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

Práctica 2: IPv6 en INET

Estudante 1: Óscar Olveira Miniño

Estudante 2: Alejandro Javier Herrero Arango

A Coruña, noviembre de 2024.

Índice general

Índice de figuras

Capítulo 1

IPv4

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 0x17c4aa2f
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 0x17c4aa2f
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 enviando un DHCPDISCOVER con el objetivo de descubrir un servidor DHCP disponible. El servidor local resulta ser el primero en responder la solicitud con un paquete DHCPOFFER con una dirección IP disponible. El cliente (host[0]), responde confirmando la elección del servidor local como su servidor DHCP y pidiendo la IP ofrecida mediante un DHCPREQUEST. El router también envía el paquete DHCPOFFER, pero lo hace más tarde, por lo que el cliente lo ignora y el router cancela la oferta (Ver 1.2). Finalmente, el servidor local contesta con un DHCPACK confirmando la recepción del DHCPREQUEST y asignando definitivamente la IP elegida (Este proceso se repiten para los clientes host[1] y host[2]).

Posteriormente, el cliente host[0] intenta establecer una conexión con el servidor local, ya que este también ha sido configurado con una TcpGenericServerApp. Para ello, el cliente manda primero un broadcast ARP para averiguar cuál es la MAC de la máquina servidor con la IP que establece en la cabecera ARP. Después, el servidor contesta al broadcast ARP que mandó el cliente identificándose su MAC.

Finalmente, la conexión sigue adelante con los mensajes TCP correspondientes y repitiéndose cada 4 segundos (Esto ocurre ya que establecemos un idleInterval de 4 segundos).

Tipos de paquetes:

1. DHCPDISCOVER: Para decubrir servidores DHCP disponibles. El cliente (en este caso host[0]) manda un paquete con IP destino de broadcast.

2. DHCPOFFER: Respuesta del servidor a un paquete DHCPDISCOVER ofreciéndose como servidor DHCP. 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, indicando además una IP ofrecida.

3. DHCPREQUEST: El cliente confirma la elección de su servidor DHCP y le solicita la IP ofrecida. Esta comunicación sigue siendo broadcast para informar al resto de servidores DHCP, si los hubiere para que cancelen sus ofertas.

4. DHCPACK: El servidor confirma la recepción de la petición por el cliente y le establece la IP solicitada. Este paquete tiene como destino la dirección de broadcast porque el cliente no ha recibido aún la IP, aunque se sabe a quién va dirigido gracias a un campo que incluye la MAC del cliente destino.

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.)

Por lo que se puede ver en la figura 1.1, en el intercambio inicial de paquetes DHCP, es el servidor local, con IP 192.168.0.10, quien envía el primer paquete DHCPOFFER. De esta manera, el servidor local, se adelanta al router, que envía su DHCPOFFER ya después de que el cliente responda con el DHCPREQUEST. Por tanto, es obvio deducir que el cliente está respondiendo y tomando la ip del servidor local. De todas formas, se pueden observar en detalle los paquetes 3 (DHCPREQUEST) y 5 (DHCPACK) para confirmar nuestra hipótesis. El paquete número 3 incluye los campos Requested IP Adress con la dirección IP solicitada por el cliente y DHCP Server Identifier con el servidor elegido para que le asigne esa IP. De igual forma, en el paquete número 5, aparece la IP asignada en el campo Your (client) IP Adress y el servidor que la asigna, de nuevo en la opción DHCP Server Identifier. El router, que es el otro servidor DHCP, sí descubre que host[0] no coge su dirección IP, abandonando su oferta y dejando libre esa dirección para más adelante. Esto es gracias a que el paquete DHCPREQUEST, en el que el cliente contesta con la IP solicitada y el servidor al que se solicita, se envía a todos los posibles destinatarios dentro de la red 192.168.0.0.

