IPv6

“Redes de Banda Ancha”
Universidad Pública de Navarra

Autores:
Belén Aldecoa Sánchez del Río
Luis Alberto Ramon Surutusa

ÍNDICE

PARTE TEÓRICA

0. ACRÓNIMOS

ARP	Address Resolution Protocol
ATM	Asynchronous Trasnfer Mode
CIDR	Classless Inter-Domain Routing
DCHPv6	Dynamic Host Configuration Protocol version 6
HW	HardWare
IEEE	Institute of Electrical and Electronics Engineers
ICMP	Internet Control Message protocol
ICMPv6	Internet Control Message protocol version 6
IGMP	Internet Group Management Protocol
Interface ID	Interface Identifier
ISP	Internet Service Provider
IPv4	Internet Protocol versión 4
IPv6	Internet Protocol versión 6
MAC	Media Access Control
MTU	Maximum Transmission Unit
ND	Neighbor Discovery
PMTU	Path Maximum Transmission Unit
RIR	Regional Internet Registries
RDISC	Router Discovery
TCP	Transmission Control Protocol

1. INTRODUCCIÓN

IPv6 es la versión 6 del Protocolo de Internet (Internet Protocol), un estándar del nivel de red encargado de dirigir y encaminar los paquetes a través de una red.

IPv6 está destinado a sustituir al estándar IPv4 y el motivo más básico por el que surgió fue la falta de direcciones. IPv4 tiene un espacio de direcciones de 32 bits mientras que el de IPv6 es de 128. Además, IPv6 soluciona o mejora algunos problemas detectados en IPv4 que comentaremos en detalle a lo largo del trabajo.

Entre las principales diferencias que encontramos entre ambos protocolos destacamos:

- No hay direcciones de broadcast (función sustituida por direcciones multicast)
- Los campos de las direcciones reciben nombres específicos, se denomina "prefijo" a la parte de la dirección hasta el nombre indicado. El prefijo nos permite conocer dónde está conectada una determinada dirección, es decir, su ruta de encaminamiento.
- Cualquier campo puede contener sólo unos o sólo ceros, salvo que explícitamente se indique lo contrario.
- Las direcciones IPv6 son asignadas a interfaces, no a nodos. Dado que cada interfaz pertenece a un único nodo, cualquiera de las direcciones unicast de las interfaces del nodo puede ser reemplazado para referirse a dicho nodo.
- Todas las interfaces han de tener, al menos, una dirección unicast link-local (alcance local. Existen también direcciones unicast site-local).
- Una única interfaz puede tener también varias direcciones IPv6 de cualquier tipo (unicast, multicast o anycast).
- Una misma dirección o conjunto de direcciones unicast pueden ser asignados a múltiples interfaces físicas, siempre que la implementación trate dichas interfaces, desde el punto de vista de internet, como una única, lo que permite balanceo de carga entre múltiples dispositivos .
- Al igual que en IPv4, se asocia un prefijo de subred con un enlace, y se pueden asociar múltiples prefijos de subred a un mismo enlace.
- Se produce una simplificación de la cabecera. Algunos campos de la cabecera del IPv4 son eliminados o pasan a ser opcionales, tanto para reducir el coste de procesamiento como el tamaño de la cabecera.
- El tamaño de la cabecera de IPv6 es fijo (40Bytes) lo que facilita el procedo en routers y conmutadores.
- Existe mayor flexibilidad para extensiones y nuevas opciones. En IPv6 no existe un campo "opciones", como tal. La gestión de opciones se realiza por un campo "siguiente cabecera". Eliminando así las limitaciones de tamaño en la cabecera, e introduciendo una gran flexibilidad en el desarrollo de nuevas opciones.
- Capacidades de control de flujo. Se añaden capacidades que permiten marcar los paquetes que pertenezcan a un determinado tipo de tráfico, para el cual el remitente demanda una calidad mayor a la especificada por defecto o servicios en tiempo real.
- IPv6 provee extensiones para soportar autenticación, e integridad y confidencialidad de datos.

2. DIRECCIONAMIENTO (rfc 4291)

En IPv6 han sido definidos varios tipos de direcciones. Éstas, son asignadas a interfaces, no a nodos. Además, varias direcciones de distintos tipos pueden ser asignadas a un mismo interfaz, por lo que un nodo podrá ser identificado por cualquiera de las direcciones asignadas a uno de sus interfaces. Como mínimo, todos los interfaces deben tener asignada una dirección del tipo “link-local unicast” (ver sección 2.5.6.).

Al igual que en IPv4, la asignación de direcciones tiene que identificar unívocamente a un nodo, pero existe un caso en que una misma dirección puede asignarse a un conjunto de interfaces. Esta excepción se puede dar cuando los distintos interfaces pertenecen a un mismo nodo y este los presenta a la capa de internet como un solo interfaz. Este mecanismo puede ser útil para balanceos de carga.

Otra similitud entre el modelo de direccionamiento de IPv4 y el de IPv6 es que existen prefijos de subred asociados a los enlaces. En IPv6, a un mismo enlace se le pueden asignar varios de estos prefijos.

2.1. Representación de direcciones

Las direcciones de Ipv6 están compuestas por 128 bits. Existen tres formas convencionales para su representación como cadenas de caracteres:

- El formato preferente es x:x:x:x:x:x:x:x, donde las 'x' representan de uno a cuatro dígitos hexadecimales pertenecientes a cada uno de los 8 grupos de 16 bits en que se divide la dirección.

  Ejemplo:

  FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

  1080:0:0:0:8:800:200C:417A

  No es necesario escribir todos los ceros en un campo individual, pero debe haber al menos un número en cada campo.

- Debido a los métodos de asignación de ciertos tipos de direcciones IPv6, es común encontrar direcciones que contengan largas cadena de bits a 0. Para escribir estas direcciones fácilmente, se puede hacer uso de la sintaxis especial "::". Con esto podemos indicar múltiples grupos de 16 bits de ceros.

  Ejemplo:

1080:0:0:0:8:800:200C:417A	*a unicast address*
FF01:0:0:0:0:0:0:101	*a multicast address*
0:0:0:0:0:0:0:1	*the loopback address*
0:0:0:0:0:0:0:0	*the unspecified addresses*

  Pueden representarse como:

1080::8:800:200C:417A	*a unicast address*
FF01::101	*a multicast address*
::1	*the loopback address*
::	*the unspecified addresses*

- Una forma alternativa, más conveniente en entornos en los que se mezclan nodos IPv4 e IPv6 es x:x:x:x:x:x:d.d.d.d, donde las 'x' representan el valor hexadecimal de los 6 primeros grupos de 16 bits y las 'd' representan el valor decimal de los cuatro grupos de 8 bits menos significativos de la dirección (representación normal de las direcciones IPv4).

  Ejemplo:

  0:0:0:0:0:0:13.1.68.3

  0:0:0:0:0:FFFF:129.144.52.38

  o en forma comprimida:

  ::13.1.68.3

  ::FFFF:129.144.52.38

2.2. Representación de prefijos de direcciones

La representación de los prefijos en IPv6 es igual que la de las direcciones IPv4 CIDR. Se representan como:

IPv6-address/prefix-length

donde:

IPv6-address: Es una dirección IPv6 en alguna de las notaciones ya descritas

prefix-length: Es un valor decimal que especifica cuantos, d los bits más significativos, son considerados prefijo.

Ejemplo

Las siguientes representaciones serían válidas para indicar el prefijo de 60-bits de longitud, 0x12AB00000000CD3:

12AB:0000:0000:CD30:0000:0000:0000:0000/60
12AB::CD30:0:0:0:0/60
12AB:0:0:CD30::/60

Cuando queramos escribir una dirección y el prefijo de esa dirección, ambas pueden combinarse como:

Dirección de nodo: *12AB:0:0:CD30:123:4567:89AB:CDEF*
Número de la subred: *12AB:0:0:CD30::/60*
Su abreviación quedará como: 12AB:0:0:CD30:123:4567:89AB:CDEF/60

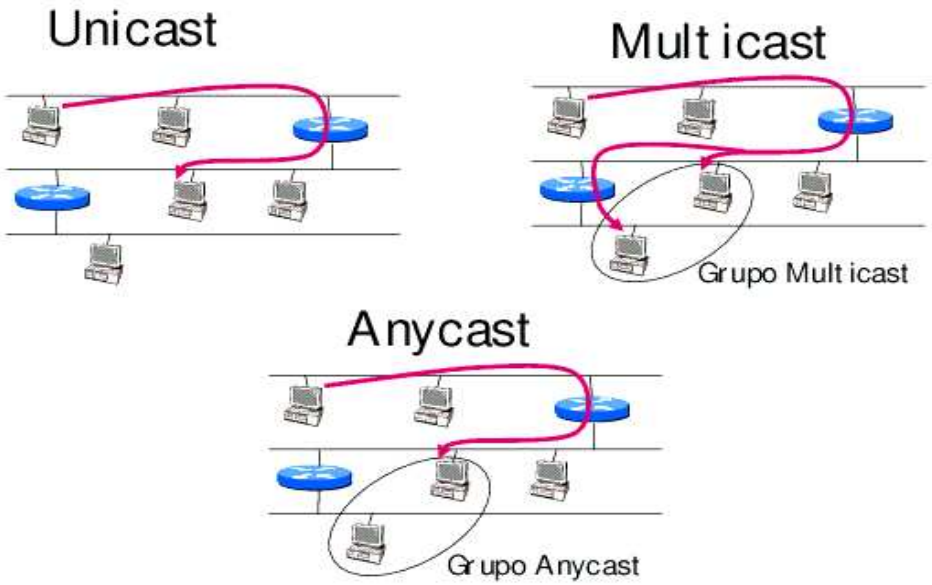
2.3. Tipos de direcciones

Existen tres tipos de direcciones:

Unicast: Identifica un único interfaz de un nodo IPv6. Un paquete enviado a una dirección unicast es entregado al interfaz identificado por dicha dirección.

Anycast: Es asignada a múltiples interfaces (normalmente de distintos nodos). Un paquete enviado a una dirección anycast es entregado a uno de los interfaces identificados con dicha dirección (normalmente el interfaz mas cercano).No tiene un formato de dirección especial, es una dirección unicast.

Multicast: Una dirección multicast identifica a un grupo de interfaces (típicamente pertenecientes a diferentes nodos). Un paquete enviado a una dirección multicast es entregado a todos los interfaces que tengan asignada dicha dirección.



En IPv6 no existe la dirección de broadcast. Su función es reemplazada por el direccionamiento multicast. Por otro lado las direcciones todo ceros y todo unos son valores legales para cualquier campo, a menos que esté específicamente excluido.

2.4. Identificación de tipos de direcciones

El tipo específico de dirección Ipv6 queda identificado por los bits más significativos de la dirección como se muestra en la siguiente tabla:

Tipo de Dirección	Prefijo Binario	Notación IPv6
Unspecified	00...0 (128 bits)	::/128
Loopback	0...01 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Global unicast	Resto	

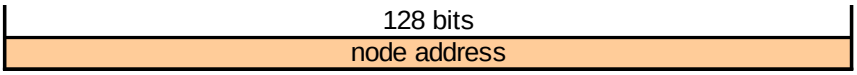
Como ya se dijo, las direcciones anycast están dentro del espacio de direcciones unicast, y no son distinguibles sintácticamente. Por otro lado, en futuras especificaciones podrán redefinirse uno o más subrangos del espacio de direcciones Global unicast para otros propósitos, pero mientras no ocurra, todas las direcciones que no presenten un prefijo de los listados en la tabla, serán consideradas direcciones Global unicast.

