

# IPv6 @ UJI

Luis Peralta  
<http://spisa.act.uji.es/~peralta/ipv6>

1 de octubre de 2002



# Índice general

<b>1. Introducción a IPv6</b>	<b>5</b>
<b>2. El protocolo IPv6</b>	<b>7</b>
2.1. La cabecera IPv6 . . . . .	7
2.2. El campo de siguiente cabecera (Next Header field) . . . . .	8
<b>3. Arquitectura de direccionamiento</b>	<b>11</b>
3.1. Direccionamiento IPv6 . . . . .	11
3.2. Modelos de direccionamiento . . . . .	12
3.3. Ámbitos . . . . .	12
3.4. Nomenclatura de las direcciones . . . . .	13
3.5. Nomenclatura de los prefijos . . . . .	14
3.6. Representación de los tipos de direcciones . . . . .	14
3.7. Direcciones unicast . . . . .	14
3.7.1. Identificadores de interfaz . . . . .	16
3.7.2. La dirección no específica . . . . .	16
3.7.3. La dirección de loopback . . . . .	16
3.7.4. Direcciones IPv6 con direcciones IPv4 embebidas . . . . .	17
3.7.5. Direcciones globales agregables . . . . .	17
3.8. Direcciones anycast . . . . .	17
3.9. Direcciones multicast . . . . .	17
3.10. Requerimientos de nodo . . . . .	18
<b>4. Autoconfiguración</b>	<b>21</b>
4.1. Objetivos del diseño . . . . .	22
4.2. El protocolo . . . . .	22
4.3. Creación de las direcciones de enlace local (link-local) . . . . .	23
4.4. Creación de direcciones globales y de 'sitio' local (site-local) . . . . .	23
4.5. Consideraciones de seguridad . . . . .	24
<b>5. Movilidad</b>	<b>27</b>
5.1. Operación . . . . .	27
5.2. Cabeceras adicionales . . . . .	28
5.3. Consideraciones de seguridad . . . . .	29

<b>6. Estrategias de implantación</b>	<b>31</b>
6.1. Túneles	31
6.1.1. Túneles estáticos	31
6.1.2. 6to4	32
6.1.3. 6over4	32
6.1.4. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)	32
6.2. Comunicación entre nodos	32
6.2.1. Doble pila	33
6.2.2. Stateless IP/ICMP Translation Algorithm (SIIT)	33
6.2.3. Network Address Translation - Protocol Translation (NAT-PT)	33
6.2.4. Bump in the Stack (BIS)	33
6.2.5. SOCKS64	33

# Capítulo 1

## Introducción a IPv6

El nacimiento de este nuevo protocolo no ha venido solo propiciado por la escasez de direcciones IPv4 en estos momentos, sino que además se añaden nuevas características y se mejoran las existentes. Sobre IPv4 las tablas de rutas de los routers se están haciendo gigantescas, tanto el multi-homing como la movilidad son tareas excesivamente complejas. Las nuevas necesidades del usuario no pueden ser satisfechas de forma sencilla: seguridad, movilidad y calidad de servicio (QoS) entre otras. De todas estas razones, la única que no tiene alternativa sobre IPv4 es el agotamiento de direcciones: en la práctica las  $2^{32}$  direcciones quedan restringidas a la configuración flexible de las subredes, con lo que el número de direcciones asignado de forma eficiente se nos queda en tan solo 200 millones [Hui94].

A continuación detallaremos un poco más cada una de estas nuevas características.

### Aumento del espacio de direcciones

El protocolo IPv4 que forma la Internet de hoy en día está basado en una arquitectura que utiliza direcciones de 32 bits. Con la nueva versión del protocolo, las direcciones constan de 128 bits. Esto significa, entre otras cosas, que soluciones al agotamiento de direcciones IPv4, como el NAT, no serán necesarias. Podemos decir que una “desventaja” de estas nuevas direcciones es su dificultad para recordarlas dado su tamaño: `3ffe:3330:2:0:2a0:c9ff:fe10:cb02` podría ser tranquilamente nuestra dirección IPv6. Es de suponer que el servicio DNS tendrá más importancia aún.

### Autoconfiguración

Pinchar y funcionar. Cuando un nodo se conecta a la red, éste recibe los datos necesarios para empezar a comunicarse por parte del router: dirección IPv6, máscara de red y rutas. Hay que recordar que este nuevo protocolo trata de simplificar. Con IPv4 tenemos el DHCP (Dynamic Host Configuration Protocol, Protocolo de Configuración Dinámica de Nodo) para conseguir algo equivalente.

## Movilidad

Con esta funcionalidad podremos “saltar” de una red a otra sin apenas percibir ningún cambio. Si bien esto ya era posible con IPv4 de una manera más bien ardua [Mon01], en IPv6 fue una de los requerimientos del diseño. Esta característica será de gran importancia cuando entren en funcionamiento las nuevas redes de telefonía con tecnología UMTS.

## Seguridad

Este fue otro de los requerimientos del diseño del nuevo protocolo: todas las aplicaciones se deben beneficiar de las facilidades de autenticación y encriptación de datos de forma transparente. El estándar escogido para esto fue IPsec [TDG98].

## Encaminamiento jerárquico

El encaminamiento bajo IPv6 es bastante similar al de IPv4 con CIDR, es decir, jerárquico y sin clases. Con esto se pretende conseguir que las entradas en las tablas de rutas en los backbones no abunden más de lo necesario. Al mismo tiempo, se consigue simplificar el enrutamiento y se espera que los routers sean más rápidos.

## Multi-Homing

Esta funcionalidad se consigue con direcciones anycast. Una dirección anycast identifica a un conjunto de distintos interfaces, encontrándose estos, por norma general, en distintos nodos. Un paquete a una dirección anycast será entregado a un solo miembro del conjunto. En principio, el paquete será entregado al miembro más cercano según el concepto de cercano de los protocolos de encaminamiento.

## Calidad de servicio (QoS)

Si bien con IPv4 tenemos unos pocos bits para el control del tipo de servicio, ToS, con IPv6 disponemos de campos más amplios para definir la prioridad y flujo de cada paquete. Según el contenido de este campo, el router deberá darle un trato más o menos especial.

## Capítulo 2

# El protocolo IPv6

En 1992, el IETF llegó a la conclusión de que haría falta un sustituto del IPv4 y formó un grupo de trabajo con el nombre de IPNG que tendría la misión de desarrollar la siguiente generación del protocolo IP. De las distintas propuestas, el IETF escogió el Protocolo IP versión 6, que más tarde sería Draft Standard.

### 2.1. La cabecera IPv6

La cabecera de un paquete IPv6 es, sorprendentemente, más sencilla que la del paquete IPv4. Y recordemos que además la funcionalidad del protocolo IPv6 es mucho mayor.