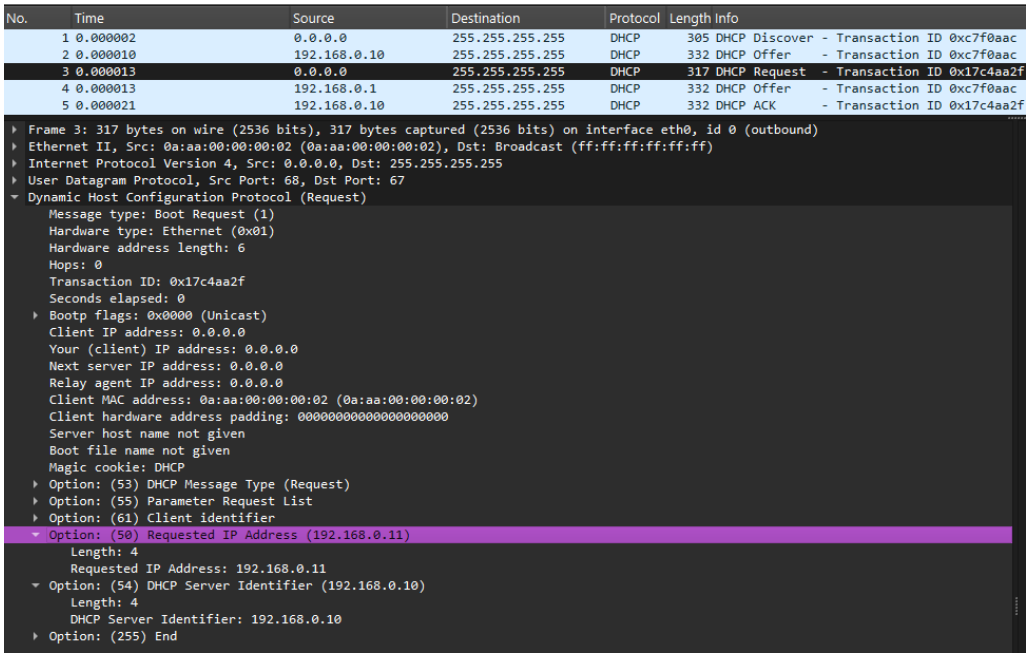


Figura 1.2: Detalle de paquete DHCPREQUEST de host[0]

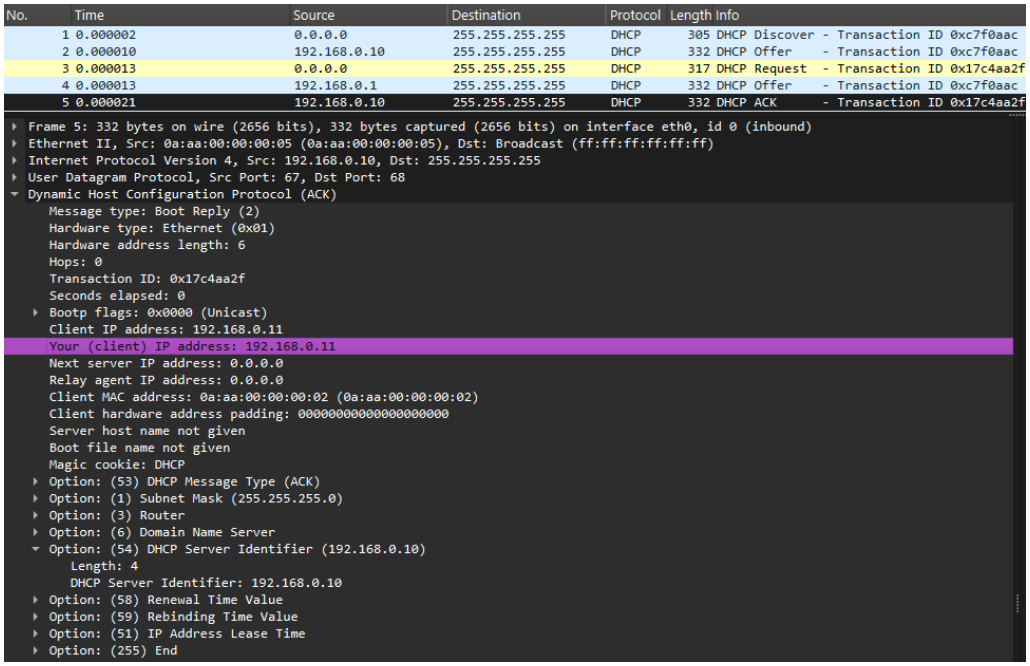


Figura 1.3: Detalle de paquete DHCPACK de server_local

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. El objetivo de estos paquetes es mapear una dirección IP con su correspondiente dirección MAC. Depués, se guarda esta información en la caché ARP de los dispositivos involucrados. El host[0] pregunta por la MAC de la máquina con la IP 192.168.0.10, que es el servidor local que funciona como servidor TCP. Posteriormente, el servidor local responde enviando un paquete ARP de respuesta unicast al host con su dirección MAC. Sin este paso, no se podería 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?

En el paquete de solicitud, la dirección de origen es la MAC del cliente TCP host[0], mientras que la dirección de destino es ff:ff:ff:ff:ff:ff, es decir, la dirección de broadcast MAC, cualquier dispositivo en la LAN. En el paquete de respuesta, la dirección origen es la MAC de server_local y la dirección de destino la MAC de host[0]. Como ARP es un protocolo de capa de enlace, se envía a nivel ethernet y, por tanto, su origen y destino son, como se menciona anteriormente direcciones MAC y no IP. Sin

embargo, las direcciones IP origen y destino sí estan incluídas en el protocolo ARP, y por lo tanto en el paquete, dentro de los campos sender/target IP address, como se aprecia en la imagen.

