



Xarxes de computadors II



Tema 2 – Administración de ISP

- ▶ a) Arquitectura y direccionamiento en Internet
- ▶ b) Encaminamiento intra-dominio
- ▶ c) Encaminamiento inter-dominio
- ▶ d) Temas de investigación
- ▶ e) Conceptos avanzados

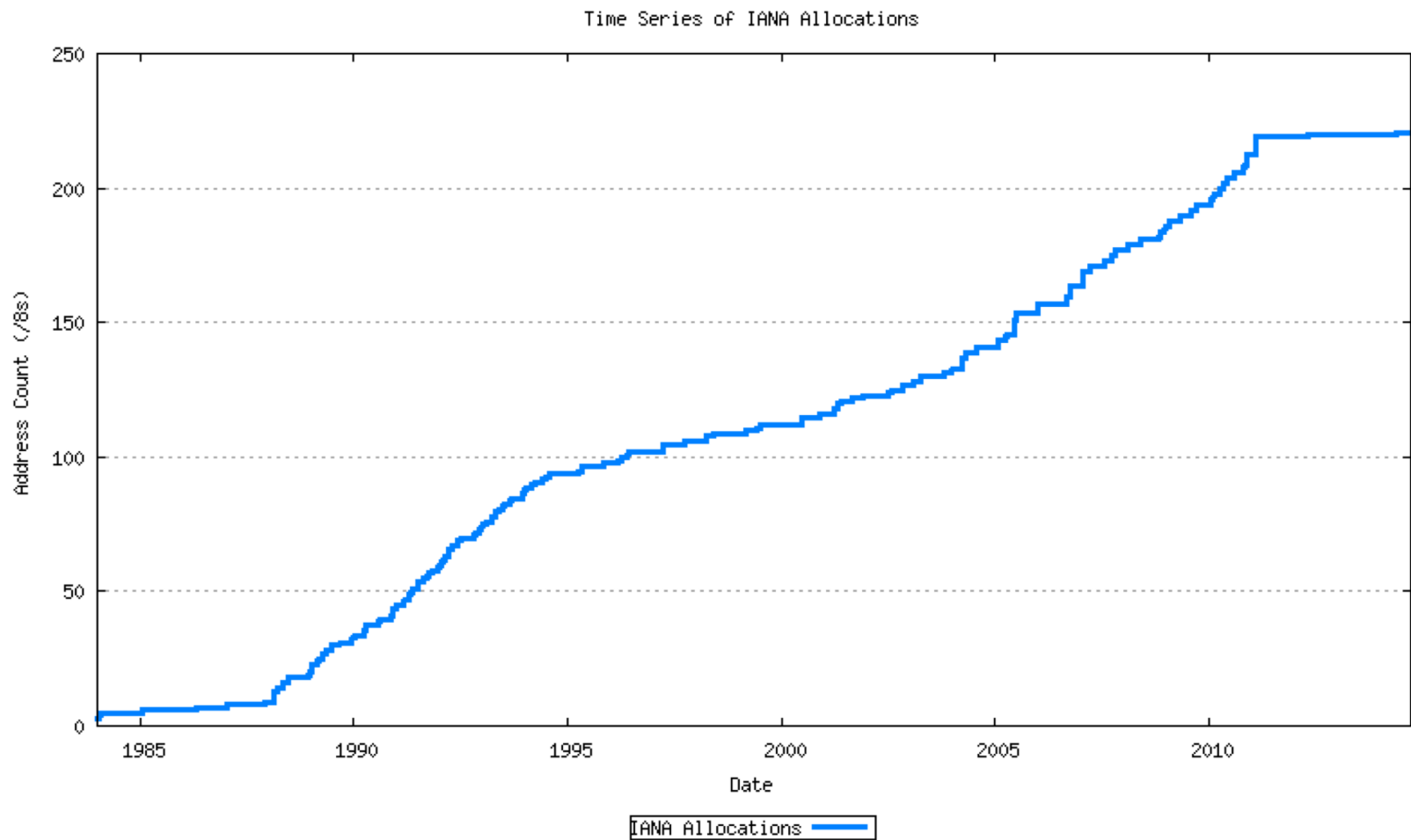


Tema 2 – Administración de ISP

- ▶ a) Arquitectura y direccionamiento en Internet
 - ▶ 1) Entender la arquitectura general de Internet
 - ▶ 2) Identificar los actores principales de Internet
 - ▶ 3) Identificar las organizaciones principales de Internet
 - ▶ 4) **Agotamiento de IPv4 y alguna noción sobre IPv6**
- ▶ b) Encaminamiento intra-dominio
- ▶ c) Encaminamiento inter-dominio
- ▶ d) Temas de investigación
- ▶ e) Conceptos avanzados



Situación en los '90



Dos soluciones

- ▶ Direcciones privadas y NAT
- ▶ Nuevo espacio de direccionamiento más amplio



Direcciones privadas y NAT

▶ Idea

- ▶ Si hay equipos en redes privadas → no necesitan una @IP publica única cada uno

▶ Solución

- ▶ Se crean 3 grupos de @IP privadas (uno por clase)
- ▶ Se pueden usar libremente en redes privadas
- ▶ Se usa NAT para ir a Internet (red publica) para mantener la unicidad de las @IP
- ▶ Se permite que varios equipos puedan compartir pocas (incluido una sola) @IP publica
- ▶ Se reduce la necesidad de @IP publicas

▶ Problemas

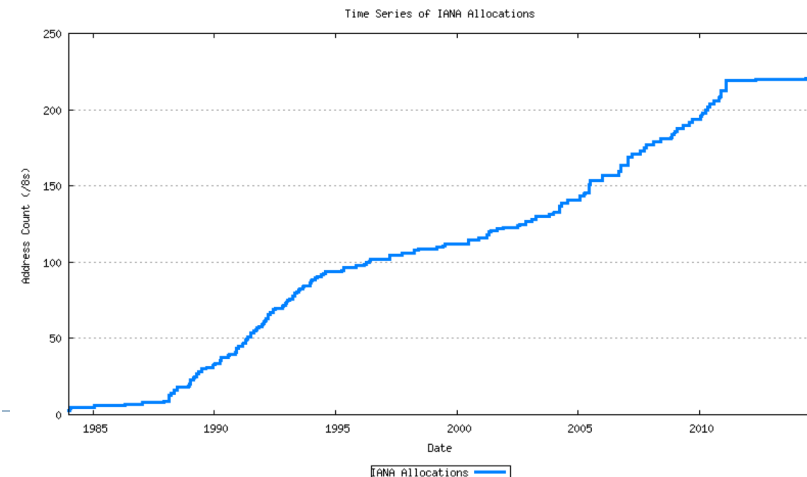
- ▶ Se necesitan tablas de traducción en los routers (la comunicación ya no es extremo-extremo ya que el router interviene los datagramas e incluido las cabeceras de transporte si implementa PAT)



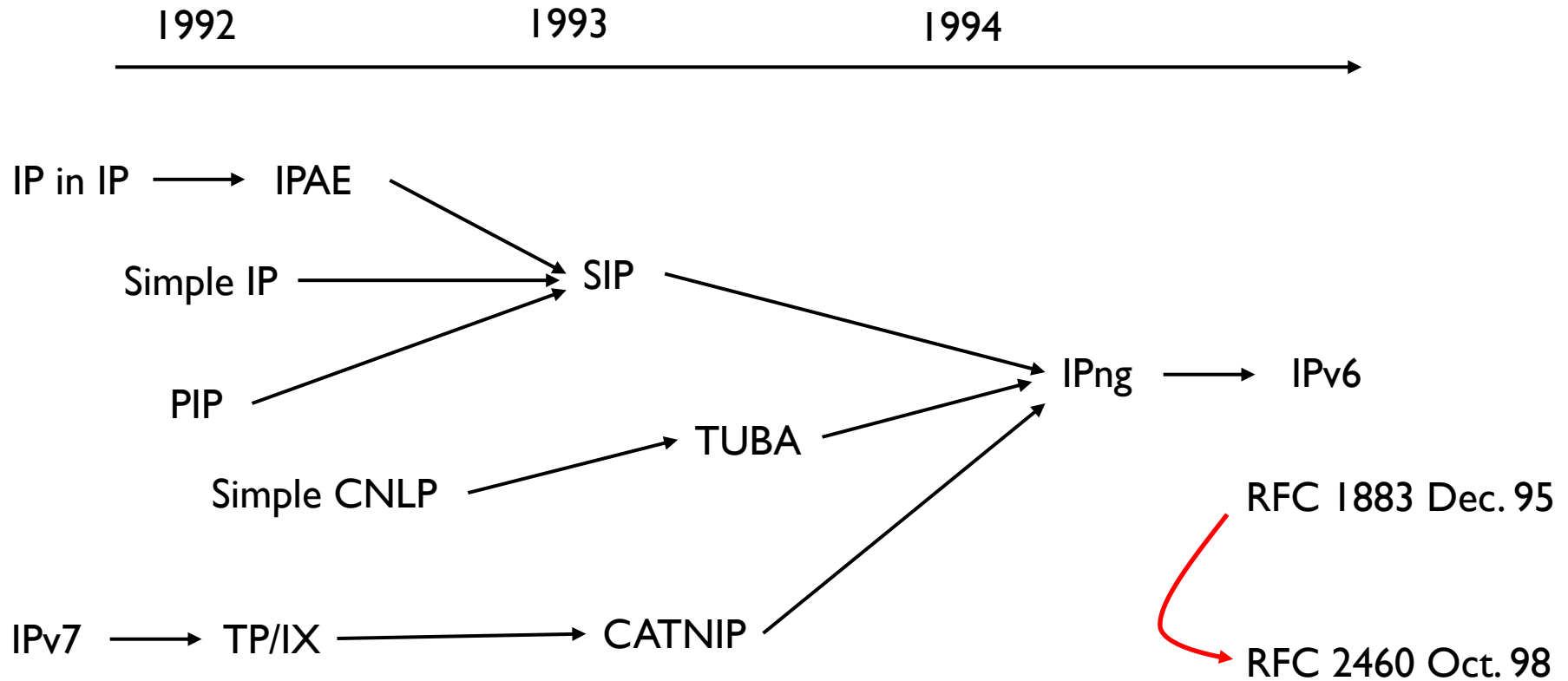
Direcciones privadas y NAT

► Problemas

- Se necesitan tablas de traducción en los routers
- La comunicación ya no es extremo-extremo ya que el router interviene los datagramas, incluida las cabeceras de transporte TCP/UDP si implementa PAT (ya no es un equipo puramente de nivel 3)
- Multicast con NAT difícil de configurar
- VPN con IPsec (autenticación y encriptación)
 - Si el router debe modificar las @IP o los puertos, debe saber como desenscriptar → posible punto vulnerable
- Algunas aplicaciones no funcionan si hay NAT en el medio (como VoIP)
- Finalmente, es una solución a corto plazo como se ve en la grafica