2.5. Direcciones Unicast

Las direcciones unicast son agrupables gracias a los prefijos de dirección, como ocurría con las direcciones IPv4 con CIDR.

Hay varias tipos de direcciones unicast en IPv6, como las direcciones globales, las site-local y el direccionamiento link-local. Hay también algunos subtipos con propósitos especiales como las “IPv6 addresses with embedded IPv4 addresses” (Direcciones IPv6 con direcciones IPv4 “encajadas”).

Los nodos IPv6 pueden tener mucho o poco conocimiento de la estructura interna de las direcciones IPv6 según su papel (ej. host, router). El mínimo conocimiento implica que un nodo considere que las direcciones unicast (incluyendo la suya) no tienen estructura interna:



Un host algo más sofisticado puede conocer los prefijos de subred (“subnet prefix”) de los enlaces a los que está unido, donde diferentes direcciones pueden tener diferentes longitudes de “subnet prefix”:

donde el identificador de interfaz (“interface ID”) identifica los distintos interfaces de un enlace.

En cuanto a los routers, si bien pueden no tener ningún conocimiento sobre la estructura interna de la dirección unicast, normalmente conocerán los límites jerárquicos para la operación de los protocolos de enrutamiento. El conocimiento de dichos límites variará de un router a otro, según la posición que ocupen en la jerarquía de enrutamiento.

2.5.1. Identificador de Interfaz (Interface ID)

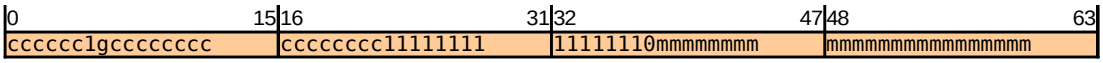
Como ya se ha comentado en el punto anterior, los “Interface ID” son usados para identificar los interfaces de un enlace y por tanto, es un requisito indispensable que sea único en el ámbito delimitado por un “subnet prefix”. Además, es recomendable que sean únicos en el enlace o incluso en un alcance más amplio. Requieren ser únicos en ese link, pudiendo ser únicos también para un alcance más amplio. En algunos casos, el identificado de interfaz se obtendrá a partir de la dirección de enlace del interfaz (su MAC).

Para las direcciones unicast (excepto las que empiezan con los bits 000) los “Interface ID” deben ser de 64 bits, por lo que, para convertir una dirección de enlace IEEE 802 48-bit MAC (la que incorporan las tarjetas ethernet), es necesaria una conversión.

Un ejemplo de dirección IEEE 802 48-bit MAC es:

donde “c” son los bits que identifican a la compañía (la que desarrolla el HW), “0” es el valor dado al bit global/local (1/0) que implica que el alcance de la dirección es local, “g” es un bit que indica si es una dirección individual o de grupo y “m” son los bits del identificado de extensión elegido por el fabricante.

A partir de esta dirección, se obtiene el “Interface ID”:



donde se ha cambiado el bit global/local para especificarlo como de alcance universal y se han añadido los dígitos 0x FFFE para alcanzar los 64 bits necesarios.

Ejemplo:

MAC: *00:0A:5E:3E:40:EC*

Interface ID: *020A:5EFF:FE3E:40EC*

2.5.2. Dirección Unspecified (::)

La dirección todo ceros es llamada Unspecified. Indica la ausencia de direcciones y no puede ser asignada a ningún nodo. Un ejemplo de uso de esta dirección es en el campo dirección origen de un paquete IPv6 enviado por un host durante su proceso de inicialización, antes de que haya obtenido su propia dirección.

La dirección de unspecified no puede ser usada como dirección origen en paquete salientes, y un paquete con la dirección unspecified como destino nunca puede ser enviado fuera del nodo ni debe ser encaminado por los routers.

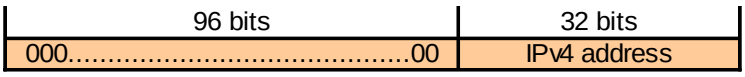
2.5.3. Dirección de Loopback (::1)

La dirección 0:0:0:0:0:0:0:1 es llamada dirección de loopback. Ésta, sirve para enviar un paquete de IPv6 de un nodo a si mismo. No puede ser asignada a ningún interfaz físico, si no que debe entenderse como la dirección link-local asignada a un interfaz virtual unido a un enlace que no va a ninguna parte.

Al igual que la dirección unspecified, la dirección de loopback no puede ser usada como dirección origen en paquete salientes, y si un paquete tiene la dirección de loopback como destino no debe ser enviado fuera del nodo ni debe ser encaminado por los routers.

2.5.4. IPv6 Addresses with Embedded IPv4 Addresses

Existen dos tipos de direcciones IPv6 que contienen en sus últimos 32 bits direcciones de IPv4. La primera de ellas, denominada “IPv4-compatible IPv6 address” fue definida para ayudar en los mecanismos de transición . Su formato es:



El segundo tipo de direcciones IPv6 que contienen direcciones IPv4, son las que representan las direcciones de nodos que sólo soportan IPv4 en formato IPv6. Este tipo de direcciones es conocida como “IPv4-mapped IPv6 address” y su formato es:

80 bits	16 bits	32 bits
000.....000	FFFF	IPv4 address

2.5.5. Direcciones Global Unicast (RFC 3587)

El formato general para las direcciones global unicast es el siguiente:

n bits	m bits	128 – n – m bits
Global routing prefix	subnet ID	Interface ID

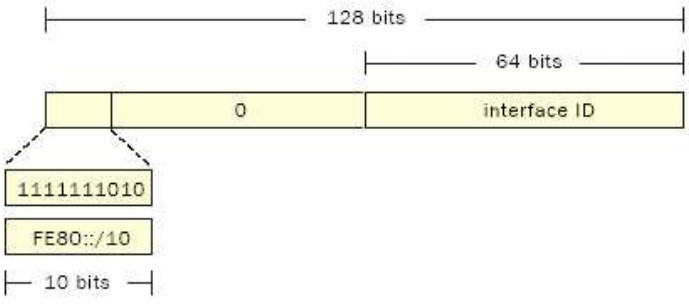
donde el “global routing prefix” (prefijo de enrutamiento global) es un valor asignado a un “site” (un conjunto de subredes o enlaces), el campo “subnet ID” es el identificador de cada una de las subredes dentro del “site”, y el “interface ID” es el identificador definido en la sección 2.5.1.

El prefijo de enrutamiento global ha sido diseñado para ser estructurado jerárquicamente por los RIR's (Regional Internet Registries) y los ISP's (Internet Service provider). El “Subnet ID” ha de ser asignado de manera jerárquica por los administradores del “site”.

Todas las direcciones global unicast que no empiecen con el prefijo 000 tienen un “interface ID” de 64 bits, mientras que las que las que empiezan por 000 no tienen ninguna restricción sobre la longitud de dicho identificador. Un ejemplo de estas últimas son las “IPv6 Addresses with Embedded IPv4 Addresses” explicadas en el punto anterior.

2.5.6. Direcciones Local-use unicast

Existen dos tipos de direcciones unicast de uso local: Link-local y Site-local.
Las direcciones Link-local tienen el siguiente formato:

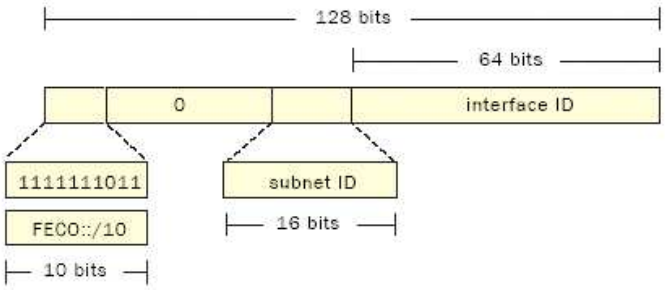


Estas direcciones están diseñadas para comunicaciones dentro de un solo enlace, para ofrecer funcionalidades como la autoconfiguración de dirección, neighbor discovery o para permitir la comunicación entre hosts cuando no hay routers presentes.

Los routers no reenvían ningún paquete con dirección local como fuente.

El otro tipo de direcciones de uso local, denominadas site-local, definidas para ser utilizadas dentro de una red local con diferentes enlaces (en un site), han sido desaprobadas debido a que, pese a quedar definidas teóricamente, en la práctica su definición es ambigua, lo que supone un problema tanto para los desarrolladores de aplicaciones como para los routers.

Aunque hayan sido desaprobadas, esto no ha impedido su uso, por lo menos hasta que se haya estandarizado el cambio y estas hayan sido reemplazadas. Tienen el siguiente formato:



Las direcciones site-local se pensaron para ser usadas dentro de un site sin necesidad de prefijo global, pero cuando se produzca el cambio, el prefijo que utilizaban este tipo de direcciones (FEC0::/10) pasará a formar parte del espacio de direcciones global, como quedaba recogido en la tabla vista en la sección 2.4.

Los routers no reenvían los paquetes con dirección origen local o de site fuera de éste.

- Link-local
- Site-Local
- global

2.6. Direcciones Anycast

Una dirección anycast es una dirección asignada a más de un interfaz (generalmente de diferentes nodos), con el propósito de que un paquete que sea enviado a una de estas direcciones sea encaminado hasta el interfaz más cercano que responda a dicha dirección.

Las direcciones anycast ocupan parte del espacio de direcciones unicast, usando alguno de los formatos definidos para el mismo. Así, un paquete anycast es sintácticamente indistinguible de un paquete unicast. Cuando una dirección unicast es asignada a más de un interfaz se convierte inmediatamente en anycast, y los nodos a los que se les asigna deben ser explícitamente configurados para saber que se trata de una dirección anycast.

Para cada dirección anycast asignada, hay un largo prefijo de dirección (P), que identifica la región topológica en la que residen todos los interfaces con una dirección anycast concreta. Dentro de la región identificada con P, cada dirección anycast debe escribirse como una entrada diferente en el sistema de enrutamiento (comúnmente denominado “host route”); fuera de la región P, las direcciones anycast deben ser englobadas en las entradas de rutas bajo el prefijo P.

Un uso esperado de las direcciones anycast es identificar conjuntos de routers pertenecientes a una organización que provea servicios de Internet (ISP's). Estas direcciones podrán ser usadas como direcciones intermedias en una “Cabecera de encaminamiento” (ver sección 3.) para provocar que el flujo de paquetes sea encaminado a través de un ISP particular o una secuencia de estos.

En un funcionamiento similar, las direcciones anycast pueden ser utilizadas también para identificar el conjunto de routers unidos a una determinada subred.

2.7. Direcciones Multicast

Una dirección multicast en IPv6, identifica a un grupo de interfaces. Además, un interfaz puede pertenecer a cualquier número de grupos multicast. Estas direcciones tienen el siguiente formato:

donde el prefijo 11111111 ó 0xFF identifica la dirección como multicast, el campo flag es un conjunto de 4 flags, el campo scope indica el alcance de cada dirección multicast en concreto (desde alcance de interfaz hasta alcance global) y el group ID identifica al grupo multicast.

Las direcciones multicast no pueden aparecer como dirección origen en un paquete, y tampoco pueden ser utilizadas en el campo “Cabecera de encaminamiento” comentado para las direcciones anycast.

Entre las direcciones multicast “well-known” se encuentran algunas de uso habitual como:

- Dirección “All Nodes”.

  FF02:0:0:0:0:0:0:1

  Esta dirección identifica a todos los nodos y, en este caso, el alcance se ha fijado de enlace (valor del campo scope 2).

- Dirección “All Routers”

  FF02:0:0:0:0:0:0:2

  FF05:0:0:0:0:0:0:2

  Estas direcciones identifican a todos los routers, dentro de un alcance de enlace (socpe=2) o de “site” (scope=5).