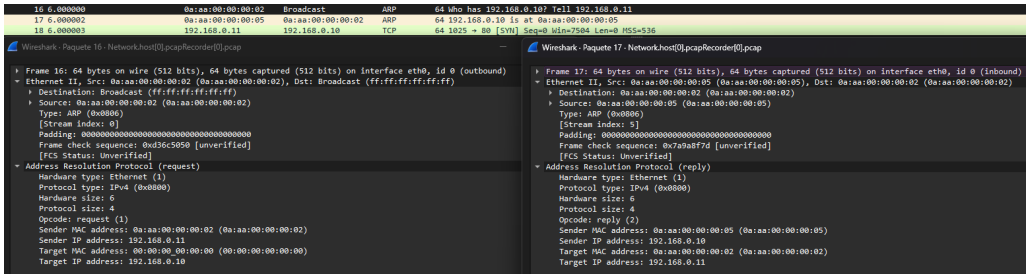


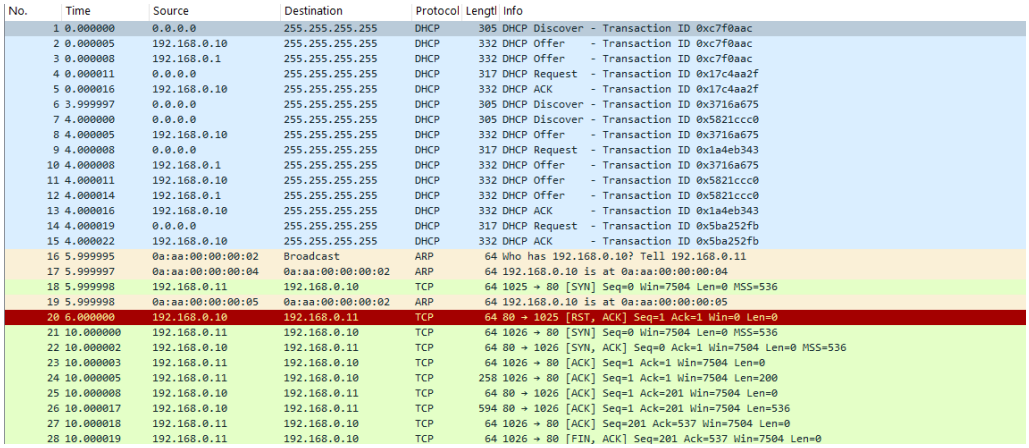
Figura 1.4: Detalle del intercambio de paquetes ARP entre host[0] y server_local

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 habería un confrito 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 disposivo), pérdida de conexión, inconsistencias en las tablas ARP, etc.

Configurando el cliente TCP en el host[0], como se puede ver en la imagen 1.5, 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 él 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	Lengtl	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 [RST, 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	258	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 1.5: Tráfico DHCP entre host[0] y los servidores cuando host[2] recibe la ip servidor local

Capítulo 2

IPv6-SLAAC

2.1 Ejercicio 2.1

2.1.1 Muestra una captura del escenario en el momento inicial de la simulación en la que se vean todas las direcciones MAC e IP. Utilizando la dirección IP de host[0] como ejemplo, explica cómo se construye, destacando los campos y bits relevantes. Utiliza para explicarlo la notación IPv6 no abreviada (16 bytes: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx)

En el principio de la simulación, la direcciones que se muestran son direcciones unicast local de enlace. Estas son direcciones se generan automáticamente una vez que un dispositivo se conecta a una red. La estructura de este tipo de direcciones tienen como prefijo FE80::, no se enrutan y son únicas en ese enlace específico.

La segunda parte de la dirección se forma a partir de la dirección MAC del dispositivo. Para esto, fijamos el séptimo bit a 1 e insertamos 0XFFFE entre las dos mitades de la dirección MAC del dispositivo.

Como ejemplo, vamos a ver la dirección unicast local de enlace que genera el dispositivo host[0]. Como vemos en la imagen 2.1, su dirección está formada como primera parte FE80:: y la segunda parte como 2E8A:21FF:FE7A:8B9C. Esa segunda parte, como se explicó antes, se forma a partir de su dirección MAC (2C-8A-21-7A-8B-9C). Como vemos, en los primeros 2 bytes de la dirección unicast local de enlace (2E8A), si lo comparamos con su dirección MAC, pasa a ser la segunda una E ya que tenemos que añadir un uno en el séptimo bit (la letra C hexadecimal, en binario es 1100, como su tercer bit corresponde al séptimo bit de la dirección unicast, pasa a ser 1 por lo que se convirte en E -> 1110). Llegados a este punto y al haber hecho el primer cambio, tenemos la siguinte estructura -> FE80::2E80:21

Como se explicó anteriormente, una vez hecho el primer cambio, ahora añadimos 0XFFFE y posteriormente los últimos 3 bytes de la dirección MAC, por lo que queda como dirección final unicast local de enlace FE80::2E8A:21FF:FE7A:8B9C