La cabecera de un paquete IPv4 es variable, por lo que necesita un campo de tamaño o length. Sin embargo, para simplificar la vida de los routers, IPv6 utiliza un tamaño de cabecera fijo de 40 bytes, que componen un total de ocho campos:

- Versión (4 bits), sirve para que el router se entere de que es un paquete IPv6.
- Dirección origen y de destino (128 bits cada una), son las direcciones de los nodos IPv6 que realizan la comunicación.
- Clase de tráfico (8 bits), para poder diferenciar entre servicios sensibles a la latencia, como VoIP, de otros que no necesitan prioridad, como tráfico http.
- Etiqueta de flujo (20 bits), permite la diferenciación de flujos de tráfico. Esto tiene importancia a la hora de manejar la calidad de servicio (QoS)
- Siguiendo cabecera (8 bits), este campo permite a routers y hosts examinar con más detalle el paquete. A pesar de que el paquete básico IPv6 tiene cabecera de tamaño fijo, el protocolo puede añadir más para utilizar otras características como encriptación y autenticación.
- Tamaño de payload (16 bits), describe el tamaño en octetos de la sección de datos del paquete. Al ser este campo de 16 bits, podremos usar paquetes de hasta más de 64000 bytes.

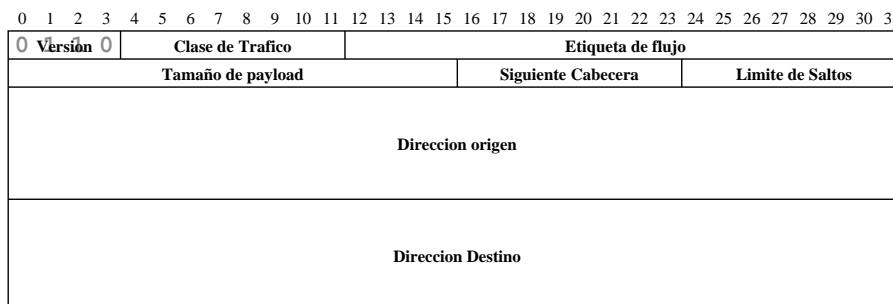


Figura 2.1: El paquete IPv6

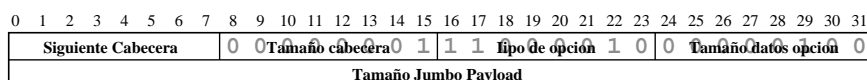


Figura 2.2: La siguiete cabecera

- Límite de saltos (8 bits), especifica el número de saltos de router que puede hacer el paquete antes de ser desechado. Con 8 bits podremos tener un máximo de 255 saltos.

## 2.2. El campo de siguiete cabecera (Next Header field)

Como hemos dicho antes, el tamaño de la cabecera IPv6 básica es fijo. Dentro de esta cabecera existe un campo llamado de siguiete cabecera que permite describir con más detalle las opciones del paquete. Esto quiere decir que en realidad tendremos una cabecera de tamaño fijo por norma general y otra cabecera de tamaño variable en caso de que utilicemos alguna de la características avanzadas.

En el campo de siguiete cabecera se codificarán las opciones presentes en la siguiete cabecera:

Siguiete cabecera	Valor del campo
Opciones de Hop-by-Hop	0
Opciones de destino	60
Encaminamiento	43
Fragmento	44
Autenticación	51
Encapsulación	50
Ninguna	59

Esta arquitectura es muy flexible, ya que cada cabecera tiene un campo de siguiete cabecera, con lo que podemos tener varias opciones agregadas. Un ejemplo ilustrativo lo podemos ver en las figuras 2.3 y 2.4.

Con la cabecera de encaminamiento conseguimos la funcionalidad equivalente de IPv4 de Source-Routing, es decir, especificar los nodos intermedios por los que ha de pasar el paquete.

## 2.2. EL CAMPO DE SIGUIENTE CABECERA (NEXT HEADER FIELD) 9

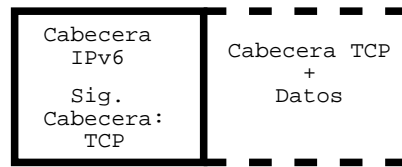


Figura 2.3: Cabecera IPv6 básica y datos

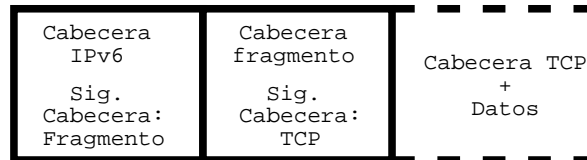


Figura 2.4: Cabecera IPv6 básica, fragmento y datos

Una cosa que ha de quedar bien clara es que los nodos intermedios o enrutadores NO deben examinar más que la cabecera IPv6 básica. Existen excepciones como en el caso de que existan cabeceras de opciones Hop-by-Hop o, como en el caso anterior, que exista una cabecera de encaminamiento en el que sólo los nodos en ella definidos deberán alterar el paquete.

La especificación recomienda además el siguiente orden para las cabeceras adicionales:

- Cabecera IPv6 básica.
- Opciones Hop-by-Hop.
- Opciones de destino.
- Encaminamiento.
- Fragmento.
- Autenticación.
- Encapsulación.
- Opciones de destino.
- Cabecera nivel superior.

Las opciones de destino pueden ser procesadas en momentos distintos dependiendo de si el paquete atraviesa un nodo intermedio o llega al nodo destino. La única restricción de la especificación es que las opciones de Hop-by-Hop han de ir siempre de la cabecera básica.

Podemos encontrar más información en [DH98].



## Capítulo 3

# Arquitectura de direccionamiento

### 3.1. Direccionamiento IPv6

Como ya hemos dicho, las direcciones IPv6 son identificadores de interfaces y/o conjuntos de interfaces de 128 bits. Tenemos tres tipos de direcciones:

- Unicast: Identificará un solo interfaz. Un paquete enviado a una dirección unicast se entregará a un solo interfaz.
- Anycast: Identificará un conjunto de interfaces, probablemente en distintos nodos. Un paquete enviado a una dirección de este tipo será entregado sólo a unos de los nodos, que debería ser, en principio, el más cercano.
- Multicast: Igual que en el caso anterior, identificará a un conjunto de interfaces que estarán seguramente en nodos distintos. Pero, en este caso, el paquete será enviado a todos los nodos del conjunto.

En las figuras podemos ver un ejemplo de comunicación entre tres nodos con direcciones A, B y C. Y los distintos comportamientos según el tipo de comunicación.

Con IPv6 dejan de existir las direcciones broadcast, cuya funcionalidad es absorbida por las direcciones multicast.

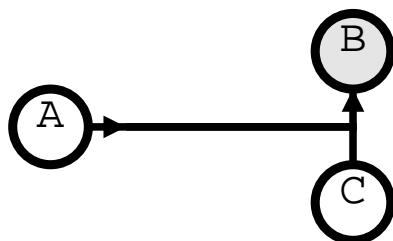


Figura 3.1: Ejemplo comportamiento unicast

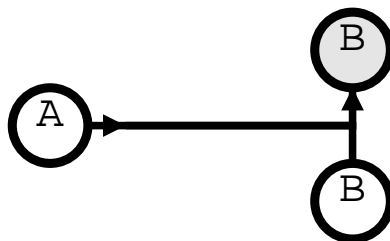


Figura 3.2: Ejemplo comportamiento anycast

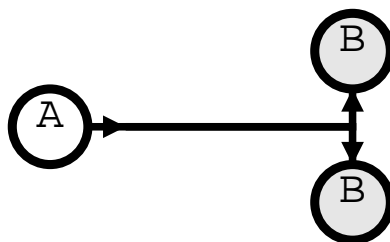


Figura 3.3: Ejemplo comportamiento multicast

## 3.2. Modelos de direccionamiento

Cualquier tipo de dirección se asigna a interfaces, no nodos. Es algo importante que no hay que olvidar. Todos los interfaces han de tener, por los menos, una dirección de enlace local (Link-Local) de tipo unicast. Un mismo interfaz puede tener asignadas múltiples direcciones de cualquier tipo (unicast, anycast, multicast) o ámbito (scope). Direcciones unicast con ámbito mayor que el de enlace no son necesarias para interfaces que no son usados como origen y destino de paquetes IPv6 hacia o desde no vecinos. Esto significa que para la comunicación dentro de una LAN no nos hacen falta direcciones IPv6 globales, sino que tenemos más que suficiente con direcciones de ámbito local. De hecho, es lo aconsejable para enlaces punto a punto.