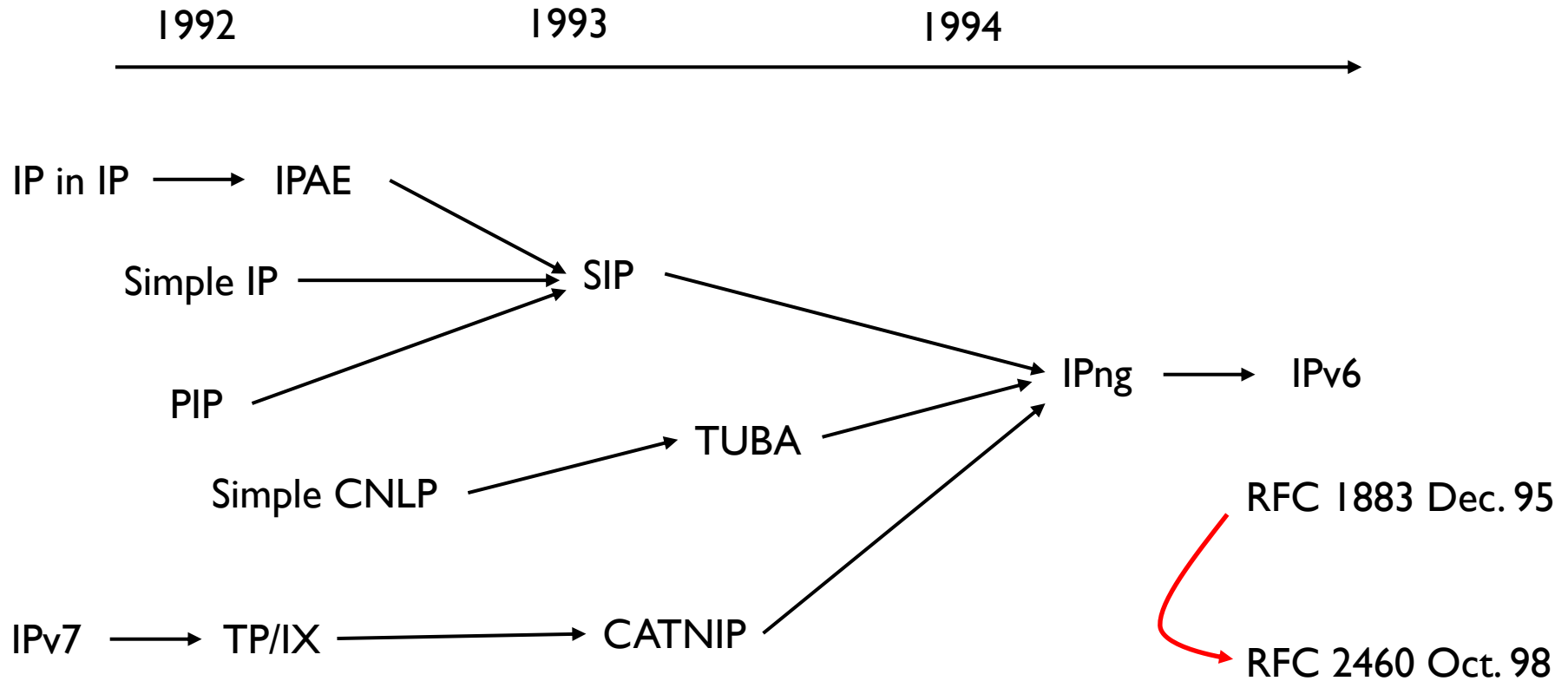


Aumentar el espacio de direccionamiento

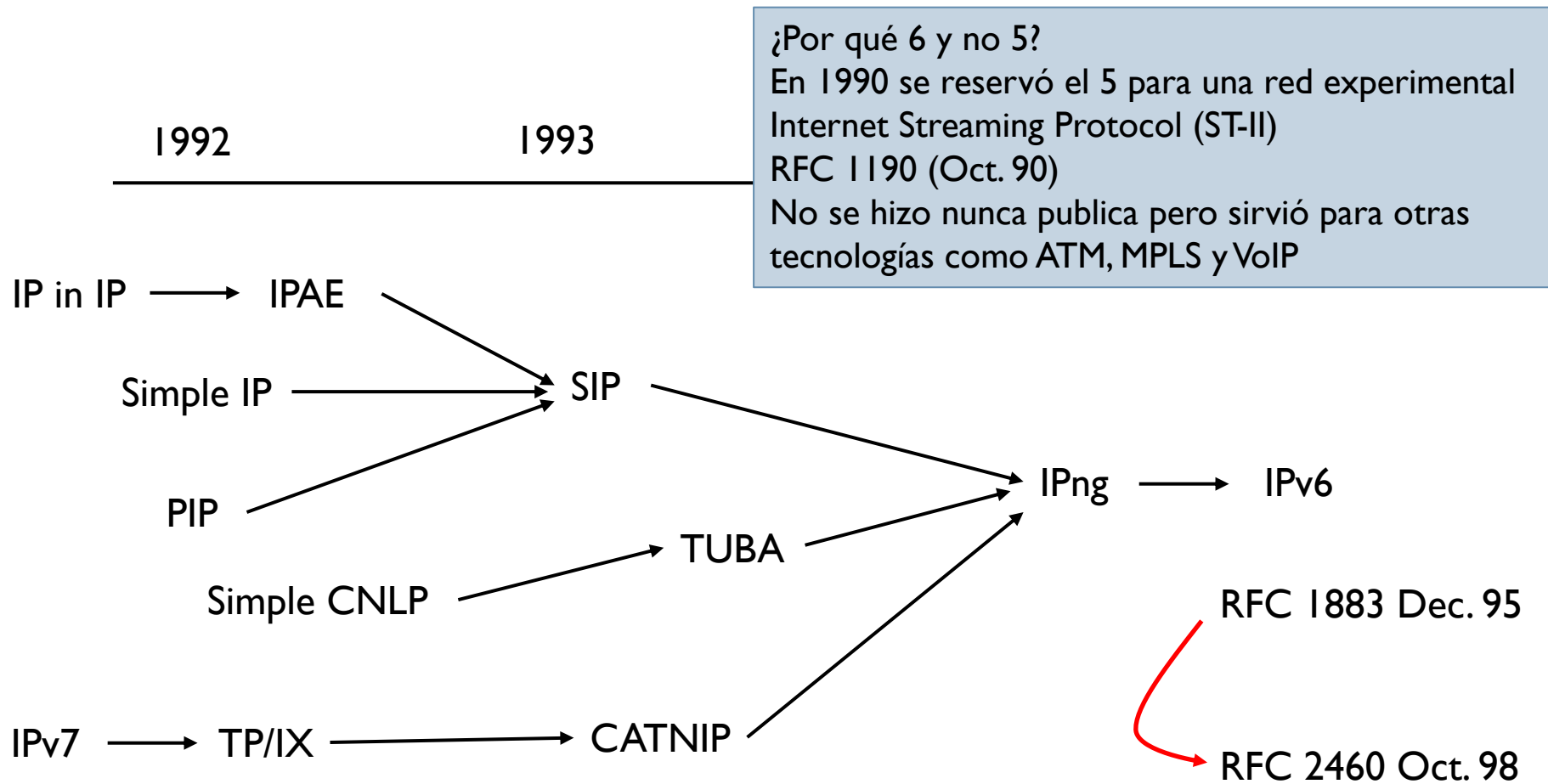


Aumentar el espacio de direccionamiento

¿Por qué 6 y no 5?



Aumentar el espacio de direccionamiento



Aumentar el espacio de direccionamiento

- ▶ Diferentes propuestas

- ▶ 64 bits para las direcciones
- ▶ Longitud variable entre 64 y 160 bits
- ▶ Finalmente se fija a 128 bits
 - $2^{128} \cong 10^{28}$ direcciones IPv6 por persona en el mundo
 - posibilidad de conectar a Internet cualquier dispositivo electrónico
 - Internet of Things (IoT) e Internet of Everything (IoE)



IPv6

- ▶ No compatible con IPv4
 - ▶ Por eso la transición es complicada IPv4 → IPv6
 - ▶ Durante este tiempo Internet funciona con ambos formato
 - ▶ La mayoría de los OS ya soportan IPv6
 - ▶ Los equipos hardware se van actualizando y/o substituyendo poco a poco
 - ▶ Los servicios se van actualizando
- ▶ Comprobar si tenemos IPv6 y estamos conectados a una red IPv6
 - ▶ <http://test-ipv6.com>



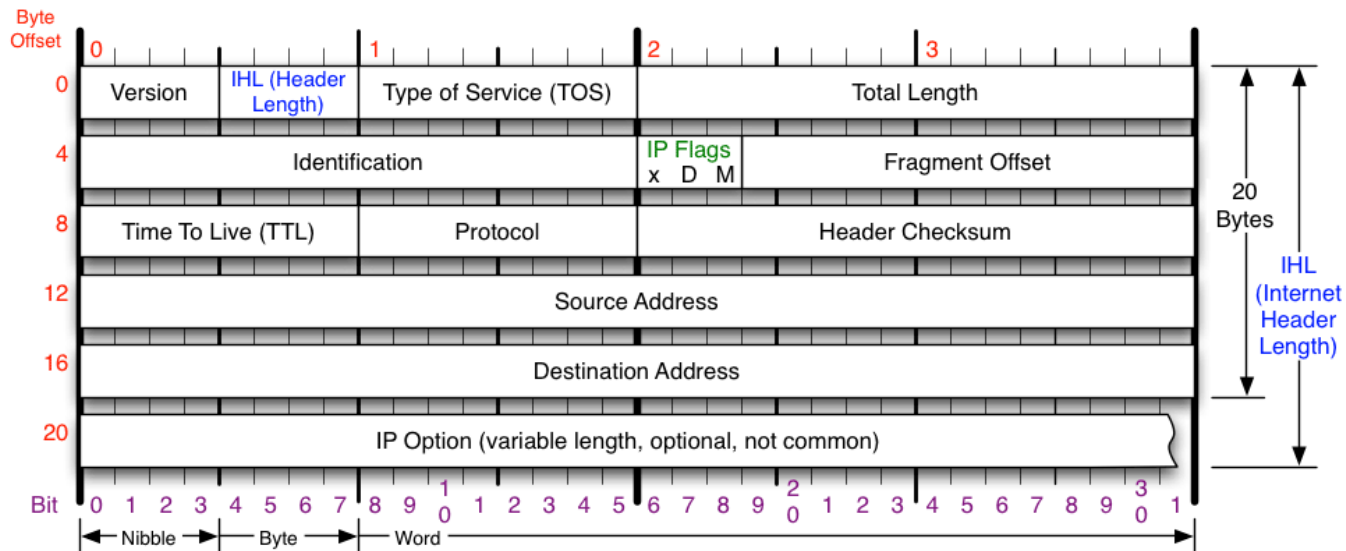
IPv6

- ▶ El 8 de Junio de 2011, Internet Society (ISOC) celebró el World IPv6 day
 - ▶ Los sitios web mas importantes (~1000)
google, youtube, facebook, yahoo, microsoft, cnn, bbc, etc.
hicieron sus paginas accesibles vía IPv6 (e IPv4 claro) durante 24 horas
 - ▶ Muchos de ellos siguen funcionando con IPv6
 - ▶ <http://www.internetsociety.org/ipv6/archive-2011-world-ipv6-day/>
- ▶ El 6 de Junio de 2012, Internet Society (ISOC) abre el World IPv6 launch
 - ▶ Una plataforma para fomentar el despliegue de IPv6 en el mundo
 - ▶ Más operadoras, servicios y fabricantes se apuntan al IPv6
 - ▶ Mantiene medidas en tiempo real sobre el despliegue
 - ▶ <http://www.worldipv6launch.org>



Cabecera IPv6

IPv4 Header



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

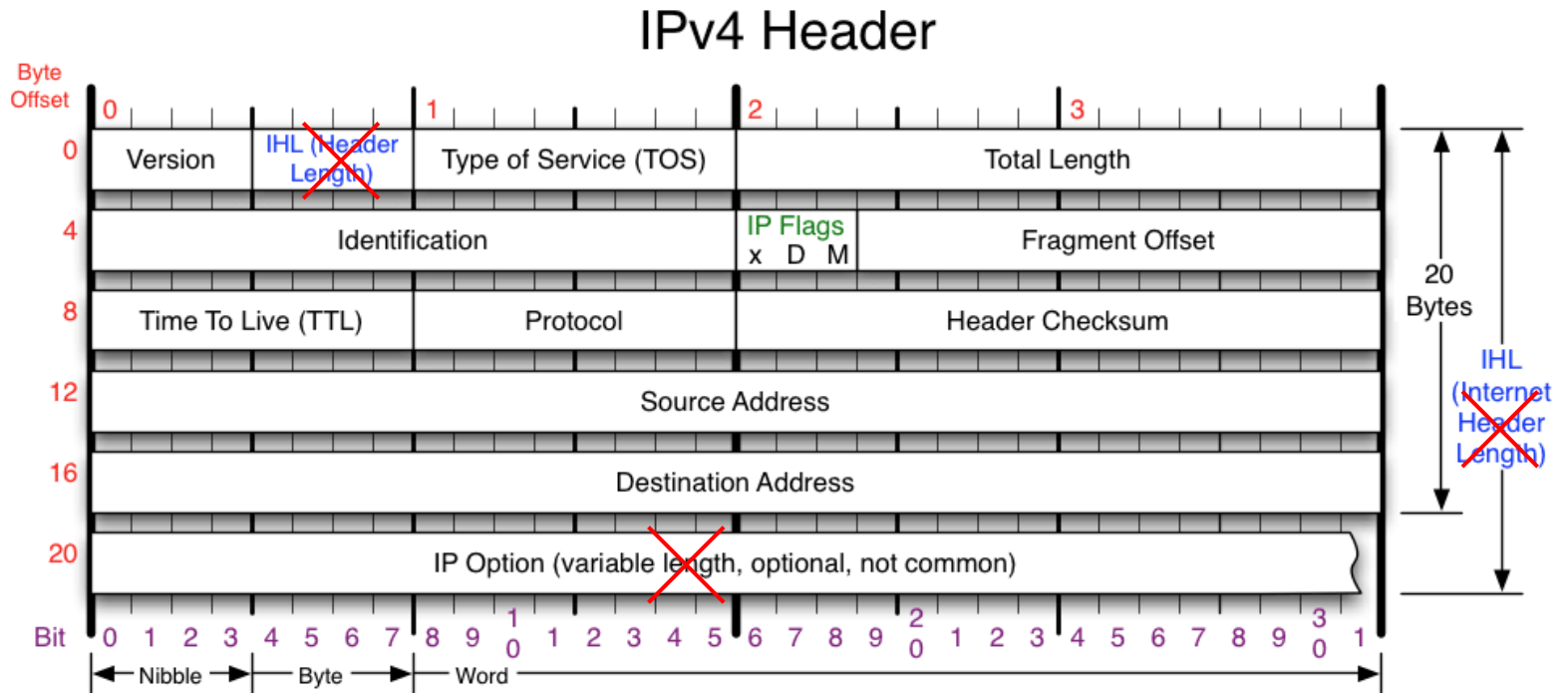
x	D	M
---	---	---

x 0x80 reserved (evil bit)
 D 0x40 Do Not Fragment
 M 0x20 More Fragments follow

RFC 791

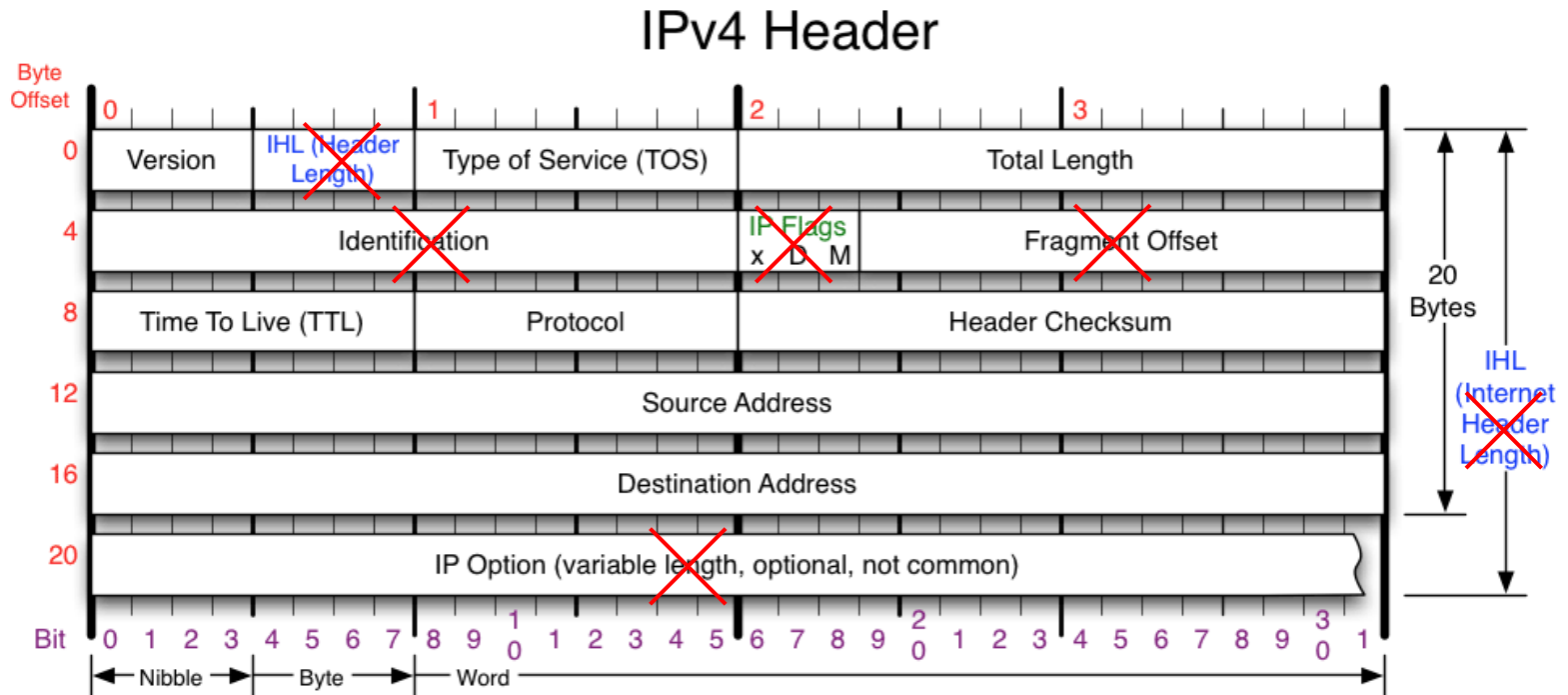
Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Cabecera IPv6