- Dirección “Solicited-Node”.

  FF02:0:0:0:0:1:FFXX:XXXX

  Es una dirección calculada a partir de una dirección unicast (o anycast).Se toman los últimos 24 bits de la dirección y se añaden al prefijo FF02:0:0:0:0:1:FF00:0/104. Así, los nodos cuyas direcciones difieran solo en los bits de orden mayor, quedarán asociados a una única dirección multicast, reduciendo el número de grupos a los que el nodo deberá unirse. Los nodos están obligados a unirse a todas las direcciones solicited-node multicast asociadas a las direcciones unicast (o anycast) que le han sido asignadas.

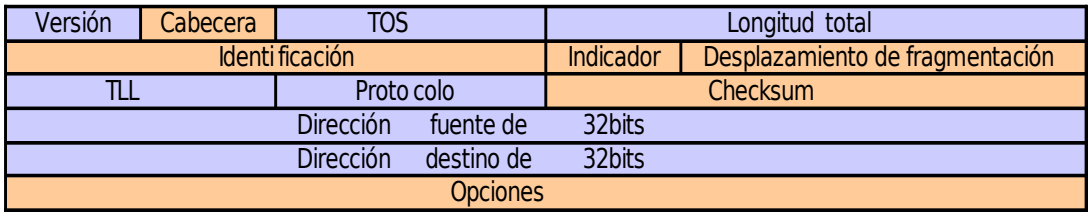
Ejemplo:

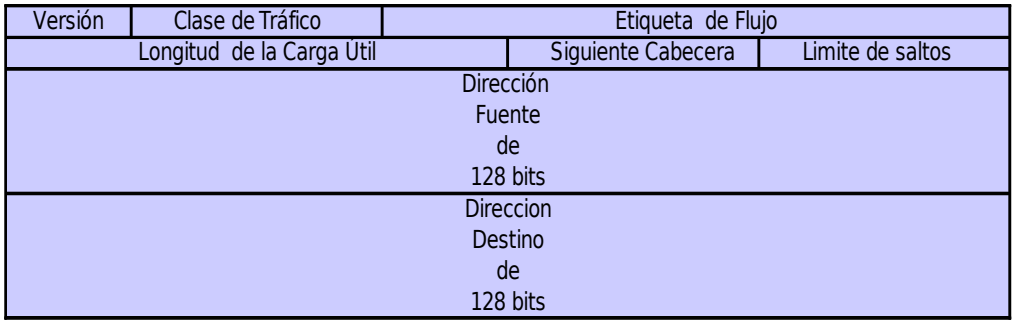
IPv6 address 4037::01:800:200E:8C6C
Solicited-node address FF02::1:FF0E:8C6C

3. REPRESENTACIÓN DE LA CABECERA

Uno de los principales motivos para la eliminación de algunos campos de la cabecera de los paquetes de IPv4 es la innecesaria redundancia. En IPv4, la misma información era ofrecida de varias formas, así por ejemplo nos encontramos con el checksum (verificación de la integridad de la cabecera), y que ya existen mecanismos de encapsulado que realizan esta función. Por otro lado la fragmentación de los paquetes ha sido totalmente modificada en IPv6 por lo que el campo "Fragmentation offset" es eliminado. En IPv6 los routers no fragmentan los paquetes, sino que de ser necesaria, se produce extremo a extremo, y además, para facilitar el procesado en los routers y conmutadores se fija la longitud de los paquetes de cabecera en 40 Bytes.

En las siguientes figuras se muestra, el formato de la cabecera de IPv4, indicando los campos modificados y los que desaparecen, y el formato de cabecera de IPv6.





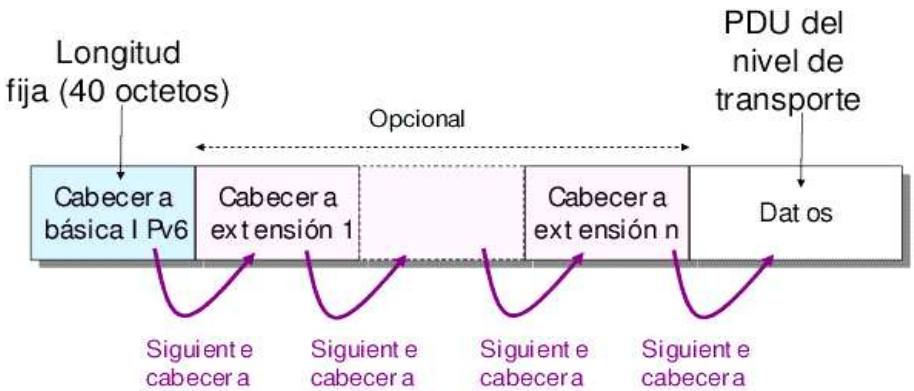
Por tanto, los campos de la cabecera de un paquete de IPv6 son:

- **Versión:** Este campo ocupa 4 bits, e indica la versión de IP. Para el formato descrito, la versión es la 6, para IPv6.
- **Clase de Tráfico:** Este campo ocupa 4 bits, e indica la prioridad que el remitente desea para los paquetes enviados, respecto a los demás paquetes enviados por él mismo. Los valores de prioridad se dividen en dos rangos, de 0 a 7, paquetes para los cuales el remitente espera una respuesta en caso de congestión (p.e. tráfico TCP). Y de 8 hasta 15, paquetes que no deben ser respondidos en caso de congestión, el valor más bajo (8), se usaría cuando el remitente está dispuesto a que sus paquetes sean descartados en caso de congestión (p.e. Video en alta calidad).

- **Etiqueta de flujo:** Este campo ocupa 24 bits, y es usado por el remitente para indicar que sus paquetes sean tratados de forma especial por los routers, como en servicios de alta calidad o en tiempo real. Todos los paquetes pertenecientes al mismo flujo deben tener valores similares en los campos dirección origen, dirección destino, prioridad, y etiqueta de flujo.
- **Longitud de la carga útil:** Este campo ocupa 16 bits, e indica la longitud del resto del paquete que sigue a la cabecera, en octetos. Si su valor es cero, indica que el tamaño de la carga vendrá especificado como "Carga Jumbo", en una opción "salto a salto".
- **Siguiente cabecera:** Este campo ocupa 4 bits, e identifica el tipo de cabecera que sigue a la cabecera IPv6. Es coherente con los valores del campo protocolo en IPv4.
- **Límite de saltos:** Este campo ocupa un octeto. Es decrementado en una unidad por cada nodo que redirige el paquete hacia su destino. El paquete es descartado si el valor del campo llega a cero. Este campo sustituye al campo tiempo de vida, de IPv4.
- **Dirección origen:** Este campo ocupa 128 bits, y corresponde a la dirección de origen.
- **Dirección destino:** Este campo ocupa 128 bits, y corresponde a la dirección de destino.

Para una mayor flexibilidad para extensiones y nuevas opciones, en IPv6 no existe un campo "opciones", como tal. La gestión de opciones se realiza por el campo "siguiente cabecera" y el uso de cabeceras de extensión, eliminando así las limitaciones de tamaño en la cabecera, e introduciendo una gran flexibilidad en el desarrollo de nuevas opciones. Este diseño aporta gran eficacia y flexibilidad ya que se pueden definir en cualquier momento a medida que se vayan necesitando entre la cabecera fija y la carga útil.

Hasta el momento, existen 8 tipos de cabeceras de extensión, donde la cabecera fija y las de extensión opcionales incluyen el campo de cabecera siguiente que identifica el tipo de cabeceras de extensión que viene a continuación o el identificador del protocolo de nivel superior. Luego las cabeceras de extensión se van encadenando utilizando el campo de cabecera siguiente que aparece tanto en la cabecera fija como en cada una de las citadas cabeceras de extensión. Como resultado de la secuencia anterior, dichas cabeceras de extensión se tienen que procesar en el mismo orden en el que aparecen en el datagrama.



Todas o parte de estas cabeceras de extensión tienen que ubicarse en el datagrama en el orden especificado:

1. Cabecera principal
2. Cabecera de opciones de salto a salto (Hop-by-Hop Header), transporta información opcional, contiene los datos que deben ser examinado por cada nodo a través de la ruta de envió de un paquete.
3. Cabecera de encaminamiento (Routing Header), se utiliza en para que un origen IPv6 indique uno o más nodos intermedios que se han de visitar en el camino del paquete hacia el destino.
4. Encaminamiento desde la fuente.
5. Cabecera de fragmentación (Fragment Header), hace posible que el origen envíe un paquete más grande de lo que cabría en la MTU de la ruta.
6. Cabecera de autenticación (Authentication Header), nos sirve para proveer servicios de integridad de datos, autenticación del origen de los datos, antireplay para IP.
7. Cabecera de encapsulado de seguridad de la carga útil (Encapsulating Security Payload Header), permiten proveer servicios de integridad de datos.
8. Cabecera de opciones para el destino (Destination Header), se usa para llevar información opcional que necesita ser examinada solamente por los nodos destino del paquete.

Cada cabecera de extensión debe aparecer como mucho una sola vez, salvo la cabecera de opción destino, que puede aparecer como mucho dos veces, una antes de la cabecera encaminamiento y otra antes de la cabecera de la capa superior.

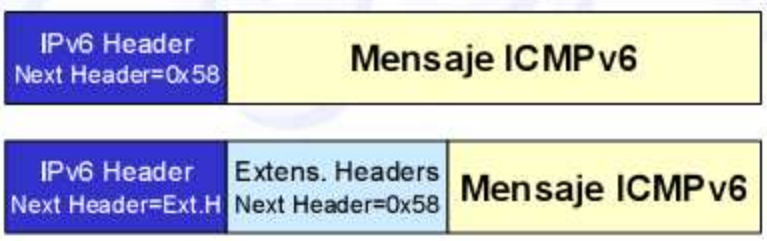
4. ICMPv6 (RFC 4443)

Ligado al nuevo protocolo IPv6 aparecen nuevas versiones de protocolos usados con IPv4 como es el caso de ICMPv6 (Internet Control Message protocol versión 6). Este protocolo es una parte integral de IPv6 y, por tanto, deberá ser implementado por todos los nodos IPv6.

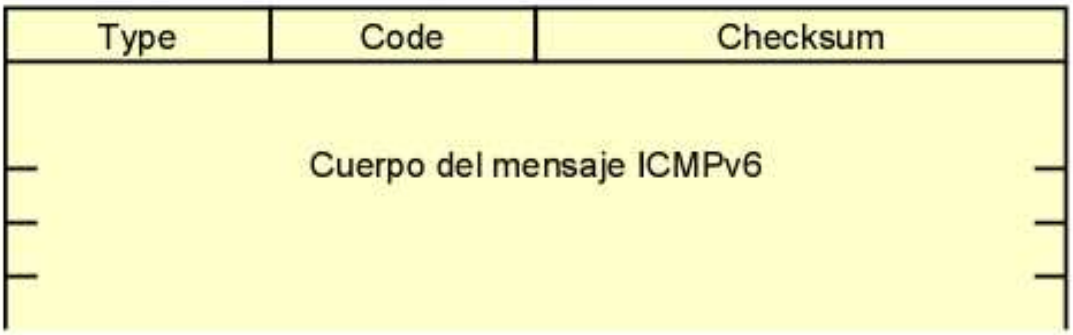
Al igual que su predecesor, ICMPv6 es el protocolo utilizado por los nodos para informar sobre errores en los paquetes procesados y para ofrecer servicios en internet a partir de mensajes de información. En esta versión los servicios ofrecidos por ICMP han sido ampliados e incluyen funcionalidades ofrecidas anteriormente por IGMP (Internet Group Management Protocol) y ARP (Address Resolution Protocol) además de nuevas funcionalidades como el descubrimiento de la máxima MTU del camino.