Respecto a los prefijos de subred, IPv6 sigue el mismo modelo que IPv4, es decir, un prefijo se asocia a un enlace, pudiendo haber varios prefijos en un mismo enlace.

## 3.3. Ámbitos

Acabamos de mencionar en la sección anterior el 'ámbito' de una dirección sin saber todavía lo que era. Vamos a explicar en qué consiste. El protocolo IPv6 añade soporte para direcciones de distintos ámbitos, lo que quiere decir que tendremos direcciones globales y no globales. Si bien con IPv4 ya habíamos empleado direccionamiento no global con la ayuda de prefijos de red privados, con IPv6 esta noción forma parte de la propia arquitectura de direccionamiento.

Cada dirección IPv6 tiene un ámbito, que es un área dentro de la cual ésta puede ser utilizada como identificador único de uno o varios interfaces. El ámbito de cada dirección forma parte de la misma dirección, con lo que vamos a poder diferenciarlos a simple vista.

Para las direcciones unicast distinguimos tres ámbitos:

- De enlace local (link-local), para identificar interfaces en un mismo enlace. Empiezan todas por `fe80::`.
- De sitio local (site-local), para identificar interfaces en un mismo 'sitio'. La definición de 'sitio' es un tanto genérica, pero en principio un 'sitio' es el área topológica de red perteneciente a un edificio o un campus, perteneciente a una misma organización. Empiezan por `fec0::`.
- Global, para identificar interfaces en toda Internet. Éstas comienzan por `2001::` o `3ffe::`.

En lo que a ámbito se refiere, las direcciones anycast siguen la misma norma que las unicast.

Sin embargo, para las direcciones multicast tenemos catorce posibles ámbitos, que identifican desde un interfaz local a una dirección global.

Nodos de un mismo ámbito y visibles entre sí definen una zona. No se permite que un router encamine tráfico entre diferentes zonas (perderían todo el sentido los ámbitos).

Una de las grandes ventajas de los ámbitos es que permitirá la reenumeración de prefijos sin mucha dificultad, ya que las direcciones de ámbito no global se mantendrán. Tenemos que esperar que se produzca alguna reenumeración de prefijos globales, ya que según crezca una organización su prefijo se puede quedar pequeño y necesitar más espacio de direcciones. Y como hemos dicho antes, se tratará siempre que sea posible de mantener las tablas de encaminamiento al mínimo. Lo que sólo se consigue dando un prefijo nuevo mayor e invalidando el anterior, porque lo que seguramente sucederá será que las redes contiguas ya estén asignadas.

Más información en [NDO<sup>+</sup>01].

### 3.4. Nomenclatura de las direcciones

Tenemos tres formas comunes de representar direcciones IPv6 en texto:

- `x:x:x:x:x:x:x` donde cada `x` es el valor en hexadecimal de cada grupo de 16 bits de la dirección.
- `x:x::x` en el caso de que haya grupos contiguos de 16 bits todos cero. Es una abreviatura que servirá para hacer más cómodo el uso de algunas direcciones. Podemos ver un ejemplo comparativo de este caso y el anterior en la tabla 3.1 (página 14).
- `x:x:x:x:x:d.d.d.d`, donde las `x` son los seis grupos de 16 bits en hexadecimal de mayor peso de la dirección y las `d` son los valores decimales de los cuatro grupos de 8 bits de menor peso de la dirección. Esta forma es a veces más conveniente a la hora de manejar entornos mixtos IPv6 e IPv4. Por ejemplo: `0:0:0:0:FFFF:129.144.52.38` y en su forma abreviada `::FFFF:129.144.52.38`.

Representación normal	Representación abreviada	Tipo
1080:0:0:8:800:200C:417A	1080::8:800:200C:417A	unicast
FF01:0:0:0:0:0:101	FF01::101	multicast
0:0:0:0:0:0:1	::1	loopback
0:0:0:0:0:0:0	::	dirección no especificada

Cuadro 3.1: Nomenclatura de direcciones IPv6

### 3.5. Nomenclatura de los prefijos

La representación de los prefijos de direcciones con IPv6 es similar a la que tenemos con CIDR con IPv4 [FLYV93]: `dirección-ipv6/tamaño-prefijo`. Donde `dirección-ipv6` es alguna de las notaciones vistas en la sección anterior y `tamaño-prefijo` es un valor decimal que especifica cuantos bits de la dirección corresponden al prefijo.

Por ejemplo, el prefijo de la UJI en hexadecimal es `3FFE33300002`, que son 48 bits, lo podemos escribir como:

```
3FFE:3330:0002:0000:0000:0000:0000:0000/48
3FFE:3330:2:0:0:0:0:0/48
3FFE:3330:2::/48
```

Si queremos escribir la dirección y el prefijo, no hace falta que escribamos los dos de forma explícita. Por ejemplo, una dirección IPv6 de la misma UJI con su prefijo asociado quedaría `3FFE:3330:2:1:250:BAFF:FE7A:E67E/48`.

### 3.6. Representación de los tipos de direcciones

El tipo específico de cada dirección IPv6 viene dado por los primeros bits de ésta, dentro de lo que se llama el campo de formato de prefijo (FP, format prefix). El tamaño de este campo es variable. La asignación de estos prefijos se puede ver en la tabla 3.2 (página 15).

Los prefijos desde 001 a 111 tienen la obligación de tener los identificadores de interfaz de 64 bits en formato EUI-64, descrito en [IEE97], excepto para las direcciones multicast (1111 1111). Las direcciones unicast se distinguen por el valor del octeto de mayor peso, que tiene algún valor distinto de '1'. Las direcciones anycast se asignan dentro del espacio de las anycast y no son distinguibles entre sí observando sus bits.

Como podemos ver, hay mucho espacio no asignado (el 85%), lo que en un futuro permitirá expandir el espacio posible o incluso dar nuevos usos.

### 3.7. Direcciones unicast

Existen varios tipos de direcciones unicast en IPv6, como las globales agregables, las site-local, las link-local, las IPX jerárquicas, la NSAP, y las compatibles IPv4. Más tipos de direcciones pueden ser definidos en el futuro.