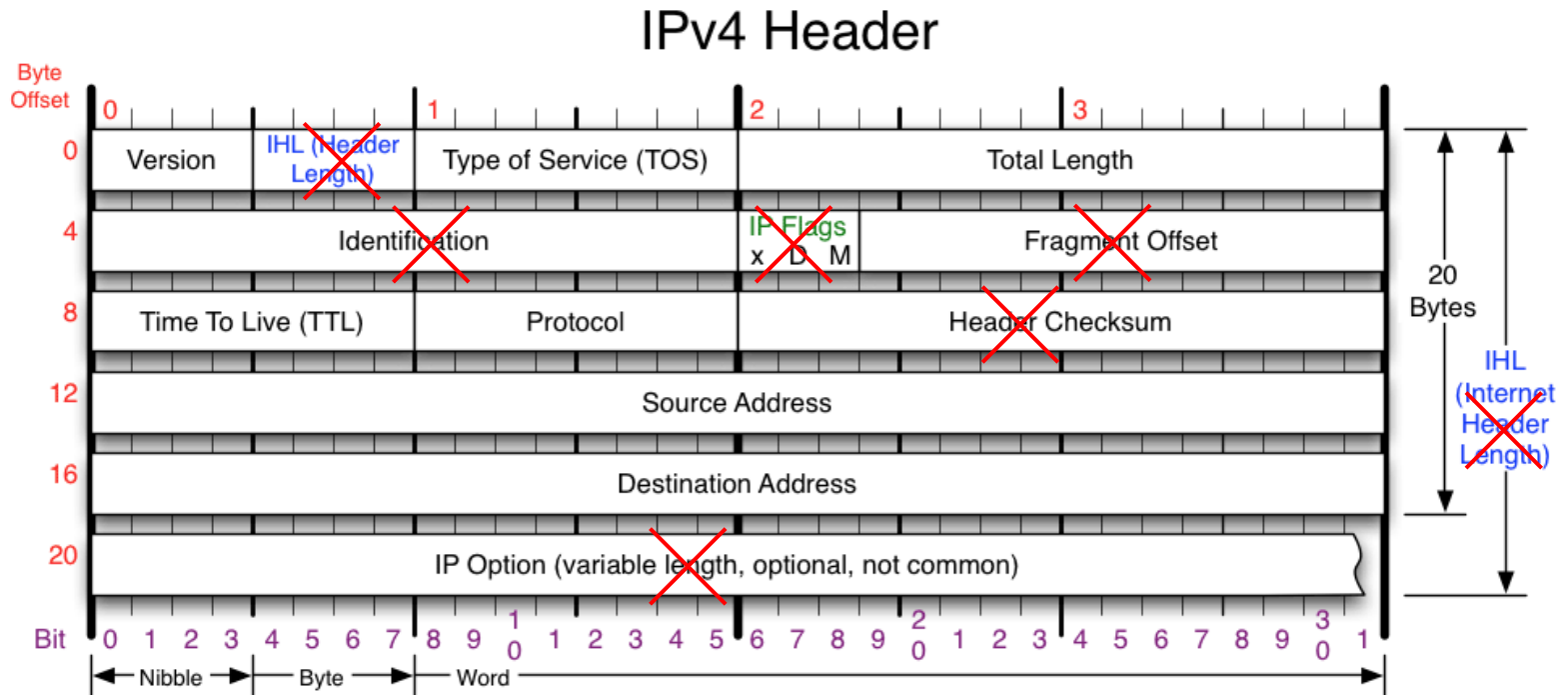
- ▶ Longitud cabecera fija → fuera Header Length y Option

Cabecera IPv6



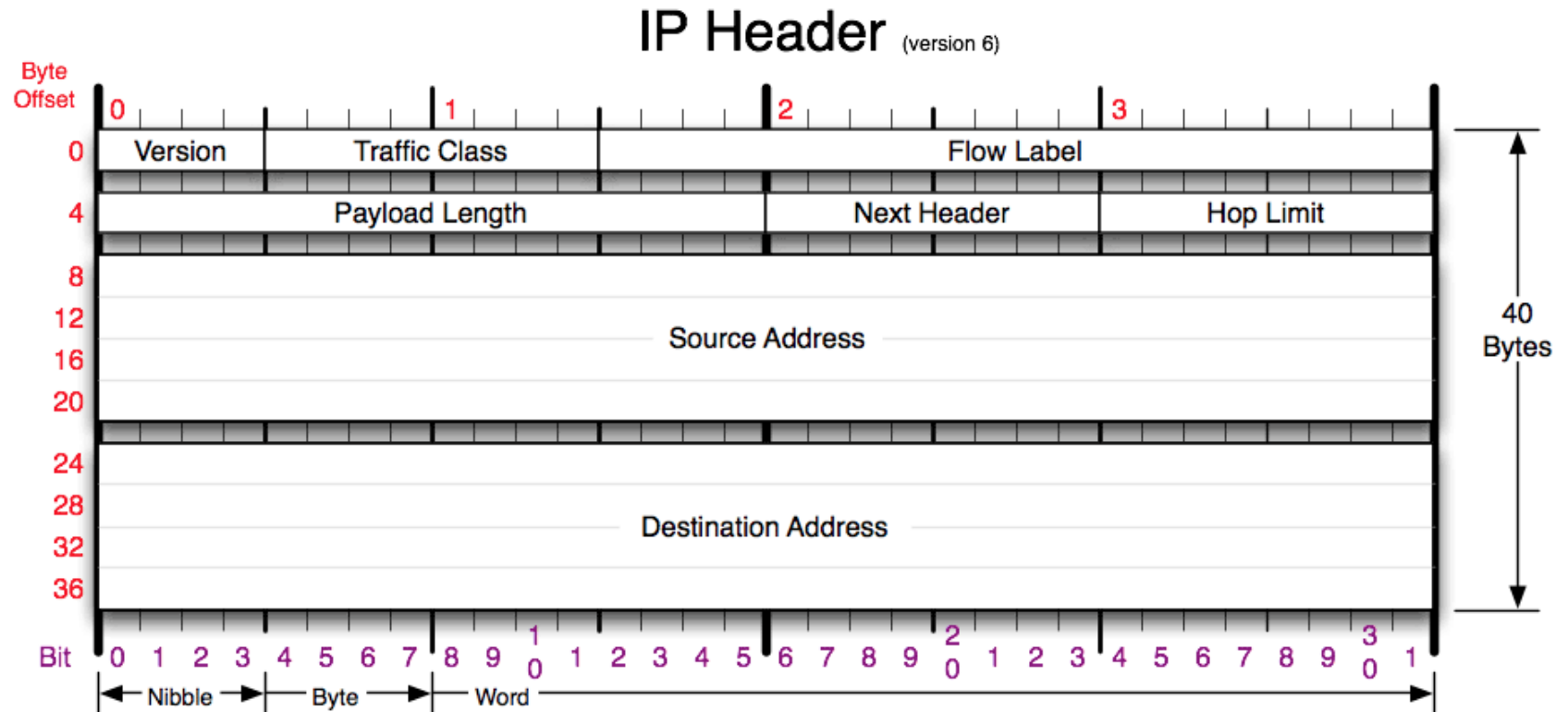
- Fragmentación se evita siempre que se pueda
→ fuera Identification, Flags y Fragment Offset

Cabecera IPv6

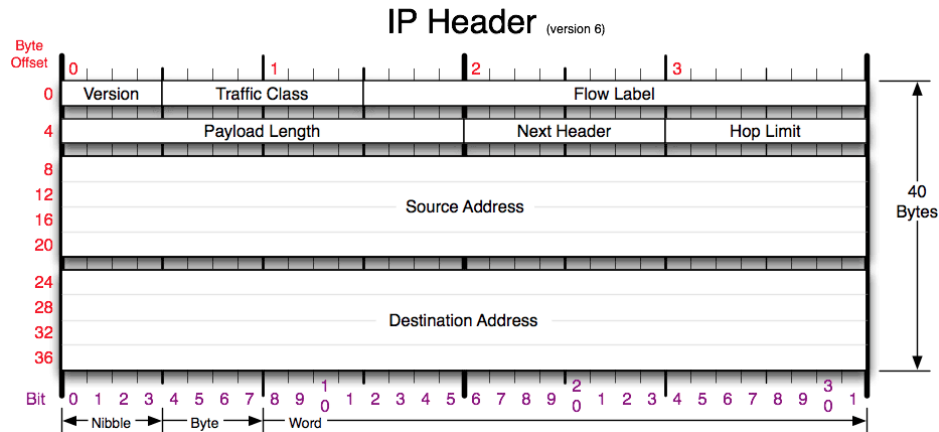
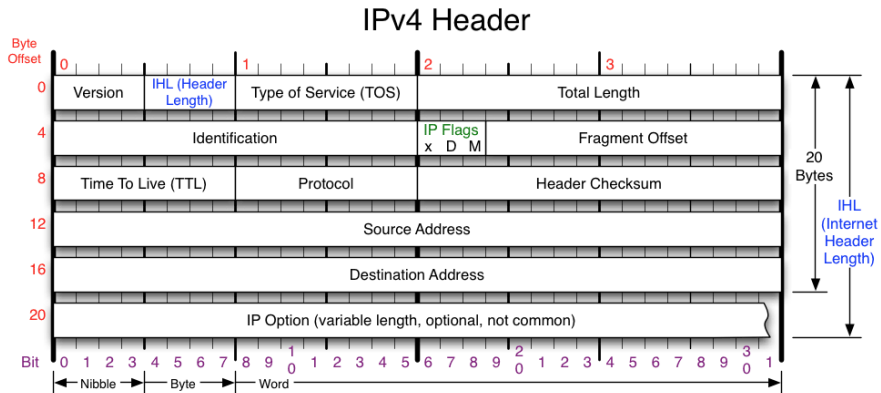


- ▶ Ya se hace un control de error en Transporte e Interfaz de Red
→ fuera Header Checksum

Cabecera IPv6

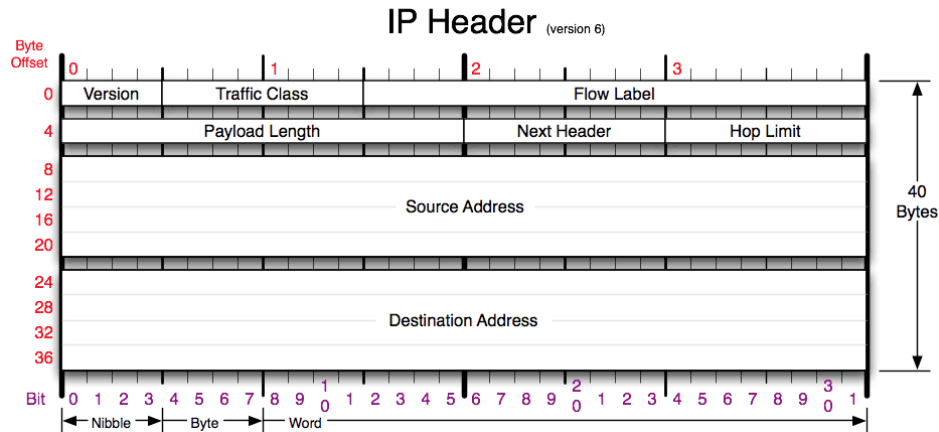
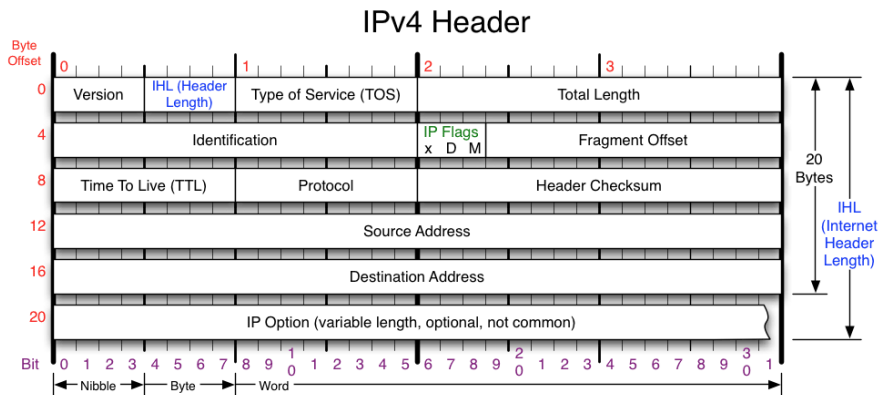