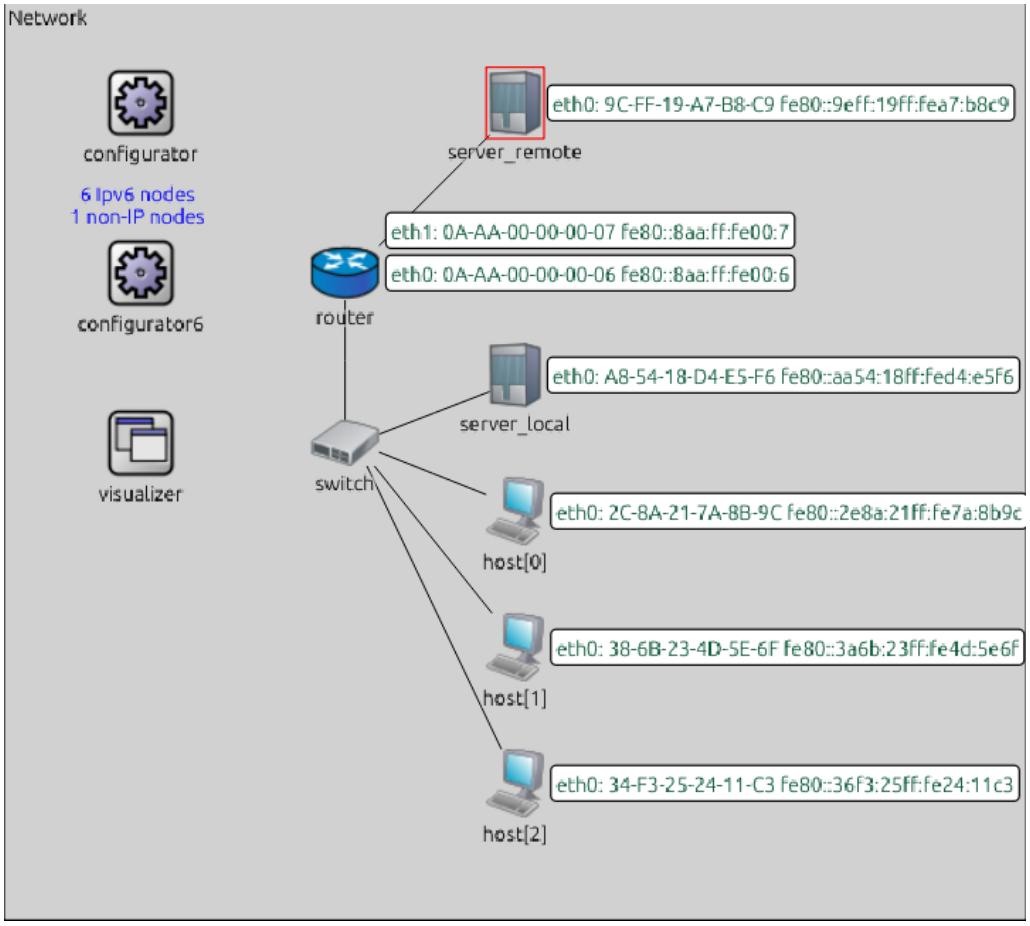


Figura 2.1: Escenario inicial dispositivos con IPv6

2.2 Ejercicio 2.2

2.2.1 Asigna al host[2] la misma dirección MAC que al host[0] y arranca la simulación. ¿Qué error ocurre antes de que haya transcurrido el primer segundo de simulación? Muestra una captura del error que aparece. ¿Qué paquete (tipo, origen y destino) provoca el error? ¿Por qué?

Tal y como se muestra en la imagen 2.2, hay un fallo de direcciones duplicadas cuando el host[2] intenta asignarse una dirección IPv6.

El paquete en concreto que provoca el error es de tipo ICMPv6 con mensaje NS (Neighbor Solicitation), donde el origen de momento es el host[2] (en ese momento sin ninguna dirección) y como destino una dirección multicast. Este error ocurre porque el host[2], cuando crea su dirección, le comunica a sus vecinos de la red (DAD) como es su dirección entonces se encuentra que el host[0] tiene la misma dirección IPv6 que el se asigno por lo que ocurre un conflito de IPs.