<b>Asignación</b>	<b>Prefijo</b>
Reservado	0000 0000
No asignado	0000 0001
Reservado para asignación NSAP	0000 001
Reservado para asignación IPX	0000 010
No asignado	0000 011
No asignado	0000 1
No asignado	0001
Direcciones Unicast Globales Agregables	001
No asignado	001
No asignado	010
No asignado	011
No asignado	100
No asignado	101
No asignado	110
No asignado	1110
No asignado	1111 0
No asignado	1111 10
No asignado	1111 110
No asignado	1111 1110 0
Direcciones Unicast Link-Local	1111 1110 10
Direcciones Unicast Site-Local	1111 1111 11
Direcciones Multicast	1111 1111

Cuadro 3.2: Tabla reserva de prefijos

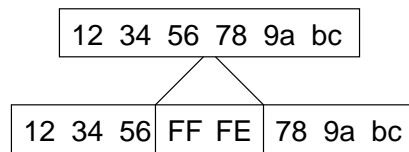


Figura 3.4: Contrucción del identificador de interfaz a partir de la MAC

Dependiendo del papel que realice cada nodo, éste puede tener más o menos conocimiento de la estructura del paquete IPv6. Por ejemplo, un nodo puede considerar una dirección IPv6 unicast como un "todo" siendo inconsciente incluso de los prefijos; algo más complejo entendería de prefijos y, yendo un poco más lejos, podría entender la jerarquía dentro del prefijo y lo que ello implica.

### 3.7.1. Identificadores de interfaz

Los identificadores de interfaz en las direcciones unicast IPv6 se utilizan para identificar interfaces en un determinado enlace (una LAN, por ejemplo). Es necesario que sean únicos en el enlace, porque si dejamos de identificar interfaces a nivel de enlace ya no hay nada más que hacer. Pero esto último no significa que puedan seguir siendo únicos en un ámbito mayor que el de enlace. Por norma general, los identificadores se obtendrán a partir de las direcciones de la capa de enlace.

Como hemos visto en la sección 3.6, unos cuantos tipos de prefijos requieren identificadores de interfaz de 64 bits y, además, estar contruidos en formato IEEE EUI-64. Estos identificadores pueden ser globales en el caso de que un token global esté disponible, como los 48 bits de la MAC, o locales en caso de que no lo esté, como un enlace por puerto paralelo o los extremos de un túnel.

Se requiere que el bit 'u' sea invertido en caso de que el identificador se haya construido a partir del formato EUI-64. Este bit, según la terminología IEEE es el que indica la localidad o universalidad del identificador. Esto, que en un principio no tiene mucho sentido, va a servir para que aquellos interfaces donde no es posible obtener un token global tengan una forma más sencilla. Por ejemplo, un extremo de un túnel, su identificador debería ser 0200:0:0:1 en vez de ::1 si este cambio no nos arreglase un poco la vida.

### 3.7.2. La dirección no específica

Así es como llamamos a la dirección 0:0:0:0:0:0:0:0. Ésta nunca debe ser asignada a ningún nodo y sólo se permite su uso en casos bien contados, como en el campo de dirección origen cuando un interfaz no conoce todavía la suya.

Bajo ningún concepto se debe usar esta dirección como dirección destino de un paquete IPv6 o en la cabecera de encaminamiento.

### 3.7.3. La dirección de loopback

La dirección unicast 0:0:0:0:0:0:0:1 recibe el nombre de loopback y su equivalente en IPv4 es 127.0.0.1. Se utiliza para la comunicación entre servicios de un mismo nodo y nunca se debe mandar un paquete con esta dirección tanto de origen como destino sobre un medio físico. Con esto queda claro que no se

puede asignar a interfaces reales, sino a interfaces virtuales (como el interfaz de loopback).

#### 3.7.4. Direcciones IPv6 con direcciones IPv4 embebidas

Dentro de los mecanismos previstos de transición de IPv4 a IPv6, existe una técnica que permite a los hosts y routers entunelar dinámicamente paquetes IPv6 sobre la infraestructura IPv4 existente. Los nodos que vayan a utilizar esta técnica recibirán una dirección unicast IPv6 un tanto especial: los 32 bits más bajos serán la dirección IPv4. A este tipo de direcciones se las llama direcciones IPv6 compatibles con IPv4.

También existe otro tipo de dirección IPv6 que contiene a una IPv4 y se utilizará para representar aquellos nodos que sólo disponen de pila IPv4. En este caso los 32 bits más bajos serán iguales que en el caso anterior (la dirección IPv4), pero los 16 bits siguientes por delante serán todos 1. Este tipo de direcciones recibe el nombre de direcciones IPv6 mapeadas IPv4.

#### 3.7.5. Direcciones globales agregables

### 3.8. Direcciones anycast

Este tipo de direcciones pueden ser asignadas a distintos interfaces de uno o varios nodos, de forma que un paquete enviado a una dirección anycast llegará a uno y sólo a uno de los interfaces.

Sintácticamente, las direcciones anycast no pueden ser distinguidas de las unicast, por lo que si un interfaz tiene asignada una dirección de este tipo se le debe decir expresamente.

En la actualidad, se tiene poca experiencia con las direcciones anycast por lo que se han impuesto una serie de restricciones:

- No se puede enviar un paquete con dirección origen que sea de tipo anycast.
- Una dirección anycast no puede ser asignada a un host, sólo a routers.

A pesar de las restricciones, las direcciones anycast ya se están utilizando por ejemplo para que un nodo móvil contacte con alguno de sus routers en su red de casa.

### 3.9. Direcciones multicast

Una dirección multicast identifica a un grupo de nodos. Un nodo puede pertenecer a varios grupos multicast.

Los distintos campos de una dirección multicast se pueden ver en la figura. Donde cada campo tiene el siguiente significado:

- 11111111 identifica a las direcciones multicast.
- **flags** es un conjunto de cuatro banderas, estando los tres primeros bits reservados. El último, si es 0 indica que la dirección es 'conocida' y ha sido asignada por la autoridad de numeración de Internet. En caso contrario indica que una dirección 'temporal', no asignada de forma permanente.

Significado	Dirección
Reservado	FF0X:0:0:0:0:0:0:0
Todos los nodos	FF0X:0:0:0:0:0:0:1
Todos los routers	FF0X:0:0:0:0:0:0:2

Cuadro 3.3: Tabla direcciones *conocidas* predefinidas

- **scope** indica el ámbito del grupo multicast.
- **group ID** identifica el grupo multicast dentro del ámbito determinado.

Dentro de las direcciones *conocidas* existen unas predefinidas que se muestran en la tabla 3.3.

### 3.10. Requerimientos de nodo

Para cumplir con la especificación se requiere que todos los nodos reconozcan como suyas:

- Su dirección de enlace local (link-local) para cada interfaz.
- Su dirección unicast asignada.
- Su dirección de loopback.
- La dirección multicast de 'Todos los nodos'.
- Las direcciones multicast de todos los grupos a los que pertenezca.

Además, si el nodo es un router, se requiere que reconozca también:

- Las direcciones anycast de cada subred para las que es router.
- Las direcciones anycast que se le han asignado.
- La dirección multicast de 'Todos los routers'.

En cuanto a prefijos, los únicos predefinidos en una implementación son:

- La dirección no específica.
- La dirección de loopback.
- El prefijo multicast (FF).
- Los prefijos locales de enlace y de 'sitio' (link-local y site-local).
- Las direcciones multicast predefinidas.
- Los prefijos compatibles IPv4.

## IPv6 práctico: montando un túnel

Cuando hablemos de estrategias de implantación veremos que la forma más sencilla de conectarse a la red IPv6 actual es a través de un servidor de túneles gratuito.

Con Linux, tenemos dos herramientas básicas para construir el túnel: `iproute2` y las clásicas `ifconfig/route`.