4.1. Formato del paquete

ICMPv6 es encapsulado por encima de IPv6. Por tanto, los mensajes ICMP contienen la cabecera IP básica, posibles cabeceras de extensión y el mensaje ICMP cuya inmediata predecesora la identifica con el campo "siguiente cabecera" igual a 0x58.



Los campos del paquete ICMP no han cambiado con respecto a los de su versión anterior:



Los 4 primero Bytes del cuerpo del mensaje quedarán reservados para distintas opciones según el tipo de mensaje. El resto ya no se completa con la cabecera IP y los 64 primeros bits del paquete que "invoque" el mensaje ICMP, si no que se introducen tantos bits como quepan sin que el paquete completo exceda del mínimo MTU definido para IPv6 (1280 Bytes).

4.2. Mensajes ICMP

Los distintos tipos de mensaje ICMP son distinguidos por los valores de Type y Code y, como ya se ha comentado en la introducción, se dividen en mensajes de error y mensajes de información. ICMPv6 elimina algunos mensajes y añade otros con respecto a ICMPv4. Como analizar todos con detalle sería muy extenso, nos limitaremos a profundizar sólo en las novedades:

4.2.1. Mensajes de error:

Type	Code	Mensaje ICMP de error
1	-	**Destino inalcanzable**
1	0	No hay ruta al destino
1	1	Comunicación con el destino administrativamente prohibida
1	2	Sin asignar
1	3	Dirección inalcanzable
1	4	Puerto inalcanzable
2	0	**Paquete demasiado grande**
3	-	**Tiempo excedido**
3	0	Hop-Limit excedido
3	1	Tiempo de reensamblado de parámetros excedido
4	-	**Problema de parámetros**
4	0	Campo erróneo en la cabecera
4	1	"Next-Header" no reconocida
4	2	Opción IPv6 no reconocida

El mensaje paquete demasiado grande (“Packet too big”), es un nuevo mensaje de error enviado por los routers en respuesta a un fallo en el encaminamiento de un paquete debido a que el tamaño de este sea superior a la MTU del enlace de salida. Este paquete es utilizado para una nueva funcionalidad ofrecida por IPv6, el Proceso de Descubrimiento del MTU del camino (PMTU, Path MTU discovery process).

4.2.2. Mensajes de información:

Type	Mensaje ICMP informativo
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect

Entre los mensajes de información se incluyen los pertenecientes a IGMP y 5 nuevos paquetes definidos para proporcionar nuevas funcionalidades: el Descubrimiento de Vecinos (Neighbor Discovery) y la autoconfiguración.

5. NEIGHBOR DISCOVERY (RFC 2461)

En IPv6, los nodos (routers o hosts) de una red usan un nuevo protocolo denominado Neighbor Discovery (descubrimiento de vecinos) para determinar las direcciones de enlace de los nodos que residen en su mismo enlace (denominados vecinos) y para eliminar rápidamente los valores almacenados en caché que queden invalidados. Por tanto, son capaces de mantener un seguimiento sobre el estado de la conectividad con sus vecinos y detectar cambios en sus direcciones unicast link-local.

Este protocolo proporciona además un mecanismo a los hosts que les permite encontrar los routers que han de encaminar sus paquetes, por lo que, teniendo en cuenta su facultad de seguimiento de estado del enlace, serán capaces de encontrar nuevas rutas ante fallos de enlaces o routers.

5.1. Direcciones utilizadas por ND

ND utiliza los siguientes tipos direcciones para ofrecer sus servicios:

All-nodes multicast address (FF02::1):
Para comunicación con todos los nodos del enlace.

All-routers multicast address (FF02::2):
Para mensajes dirigidos a todos los routers del enlace.

Solicited-node multicast address:
Para referirse a un nodo en concreto.

Link-local unicast address:
La dirección unicast que identifica cada interfaz.

Unspecified address (0::0):
Utilizada como dirección de origen cuando aún no se posea ninguna

5.1.1. Terminología

on-link: Un destino se considera on-link cuando se encuentra en el mismo enlace y el prefijo de su dirección coincide con alguno de los definidos en el interfaz del origen. Además, los hosts considerarán un destino como on-link cuando un router se lo anuncie como siguiente salto sin realizar ninguna comprobación.

off-link: Lo contrario a on-link

5.2. Funcionalidades

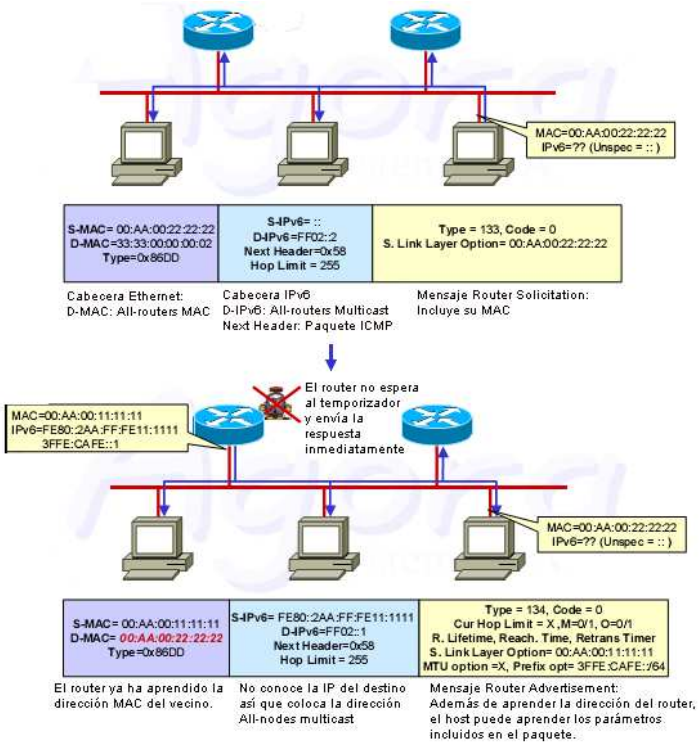
ND facilita la interacción entre los nodos unidos en un mismo enlace. Define para ello mecanismos que hacen uso de cinco nuevos paquetes recogidos en ICMPv6:

- **Solicitud de Router (Router Solicitation):** Peticiones a los routers.
- **Aviso de Router (Router Advertisement):** Enviado por los routers periódicamente y en respuesta un mensaje de Router Solicitation. El tiempo entre avisos periódicos debe ser suficientemente frecuente para que los hosts conozcan la presencia de los routers en unos pocos minutos, pero no es necesario que sean capaces de detectar fallos de los routers por la ausencia de estos paquetes.
- **Solicitud a Vecino (Neighbor Solicitation):** Peticiones a los hosts.
- **Aviso de Vecino (Neighbor Advertisement):** Se usa en respuesta a un mensaje de Neighbor Solicitation, o también cuando un nodo cambia su dirección de enlace, en cuyo caso enviará los avisos inmediatamente sin esperar a que le sean solicitados.
- **Redirección (Redirect):** Utilizados por los routers para informar de mejores rutas a los hosts. El siguiente salto proporcionado por estos paquetes será considerado automáticamente como un destino on-link por parte de los nodos.

Algunas de las funcionalidades cubiertas por este protocolo son:

– **Descubrimiento de los routers vecinos por parte de los hosts.**

Cuando un interfaz de un host es habilitado, este debe mandar mensajes de Router Solicitation para invocar inmediatamente mensajes de Router Advertisement provenientes de los routers del enlace. Estos mensajes contienen, entre otras cosas, información sobre los routers.



- **Descubrimiento del prefijo (dirección de la subred).**

La información sobre el prefijo puede encontrarse también en los mensajes Router Advertisement (aunque opcionalmente los routers podrán optar por no anunciar uno, varios o todos los prefijos de la red). Con esta información, los nodos podrán diferenciar los destinos on-link de los off-link.

- **Descubrimiento de parámetros de enlace (MTU, etc.) o de Internet (Hop-Limit, etc.).**

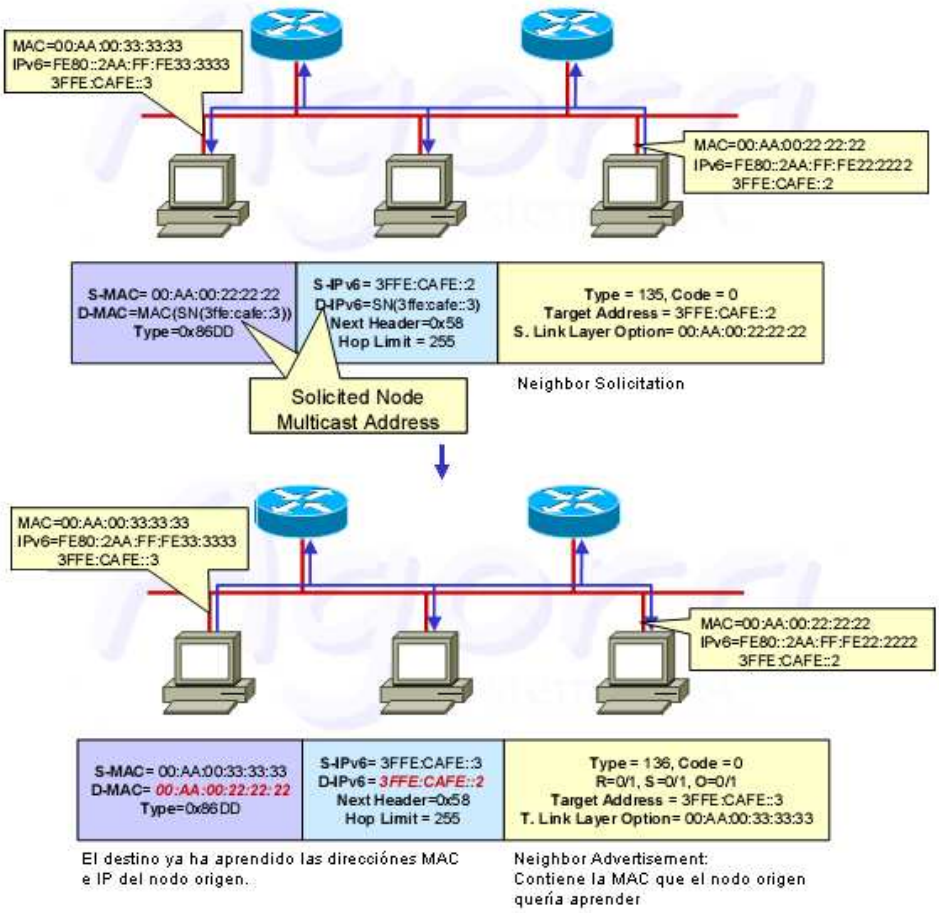
Estos parámetros se encuentran también en los paquetes Router Advertisement repartidos por los routers. Este mecanismo facilita la administración centralizada de parámetros críticos, que pueden ser configurados en routers y, automáticamente, propagados al resto de hosts pertenecientes al enlace. A partir de esta información los hosts fijarán los parámetros de sus paquetes salientes.

- **Autoconfiguración de dirección de los interfaces de los nodos.**

Los paquetes Router Advertisement servirán también a los routers para informar a los hosts sobre como llevar a cabo la autoconfiguración de dirección. Por ejemplo, los routers pueden especificar si los hosts deberán usar una configuración de dirección stateful (DHCPv6) o stateless.