Cabecera IPv6



- Versión: ahora 6
- Longitud cabecera: en IPv6 la cabecera se fija a 40 bytes
- Tipo de servicio: ahora clase de servicio en IPv6 (8 bits)
 - Como en IPv4, este campo depende si los routers están configurados para soportar este servicio
 - 6 bits → Differentiated Service (DiffServ) RFC 2474 → paquetes con prioridades diferentes según unos Code Point (DSCP)
 - 2 bits → Explicit Congestion Notification (ECN) RFC 3168
 - un router puede marcar un paquete cuando está en congestión
 - cuando el destino recibe este paquete y tiene que contestar al origen, crea un eco de este nivel de congestión y lo incluye en su paquete para el origen
 - Al recibir este paquete, el origen baja su tasa de envío (funciona conjuntamente con TCP bajando el tamaño de la ventana de transmisión)

Cabecera IPv6



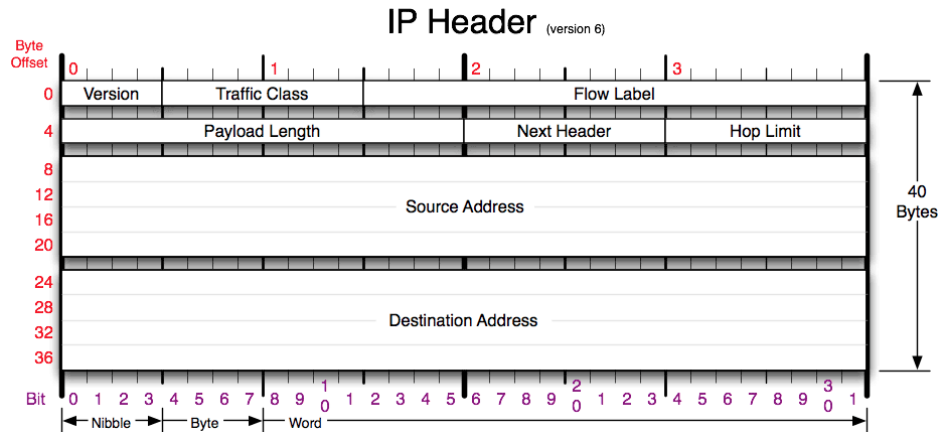
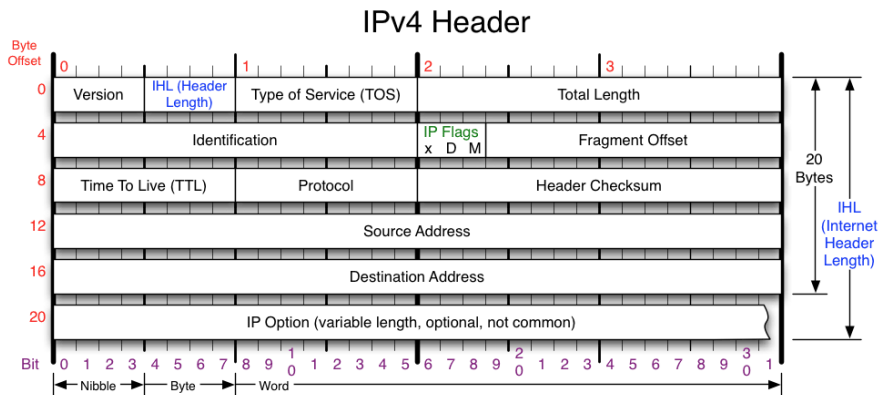
► Etiqueta de flujo: nuevo en IPv6

- Para facilitar el reconocimiento de paquetes que pertenecen a un mismo flujo
 - Secuencia de paquetes relacionados entre sí (por ejemplo de una misma aplicación o de un mismo servicio)
 - Paquetes de una misma sesión TCP
- Todos los paquetes de un mismo flujo se marcan con el mismo valor
- Un valor 0 significa que no se está usando este campo

→ Permite aplicar un control/filtrado basado en flujo y no por paquete (en principio más rápido)

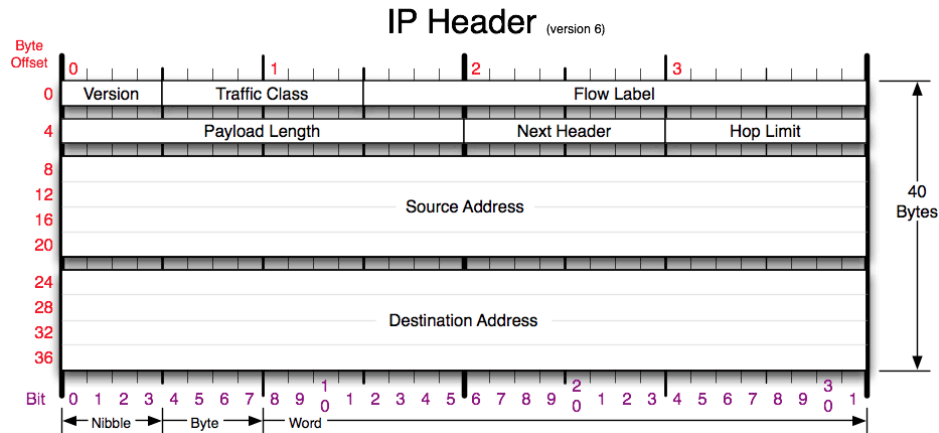
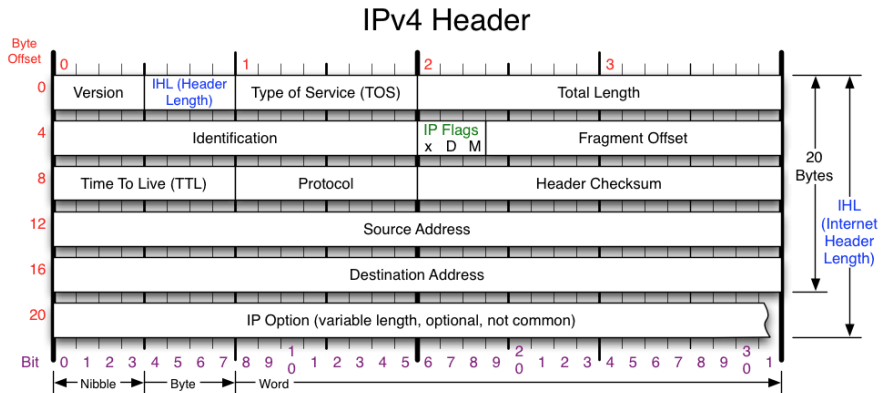
- Longitud total: en IPv6 no se consideran los 40 bytes de la cabecera ya que son fijos y solo cuentan los bytes del payload

Cabecera IPv6



- **Identificación, flags, fragmentos:** se usan para fragmentar IPv4
 - En IPv6 se usa un método distinto y se eliminan de la cabecera
 - Ya que se intenta evitar al máximo la fragmentación y solo puntualmente se necesita fragmentar, es ineficiente tener siempre estos campos en la cabecera
- **Tiempo de vida:** ahora se llama **Limite de Saltos**
 - Mismo funcionamiento que en IPv4
 - El origen pone un valor que indica el número máximo de routers por donde puede pasar el datagrama
 - Cada router disminuye este valor de 1
 - Si al hacer esta operación este campo vale 0, el datagrama se descarta

Cabecera IPv6



- ▶ **IP origen y destino**
 - ▶ Pasa de 32 a 128 bits
- ▶ **Opciones**
 - ▶ No se incluyen en la cabecera IPv6
 - ▶ Si se quieren añadir opciones, se usa un método diferente (luego veremos)

Siguiente Cabecera IPv6

► Funcionamiento “normal”

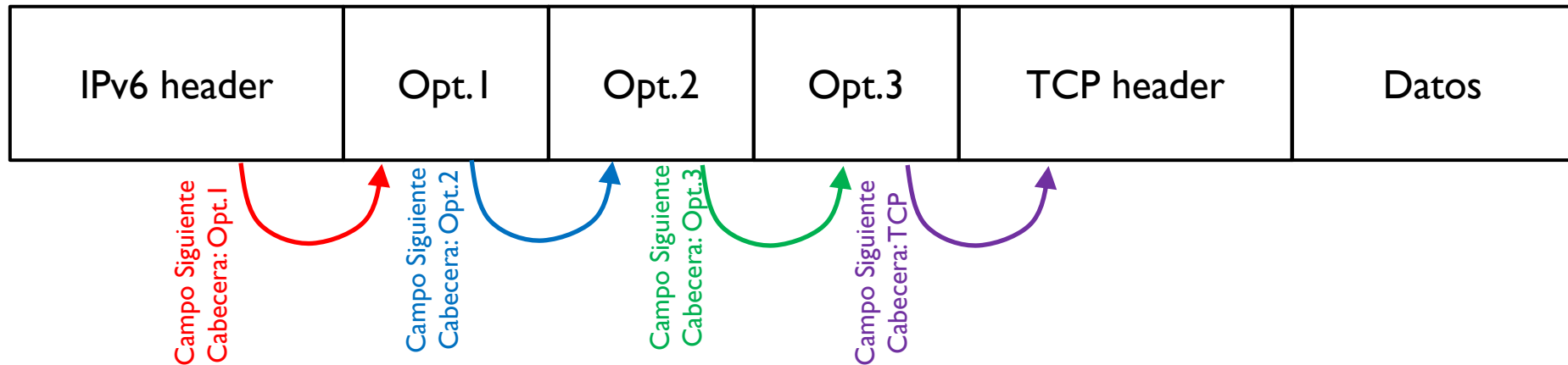


Campo Siguiente
Cabecera: TCP



Siguiente Cabecera IPv6

► Añadir opciones en cascada



- Cada opción tendrá su propio formato, con sus campos, pero siempre hay un campo Siguiente Cabecera que indica que cabecera hay a continuación

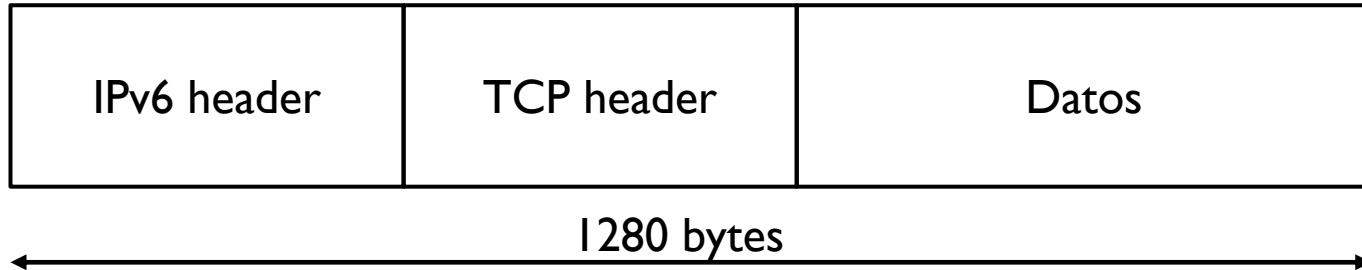
Siguiente Cabecera IPv6

- ▶ Opciones estandarizadas y más usadas
 - ▶ Opciones hop-by-hop
 - ▶ Encaminamiento
 - ▶ Fragmentación
 - ▶ Autenticación
 - ▶ Encapsulamiento seguro
 - ▶ Opciones del destino



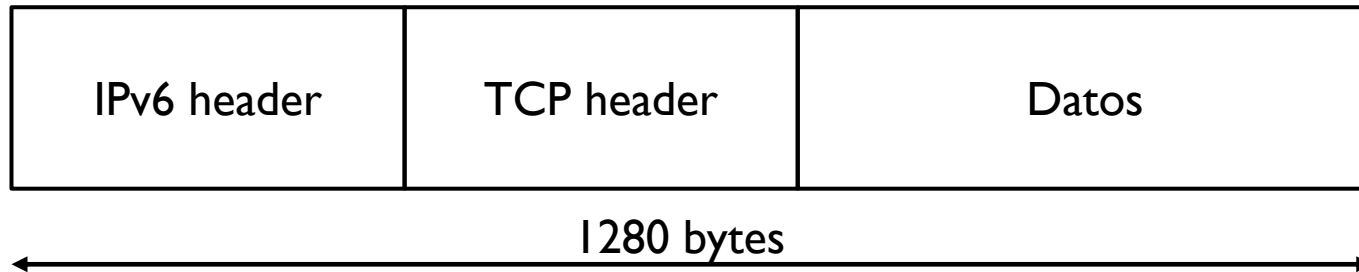
Siguiente Cabecera IPv6

- ▶ Longitud mínima de un IPv6



Siguiente Cabecera IPv6

► Longitud mínima de un IPv6



- ¿Y su MTU de nivel enlace de una tecnología XYZ es menor de 1280 bytes?
- Hay que crear un estándar IPv6overXYZ que separe y junte los IPv6 a nivel enlace (nivel 2, menor que IPv6) para la transmisión con esta tecnología
- De manera que el nivel 3 (IPv6) no se entera y este nivel siempre trate datagramas superiores a 1280 bytes

Notación IPv6

- ▶ Se usan números hexadecimales separados por dos puntos

2031 : 0000 : 130f : 0000 : 0000 : 09c0 : 876a : 130b

- ▶ Se simplifica quitando los 0 no significativos

2031 : 0 : 130f : 0 : 0 : 9c0 : 876a : 130b

- ▶ Se simplifica quitando en un único lugar bloques de 0 seguidos y sustituyéndolo por ::