Supongamos que un servidor de túneles nos ha asignado la dirección IPv6 `2001:720:1010:5::31/128`, nos dice que su extremo del túnel IPv6 es `2001:720:1010:2::1` y que el IPv4 es `150.128.81.246`. Para construir nuestro extremo del túnel tendremos que hacer lo siguiente:

```
linux# ip tunnel add sixbone mode sit remote 150.128.81.246
linux# ip link set sixbone up
linux# ip addr add 2001:720:1010:5::31/128 dev sixbone
linux# ip -f inet6 route add default via 2001:720:1010:2::1
```

En primer lugar, hemos creado el interfaz virtual del túnel, `sixbone`, y le hemos dado la dirección que el servidor de túneles nos había dado. En segundo lugar hemos añadido la ruta por defecto para el protocolo IPv6, que obviamente ha de ser el otro extremo del túnel. Debemos tener un kernel de Linux reciente para añadir rutas por defecto para IPv6, ya que podría dar problemas.

Después de haberlo configurado, podemos probar hacer un ping bajo IPv6:

```
linux# ping6 www.ipv6.uji.es
PING www.ipv6.uji.es(2001:720:1010:1::1) 56 data bytes
64 bytes from 2001:720:1010:1::1: icmp_seq=0 hops=64 time=301 usec
64 bytes from 2001:720:1010:1::1: icmp_seq=1 hops=64 time=207 usec
64 bytes from 2001:720:1010:1::1: icmp_seq=2 hops=64 time=187 usec
--- www.ipv6.uji.es ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.187/0.231/0.301/0.052 ms
linux#
```

Con esto tendremos comprobado que el túnel funciona.

En [Fem01] podremos encontrar ejemplos de configuración de túneles para otros sistemas operativos.



## Capítulo 4

# Autoconfiguración

Como ya hemos dicho, ésta es otra de las nuevas características que trae consigo IPv6 respecto a IPv4. El proceso de autoconfiguración consiste en lo siguiente: creación de una dirección de enlace local (link-local) y comprobar su unicidad en el enlace, determinar qué tipo de información se debe autoconfigurar (direcciones, otros datos o los dos) y, en el caso de direcciones, determinar qué mecanismo se debe usar para obtenerlas (con control de estado, sin él o con los dos). La obtención automática de configuración sólo se aplica a hosts y nunca a routers (si bien éstos pueden generar su propia dirección de enlace local), ya que los hosts necesitan información anunciada por éstos y alguien tendrá que configurarlos. Se puede consultar [TN98] para más información.

IPv6 define tanto el mecanismo de autoconfiguración con control de estado como el sin control de estado. En el caso de sin control de estado, la configuración necesaria es nula en los nodos y prácticamente nula en los routers. El mecanismo permite al host obtener una dirección a partir de información local (el identificador de interfaz) e información anunciada por los routers (el prefijo de subred). En caso de que no haya routers en la red, los hosts pueden generar sus propias direcciones de enlace local (link-local). Suficiente para comunicarse entre sí.

En el caso de autoconfiguración con control de estado, los hosts obtienen sus direcciones y otra información de algún servidor. Este servidor puede mantener un control preciso de qué direcciones han sido asignadas a cada host. Se ha desarrollado una versión específica de DHCP para IPv6 llamada DHCPv6 para tal efecto. No vamos a detallar en exceso este caso.

No necesitamos control de estado cuando un 'sitio' no se preocupa en exceso por las direcciones que usa cada host mientras éstas sean válidas y encaminables. Sí lo necesitamos en el caso de que necesitemos saber qué host usa qué dirección en todo momento por cualquier razón (quizá inventario). En cualquier caso, siempre podemos usar los dos métodos para configurar según qué cosas.

Para asegurarnos de que las direcciones sean únicas, los nodos ejecutan un algoritmo de detección de direcciones duplicadas.

A partir de ahora hablaremos exclusivamente del mecanismo sin control de estado, lo que no significa que algunas de las cosas sean comunes.

## 4.1. Objetivos del diseño

El método de configuración sin control de estado se diseñó con los siguientes objetivos:

- No debe ser necesaria la configuración manual de los hosts para poder comunicarse a través de la red. Por esto mismo, un nodo debe de ser capaz de generar una dirección para cada interfaz. El mecanismo asume que cada interfaz tiene al menos un identificador único. Un identificador y un prefijo permiten obtener una dirección.
- 'Sitios' de tamaño pequeño no deben de necesitar ni un servidor ni un router para comunicarse entre sí. Con las direcciones de enlace local (link-local) se consigue esto. Éstas se obtienen añadiendo el identificador de interfaz al prefijo de enlace local.
- 'Sitios' grandes no deben necesitar un servidor de autoconfiguración con control de estado si no lo desean. Los routers han de ser capaces de anunciar los datos necesarios para obtener una dirección correcta, no duplicada y encaminable a través de Anuncios de Router (Router Advertisements, RA).
- La reenumeración de hosts ha de ser sencilla. Como ya hemos dicho, puede ocurrir que una subred sea reenumerada. El mecanismo de autoconfiguración debe proveer una forma sencilla de realizar esto e incluso permitir más de una dirección durante la posible transición. Esta parte de la autoconfiguración se describe con detalle en [Nor01].

## 4.2. El protocolo

Vamos a ver un poco más de cerca el proceso de autoconfiguración en sí.

La autoconfiguración en sí sólo es posible en enlaces o medios que permitan la multidifusión (multicast). El mecanismo empieza a funcionar en el momento en que un interfaz se levanta (reconexión a la red, encendido del sistema, ...). Los nodos, tanto hosts como routers, intentan generar una dirección de enlace local (link-local) para el interfaz. Ésta se forma añadiendo el identificador de interfaz al prefijo de enlace local.

Antes de que esta dirección pueda ser asignada al interfaz se ha de comprobar que sea única en el enlace. Lo que se hace es mandar un mensaje multicast destinado a la dirección que acabamos de calcular para nosotros. Y si ésta está en uso, el nodo correspondiente nos devolverá el mensaje.

Si resulta que la dirección que queríamos obtener para nosotros está en uso, el mecanismo se para y espera configuración manual. Si no es así, el nodo asigna definitivamente la dirección al interfaz y, a partir de este momento, obtiene conectividad a nivel IP. Los pasos siguientes sólo los realizarán los hosts, los routers se quedan aquí.

El siguiente paso consiste en obtener un Anuncio de Router (RA) o darnos cuenta de que no hay routers en el enlace. Si hay algún router, éste nos dirá que tipo de configuración tendremos que seguir. Si no es el caso, debemos pasar a autoconfiguración con control de estado.

### 4.3. CREACIÓN DE LAS DIRECCIONES DE ENLACE LOCAL (LINK-LOCAL)<sup>23</sup>

Los routers envían Anuncios de Router (RA) periódicamente, pero no lo suficientemente consecutivos en el tiempo como para obtener una funcionalidad estilo pinchar y funcionar. Por eso mismo, cuando el interfaz se levanta, el host manda una o más Solicitaciones al Router (Router Solicitations, RS) al grupo multicast de 'Todos los routers'. Los RA contienen información sobre el tipo de autoconfiguración que ha de seguir el host, así como cero o más prefijos y algún campo más.

Un host ya autoconfigurado seguirá recibiendo RA por parte del router, que deberá procesar añadiendo y/o actualizando la información antes recibida.