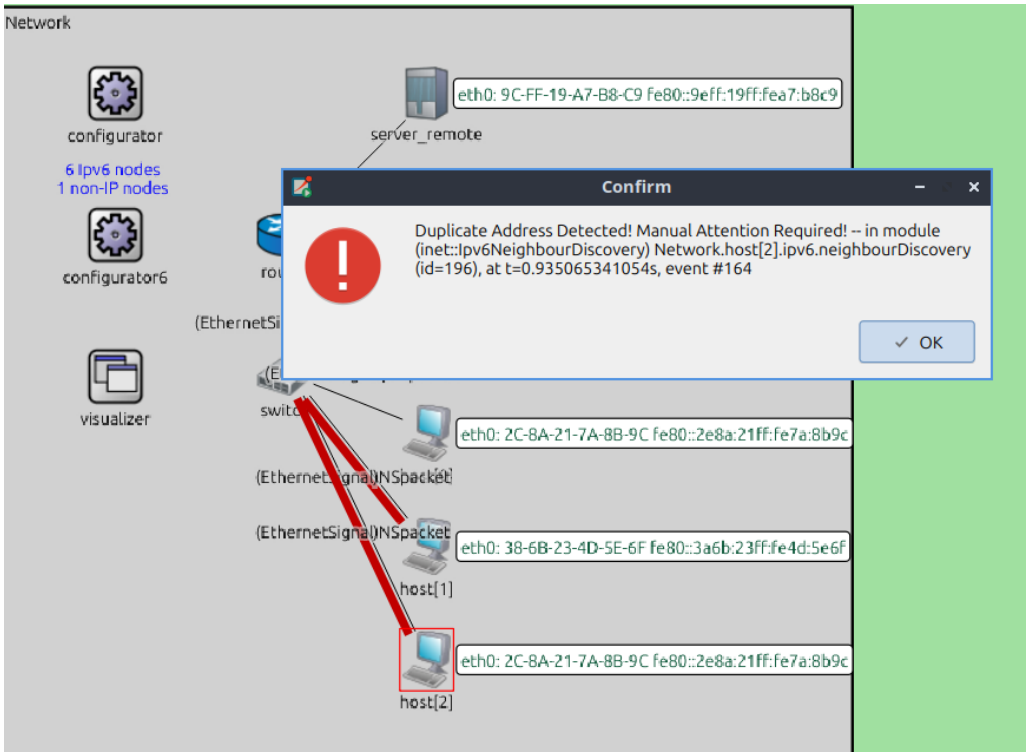


Figura 2.2: Fallo en la red con MAC host[2] igual a la de host[0]

2.3 Ejercicio 2.3

2.3.1 Cambia la MAC del host[2] de manera que coincida con la de host[0] en los últimos 3 bytes y difiera en los 3 primeros bytes (mantén esta MAC para el resto de las cuestiones). Asigna a serverremote la misma MAC que a host[0]. ¿Vuelve a ocurrir el error de dirección duplicada con serverremote y host[0]? ¿Por qué?

En este caso, al poner la misma MAC al host[0] y al server remote no produce ningún error porque son dispositivos que están en diferentes redes. El conflito que sucedía en el ejercicio 2.2 es porque los dos dispositivos estaban en la misma red por lo que salta el error de direcciones duplicadas. Esto es como en IPv4 con las IPs privadas, en la misma red no puede haber dos IPs iguales pero si hay dos redes diferentes, puede coincidir una IP privada de un dispositivo que está en la red1 con otra IP privada de otro dispositivo que está en la red2

2.4 Ejercicio 2.4

2.4.1 ¿Cuánto tiempo transcurre desde el principio de la simulación hasta que el host[0] su IP link-local definitiva (i.e., fin de DAD)? Muestra la tabla de interfaces del nodo host[0] en la que se vea su estado antes y después del DAD timeout y explica qué cambia. (Nota: Qtenv muestra toda la información de cada interfaz en una línea; para verla correctamente copia el contenido con botón derecho → Copy Value y pégalo en la memoria como texto, en lugar de usar capturas de pantalla.)

2.5 Ejercicio 2.5

2.5.1 ¿En qué instante de la simulación obtienen los equipos sus direcciones IP globales? ¿Cómo obtienen esta última? Muestra la tabla de interfaces de nodo host[0] en la que se vea su estado antes y después de obtener la dirección global y explica qué cambia.

2.6 Ejercicio 2.6

2.6.1 Explica cómo se construye la IP global usando el nodo host[0] como ejemplo, de nuevo usando la notación IPv6 no abreviada

Una vez que el equipo manda un mensaje ICMPv6 con mensaje NS (Neighbor Solicitation) y no ocurre ningún fallo (no hay conflicto de IPs), procede a mandar un paquete ICMPv6 con mensaje RS (Router Solicitation), para que el router le asigne unha IPv6 global (este tipo de IPs ya son enrrutables y únicas a nivel global). El router contesta con un paquete ICMPv6 con mensaje RA (Router Advertisement) indicando el prefijo global de red (los primeros 64 bits de la dirección de la red al que está conectada host[0], en este caso AAAA:0006:0065:0000). Una vez que el host[0] obtiene el prefijo de la red, construye su IPv6 global substituyendo el prefijo de red del enlace local por la de la red quedando su dirección como: AAAA:0006:0065:0000:2E8A:21FF:FE7A:8B9C. Esto se puede ver en la imagen 2.3