- **Resolución de la dirección de enlace de los vecinos a partir de su dirección IP.**

Los nodos aprenden la dirección de enlace de un vecino mandando un mensaje Neighbor Solicitation a la dirección solicited-node multicast (construida a partir de la dirección unicast) del destino. El nodo destino devolverá su dirección a través de un mensaje unicast de Neighbor Advertisement, y en una sola comunicación petición-respuesta, ambos (origen y destino) resolverán la dirección de enlace del otro, ya que el nodo origen incluye en su mensaje de Neighbor Solicitation su propia dirección de enlace. Podemos ver que el funcionamiento es parecido a ARP en IPv4.



- **Construcción de una tabla de siguiente salto que mapee las direcciones IP de los destinos con las de los vecinos hacia los que enviar el tráfico.**

Los mensajes Router Advertisement contendrán prefijos con los que el nodo será capaz de discernir si el destino se encuentra en su mismo enlace y así mapear en la tabla el destino como siguiente salto. Además, gracias a estos mismos mensajes, el nodo será capaz de guardar una relación de los routers a los que dirigir cada uno de los distintos destinos a los que no tenga conectividad directa.

- **Detección de vecinos inalcanzables.**

El algoritmo “Neighbor Unreachability Detection” proporciona un mecanismo de detección de fallos en los vecinos o en los caminos hacia ellos. Esto requiere confirmación de los paquetes enviados a los vecinos para asegurar que llegan y se procesan correctamente. Este mecanismo utiliza dos tipos de confirmación: si es posible, obtiene la confirmación de protocolos de capas superiores que sepan que los datos enviados anteriormente han sido entregados correctamente (por ejemplo si se han recibido confirmaciones recientemente). Si esto no es posible, el nodo envía un mensaje de Neighbor Solicitation (a la dirección unicast almacenada en caché) para obtener un Neighbor Advertisement que confirme que el siguiente salto está activo. Para no llenar la red con tráfico innecesario, estos mensajes de prueba son enviados solamente a vecinos con los cuales el nodo tiene una comunicación frecuente.

En caso de que un router quede inalcanzable el nodo tratará de elegir una

nueva ruta, y en caso de que sea un host el que quede inalcanzable, se procederá a una nueva resolución de dirección de enlace (por si el destino hubiera cambiado su dirección de enlace).

- **Detección de direcciones duplicadas.**

  Los nodos son capaces de evitar la utilización de una dirección ya en uso en la autoconfiguración de dirección de su interfaz. Se envía para ello un mensaje de Neighbor Solicitation y si nadie lo responde, el nodo asumirá que la dirección que desea colocar en su interfaz no está ocupada.

- **Redirección de los paquetes provenientes de un nodo hacia otro mejor primer salto.**

  Los routers utilizan mensajes de Redirect para avisar a los hosts sobre mejores primeros saltos hacia un determinado destino. Además, puede darse el caso en que un nodo del enlace no quede “cubierto” por ninguno de los prefijos anunciados por los routers, de manera que será considerado por el resto de nodos como un destino off-link. En este caso, los routers podrán enviar paquetes de redirección para informar a los nodos que intenten comunicar con este de que se trata de un destino on-link.

- **Actualización de direcciones inválidas.**

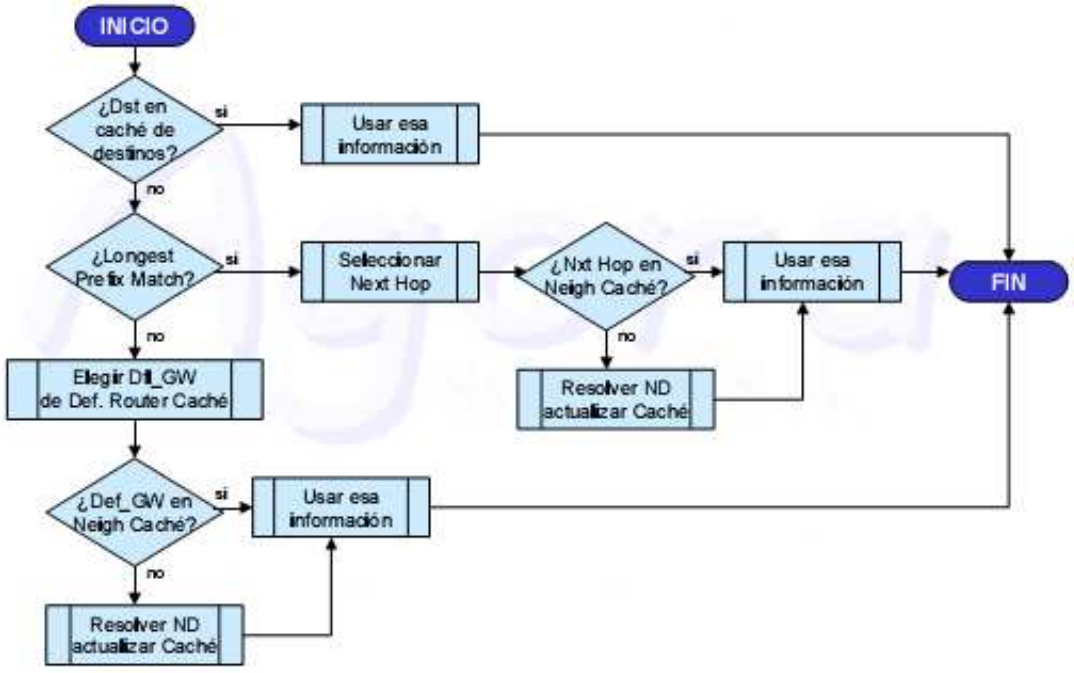
  Un nodo cuya dirección de enlace haya sido cambiada puede enviar a la dirección all-nodes multicast unos pocos paquetes de Aviso de Vecino (no solicitados por ningún otro vecino de la red) para actualizar rápidamente la dirección de enlace inválida en la caché del vecindario. Sin embargo, esto es tan solo un funcionamiento añadido que no asegura que todos los vecinos actualicen su caché. El algoritmo “Neighbor Unreachability Detection” será el encargado de asegurar que todos los nodos descubran la nueva dirección, aunque el retardo será algo mayor.

5.3. Estructuras de datos en los hosts

Cada host deberá mantener en memoria ciertos datos para interactuar con los nodos vecinos:

- **Caché de vecinos.** Una lista de los vecinos hacia los que se ha enviado tráfico recientemente. Para cada vecino se guarda información sobre su dirección unicast link-local, un flag para distinguir entre hosts y routers, etc. Contiene además información utilizada por el algoritmo "Neighbor Unreachability Detection".
- **Caché de destinos.** Una lista de los destinos a los que se ha enviado tráfico recientemente. Se almacenan tanto destinos on-link como off-link, mapeando cada dirección IP de destino junto con la dirección IP del siguiente salto (que podrá ser el propio destino).
- **Lista de Prefijos.** Una lista generada a partir de los avisos de los routers con contiene cada uno de los prefijos anunciados para el enlace junto con un tiempo de expiración. Una vez agotado el temporizador, el prefijo quedará invalidado. El valor del temporizador puede fijarse como "infinito" y permanecer así a no ser que se modifique con un posterior aviso de router.
- **Lista de Routers por defecto.** Una lista de los routers hacia los que los paquetes pueden ser enviados. Esta lista está enlazada con la caché de vecinos. Cada entrada tendrá asociado un contador de expiración para borrar las entradas no anunciadas en un periodo largo de tiempo.

Dada esta información, el diagrama de estados para cada intento de transmisión por parte de un host, quedará como muestra la siguiente figura:



5.4. Comparación con IPv4

El protocolo ND corresponde a una combinación de los protocolos de IPv4 ARP, ICMP(v4) Router Discovery (RDISC) e ICMP(v4) Redirect. De hecho, en IPv4 no existía un método estándar mediante el cual fijar mecanismos de detección de vecinos.

El protocolo ND provee multitud de mejoras con respecto al conjunto de protocolos utilizados para funciones parecidas en IPv4:

1. Router Discovery es parte del protocolo, por lo que no es necesario que los hosts tengan ningún conocimiento de los protocolos de enrutamiento.

2. Los mensajes Router Advertisement contienen información acerca de sus direcciones de enlace, por lo que no son necesarios intercambios adicionales de paquetes para que el resto de nodos tengan conocimiento de estas.

3. La información sobre el prefijo del enlace que contienen estos mismos mensajes evitan tener que implementar otros mecanismos para configurar las máscaras de la red.

4. Estos mensajes proporcionan también servicios de autoconfiguración de dirección.

5. Los routers pueden indicar la MTU a los hosts del enlace, asegurándose así de que todos los nodos usen el mismo valor en enlaces que no tengan una MTU bien definida.

6. Los mensajes Redirect contienen la dirección de enlace del nuevo Primer Salto, por lo que de nuevo no será necesario un posterior intercambio de paquetes para obtenerla.

7. Un mismo enlace puede ser asociado a múltiples prefijos. Por defecto, los hosts aprenden todos los prefijos del enlace al que están conectados a partir de los Router Advertisement. En algunos casos, los routers pueden estar configurados para omitir algunos de los prefijos (o todos) en sus avisos, de manera que los hosts asuman que destinos no se encuentran en su enlace y manden el tráfico a través de los routers, que serán los encargados de redireccionarlos apropiadamente.

8. El receptor de un paquete de Redirección asume que el nuevo Next-Hop pertenece al enlace. En IPv4, los hosts ignoraban los paquetes de redirección si consideraban que el siguiente salto no estaba en el enlace, basándose en su máscara. Esto era un problema en enlaces de medio compartido o en los que no se soportase broadcast (ATM, Frame Relay, AX.25...).

9. El algoritmo "Neighbor Unreachability Detection" es también parte del protocolo, lo que aumenta la robustez del reparto de paquetes frente a fallos de routers o cambios de direcciones de enlace por parte de los nodos. Esto permite, por ejemplo, que nodos móviles puedan cambiar de red (abandonando el vecindario) sin perder conectividad como ocurría con las cachés de ARP.

10. El uso de direcciones link-local para identificar unívocamente los routers hace posible que los hosts mantengan las asociaciones de los routers en el caso de que se establezca un nuevo prefijo global.

11. ND es inmune a los ataques por parte de usuarios no pertenecientes al enlace que envíen, accidental o intencionadamente, mensajes de ND con el campo Hop-Limit igual a 255 (debido a que los mensajes deberán proceder de direcciones de alcance local). En IPv4 estos eran capaces de enviar tanto mensajes ICMP de redirección como Avisos de Router.

6. AUTOCONFIGURACIÓN (RFC 2462)

El proceso de autoconfiguración incluye la creación de una dirección unicast link-local que no esté ya en uso en el mismo enlace, la determinación de la información a autoconfigurar (direcciones y/o de otro tipo) y, en caso de necesitar autoconfigurar una dirección distinta de la unicast link-local, la determinación de qué mecanismo, stateless, stateful o ambos, se utilizará para su obtención.

Ambos mecanismos, stateless y stateful, son definidos por IPv6 para la autoconfiguración de dirección:

La autoconfiguración stateless no requiere configuración manual de los hosts, mínima (o nula) configuración de los routers y no necesita servidores adicionales. Este mecanismo, permite a los hosts generar su propia dirección usando una combinación de información local e información repartida por los routers. Los routers avisarán a los hosts de los prefijos de la subred (o subredes), mientras que los hosts generarán un "identificador de interfaz" que los identifique unívocamente en la subred. Con la combinación de ambas partes se formará la dirección. Por tanto, en ausencia de routers, los hosts solo podrán generar direcciones de enlace local. Sin embargo no es un problema, ya que estas son suficientes para la comunicación entre los nodos del mismo enlace.