### 4.3. Creación de las direcciones de enlace local (link-local)

Un nodo construye una dirección de enlace local cuando alguno de sus interfaces se activa. Se considera que un interfaz se activa cuando:

- El interfaz se levanta al arrancar el sistema.
- El interfaz es reiniciado después de haber sido desactivado.
- El interfaz se engancha al enlace por primera vez.

Como ya hemos dicho anteriormente, la dirección de enlace local (link-local) se construye añadiendo el identificador de interfaz al prefijo FE80::0 (del tamaño adecuado). Si el identificador de interfaz tiene una longitud de N bits, el identificador reemplazará los N bits más a la derecha del prefijo. En caso de que el identificador de interfaz sea mayor de 118 bits, el mecanismo de autoconfiguración falla y requerirá intervención manual. Por norma general, esto no sucederá ya que el identificador de interfaz seguirá la norma EUI-64 y tendrá un tamaño de 64 bits.

### 4.4. Creación de direcciones globales y de 'sitio' local (site-local)

Las direcciones de tipo global y de 'sitio' local se construyen a partir de un prefijo anunciado en los RA y el identificador del interfaz.

Los routers mandan de forma periódica RA a la dirección multicast predefinida de 'Todos los nodos'. Si un nodo desea recibir un RA más pronto puede enviar uno o más RS.

Para saber si hay o no hay routers en el enlace, un nodo debe haber enviado varios RS y no haber obtenido ningún RA en un periodo razonable de tiempo. En este caso el nodo debe probar autoconfigurarse con el mecanismo de control de estado.

Estos son los pasos a seguir por un nodo a la hora de procesar las opciones de información de prefijo de cada Anuncio de Router:

- Si el prefijo es el de enlace local (link-local), debe descartarlo de forma silenciosa.

- Si el tiempo de vida del prefijo es mayor que el tiempo válido de vida, debe ignorar la información del prefijo de forma silenciosa.
- Si el prefijo anunciado tiene un tiempo válido de vida mayor que 0 y no ha formado ya una dirección a partir de este prefijo, debe contruirla y añadirla.
- Si el prefijo anunciado coincide con alguno a partir del cual hemos construido alguna dirección las acciones a tomar dependerán del tiempo válido de vida del prefijo.

## 4.5. Consideraciones de seguridad

En principio, es factible que un nodo cualquiera envíe información errónea o falsa de forma o no deliberada. La solución es usar RA y RS autenticados.

## IPv6 práctico: autoconfiguración

En caso de que el servidor de túneles nos ofrezca también la posibilidad de delegarnos una subred, podremos hacer uso de la autoconfiguración para anunciar nuestro prefijo de red y asignar direcciones globales a nuestros nodos de red.

Veamos primero como encaminamos una subred a través de nuestro túnel, suponiendo que ésta es `2001:720:1010:9::/64` y que la vamos a tener colgada en nuestro interfaz `eth1`:

```
linux# ip link set eth1 up
linux# ip addr add 2001:720:1010:9::1/64 dev eth1
linux# echo 1 >/proc/sys/net/ipv6/conf/all/forwarding
linux#
```

De esta forma hemos asignado una dirección a nuestro interfaz `eth1`. Al haber especificado una longitud de prefijo, el kernel añadirá por sí solo la ruta para la red. Además, hemos activado el reenvío de paquetes (`forwarding`), que no está activo por defecto.

El siguiente paso va a ser configurar el demonio de anuncios de router, `radvd` en Linux. Veamos como quedaría el archivo de configuración, `radvd.conf`, en nuestro caso:

```
interface eth1 {
    AdvSendAdvert on;          # queremos mandar
                              # anuncios periódicamente

    prefix 2001:720:1010:9::/64
    {
        AdvOnLink on; # mandar anuncio si
                      # detectamos nuevo nodo
                      # en la red
        AdvAutonomous on;
        AdvRouterAddr on; # anunciar ruta por defecto
    }
}
```

```
    };  
};
```

Ahora podríamos arrancar el demonio y ver como los nodos reciben su dirección global IPv6 uniendo el prefijo de red anunciado con el identificador de interfaz.



## Capítulo 5

# Movilidad

A día de hoy prácticamente todo es móvil, desde el teléfono, al ordenador, al PDA, . . . Algún soporte para la movilidad por parte del protocolo IP no sería nunca mala idea. IPv6 soporta esta característica de serie, sin parches como es necesario con IPv4.

Entenderemos movilidad como la facilidad para cambiar de red, tanto a nivel físico como a nivel lógico, sin perder el transporte ni las conexiones establecidas por capas de nivel superior al IP. Para que esto sea posible, deberemos mantener una misma dirección IPv6 estemos donde estemos y los paquetes enviados a nosotros tendrán que ser encaminados hacia nosotros estemos donde estemos.

### 5.1. Operación

Todo nodo móvil (Mobile Node, MN) tendrá una dirección 'de casa' (Home Address, HA), que será su dirección en su red origen. Esta dirección se mantendrá aunque cambiemos de red. Los paquetes que se envíen al nodo móvil estando éste en su red origen serán encaminados de forma normal, como si el soporte de movilidad no existiese.

En el momento en que el nodo móvil pasa a una red que no es la suya de origen, éste obtendrá una nueva dirección 'de invitado' (Care-of-Address, CoA). A partir de ahora el nodo podrá ser contactado también a través de esta CoA. Lo siguiente que hará el nodo móvil es contactar con un router de su red origen (Home Agent, HA) y comunicarle cual es su CoA actual. De esta forma, cuando un paquete sea enviado a la 'dirección de casa', el router sabrá que tendrá que interceptarlo y entunelarlo con destino a la CoA del nodo móvil.

Lo que en realidad hace el MN cuando se mueve es mandar un mensaje de Binding Update (BU) al HA. El BU asocia la CoA con la dirección 'de casa' del nodo móvil durante un cierto periodo de tiempo.

Llamaremos nodo correspondiente (Correspondent Node, CN) a cualquier nodo, ya sea fijo o móvil que se comunique con un MN.

Cuando un nodo móvil se comunica con un CN, el MN envía directamente los paquetes utilizando la dirección 'de invitado' que ha obtenido en la red que se encuentre. Sin embargo, el CN envía los paquetes a la dirección 'de casa' del MN, que serán interceptados por el HA y reenviados a la CoA del nodo móvil. Tendríamos un caso de ruta triangular, que no es ningún problema, pero es

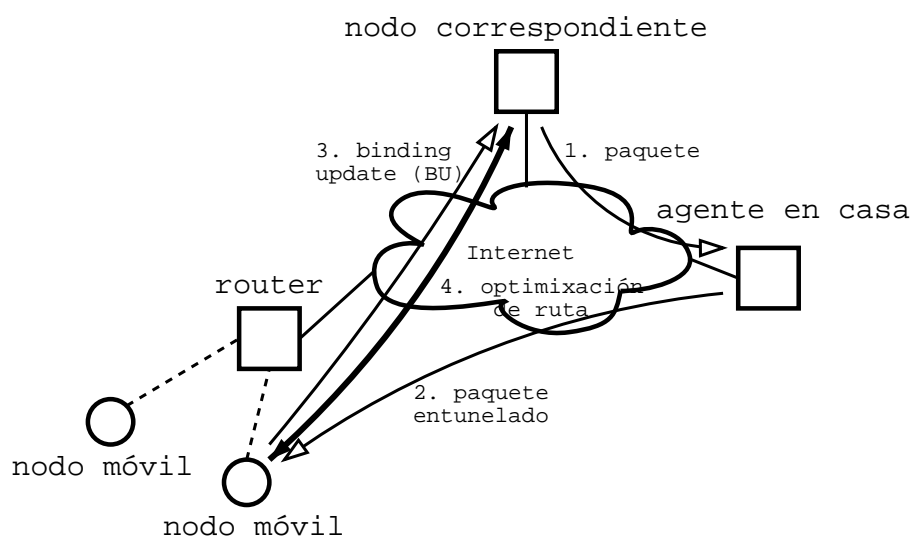


Figura 5.1: Operación de MobileIPv6

ineficiente. Para resolver esto, MobileIPv6 presenta el concepto de optimización de ruta. Este mecanismo permite al MN avisar al CN de que puede enviarle los paquetes directamente a su CoA utilizando para ello mensajes de Binding Update.