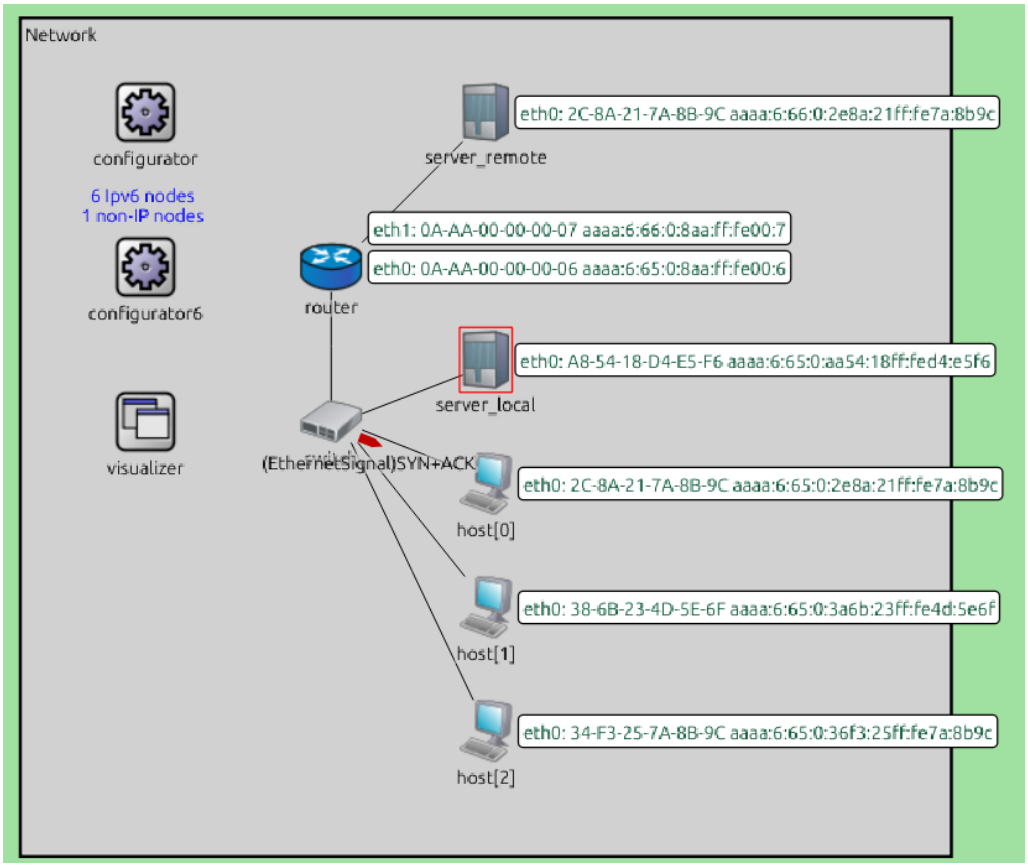


Figura 2.3: Foto red con direcciones IPv6 globales asignadas

2.7 Ejercicio 2.7

2.7.1 Configura host[0] para que se conecte al servidor server remote usando su dirección fe80::x:x:x:x: (asegúrate de que la dirección MAC de server remote es única). ¿Qué ocurre? Repite lo mismo para server local. ¿Qué ocurre?

Capítulo 3

IPv6-Multicast

3.1 Ejercicio 3.1

3.1.1 En los primeros 2 segundos de simulación se envían varios paquetes NS. Muestra una captura del tráfico de paquetes en Wireshark en la que se vean las direcciones IP y MAC origen y destino de los NS enviados desde cada uno de los nodos host[*]. (Nota: para mostrar las direcciones MAC añade dos nuevas columnas: Hw src addr (unresolved) y Hw dest addr (unresolved). Colócalas a la derecha de las direcciones IP.) ¿De qué tipo son las direcciones IP? ¿Cómo se construyen?

	Time	Source	MAC src	Destination	MAC dest	Protocol	Length Info
1	0.000000	::	0a:aa:00:00:00:06	ff02::1:ff00:6	ff:ff:ff:ff:ff:ff	ICMPv6	82 Neighbor Soli
2	0.196510	::	38:6b:23:4d:5e:6f	ff02::1:ff4d:5e6f	ff:ff:ff:ff:ff:ff	ICMPv6	82 Neighbor Soli
3	0.326076	::	a8:54:18:d4:e5:f6	ff02::1:ffd4:e5f6	ff:ff:ff:ff:ff:ff	ICMPv6	82 Neighbor Soli
4	0.697545	::	2c:8a:21:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82 Neighbor Soli
5	0.740680	::	34:f3:25:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82 Neighbor Soli
6	1.747523	fe80::3a6b:23ff:fe4…	38:6b:23:4d:5e:6f	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74 Router Solici
7	1.791090	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122 Router Advert
8	1.914339	fe80::aa54:18ff:fed…	a8:54:18:d4:e5:f6	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74 Router Solici
9	2.098462	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122 Router Advert
10	3.446770	fe80::36f3:25ff:fe7…	34:f3:25:7a:8b:9c	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74 Router Solici
11	3.493229	fe80::2e8a:21ff:fe7…	2c:8a:21:7a:8b:9c	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74 Router Solici
12	3.730036	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122 Router Advert

Figura 3.1: Captura MAC origen y destino host[0]

Como las direcciones IPs de destino son multicast, en todos los hosts se ve la misma paquetería, por lo que solo subimos la captura de tráfico en el host[0].