En la autoconfiguración stateful, los hosts obtendrán las direcciones de interfaz, la información de configuración y los parámetros de un servidor. Los servidores mantendrán una base de datos que realice un seguimiento sobre qué direcciones han sido asignadas y a qué hosts.

Los dos mecanismos de autoconfiguración son complementarios. El stateless se utiliza en entornos en los que no sea necesario conocer con exactitud qué direcciones utiliza cada host mientras que sean únicas y enrutables. Por el contrario, la autoconfiguración stateful encuentra su uso en aquellos entornos en los que se requiera un fuerte control sobre la asignación de direcciones. Ambos pueden ser utilizados simultánamente en una misma red, y será el administrador de la red el que especifique que mecanismo se ha de utilizar mediante la configuración de los campos apropiados en los mensajes Router Advertisement.

7. Mecanismos de transición a IPv6

El cambio de IPv4 a IPv6 ya ha comenzado, pero no puede hacerse instantáneamente, sino que la implantación de IPv6 es paulatina y durante unos 20 años se espera que convivan ambos protocolos. Existe una serie de mecanismos que permitirán la convivencia y la migración progresiva tanto de las redes como de los equipos de usuario. Estas técnicas pueden ser utilizadas incluso de forma combinada.

7.1. Pila dual (RFC 2893)

La **pila** dual hace referencia a una *solución de nivel IP con pila dual*, que implementa las pilas de ambos protocolos, IPv4 e IPv6, en cada nodo de la red. Cada nodo de pila dual en la red tendrá dos direcciones de red, una IPv4 y otra IPv6.

- **Pros**: Fácil de desplegar y extensamente soportado.
- **Contras**: La topología de red requiere dos tablas de encaminamiento y dos procesos de encaminamiento. Cada nodo en la red necesita tener actualizadas las dos pilas.

7.2 Túneles

Los **túneles** permiten conectarse a redes IPv6 "saltando" sobre redes IPv4. Estos túneles trabajan encapsulando los paquetes IPv6 en paquetes IPv4 teniendo como siguiente capa IP el protocolo número 41. De esta manera, los paquetes IPv6 pueden ser enviados sobre una infraestructura IPv4.

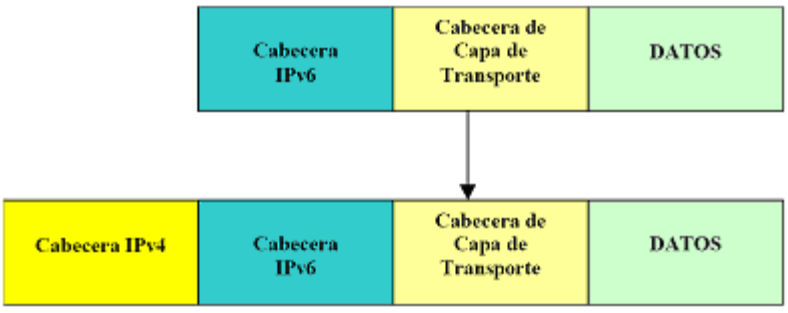
Los extremos finales del túnel son siempre los encargados de realizar la operación de encapsulado del paquete/es IPv6 en IPv4.

Estos túneles pueden ser utilizados de distintas formas:

- Router a router: Routers con doble pila (IPv4/IPv6) se conectan mediante una infraestructura IPv4 y transmiten tráfico IPv6. El túnel comprende un segmento que incluye la ruta completa, extremo a extremo, que siguen los paquetes.
- Host a router: Host con doble pila se conectan a un router intermedio (también con doble pila), alcanzable mediante una infraestructura IPv4. El túnel comprende el primer segmento de la ruta seguido por los paquetes.
- Host a Host: Host con doble pila interconectados por una infraestructura IPv4. El túnel comprende la ruta completa que siguen los paquetes.
- Router a Host: Routers con doble pila que se conectan a host también con doble pila. El túnel comprende el último segmento de la pila.

Los túneles se clasifican según el mecanismo por el que el nodo que realiza el encapsulado determina la dirección del nodo extremo del túnel. En los dos primeros casos (router a router y host a router), el paquete IPv6 es tunelizado a un router. El extremo final de este tipo de túnel, es un router intermedio que debe desencapsular el paquete IPv6 y reenviarlo a su destino final. En este caso, el extremo final del túnel es distinto del destino final del paquete, por lo que la dirección en el paquete IPv6 no proporciona la dirección IPv4 del extremo final del túnel. La dirección del extremo final del túnel ha de ser determinada a través de información de configuración en el nodo que realiza el túnel. Es lo que se denomina “túnel configurado”, describiendo aquel tipo de túnel donde el extremo final del túnel es explícitamente configurado.

En los otros casos (host a host y router a host), el paquete IPv6 es tunelizado, durante todo el recorrido, a su nodo destino. El extremo final del túnel es el nodo destino del paquete, y por tanto, la dirección IPv4 está contenida en la dirección IPv6. Este caso se denomina “túnel automático”.



El desencapsulado, en el extremo final del túnel, realiza la función opuesta.

7.3 6 over 4 (Transmisión de IP6 sobre dominios IPv4, RFC 2529)

Este mecanismo permite a hosts IPv6 aislados, sin conexión directa a routers IPv6, ser totalmente funcionales como dispositivos IPv6.

Para ello se emplean dominios IPv4 que soportan multicast como su enlace local virtual. Es decir, usamos multicast IPv4 como su "ethernet virtual". De esta forma, estos host IPv6 no requieren direcciones IPv4 compatibles, ni túneles configurados.

Los extremos finales del túnel se determinan mediante ND. Es imprescindible que la subred IPv4 soporte multicast.

7.4 6 to 4 (Conexión de dominios IPv6 sobre redes IPv4)

Es un mecanismo para asignar un prefijo de dirección IPv6 a cualquier sitio que tenga al menos una dirección IPv4 pública.

De esta forma, dominios o host IPv6 aislados, conectados a infraestructuras IPv4 (sin soporte para IPv6), pueden comunicar con otros dominios o host IPv6 con una configuración manual mínima.

Este mecanismo funciona aún cuando la dirección IPv4 pública es única y se accede a la red mediante mecanismos NAT, que es el caso mas común en las redes actuales para el acceso a Internet a través de ISP’s.

7.5 “Tunnel Server” y “Tunnel Broker”

El “tunnel broker”es el lugar donde el usuario se conecta para registrar y activar “su túnel”. El “broker” gestiona (crea, modifica, activa y desactiva) el túnel en nombre del usuario.

El “tunnel server” es un router con pila doble (IPv4 e IPv6), conectado a Internet, que siguiendo órdenes del “broker” crea, modifica o borra los servicios asociados a un determinado túnel/usuario.

El mecanismo para su configuración es tan sencillo como indicar, en un formulario Web, datos relativos al S.O. La dirección IPv4, un “apodo” para la máquina, y el país donde está conectada. El servidor de túneles crea los registros DNS, el extremo final del túnel, y genera un script para la configuración del cliente.

Estos mecanismos se hacen indispensables para labores de investigación, dado que se requisen direcciones IPv6 y nombres DNS permanentes.

Hemos encontrado ejemplos de estos sistemas en www.freenet6.net y carmen.cselt.it/ipv6/download.html**.**

PARTE PRÁCTICA

1. SOPORTE IPv6

La parte práctica la llevaremos a cabo en un entorno Linux, y el primer paso será comprobar que el módulo de IPv6 es soportado por nuestra versión del Kernel y si está o no cargado.

En una máquina UNIX, el kernel debe ser al menos de la serie 2.6.x para soportar IPv6. Lo comprobaremos en consola:

```
[rba10@tlm56 rba10]$ uname -r
2.6.10-1.771_FC2smp
```

Ahora pasaremos a comprobar si el módulo de IPv6 ha sido cargado, mediante:

```
[rba10@tlm56 rba10]$ lsmod | grep ipv6
```

Además, podemos buscar algunos archivos relacionados con este módulo, como el archivo /proc/net/if_inet6. Su existencia también nos indicaría que el modulo de IPv6 ha sido cargado en el sistema.

Por último, además de tener el sistema operativo preparado, debemos comprobar que contemos con las utilidades adecuadas para configurar nuestra máquina en IPv6. Trabajaremos con utilidades como ifconfig o route, por lo que nos aseguraremos de que estén adaptadas a ipv6. Al consultar la ayuda de estas utilidades (con el parámetro -?), encontramos alusiones a IPv6, por lo que están preparadas.

2. CONFIGURANDO TU INTERFAZ DE RED PARA IPv6

Una vez comprobado que el equipo soporta IPv6 vamos a configurar uno de sus interfaces. Para activar un interfaz usamos el comando:

ifconfig <interface> up

y para desactivarlo:

ifconfig <interface> down

En nuestro caso, activamos el interfaz eth0 (haciendo uso del comando sudo por cuestión de permisos) de la siguiente manera:

sudo ifconfig eth0 up

Para añadir nuevas direcciones se utiliza:

ifconfig <interface> add <addr>/<prefixlen>

y para eliminarlas:

ifconfig <interface> del <addr>/<prefixlen>

Por ejemplo:

sudo ifconfig eth0 add fe80::1/64 (en ordenador A)
sudo ifconfig eth1 add fe80::2/64 (en ordenador B)

Si comprobamos ahora las direcciones definidas en el PC A por ejemplo, obtenemos:

```
[rba@localhost rba]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:5E:3E:34:8C
          inet6 addr: fe80::1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:10
          collisions:0 txqueuelen:1000
          RX bytes:776 (776.0 b)  TX bytes:1354 (1.3 Kb)
          Interrupt:9 Base address:0xa400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1814 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1814 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1301452 (1.2 Mb)  TX bytes:1301452 (1.2 Mb)
```

Podemos ver como el alcance de la dirección asignada a nuestro interfaz ha quedado fijado como de enlace. Esto se debe a que los primeros bits utilizados en la dirección (FE80) corresponden con las direcciones link-local unicast, como se dijo en la sección 2.4 de la parte teórica.

3. AUTOCONFIGURACIÓN DE INTERFAZ

Podemos comprobar también como un interfaz genera automáticamente una dirección unicast link-local cuando es activado. Así , tras levantar el interfaz eth0 (sudo ifconfig eth0 up), comprobamos:

```
[rba@localhost rba]$ sudo ifconfig eth0 up
[rba@localhost rba]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:5E:3E:40:EC
          inet6 addr: fe80::20a:5eff:fe3e:40ec/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:2
          collisions:0 txqueuelen:1000
          RX bytes:314 (314.0 b)  TX bytes:168 (168.0 b)
          Interrupt:9 Base address:0xa400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1702 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1702 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1723538 (1.6 Mb)  TX bytes:1723538 (1.6 Mb)
```

Vemos que como se dijo en la sección 2.5.1., el identificador de la dirección se forma a partir de la dirección MAC del interfaz. Además, el prefijo es fe80:: ya que no hay routers que anuncien ningún otro prefijo en nuestra red, por lo que el host configura su dirección para poder trabajar en un enlace.

4. RED DE DOS HOSTS TRABAJANDO EN IPv6

Con los interfaces ya configurados, vamos a unirlos en red a través de un switch (o directamente con un cable cruzado) y haremos uso del comando ping6 para realizar la petición de eco (equivalente al comando ping de IPv4) y comprobar así la conexión entre ambos.