2031 : 0 : 130f :: 9c0 : 876a : 130b



Notación IPv6

- ▶ Solo se puede hacer esta sustitución en un único lugar ya que de lo contrario, la notación sería ambigua

- ▶ Esta @IPv6

2031 :: 130f :: 09c0 : 876a : 130b

- ▶ Podría ser cualquiera de estas dos

2031 : 0000 : 130f : 0000 : 0000 : 09c0 : 876a : 130b

2031 : 0000 : 0000 : 130f : 0000 : 09c0 : 876a : 130b

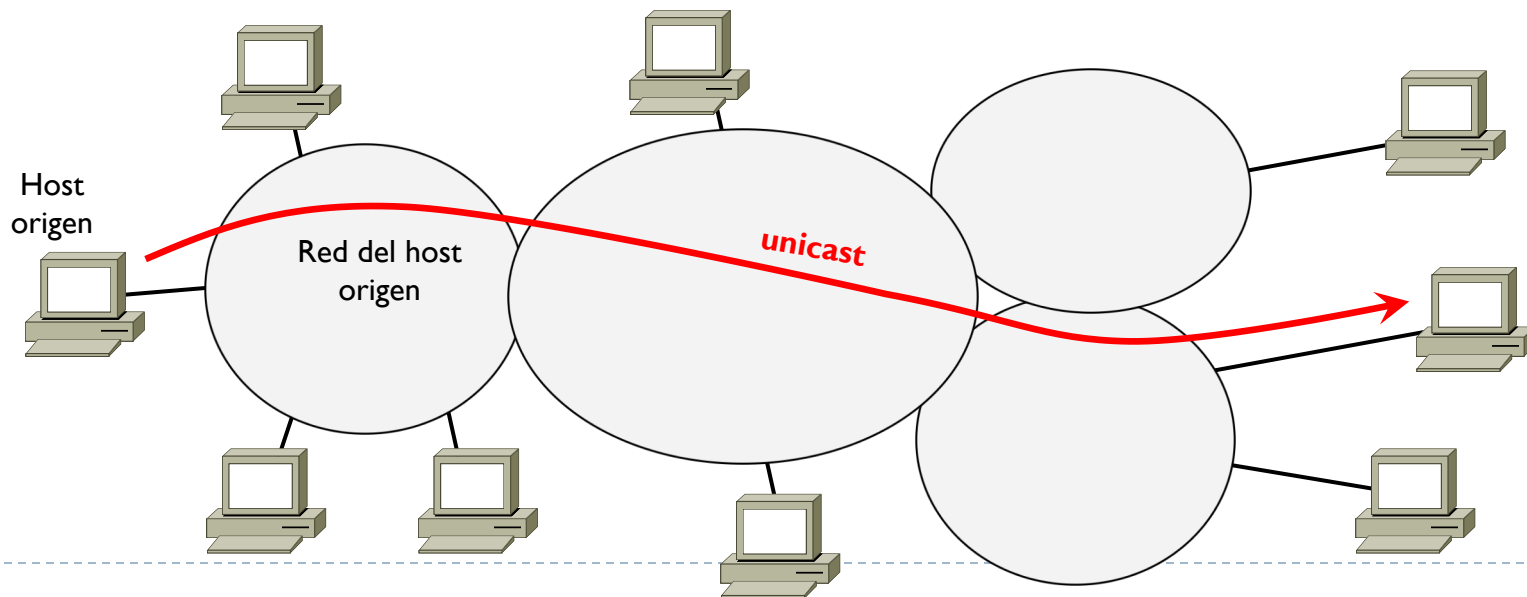
- ▶ Aunque simplificado, una @IPv6 es difícil de recordar

→ El DNS se hace aún mas fundamental



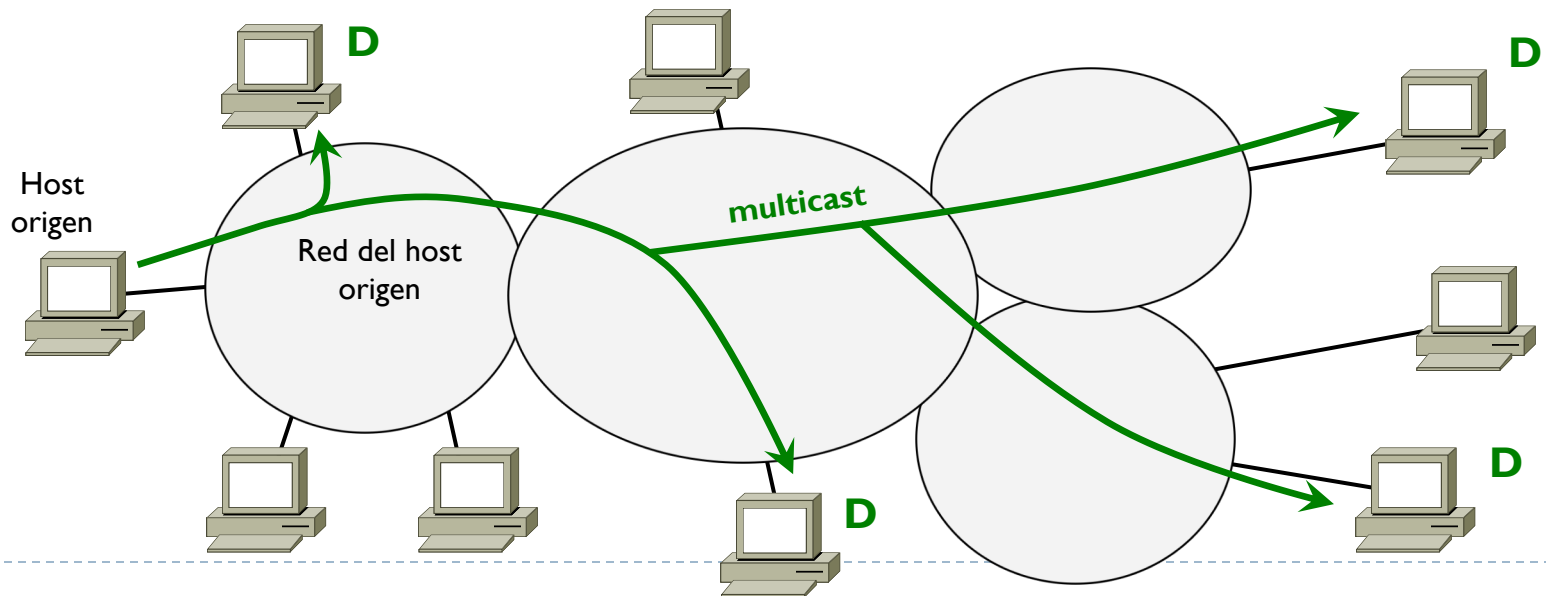
Tipo de datagrama según el destino

- ▶ **Unicast**, un datagrama con un único destino
- ▶ **Multicast**, un datagrama que se replica en la red y alcanza un grupo bien definido de destinos
- ▶ **Anycast**, un datagrama que hay que entregar a un cualquier único destino de un grupo bien definido
 - ▶ Generalmente el que está más cerca



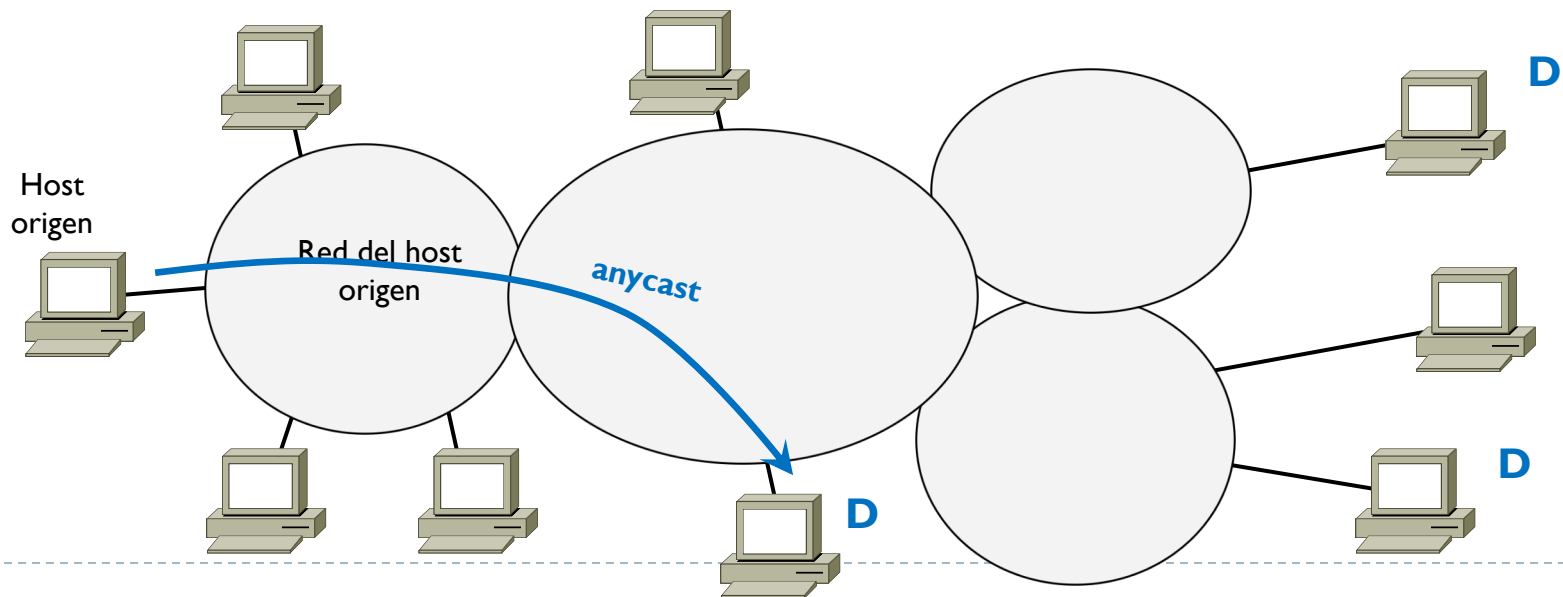
Tipo de datagrama según el destino

- ▶ **Unicast**, un datagrama con un único destino
- ▶ **Multicast**, un datagrama que se replica en la red y alcanza un grupo bien definido de destinos
- ▶ **Anycast**, un datagrama que hay que entregar a un cualquier único destino de un grupo bien definido
 - ▶ Generalmente el que está más cerca



Tipo de datagrama según el destino

- ▶ **Unicast**, un datagrama con un único destino
- ▶ **Multicast**, un datagrama que se replica en la red y alcanza un grupo bien definido de destinos
- ▶ **Anycast**, un datagrama que hay que entregar a un cualquier único destino de un grupo bien definido
 - ▶ Generalmente el que está más cerca



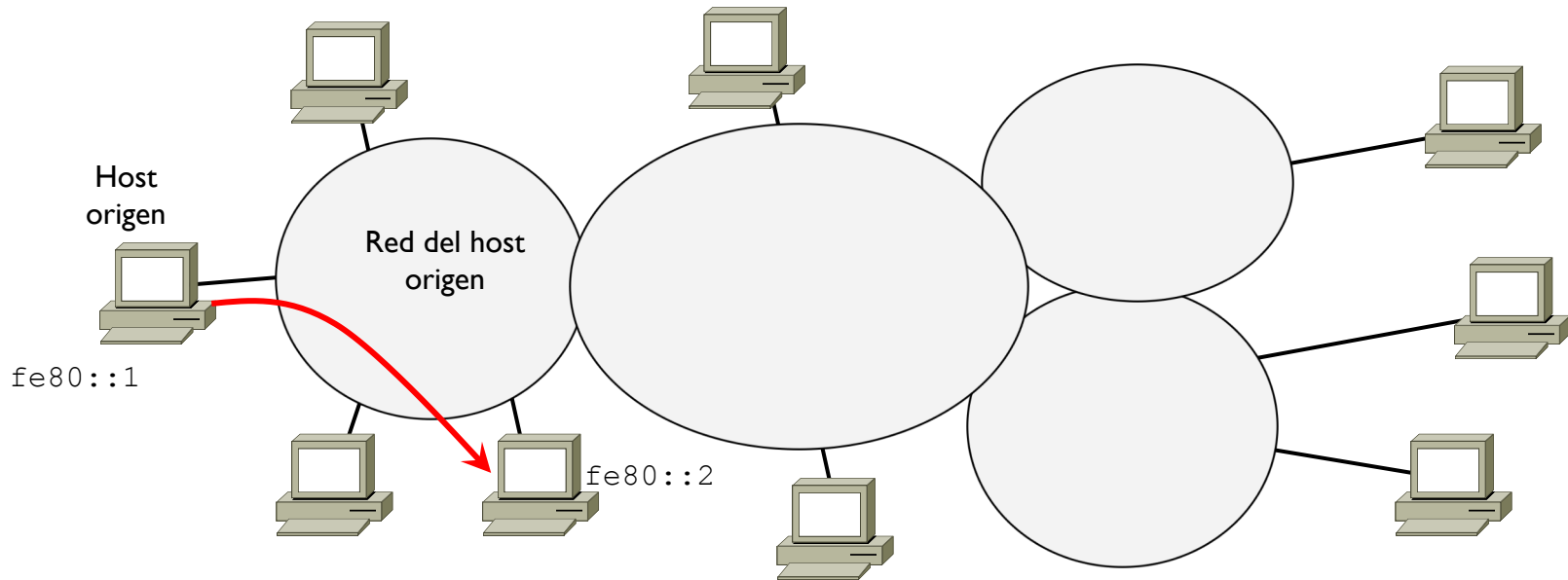
Tipos de @IPv6

- ▶ Inicialmente se crearon 3 tipos que se pueden asignar a cada interfaz
- ▶ **Link-local**
- ▶ **Site-local**
- ▶ **Global**