## 5.2. Cabeceras adicionales

Para conseguir toda esta funcionalidad añadida, MobileIPv6 aprovecha las cabeceras de opción de destino. Esto permite enviar información de señalización en el mismo paquete de datos. Los nuevos tipos de opciones de destino creadas para soportar la movilidad son:

- Home Address Option, indica cual es la dirección 'de casa' del nodo móvil cuando éste se encuentra fuera de su red origen.
- Binding Update Option, que sirve para crear, actualizar y eliminar entradas de las asociaciones que se mantienen entre MN y CoA. Un paquete con esta opción hará que se produzca una asociación en el CN o en el HA entre la dirección origen del paquete y la dirección contenida en el campo de Home Address Option.
- Binding Acknowledgement (BA) Option, que es enviada por el HA y por los CN como respuesta a los BU enviados por el nodo móvil.
- Binding Request (BR) Option, enviada por el CN para solicitar al nodo móvil refrescar su entrada en la lista de asociaciones actual del MN.

### 5.3. Consideraciones de seguridad

Tanto los Binding Updates como los Binding Acknowledgements provocan un cambio de estado en los nodos, por lo que deben de ser autenticados. MobileIPv6 utiliza autenticación de cabeceras (Authentication Header, AH) para evitar cualquier ataque.

Sin embargo, la autenticación no es el único problema. La autorización, es decir, qué CN puede alterar qué asociaciones en la tabla de un MN (que afecta a las tablas de enrutamiento), es el otro. Una posible forma de solventar esto es utilizar utilizar IKE (Internet Key Exchange) junto a DNSSEC, asumiendo que tanto el nodo móvil como el CN utilizan la misma infraestructura de llave pública.

Para más información podemos consultar [JP01].

### IPv6 práctico: moviéndonos

La configuración de todo el entorno en este caso es un poco compleja, ya que debemos preparar los dos componentes básicos: el nodo móvil y el Agente en Casa.

En lo que al Agente en Casa se refiere, tendremos que retocar la configuración del demonio de anuncios de router, `radvd.conf`, y dejarlo de la siguiente manera:

```
interface eth1 {
    AdvSendAdvert on;          # queremos mandar
                              # anuncios periódicamente

    AdvHomeAgentFlag on;     # somos Agente en Casa
    AdvHomeAgentInfo on;     # propagar información de
                              # Agente en Casa

    prefix 2001:720:1010:9::/64
    {
        AdvOnLink on; # mandar anuncio si
                      # detectamos nuevo nodo
                      # en la red
        AdvAutonomous on;
        AdvRouterAddr on; # anunciar ruta por defecto
    };
};
```

Aparte de esto deberemos poner en orden los siguientes archivos:

- `network-mipv6.conf` para definir cual va a ser la función del nodo, que en nuestro caso será de Agente en Casa (HA). También podremos definir el nivel de traza (debug) que deseamos.
- `mipv6_acl.conf` para definir las listas de control de acceso (ACL). De esta forma podremos decidir a qué nodos móviles damos servicio.
- `mipv6_sas.conf` en caso de que queramos utilizar IPSEC para mandar los mensajes de señalización, aquí definiremos las asociaciones de seguridad (Security Associations).

Por parte del nodo móvil la configuración se restringe a dos de los tres archivos antes mencionados:

- `network-mipv6.conf` para definir la función del nodo, que esta vez será Nodo Móvil (MN), nuestra dirección en la red de casa y nuestro Agente en Casa.
- `mipv6_sas.conf` que tendrá el mismo propósito que en el caso del Agente en Casa.

Podemos consultar [CR01] para obtener instrucciones de una maqueta de prueba.

## Capítulo 6

# Estrategias de implantación

Puesto que Internet no va a amanecer un día utilizando de repente IPv6 en vez de IPv4, se han debido desarrollar una serie de métodos que permitan la convivencia y comunicación entre nodos, sea cual sea su versión de protocolo IP. Como pronto veremos, se han desarrollado unos cuantos, cada uno de ellos con sus ventajas e inconvenientes, pero sobretodo pensados en un principio para casos de migración distintos.

No utilizar un mecanismo de los aquí descritos u otro no tiene mucho sentido dada la pequeña cantidad de servicios que se están ofreciendo bajo la Internet IPv6 actual. Pongamos el ejemplo de los servicios web, actualmente los más desarrollados, que no llegarán al 1% comparado con lo que existe bajo IPv4. Pongamos otro más: los servidores CVS del proyecto USAGI no se encuentran más que en IPv4. Ya queda todo dicho.

### 6.1. Túneles

Encapsular un paquete IP dentro de otro es un mecanismo conocido y se usa en la actualidad sobretodo para crear redes privadas virtuales. La utilidad que le daremos aquí es para enlazar nubes o islas IPv6 en una Internet basada prácticamente en su totalidad en IPv4.

Tenemos dos tipos básicos de túneles: estáticos y dinámicos. El 6bone actual está formado en su mayoría por túneles estáticos.

#### 6.1.1. Túneles estáticos

Esta es la solución más sencilla y la menos intrusiva si queremos tener acceso tanto a IPv6 como a IPv4.

El caso más común será un host con IPv4 que desee tener acceso a la red IPv6 existente. Para ello deberá crear un túnel con un router a través de IPv4 que tenga tanto acceso a IPv6 como a IPv4. Un caso un poco menos común para el usuario de a pie es en el que se deseen unir 'islas' IPv6, osease, unir redes IPv6, utilizando para ello la infraestructura IPv4 existente.

Este método se está utilizando en la actualidad por parte de algunos proveedores de servicios para que cualquiera pueda tener acceso a la red IPv6. En el caso de la UJI, se tiene un túnel contra la Universidad de Valencia sobre la

infraestructura IPv4 que probablemente desaparecerá en un futuro a largo plazo (entonces quizá tendremos túneles IPv4 sobre IPv6 para mantener algún tipo de servicio).

Dentro de esta categoría podemos considerar también la de los servidores de túneles, que en estos momentos son interfaces web que permiten la creación de túneles bajo demanda a cualquier usuario.

### 6.1.2. 6to4

Este mecanismo se puede aplicar para comunicar redes IPv6 aisladas por medio de la red IPv4. El router extremo de la red IPv6 crea un túnel sobre IPv4 para alcanzar la otra red IPv6. Los extremos del túnel son identificados por el prefijo del sitio IPv6. Este prefijo consiste en 16 bits fijos que indican que estamos utilizando la técnica 6to4 más 32 bits que identifican al router externo del 'sitio'.