Por ejemplo, en el ordenador A escribimos:

ping6 -I eth0 fe80::2

y así recibimos las respuestas de eco del ordenador B:

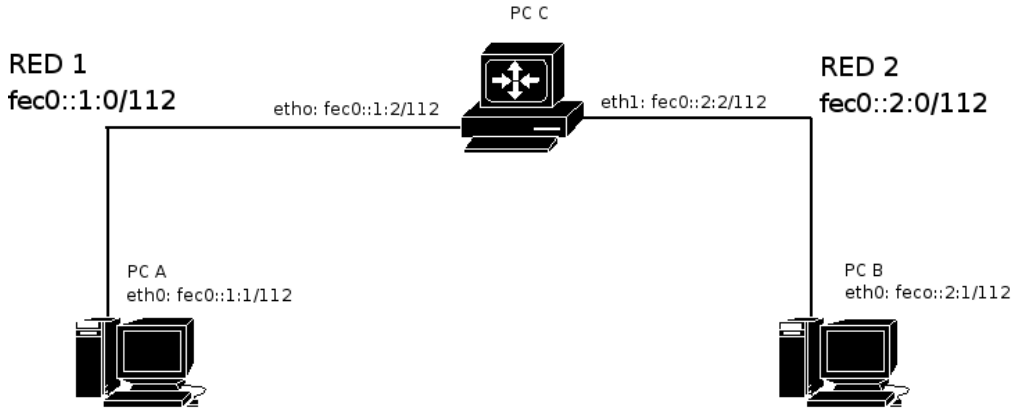
```
[rba@localhost rba]$ ping6 -I eth0 fe80::2
PING fe80::2(fe80::2) from fe80::1 eth0: 56 data bytes
64 bytes from fe80::2: icmp_seq=0 ttl=64 time=0.929 ms
64 bytes from fe80::2: icmp_seq=1 ttl=64 time=0.167 ms
64 bytes from fe80::2: icmp_seq=2 ttl=64 time=0.165 ms
64 bytes from fe80::2: icmp_seq=3 ttl=64 time=0.164 ms

--- fe80::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.164/0.356/0.929/0.330 ms, pipe 2
```

La opción -I define el interfaz local a través del que se envía la petición de eco, y es necesaria cuando el destinatario es una dirección multicast o una dirección unicast link-local o site-local.

5. INTERCONEXIÓN DE REDES A TRAVÉS DE UN PC FUNCIONANDO COMO ROUTER

Para esta prueba haremos uso de 3 PC's y uno de ellos trabajará como router. Definimos para ello 2 redes distinguidas, y las direcciones, teniendo en cuenta que deberán ser de alcance de sitio (site-local unicast), y conexiones, quedarán como muestra el siguiente gráfico:



A continuación añadimos las rutas por defecto: en el PC A será el interfaz eth0 de C y en el PC B será el interfaz eth1. Para ello utilizamos el comando route:

route -A inet6 add <ipv6network>/<prefixlen> gw <ipv6addr> [dev <device>]

La opción -A indica la familia de direcciones (en nuestro caso las direcciones de IPv6), y si la dirección del gateway fuese una dirección link-local unicast, necesitaríamos incluir la opción dev indicando el interfaz de fuente como en el caso del ping.

Así conseguiremos que los paquetes dirigidos a direcciones de la <ipv6network> sean dirigidos al la <ipv6addr>, que será el interfaz del router en nuestra red. En nuestro caso haremos:

PC A:

sudo route -A inet6 add fec0::2:0/112 gw fec0::1:2

Así, la tabla de rutas del PC A quedará:

```
[rba@localhost rba]$ route -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flags Metric Ref    Use Iface
::1/128                        *                          U     0      78       2 lo
fe80::20a:5eff:fe3e:343e/128   *                          U     0      2        2 lo
fe80::/64                      *                          U     256    0        0 eth0
fec0::1:1/128                  *                          U     0      24       2 lo
fec0::1:2/128                  fec0::1:2                  UC    0      7        0 eth0
fec0::1:0/112                  *                          U     256    0        0 eth0
fec0::2:0/112                  fec0::1:2                  U     1      0
```

0 eth0
fec0::2:0/112 fec0::1:2 UG 1 0 0 eth0
ff00::/8 * U 256 0 0 eth0

PC B:

sudo route -A inet6 add fec0::1:0/112 gw fec0::2:2

Ahora si comprobamos la comunicación (mediante peticiones de eco) entre PC A con C, PC B con C o PC C con A y B obtendremos respuesta, pero la comunicación entre PC A y B aún no será posible. Esto es porque por defecto, el PC C desechará todos los paquetes que no sean para él, no pudiendo por tanto encaminarlos por el interfaz apropiado. Para que no los descarte habrá que activar la opción de encaminamiento. Para ello, deberemos poner a 1 la variable lógica encargada de ello. Podremos encontrarla con el comando:

sysctl -A | grep forwarding

Vemos en el listado las referentes al encaminamiento de IPv6:

```
[rba@localhost rba]$ sysctl -A | grep forwarding
net.ipv6.conf.eth0.forwarding = 0
error: permission denied on key 'kernel.cad_pid'
error: permission denied on key 'kernel.cap-bound'
net.ipv6.conf.eth3.forwarding = 0
net.ipv6.conf.default.forwarding = 0
net.ipv6.conf.all.forwarding = 0
net.ipv6.conf.lo.forwarding = 0
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.conf.lo.forwarding = 0
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 0
```

En nuestro caso, el flag de interés será el de activación de forwarding para todos los interfaces. Un valor de 1 indicaría que el PC está actuando como router de IPv6, y un valor de 0 indicaría lo contrario. Por tanto, deberemos poner a uno esta variable. Para ello hacemos uso de la siguiente línea de comando (precedida de sudo en nuestro caso por no tener permisos de administrador):

sudo sysctl -w net.ipv6.conf.all.forwarding=1

El parámetro -w indica que queremos modificar su valor. Ahora ya podemos realizar una petición de eco desde el PC A al PC B y el PC C no descartará el paquete si no que lo encaminará por el interfaz adecuado.

Comprobamos que la variable contenga el valor 1:

```
[rba@localhost rba]$ sysctl -A | grep forwarding
net.ipv6.conf.eth0.forwarding = 1
net.ipv6.conf.eth1.forwarding = 1
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.lo.forwarding = 1
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.conf.lo.forwarding = 0
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 0
```

6. Router CISCO

Por último, pasaremos a realizar en la práctica alguna de las tareas soportadas por el Router CISCO 1760 (cabe destacar que IPv6 solo es soportado por estos en el laboratorio, y solo los correspondientes a las torres de la derecha), identificado en el servidor de consola (PC-SC) como Router1.

Su versión de firmware es la 12.3(2) y está compilada para que soporte IPv6. Para comprobar esto y realizar la configuración apropiada en el router, haremos uso de minicom para acceder al puerto de consola del router a través de un enlace serie desde el puerto serie del servidor de consola (PC-SC). Escribimos en el terminal:

minicom router1

Ahora, tras comprobar que los parámetros de comunicación con el puerto del router son los apropiados (9600 bps, 8 bits de datos, 1 de parada, sin paridad) a través de la opción P (Crtl+A P) del minicom, nos encontraremos la línea de comandos con el prompt "Router> "

Tecleando:

Router>**show version**

Cisco Internetwork Operating System Software
IOS (tm) C1700 Software (C1700-SV8Y7-M), Version 12.3(2)XA, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
Synched to technology version 12.3(1.6)T
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Fri 08-Aug-03 12:57 by ealyon
Image text-base: 0x80008120, data-base: 0x816DC430

ROM: System Bootstrap, Version 12.2(7r)XM2, RELEASE SOFTWARE (fc1)
ROM: C1700 Software (C1700-SV8Y7-M), Version 12.3(2)XA, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)

Router uptime is 1 week, 6 days, 1 hour, 31 minutes
System returned to ROM by reload
System image file is "flash:c1700-sv8y7-mz.123-2.XA.bin"

cisco 1760 (MPC860P) processor (revision 0x500) with 60192K/5344K bytes of memory.
Processor board ID FOC08153HFX (3040516870), with hardware revision 0000
MPC860P processor: part number 5, mask 2
Bridging software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
1 ATM network interface(s)
2 Voice FXO interface(s)
2 Voice FXS interface(s)
32K bytes of non-volatile configuration memory.
32768K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102

obtenemos información acerca del router, como la versión de su firmware.

6.1. Configuración del interfaz FastEthernet

Vamos a configurar una dirección IPv6 en el interfaz Fast Ethernet del router. Para ello, escribimos:

Router>**enable** *(entramos en modo privilegiado)*

Router#**configure terminal** *(entramos en el modo de configuración)*

Router(config)#**interface FastEthernet 0/0** *(configuración del interface Fast Ethernet)*

Ahora debemos "levantar" el interface, pero antes debemos unirlo a una red para que no se desactive automáticamente. Lo conectaremos mediante a un cable recto a un switch, donde también conectaremos un PC con un interfaz configurado para IPv6.

Router(config-if)#**no shutdown** *(levantamos el interfaz)*

Para dar una dirección al interfaz contamos con cuatro opciones:

Router(config-if)#ipv6 enable *(autoconfigura una dirección link-local)*

Router(config-if)#ipv6 address autoconfig *(igual que el anterior)*

Router(config-if)#ipv6 address <fe8x:x:x:x:x:x:x> link-local

Router(config-if)#ipv6 address <x:x:x:x:x:x:x:x>/<prefix>

Con las dos últimas opciones configuramos manualmente una dirección link-local unicast o global unicast respectivamente, fijando además con la segunda de ellas un prefijo de red que podrá ser anunciado por el router.

Una vez configurada la dirección, desde el modo privilegiado (saldremos de los modos con el comando "exit") podemos comprobar la configuración del interfaz tecleando:

Router#**show ipv6 interface**

```
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:24FF:FEAF:16C8
  Global unicast address(es):
    FEC0::2, subnet is FEC0::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
    FF02::1:FFAF:16C8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
```

```
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Podemos comprobar de nuevo el buen funcionamiento de la red realizando un ping desde el ordenador a alguna de las direcciones. Por ejemplo:

ping6 -I eth0 fec0::2

6.2. Interconexión de redes virtuales

A continuación, crearemos una topología que contendrá dos redes ipv6 distintas interconectadas mediante el router CISCO. Al tener un único interfaz FastEthernet, lo primero que necesitaremos será definir dos subinterfaces asignados a dos VLAN's distintas para crear dos interfaces virtuales.

Tras conectar el interfaz del router al interfaz 1 del switch1 (de CISCO), lo siguiente que debemos hacer es configurar dos subinterfaces en el router1. Desde el modo de configuración de terminal tecleamos:

Router(config)#**interface FastEthernet 0/0.1**

y lo configuramos para que soporte tramas IEEE 802.1Q, asignándole el identificador de VLAN (VLAN id) 10 con el siguiente comando:

Router(config-subif)#**encapsulation dot1Q 10**

Repetiremos los pasos para asignar el otro subinterfaz a la VLAN 20:

Router(config)#**interface FastEthernet 0/0.2**
Router(config-subif)#**encapsulation dot1Q 20**

Configuraremos cada uno de los interfaces virtuales, asignándoles las direcciones fec0:0:0:1::2/64 y fec0:0:0:2::2/64 respectivamente, siguiendo los pasos descritos antes, y habilitaremos el encaminamiento de paquetes unicast:

Router(config)#**interface FastEthernet 0/0.1**
Router(config-subif)#**ipv6 address fec0:0:0:1::2/64**
Router(config-subif)#**exit**
Router(config)#**interface FastEthernet 0/0.2**
Router(config-subif)#**ipv6 address fec0:0:0:2::2/64**
Router(config-subif)#**exit**
Router(config)#**ipv6 unicast-routing**
Router(config)#**exit**
Router#**show ipv6 interface**

```
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

FastEthernet0/0.1 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20F:24FF:FEAF:2C86
Global unicast address(es):
FEC0:0:0:1::2, subnet is FEC0:0:0:1::/64
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF00:2
FF02::1:FFAF:2C86
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
FastEthernet0/0.2 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20F:24FF:FEAF:2C86
Global unicast address(es):
FEC0:0:0:2::2, subnet is FEC0:0:0:2::/64
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF00:2
FF02::1:FFAF:2C86
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.