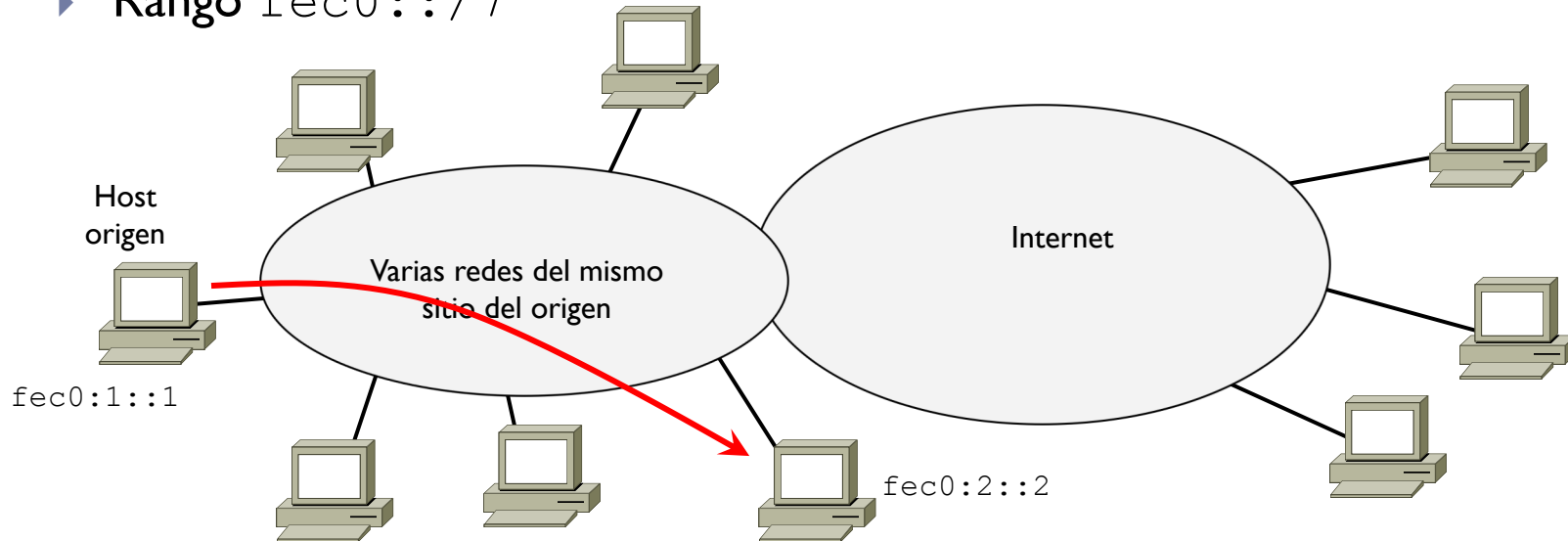
Link-local

- ▶ Para transmisión a destinos de la misma red del origen
- ▶ Origen y destino usan un routing-prefix del rango
- ▶ `fe80::/10`
- ▶ completado con otros 54 bits y luego el InterfacelD de 64 bits



Site-local

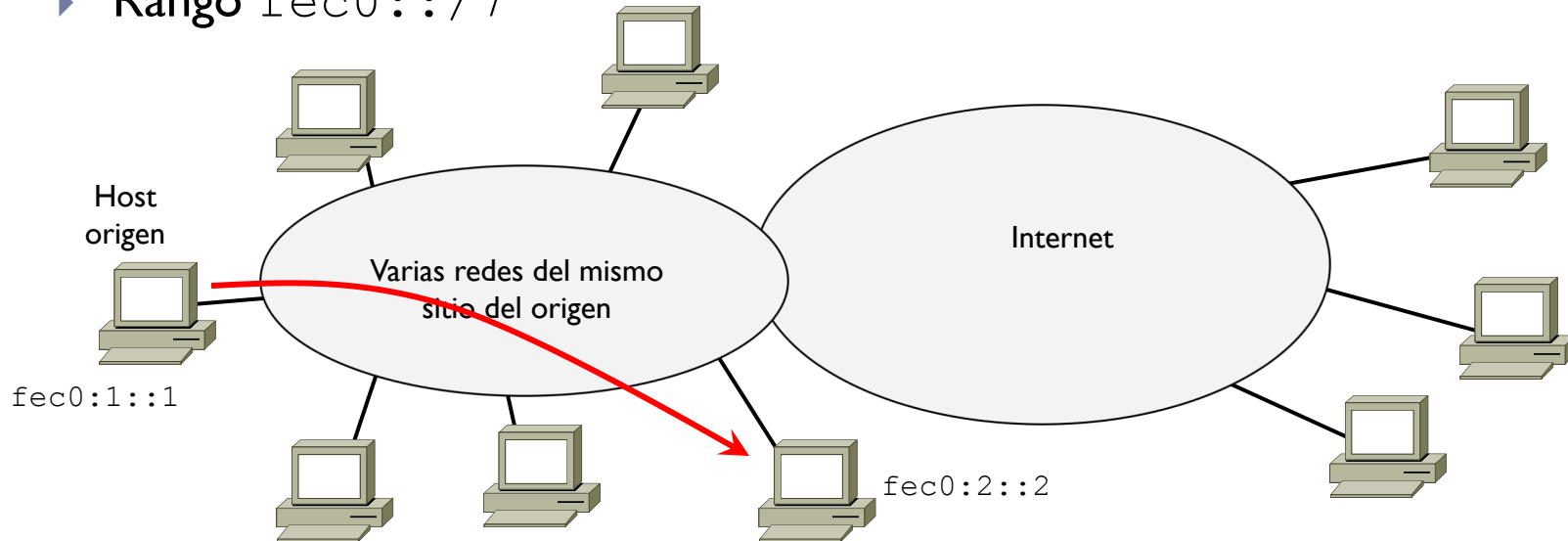
- ▶ Para transmisión entre un origen y un destino dentro de un mismo “sitio”
- ▶ Rango `fec0::/7`



- ▶ Son direcciones privadas no enrutables en Internet

Site-local

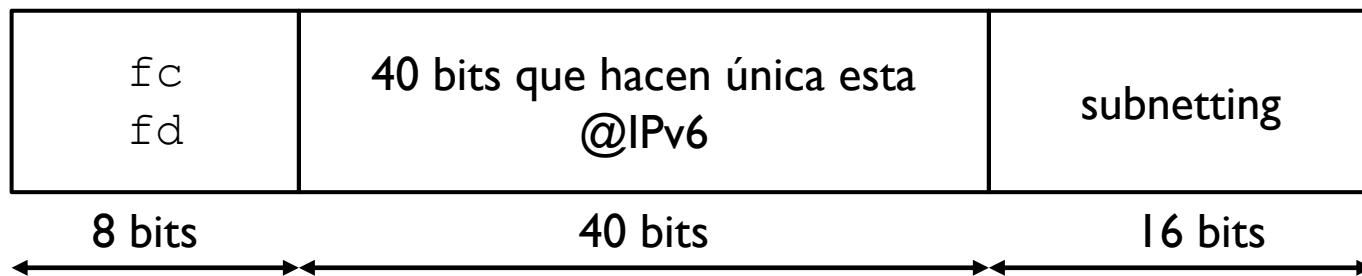
- ▶ Para transmisión entre un origen y un destino dentro de un mismo “sitio”
- ▶ Rango `fec0::/7`



- ▶ En septiembre 2004 (RFC 3879), esta @IPv6 ya no se soporta porque ambigua
 - ▶ ¿Qué es un site y cual es su limite?
- ▶ Y porque no facilita una administración de red simple con poca intervención en caso de cambios en la infraestructura
 - ▶ por ejemplo, al juntar dos sistemas, hay que reconfigurar todo el site otra vez

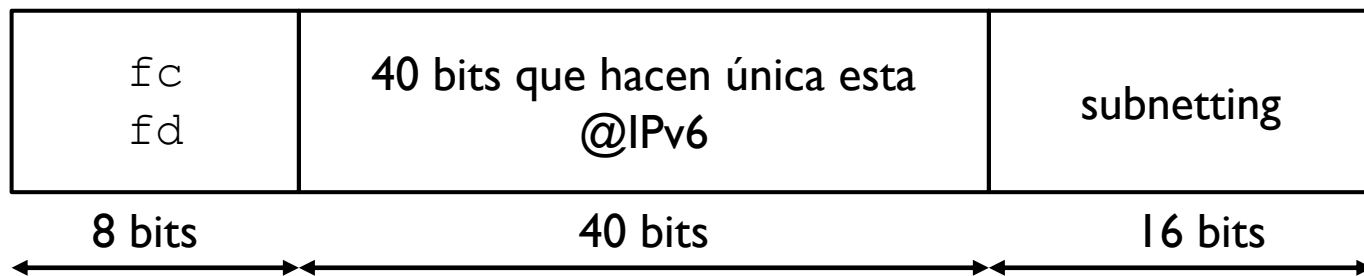
Unique Local Address (ULA)

- ▶ Unique Local Address (ULA)
 - ▶ En octubre 2005 (RFC 4193), se crea la ULA como sustituida de site-local
 - ▶ Pensada como dirección privada (equivalente a las de IPv4) pero que además sean únicas
 - ▶ Facilita la administración de red ya que eventuales cambios no afectan todo el sistema porque se garantiza que no haya redes duplicadas
- ▶ Rango `fc00::/7`
- ▶ Se crearon 2 grupos con 2 soluciones diferentes
 - ▶ Rango `fc00::/8`
 - ▶ Rango `fd00::/8`



ULA

- ▶ **Rango** `fc00::/8`
 - ▶ Los 40 bits únicos los proporciona una entidad centralizada que controla esta unicidad
 - ▶ Esta entidad actualmente aún no está disponible
- ▶ **Rango** `fd00::/8`
 - ▶ Los 40 bits únicos se generan con un algoritmo definido en el RFC 4193
 - ▶ Método usado actualmente



Tipos de @IPv6

- ▶ Otras direcciones reservadas (principales)
- ▶ `::/128` → Dirección no especificada, valor inicial de las tarjetas
- ▶ `::1/128` → Loopback
- ▶ `ff00::/8` → Multicast

- ▶ Y muchísimas más
 - ▶ `::ffff:0:0/96` reserved for IPv4-mapped Address
 - ▶ `64:ff9b::/96` is used in an algorithmic mapping between IPv4 to IPv6 addresses
 - ▶ `2001:0000::/32` reserved for TEREDO
 - ▶ `2001:0002::/48` reserved for Benchmarking
 - ▶ `2001:5::/32` reserved for EID Space for LISP
 - ▶ `2001:db8::/32` reserved for Documentation
 - ▶ `2002::/16` reserved for 6to4
 - ▶ etc.

<https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml>



Multicast IPv6

- ▶ **Bloque asignado a Multicast:** `ff00::/8`
 - ▶ Es decir, los primeros 8 bits son todos 1 en binario
- ▶ **Existen grupos multicast ya definidos**
 - ▶ Al enviar un datagrama con destino estas @IPv6, el datagrama alcanza todos estos destinos
 - ▶ `ff02::1` → todos los nodos de una LAN
 - ▶ `ff02::2` → todos los routers de una LAN
 - ▶ `ff02::9` → todos los routers RIP de una LAN
 - ▶ `ff02::1:2` → todos los servidores DHCP de una LAN
 - ▶ Y muchos más

<https://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml>



Anycast IPv6

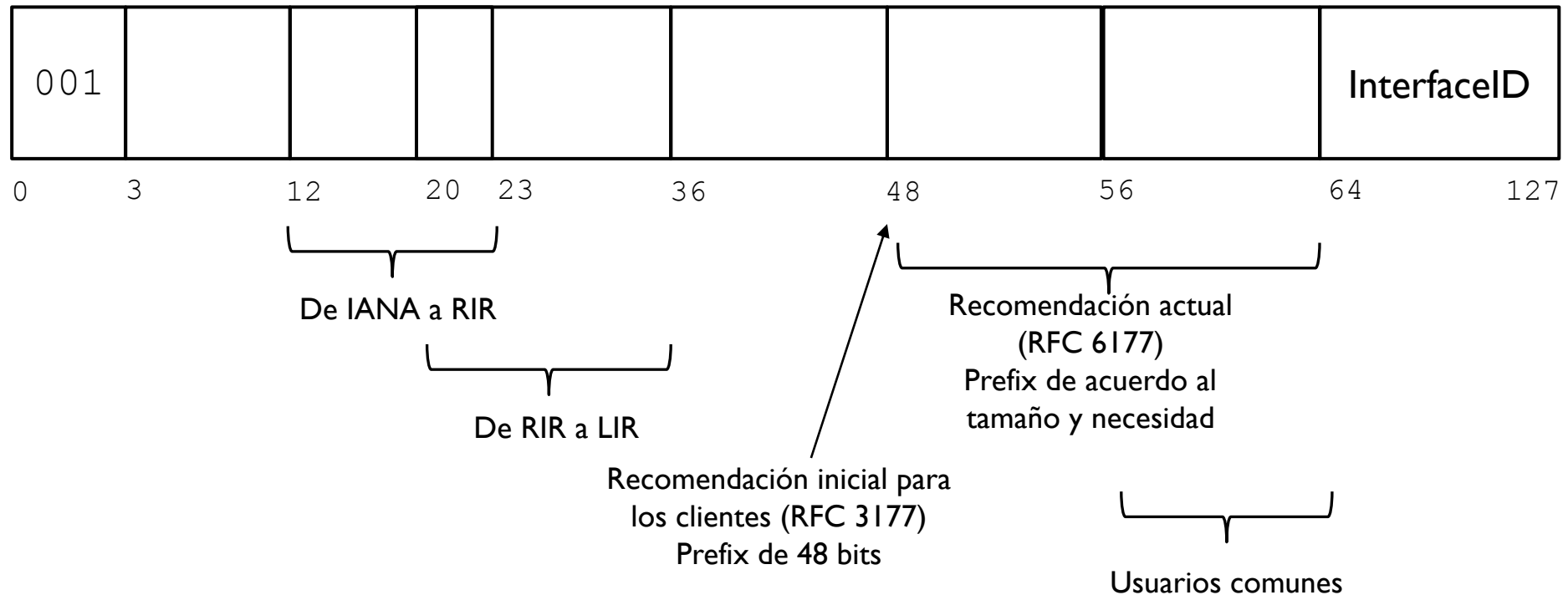
- ▶ Se usan @IPv6 unicast y globales
- ▶ Las direcciones anycast son sintácticamente indistinguibles de las direcciones unicast.
- ▶ Cuando una dirección unicast se asigna a más de una interfaz, se convirtié en una dirección anycast y los nodos a los que se asigna la dirección deben configurarse explícitamente para saber que es una dirección anycast