Las direcciones, como hemos mencionado anteriormente son multicast y se construyen con el prefijo multicast link-local (FF02::1, donde FF02 quiere decir que es tipo multicast, siendo el 2 un scope de local de enlace y el ultimo 1 quiere decir todos los dispositivos, si fuera un 2 serian todos los routers). Por último tendriamos como sufijo, los 3 bytes menos significativos de la MAC del dispositivo de ese host precedido de FF.

Por ejemplo, en el primer caso tenemos como destino FF02::1:FF00:0006 donde F002::1 es como explicamos antes, el prefijo multicast link-local y como sufijo FF00:0006, donde 000006 son los 3 últimos

bytes de la MAC de la interfaz del router en ese enlace (0A:AA:00:00:00:06)

3.2 Ejercicio 3.2

3.2.1 ¿Coincide alguna de las direcciones IP destino de los diferentes paquetes NS? ¿Por qué? ¿Qué consecuencia tiene esto?

Coinciden las direcciones IP del host[0] y host[2] ya que los dos tienen los 3 últimos bytes de la MAC iguales por lo que la dirección multicast se construye de la misma forma.

Con respeto a las consecuencias que puede se puede tener al haber dos direcciones multicast iguales, no es ninguna ya que los mensajes ICMPv6 de tipo NS (Neighbor Solicitation), guarda en su cabecera la dirección unicast de destino por lo que no se daría ningún conflito (target address). Aunque no pase nada, lo normal es que las direcciones multicast de nodo solicitado (FF02::1) sean únicas.

3.3 Ejercicio 3.3

3.3.1 ¿Por qué se envían los NS a esas direcciones IP?

Se envian ICMPv6 de tipo NS a esas direcciones ya que haciendolo con direcciones multicast, se puede mandar un único paquete a uno o varios destinos, de esta forma optimizamos la comunicación y reducimos el tráfico de la red para que no se congestione por lo que aumenta la eficiencia.

3.4 Ejercicio 3.4

3.4.1 ¿Qué direcciones MAC destino tienen los paquetes NS anteriores? ¿Cuáles deberían tener según lo visto en clases de teoría? (Utiliza el paquete NS que sale desde host[1] como ejemplo y escribe los 6 bytes que debería tener la dirección MAC en formato hexadecimal.)

No.	Time	Source	MAC src	Destination	MAC dest	Protocol	Length	Info
1	0.000000	::	0a:aa:00:00:00:06	ff02::1:ff00:6	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor So
2	0.196509	::	38:6b:23:4d:5e:6f	ff02::1:ff4d:5e6f	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor So
3	0.326076	::	a8:54:18:d4:e5:f6	ff02::1:ffd4:e5f6	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor So
4	0.697546	::	2c:8a:21:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor So
5	0.740680	::	34:f3:25:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor So
6	1.747522	fe80::3a6b:23ff:fe4…	38:6b:23:4d:5e:6f	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Soli
7	1.791090	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Adve
8	1.914339	fe80::aa54:18ff:fed…	a8:54:18:d4:e5:f6	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Soli
9	2.098462	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Adve
10	3.446770	fe80::36f3:25ff:fe7…	34:f3:25:7a:8b:9c	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Soli

Figura 3.2: Captura MAC origen y destino paquetes NS en host[1]

Como se puede observar en la imagen 3.2, todos los paquetes tienen como MAC destino FF:FF:FF:FF:FF:FF. Según se ha visto en la clase de teoría, las MAC tiene la estrutura 33:33:FF como prefijo de la MAC y después los 3 últimos bytes son los 3 bytes menos significativos de la dirección multicast. En este caso

el destino es como dirección MAC broadcast ya que Omnet no sabe como construir este tipo de direcciones MAC como se explica en la teoría, por eso todos los hosts reciben la misma paquetería. De la otra forma cada uno recibiría su propia paquetería

3.5 Ejercicio 3.5

3.5.1 ¿Qué consecuencia tiene esta diferencia con respecto a lo visto en clase de teoría?

Resulta que las consecuencias de usar dirección MAC broadcast es que todo el mundo recibe paquetes que en realidad no tendrian que recibir por lo que se sobrecarga la red. Usando la direccion MAC que se especifica en la teoria, solamente se mandaría el paquete a ese host.

3.6 Ejercicio 3.6

3.6.1 Muestra las direcciones IP y MAC destino de los mensajes RS y RA que aparecen en torno al segundo 2 de simulación. ¿De qué tipo son las direcciones IP?