Un efecto secundario de 6to4 es que deriva automáticamente un prefijo /48 de una dirección IPv4. De esta forma, los 'sitios' pueden empezar a utilizar IPv6 sin solicitar nuevo espacio de direccionamiento a la autoridad competente.

El método se describe con más detalle en [Moo01].

### 6.1.3. 6over4

Puede que no tengamos una red de sitio homogénea en el aspecto de que todos los nodos puedan comunicarse entre sí con la misma versión de protocolo IP. Con este método podremos comunicar nodos IPv6 aislados dentro de nuestro 'sitio' con el resto de nodos IPv4. Esta técnica también se emplea en casos en los cuales el router IPv6 no tiene acceso o permiso para transmitir paquetes IPv6 sobre en enlace. Para salvar este escollo se creará un enlace virtual utilizando un grupo multicast IPv4, mapeando las direcciones IPv6 sobre este grupo multicast.

### 6.1.4. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

Como su nombre indica, este método también está pensado para la comunicación entre nodos de un mismo 'sitio'. Tiene algunas ventajas respecto a 6over4, como que no necesita multicast IPv4 y que soluciona los problemas que se dan cuando una misma organización no tiene toda su red en un mismo lugar, como la baja escalabilidad en la agregación.

La técnica funciona empotrando la dirección IPv4 del nodo en el identificador EUI-64 del interfaz. Puesto que este método viene a solucionar los problemas de comunicación dentro de un 'sitio', las direcciones IPv4 no tienen por qué ser globales. Esto significa que aunque exista NAT, el mecanismo seguirá funcionando correctamente.

## 6.2. Comunicación entre nodos

Una vez tenemos unidas varias islas IPv6 el problema que se plantea es el de que todos los nodos puedan acceder a la Internet IPv6 como a la IPv4. La

solución va a consistir en o bien a nivel de aplicación, transformando la capa de enlace o asignando temporalmente direcciones IPv4 a nodos IPv6.

### 6.2.1. Doble pila

Para que un nodo se pueda comunicar tanto con nodos IPv6 como IPv4, la solución más rápida es pensar en la doble pila de protocolos. Teniendo cada nodo una dirección IPv4 e IPv6 enrutable, se conseguirá que se produzca la comunicación.

### 6.2.2. Stateless IP/ICMP Translation Algorithm (SIIT)

El protocolo SIIT permite traducir entre IPv6 e IPv4. Esta traducción queda limitada a la cabecera IP. Como su propio nombre dice, no se realiza un control de estado, por lo que la traducción se debe realizar para cada paquete.

El método se describe con detalle en [Nor00].

### 6.2.3. Network Address Translation - Protocol Translation (NAT-PT)

En caso de que tengamos nodos o bien con IPv6 o bien con IPv4 de forma exclusiva, esta puede ser una buena solución. La comunicación se realiza a través de un dispositivo específico (un router que soporte NAT-PT) y que soporta el control de estado de las conexiones. Este método necesita también cambios a nivel de aplicación para controlar las peticiones de resolución de nombre en el DNS.

El método se describe con detalle en [TS00].

### 6.2.4. Bump in the Stack (BIS)

Si pensamos en el método de NAT-PT a nivel particular de cada host llegamos a tener una idea de en qué consiste este mecanismo, que se utilizará en caso de que las aplicaciones que utilicen los nodos no soporten IPv6.

Añadiendo tres módulos (una extensión para la resolución de nombres, un mapeador de direcciones y un traductor) entre el nivel de aplicación y el de red se consigue un acceso transparente a nodos IPv6.

La idea es la siguiente: cuando una aplicación exclusiva IPv4 necesita comunicarse con un nodo IPv6, la dirección IPv6 de ese nodo se mapea a una dirección IPv4. Los paquetes IPv4 generados se transforman en paquetes IPv6 utilizando SIIT.

El método se describe con detalle en [THA00].

### 6.2.5. SOCKS64

Esta solución puede llegar a ser la ideal en caso de que el 'sitio' esté utilizando ya SOCKS. Con un gateway de tipo SOCKS64 se puede permitir conectar a los clientes tanto a nodos IPv4 como IPv6, sin los típicos problemas asociados a los túneles (fragmentación y límite de saltos).

Si se desea profundizar más en el tema se puede consultar [Kit01].



# Bibliografía

- [CR01] Diego Chaparro and Raúl Rodríguez. Proyecto mobiquo/MIND. Technical report, Grupo de Sistemas y Comunicaciones, Universidad Rey Juan Carlos, June 2001. <http://mobiquo.dat.escet.urjc.es>.
- [DH95] S. Deering and R. Hinden. RFC 1883: Internet Protocol, version 6 (IPv6) specification, December 1995. Obsoleted by RFC2460 [DH98]. Status: PROPOSED STANDARD.
- [DH98] S. Deering and R. Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) specification, December 1998. Obsoletes RFC1883 [DH95]. Status: DRAFT STANDARD.
- [Fem01] José M. Femenía. IPv6 práctico, May 2001. <http://www.rediris.es/red/reuniones/IPv6practico.pdf>.
- [FLYV93] V. Fuller, T. Li, J. Yu, and K. Varadhan. RFC 1519: Classless interdomain routing (CIDR), September 1993.
- [Hui94] C. Huitema. RFC 1715: The H ratio for address assignment efficiency, November 1994. Status: INFORMATIONAL.
- [IEE97] IEEE. Guidelines for 64-bit global identifier (EUI-64) registration authority, March 1997. <http://standards.ieee.org/db/oui/tutorials/EUI64.html>.
- [JP01] David B. Johnson and Charles Perkins. draft-ietf-mobileip-ipv6-15: Mobility support in IPv6, July 2001.
- [Kit01] H. Kitamura. RFC 3089: A SOCKS-based IPv6/IPv4 gateway mechanism, April 2001.
- [Mon01] G. Montenegro. RFC 3024: Reverse tunneling for mobile ip, January 2001.
- [Moo01] Keith Moore. draft-ietf-ngtrans-6to4-dns: 6to4 and DNS, November 2001.
- [NDO<sup>+</sup>01] Erik Nordmark, Steve Deering, A. Onoe, Brian Zill, and Tatsuya Jinmei. draft-ietf-ipngwg-scoping-arch-03: IP version 6 scoped address architecture, November 2001.
- [Nor00] E. Nordmark. RFC 2765: Stateless IP/ICMP translation algorithm (SIIT), February 2000.

- [Nor01] Erik Nordmark. draft-ietf-ipngwg-site-prefixes-05: Site prefixes in neighbor discovery, February 2001.
- [TDG98] R. Thayer, N. Doraswamy, and R. Glenn. RFC 2411: IP security document roadmap, November 1998. Status: INFORMATIONAL.
- [THA00] K. Tsuchiya, H. Higuchi, and Y. Atarashi. RFC 2767: Dual hosts using the Bump-In-the-Stack technique (BIS), February 2000.
- [TN98] S. Thomson and T. Narten. RFC 2462: IPv6 stateless address auto-configuration, December 1998. Obsoletes RFC1971. Status: DRAFT STANDARD.
- [TS00] G. Tsirtsis and P. Srisuresh. RFC 2766: Network Address Translation - Protocol Translation (NAT-PT), February 2000.