Pasamos ahora a configurar el switch1 para asignar las diferentes VLANs a sus interfaces. El resultado final debe ser que el interfaz 1 del switch funcione en modo trunk (todos los paquetes dirigidos a cualquier VLAN serán encaminados por este interfaz) y que los interfaces 2 y 3 tengan configuradas las VLAN's 10 y 20 respectivamente, donde conectaremos 2 PC's con sus interfaces configurados en ipv6. Así, toda comunicación entre estos PC's deberá pasar forzosamente a través del router (por encontrarse en redes virtuales distintas) y podremos comprobar el correcto encaminamiento del router.

Para configurar el switch, haremos uso, al igual que con el router, del minicom. Tecleamos:

minicom switch1

Una vez que aparezca el prompt, entramos en el modo de configuración del primer interfaz del router, para configurarlo en modo trunk:

Switch>**enable**
Switch#**configure terminal**

Switch(config)#**interface FastEthernet 0/1**
Switch(config-if)#**switchport mode trunk**

Ahora, configuraremos el interfaz FastEthernet 0/2 para que pertenezca a la VLAN 10:

Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet 0/2**
Switch(config-if)#**switch access vlan 10**

y repetimos los pasos para asignar la VLAN 20 al tercer interfaz:

Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet 0/3**
Switch(config-if)#**switch access vlan 20**

Comprobamos todas las asignaciones de VLANs desde el modo privilegiado:

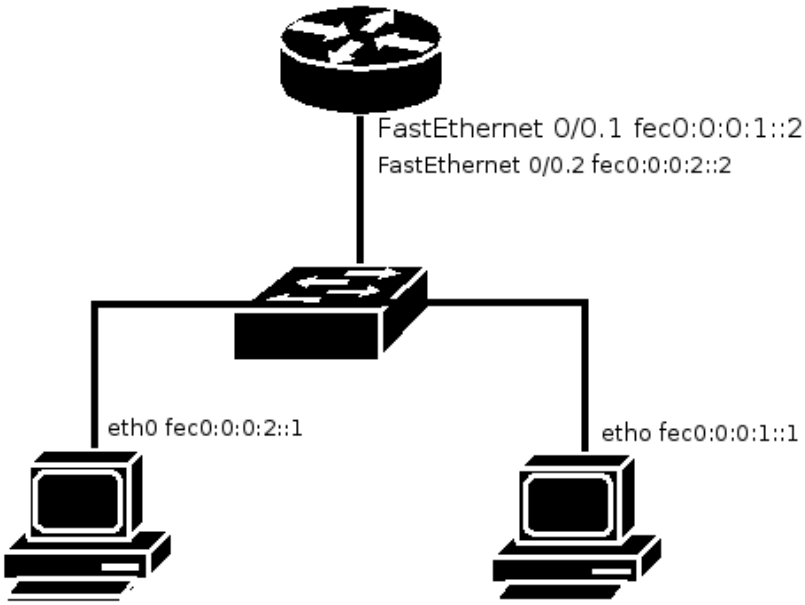
Switch#show vlan

VLAN	Name	Status	Ports
1	default	active	Fa0/4, Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24
2	VLAN0002	active	
10	**accounting**	**active**	**Fa0/2**
20	**marketing**	**active**	**Fa0/3**
30	engineering	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

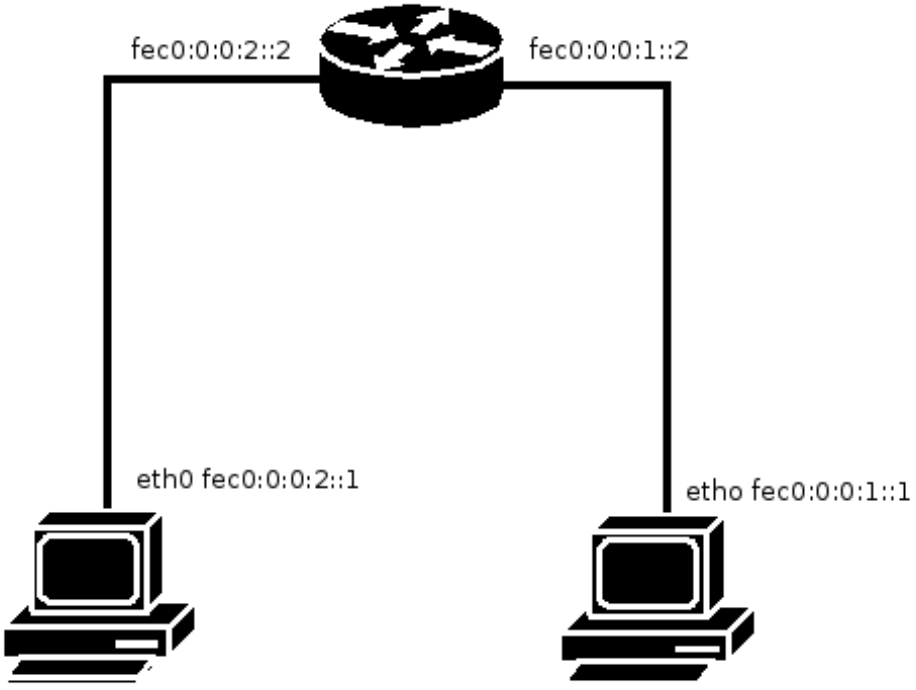
Podemos ver como han sido correctamente asignados los interfaces dos y tres a sus correspondientes VLAN's, y el interfaz 1, al haber sido configurado en modo trunk, pertenece a todas ellas y por lo tanto no aparece listado.

Por último, configuraremos los interfaces de 2 PC's, asignándoles direcciones que se encuentren en la misma red que la asignada al subinterfaz del router al que estén conectados virtualmente, y les asignaremos rutas por defecto hacia dicha dirección.

El montaje físico de la red será el mostrado en la siguiente figura:



Sin embargo, virtualmente la red desarrollada tendrá el siguiente aspecto:



Para comprobar la conectividad, utilizaremos la herramienta ping6 para llegar de un PC a otro.

6.3. Otros parámetros y opciones

Además de la interconexión de redes, el router CISCO 1760 nos ofrece varios parámetros de configuración que, gracias a Neighbor Discovery, serán difundidos por toda la red. Vamos por ejemplo a cambiar la MTU que anunciará y utilizará el router. Podemos observar en la información de interfaz obtenida anteriormente, que la MTU configurada es de 1500 bytes. La Configuraremos a 1300 bytes. Para ello, desde el modo de configuración de interfaz, escribimos:

Router(config-if)#**ipv6 mtu 1300**

y lo comprobamos mostrando de nuevo la información del interfaz:

```
Router#show ipv6 interface
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2
  Global unicast address(es):
    FEC0:0:0:1::2, subnet is FEC0:0:0:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
  MTU is 1300 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Otros parámetros interesantes que difundirán los routers son, como ya se dijo, los prefijos de red en los que los hosts podrán autoconfigurar sus direcciones. Procederemos ahora a observar dicha autoconfiguración. Con el interfaz del router configurado con alguna dirección global con su correspondiente prefijo, conectamos un PC al mismo conmutador que el router, y comprobamos como al levantar uno de sus interfaces, el host autoconfigura una dirección global en la red anunciada por el router (por ejemplo fec0:0:0:1::/64).

Toda la información repartida por los routers a la red, viaja dentro de los paquetes de ICMP, “Router Advertisement”, por lo que otra configuración interesante es el intervalo de tiempo transcurrido entre envíos de estos mensajes. Probaremos a establecer un temporizador de 15 segundos. Para ello:

```
Router>enable
Router#configure terminal
Router(config)#interface FastEthernet 0/0
Router(config-if)#ipv6 nd ra-interval 15
```

Ahora, mediante ethereal, escuchando en el interfaz activado en algún host y conectado al router, podremos ver como cada 15 segundos llega un nuevo paquete ICMP Router Advertisement, con la información configurada en el router. El PC guardará también la dirección del router en su tabla de “Next-Hop”, y almacenará la dirección del vecino, marcándolo como router, y estableciendo el tiempo de expiración de la entrada según el valor que encuentre en los paquetes. Podemos

encontrar como configurar todos estos parámetros otras muchas opciones con el comando ayuda (?) del router:

Router(config)#**ipv6 ?** *(desde el modo de configuración del terminal)*

```
  access-list        Configure access lists
  cef                Cisco Express Forwarding for IPv6
  hop-limit          Configure hop count limit
  host               Configure static hostnames
  icmp               Configure ICMP parameters
  local              Specify local options
  mfib               IP MFIB forwarding
  multicast-routing  Enable IP multicast
  neighbor           Neighbor
  ospf               OSPF
  pim                Configure Protocol Independent Multicast
  prefix-list        Build a prefix list
  route              Configure static routes
  router             Enable an IPV6 routing process
  source-route       Process packets with source routing header options
  unicast-routing    Enable unicast routing
```

Router(config-if)#**ipv6 ?** *(desde el modo de configuración de interfaz)*

```
IPv6 interface subcommands:
  address        Configure IPv6 address on interface
  cef            Cisco Express Forwarding for IPv6
  enable         Enable IPv6 on interface
  mfib           IP MFIB forwarding
  mld            interface commands
  mtu            Set IPv6 Maximum Transmission Unit
  nd             IPv6 interface Neighbor Discovery subcommands
  ospf           OSPF interface commands
  pim            PIM interface commands
  redirects      Enable sending of ICMP Redirect messages
  rip            Configure RIP routing protocol
  traffic-filter Access control list for packets
  unnumbered     Preferred interface for source address selection
  verify         Enable per packet validation
```

BIBLIOGRAFÍA

1. RFC's:

"RFC 2461 - "Neighbor Discovery and Autoconfiguration".
"RFC 2462 - Stateless address Autoconfiguration".
"RFC 3587 - IPv6 Global Unicast Address Format".
"RFC 4291 - IP Version 6 Addressing Architecture".
"RFC 4443 – ICMPv6".

(www.ietf.org/rfc.html)

2. Sylvia Hagen, "IPv& Essentials", O'REILLY, 2002.

3. Linux IPv6 HOWTO.
(http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Linux+IPv6-HOWTO.html)

4. Temario y plan de prácticas de "Laboratorio de Programación de Redes".
(https://www.tlm.unavarra.es/~daniel/docencia/lpr/lpr04_05/practicas.html)

5. Start Here: Cisco IOS Software Release Specifics for IPv6 Features
(http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_configuration_guide_chapter09186a00801d65ed.html)

6. ConsulIntel: Tutorial de IPv6.
(http://www.consulintel.es/Html/ForoIPv6/Documentos/Tutorial%20de%20IPv6.pdf)

7. Evolución de internet desde IPv4 hasta IPv6.
(http://internetng.dit.upm.es/ponencias-jing/2002/fernandez/Evolucion-IPv4-IPv6-David-Fernandez.PDF)

8. Luis Peralta, "IPv6 ".
(http://spisa.act.uji.es/~peralta/static/ipv6/ipv6.pdf)

9. IPv6 Tunnel Broker.
(http://www3.ietf.org/proceedings/99nov/I-D/draft-ietf-ngtrans-broker-02.txt)

10. Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels.
(http://www3.ietf.org/proceedings/99jul/I-D/draft-ietf-ngtrans-6to4-02.txt)