No.	Time	Source	MAC src	Destination	MAC dest	Protocol	Length	Info
1	0.000000	::	0a:aa:00:00:00:06	ff02::1:ff00:6	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor Solicitatio
2	0.196510	::	38:6b:23:4d:5e:6f	ff02::1:ff4d:5e6f	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor Solicitatio
3	0.326076	::	a8:54:18:d4:e5:f6	ff02::1:ffd4:e5f6	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor Solicitatio
4	0.697545	::	2c:8a:21:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor Solicitatio
5	0.740680	::	34:f3:25:7a:8b:9c	ff02::1:ff7a:8b9c	ff:ff:ff:ff:ff:ff	ICMPv6	82	Neighbor Solicitatio
6	1.747523	fe80::3a6b:23ff:fe4…	38:6b:23:4d:5e:6f	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Solicitation
7	1.791090	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Advertisement
8	1.914339	fe80::aa54:18ff:fed…	a8:54:18:d4:e5:f6	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Solicitation
9	2.098462	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Advertisement
10	3.446770	fe80::36f3:25ff:fe7…	34:f3:25:7a:8b:9c	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Solicitation
11	3.493229	fe80::2e8a:21ff:fe7…	2c:8a:21:7a:8b:9c	ff02::2	ff:ff:ff:ff:ff:ff	ICMPv6	74	Router Solicitation
12	3.730036	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Advertisement
13	3.936082	fe80::8aa:ff:fe00:6	0a:aa:00:00:00:06	ff02::1	ff:ff:ff:ff:ff:ff	ICMPv6	122	Router Advertisement
14	5.762481	aaaa:6:65:0:2e8a:21…	2c:8a:21:7a:8b:9c	ff02::1:ffd4:e5f6	ff:ff:ff:ff:ff:ff	ICMPv6	90	Neighbor Solicitatio
15	5.762484	aaaa:6:65:0:aa54:18…	a8:54:18:d4:e5:f6	aaaa:6:65:0:2e8a:21…	ff:ff:ff:ff:ff:ff	ICMPv6	90	Neighbor Advertiseme
16	5.762485	aaaa:6:65:0:2e8a:21…	2c:8a:21:7a:8b:9c	aaaa:6:65:0:aa54:18…	a8:54:18:d4:e5:f6	TCP	82	1025 → 80 [SYN] Seq=

Figura 3.3: Captura paquetes RS y RA host[0]

Como se puede observar en la imagne 3.3, en el caso de los paquetes ICPMv6 con mensaje RS tienen como dirección destino la direccion multicast FF02::2 y como MAC de destino tiene la dirección FF:FF:FF:FF:FF:FF. En este caso la dirección multicast son de este tipo ya que son direcciones a las que van dirigidos los paquetes a todos los routers de la misma red de en enlace para asi optimizar el proceso de descubrimiento de routers y no malgastar ancho de banda.

En el caso de los paquetes ICMPv6 con mensaje RA tienen como dirección destino la direccion multicast FF02::1 y como MAC de destino tiene la dirección FF:FF:FF:FF:FF:FF. En este caso la dirección multicast tiene esta estructra ya que va dirigido a todo dispositivo que está en el mismo enlace de red.

3.7 Ejercicio 3.7

3.7.1 ¿Por qué el RA de respuesta a un RS no usa IP destino unicast?

Este enfoque permite enviar la información de configuración a todos los dispositivos de la red sin necesidad de identificar la dirección específica de cada uno. Si se utilizara una dirección unicast, sería necesario conocer la dirección única de cada dispositivo, lo que resultaría poco práctico en redes grandes o cuando los dispositivos conectados pueden cambiar con frecuencia. Así de esta forma, el tráfico no se sobrecarga y resultaría todo mas eficiente.

3.8 Ejercicio 3.8

3.8.1 ¿Qué direcciones MAC destino deberían tener los RS según lo visto en clases de teoría? ¿Y los RA?

La MAC destino del paquete RS debería ser 33:33:00:00:00:02, asegurando que solo los routers reciban las solicitudes de descubrimiento de nodos, y los paquetes RA deberían tener como MAC destino 33:33:00:00:00:01, por lo que así mapearía a todos los dispositivos IPv6 de esa misma red.

3.9 Ejercicio 3.9

3.9.1 Aproximadamente en t = 6 s el router envía dos NS. ¿De qué tipo son las direcciones IP destino de estos paquetes? Explica cómo se construyen (notación IPv6 no abreviada).

3.10 Ejercicio 3.10

3.10.1 ¿En qué equipos llega el segundo paquete NS enviado por el router al módulo ipv6? ¿Y al submódulo ipv6.neighborDiscovery? ¿Por qué? Nota: para ver el recorrido del paquete en el módulo ipv6, haz doble click en el nodo deseado y luego en el módulo ipv6. Puedes mostrar varios nodos a la vez en diferentes ventanas con botón derecho → Open Graphical View for 'ipv6' una vez dentro de ese nivel.)

3.11 Ejercicio 3.11

3.11.1 ¿En qué equipos llegaría el mensaje al módulo ipv6 si INET implementase direcciones MAC multicast (33- 33-xx-xx-xx-xx)? ¿Por qué?