdida de conexión, inconsistencias en las tablas ARP, etc.

Configurando el cliente TCP en el host[0], como se puede ver en la imagen 1.2, el host[0] manda un paquete ARP modo broadcast para averiguar la MAC del dispositivo con la ip 192.168.0.10. Le contesta primero el servidor local por lo que empieza la conexión en capa 4 pero poco después el host[2] contesta también al broadcast mandado por el host[0] ya que el tiene la misma IP asignada que el servidor local por lo que se interrumpe la conexión del host[0] con el servidor local mandando un RST,ACK. Pasados 4 segundos (ya que establecemos un idleInterval de 4 segundos), se vuelve establecer la conexión entre host[0] y el server local haciendose esta de manera exitosa ya que en la tabla ARP del host[0] se almacenó como dispositivo correspondiente a la IP 192.168.0.10, la MAC de servidor local ya que fue el primer dispositivo en responde al mandar paquetes ARP en modo broadcast

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------------|-------------------|----------|--------|---|
| 1 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0xc7f0aac |
| 2 | 0.000005 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0xc7f0aac |
| 3 | 0.000008 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0xc7f0aac |
| 4 | 0.000011 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0x17c4aa2f |
| 5 | 0.000016 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0x17c4aa2f |
| 6 | 3.999997 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0x3716a675 |
| 7 | 4.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | 305 | DHCP Discover - Transaction ID 0x5821ccc0 |
| 8 | 4.000005 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x3716a675 |
| 9 | 4.000008 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0x1a4eb343 |
| 10 | 4.000008 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x3716a675 |
| 11 | 4.000011 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x5821ccc0 |
| 12 | 4.000014 | 192.168.0.1 | 255.255.255.255 | DHCP | 332 | DHCP Offer - Transaction ID 0x5821ccc0 |
| 13 | 4.000016 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0x1a4eb343 |
| 14 | 4.000019 | 0.0.0.0 | 255.255.255.255 | DHCP | 317 | DHCP Request - Transaction ID 0x5ba252fb |
| 15 | 4.000022 | 192.168.0.10 | 255.255.255.255 | DHCP | 332 | DHCP ACK - Transaction ID 0x5ba252fb |
| 16 | 5.999995 | 0a:aa:00:00:00:02 | Broadcast | ARP | 64 | Who has 192.168.0.10? Tell 192.168.0.11 |
| 17 | 5.999997 | 0a:aa:00:00:00:04 | 0a:aa:00:00:00:02 | ARP | 64 | 192.168.0.10 is at 0a:aa:00:00:00:04 |
| 18 | 5.999998 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1025 → 80 [SYN] Seq=0 Win=7504 Len=0 MSS=536 |
| 19 | 5.999998 | 0a:aa:00:00:00:05 | 0a:aa:00:00:00:02 | ARP | 64 | 192.168.0.10 is at 0a:aa:00:00:00:05 |
| 20 | 6.000000 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1025 [ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 21 | 10.000000 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [SYN] Seq=0 Win=7504 Len=0 MSS=536 |
| 22 | 10.000002 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1026 [SYN, ACK] Seq=0 Ack=1 Win=7504 Len=0 MSS=536 |
| 23 | 10.000003 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=0 |
| 24 | 10.000005 | 192.168.0.11 | 192.168.0.10 | TCP | 256 | 1026 → 80 [ACK] Seq=1 Ack=1 Win=7504 Len=200 |
| 25 | 10.000008 | 192.168.0.10 | 192.168.0.11 | TCP | 64 | 80 → 1026 [ACK] Seq=1 Ack=201 Win=7504 Len=0 |
| 26 | 10.000017 | 192.168.0.10 | 192.168.0.11 | TCP | 594 | 80 → 1026 [ACK] Seq=1 Ack=201 Win=7504 Len=536 |
| 27 | 10.000018 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [ACK] Seq=201 Ack=537 Win=7504 Len=0 |
| 28 | 10.000019 | 192.168.0.11 | 192.168.0.10 | TCP | 64 | 1026 → 80 [FIN, ACK] Seq=201 Ack=537 Win=7504 Len=0 |

Figura 1.2: Tráfico DHCP entre host[0] y los servidores cuando host[2] recibe la ip servidor local