IPv6 global unicast

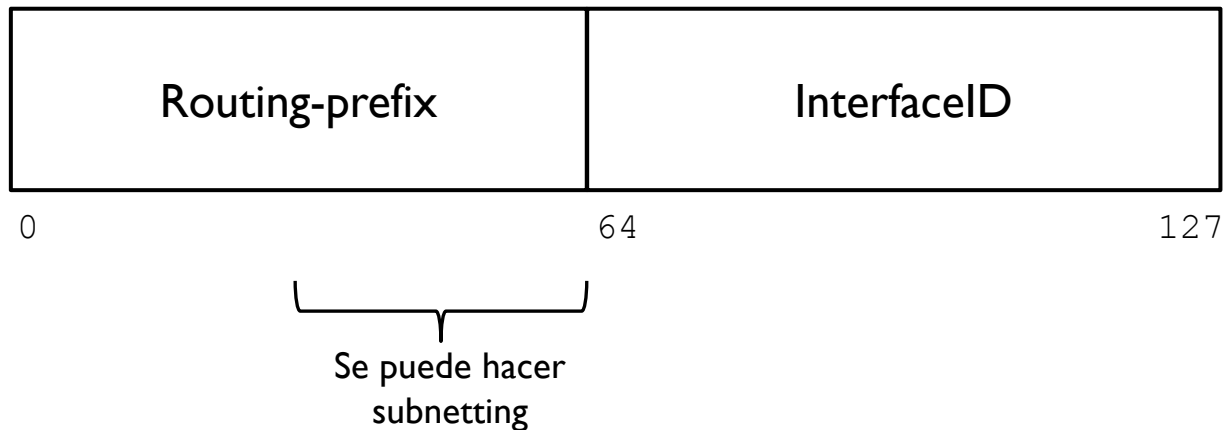
- ▶ Todo lo que queda disponible
- ▶ Actualmente IANA solo está asignado @IPv6 de este rango

2000::/3



Como se asignan @IPv6

- ▶ No existen las direcciones de red y broadcast como en IPv4

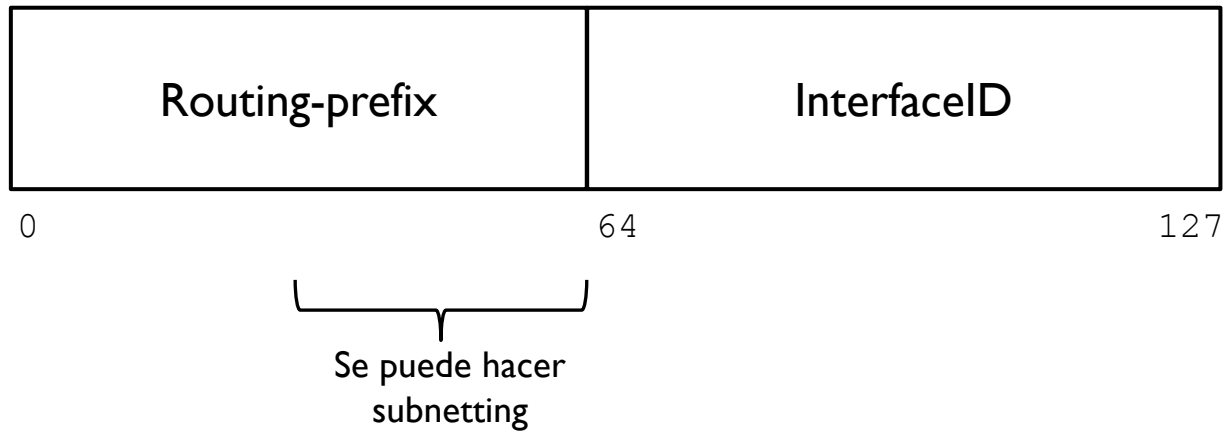


- ▶ 2002:4c0::74:1:27/52
 - ▶ Routing prefix: ?
 - ▶ InterfaceID: ?



Como se asignan @IPv6

- ▶ No existen las direcciones de red y broadcast como en IPv4



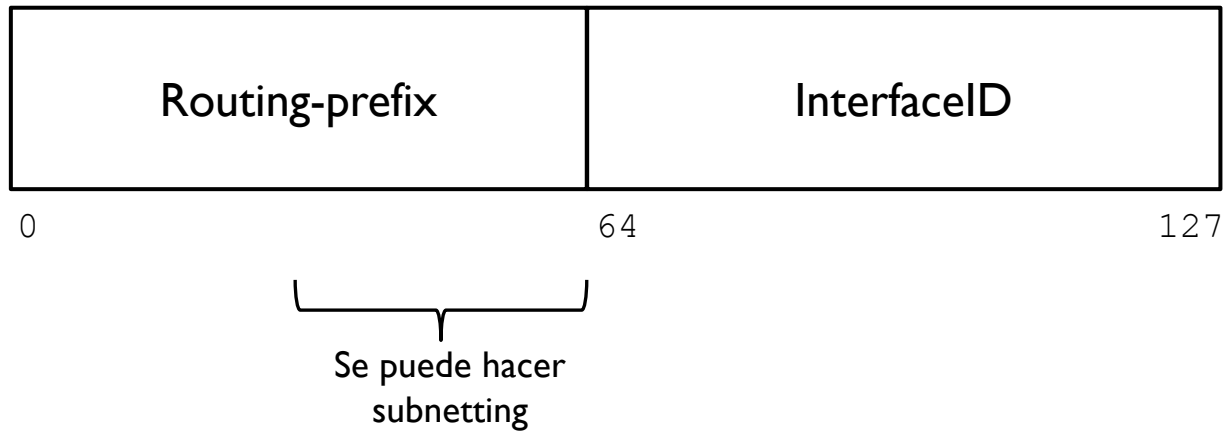
- ▶ `2002:4c0::74:1:27/52`

52 bits de routing-prefix $\rightarrow 64 - 52 = 12$ bits de subnetting



Como se asignan @IPv6

- ▶ No existen las direcciones de red y broadcast como en IPv4

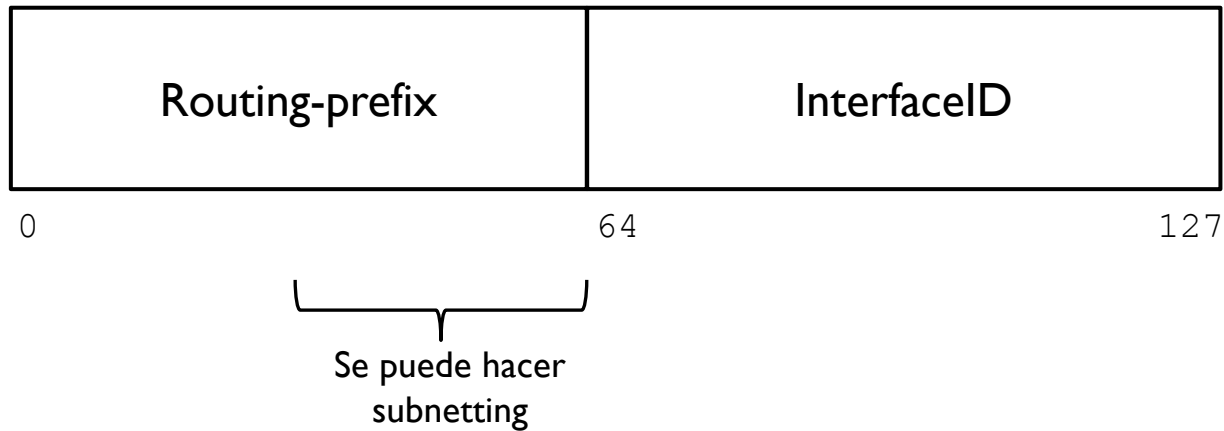


- ▶ 2002:4c0::74:1:27/52

52 bits de routing-prefix $\rightarrow 64 - 52 = 12$ bits de subnetting

Como se asignan @IPv6

- ▶ No existen las direcciones de red y broadcast como en IPv4



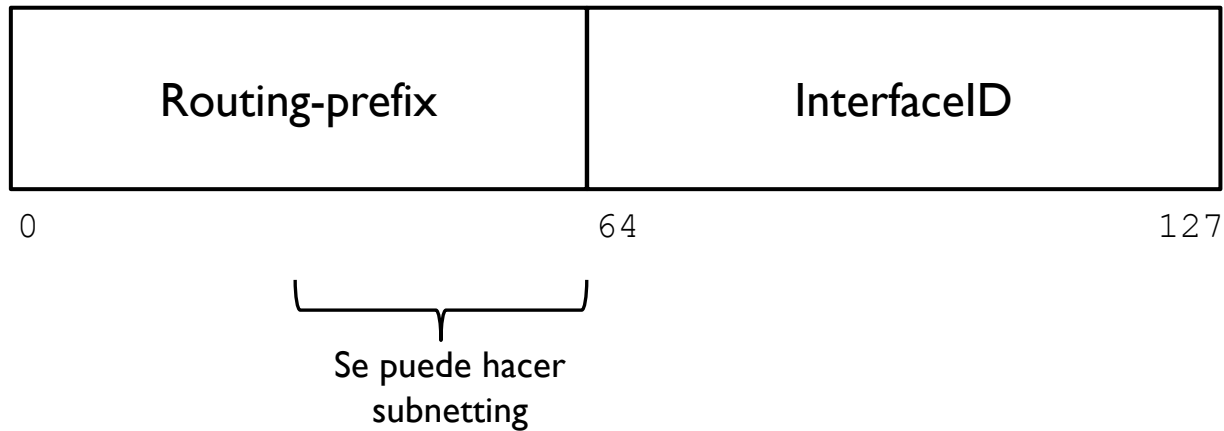
- ▶ 2002:4c0::74:1:27/52

52 bits de routing-prefix $\rightarrow 64 - 52 = 12$ bits de subnetting

64 bits de interfaceID $\rightarrow 4$ grupos de 16 bits

Como se asignan @IPv6

- ▶ No existen las direcciones de red y broadcast como en IPv4



- ▶ 2002:4c0::74:1:27/52
 - ▶ Routing prefix: 2002:04c0:0000:0
 - ▶ Subnet Routing prefix: 000
 - ▶ InterfaceID: :0000:0074:0001:0027

Como se asignan @IPv6: 2 formas

▶ Stateful

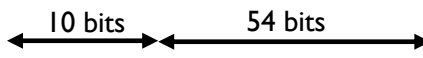
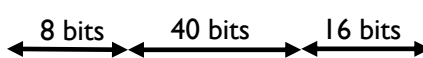
- ▶ Conocimiento completo de los “estados”
- ▶ Una entidad mantiene todas las IP de los hosts y evita que se dupliquen
- ▶ Por ejemplo servidores DHCP o manualmente

▶ Stateless

- ▶ Sin conocer todos los “estados”
- ▶ Cada host se autoconfigura correctamente sin duplicar @IP

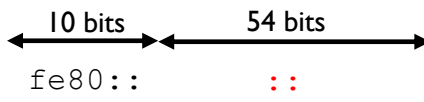
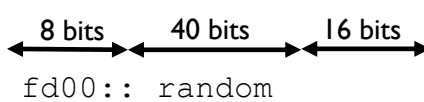


¿Qué es lo que hay que asignar?

		Routing-prefix	interfaceID
Link-local		 fe80::	
(opcional)	ULA	 fd00:: random	
Global			



¿Qué es lo que hay que asignar?

	Routing-prefix 64 bits	interfaceID 64 bits
Link-local	 fe80:: ::	
ULA	 fd00:: random	
Global		

Como el alcance está limitado a la misma red y si no hay alguna razón específica para que sea diferente, estos 54 bits suelen ser todos 0



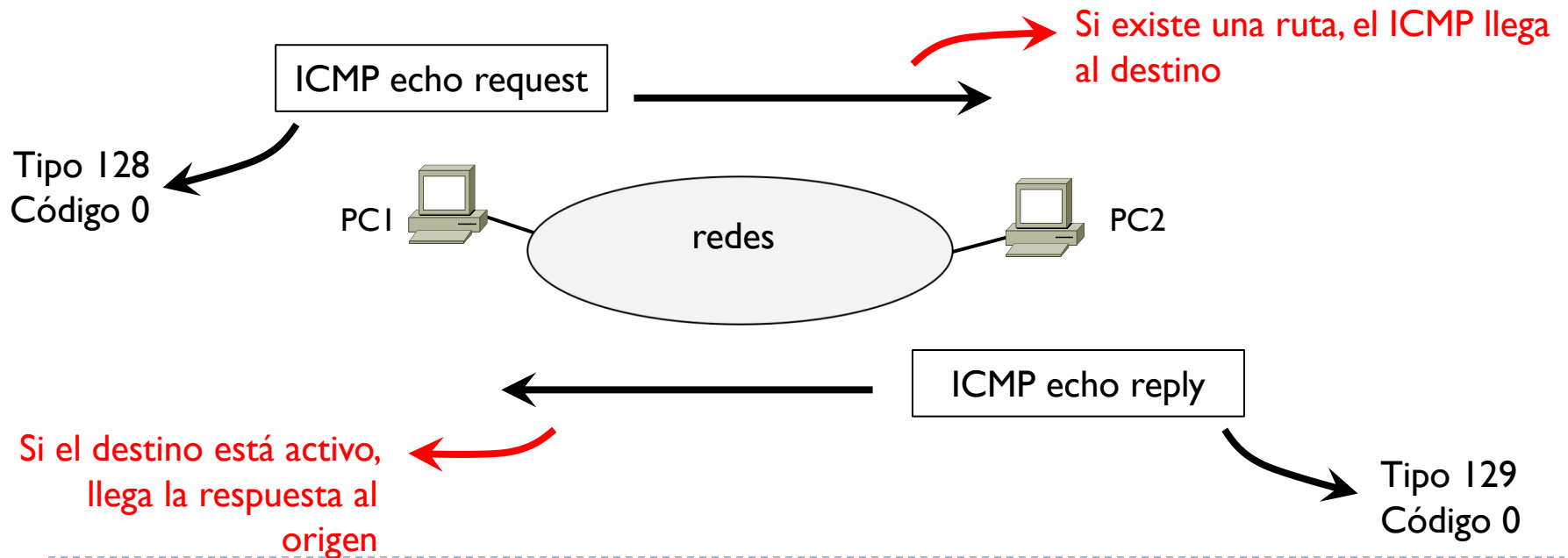
ICMPv6

- ▶ ICMPv4 + ARP + nuevas funciones



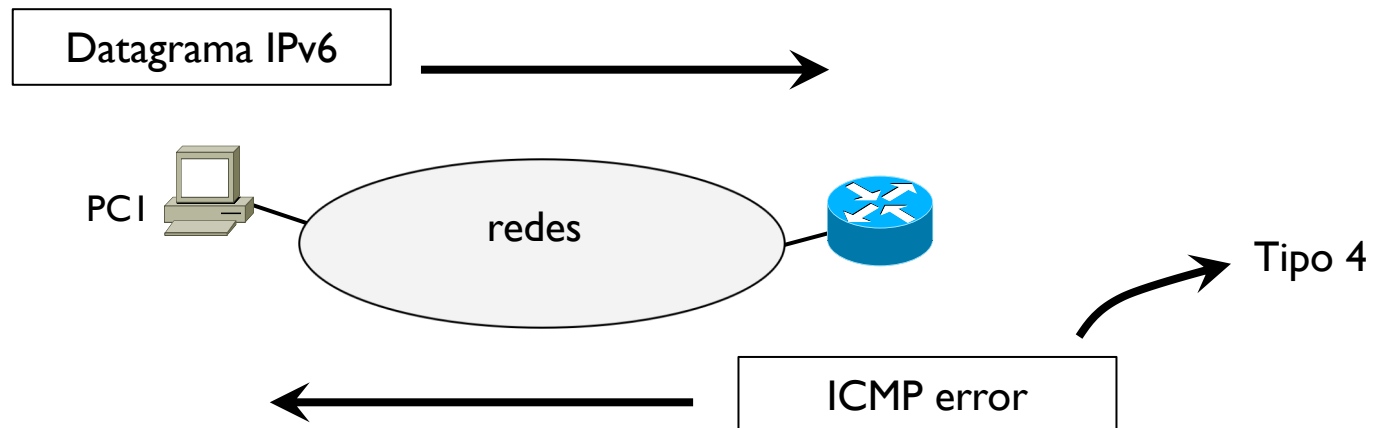
ICMPv6: ICMPv4

- ▶ ICMPv4: protocolo de control de IPv4
 - ▶ Envío de mensajes de supervisión (echo request/reply) y de error
- ▶ ICMPv6: supervisión



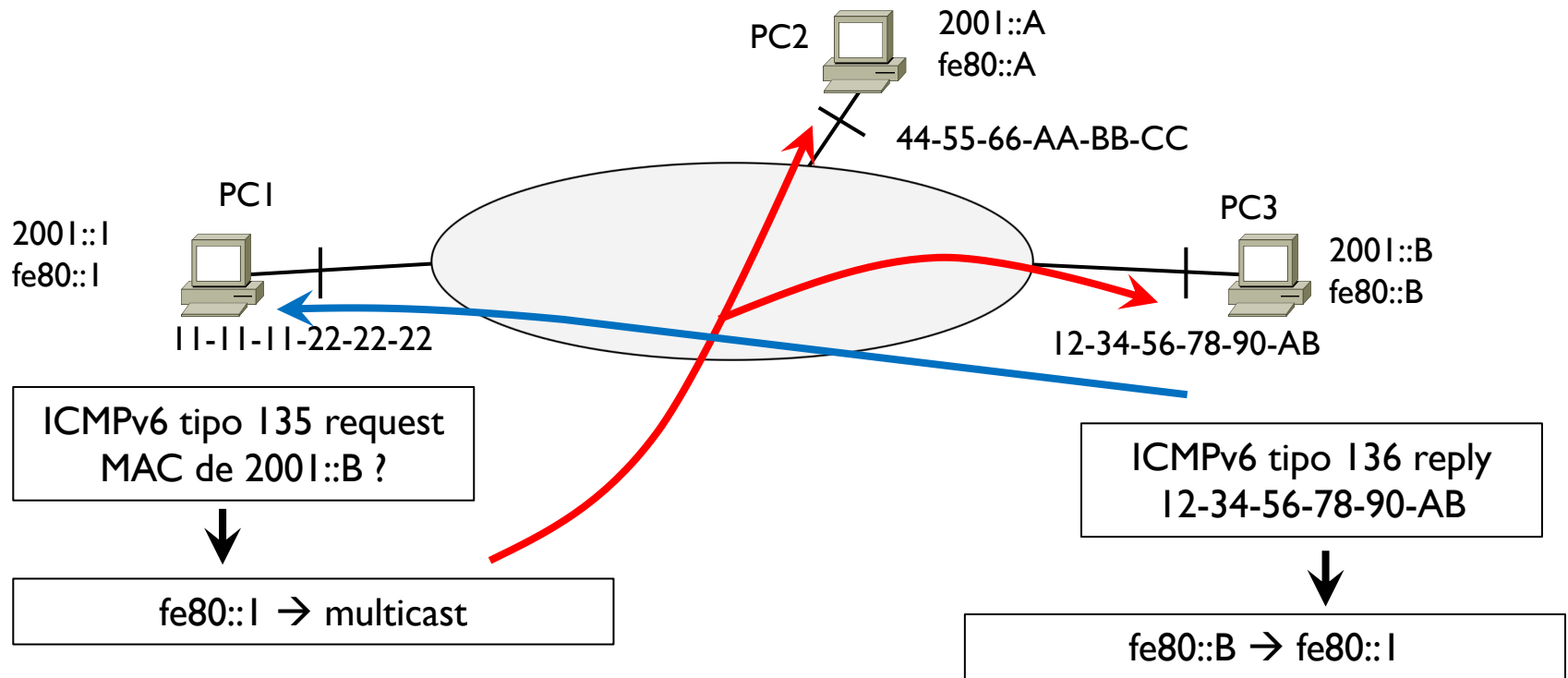
ICMPv6: ICMPv4

- ▶ ICMPv4: protocolo de control de IPv4
 - ▶ Envío de mensajes de supervisión (echo request/reply) y de error
- ▶ ICMPv6: nuevos errores
 - ▶ Tipo 4: Problema con los parámetros
 - ▶ Código 1: campo Siguiente Cabecera no reconocido
 - ▶ Código 2: opción IPv6 no reconocida



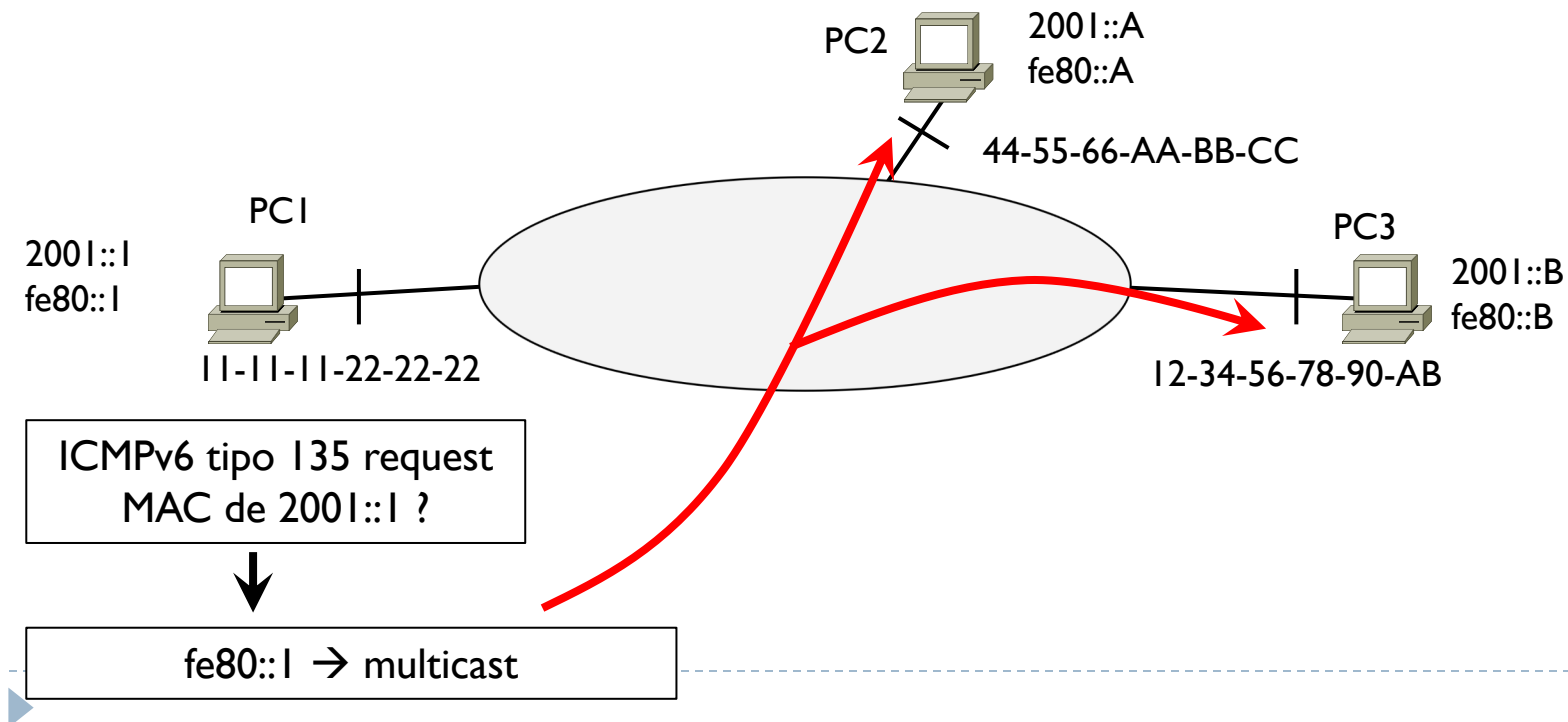
ICMPv6: ARP

- ▶ ARP: protocolo de resolución de dirección MAC
 - ▶ Conocida una @IP, se quiere descubrir su dirección MAC



ICMPv6: ARP

- ▶ Incluidos los ARP gratuitos ahora llamado DAD
- ▶ En un ARP gratuito
 - ▶ Se envía un ARP request indicando que se busca la MAC de la @IP del que envía el mensaje



ICMPv6: ARP

- ▶ Incluidos los ARP gratuitos ahora llamado DAD
- ▶ En un ARP gratuito
 - ▶ Se envía un ARP request indicando que se busca la MAC de la @IP del que envía el mensaje
 - ▶ Si alguien contesta, es que este tiene la misma @IP del que ha enviado el request
→ @IP duplicada
- ▶ DAD: Duplicate Address Discovery



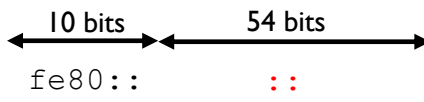
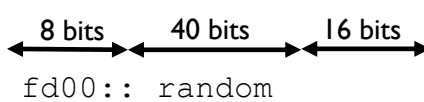
Configuración stateless

► Proceso

- 1) Se genera una @IPv6
- 2) Se verifica que la @IPv6 es única en la red (se envía un DAD)
- 3) Si es única se asigna a la interfaz
Si no es única se vuelve al punto 1)



¿Qué es lo que hay que asignar?

	Routing-prefix 64 bits	interfaceID 64 bits
Link-local	 fe80:: ::	
ULA	 fd00:: random	
Global		

Como el alcance está limitado a la misma red y si no hay alguna razón específica para que sea diferente, estos 54 bits suelen ser todos 0



Routing-prefix Global + ULA

- ▶ Al arrancar un host tiene la @IPv6 ::/128
- ▶ Envía un ICMPv6 (nueva función) de tipo Router Solicitation (RS, tipo 133) en multicast en su red pidiendo la configuración global y ULA
- ▶ Un router envía periódicamente un ICMP de tipo Router Advertisement (RA, tipo 134) o como respuesta a un ICMP RS en multicast en la red informando de la configuración del routing-prefix Global y si se usa y cual es el routing-prefix ULA



¿Qué es lo que hay que asignar?

	Routing-prefix 64 bits	interfaceID 64 bits
Link-local	<div>10 bits54 bits</div> <div>fe80:: ::</div>	
ULA	<div>8 bits40 bits16 bits</div> <div>fd00:: ICMP RA</div>	
Global	ICMP RA	

Como el alcance está limitado a la misma red y si no hay alguna razón específica para que sea diferente, estos 54 bits suelen ser todos 0

Dos métodos para InterfaceID

- ▶ Usando la MAC
- ▶ Aleatorio

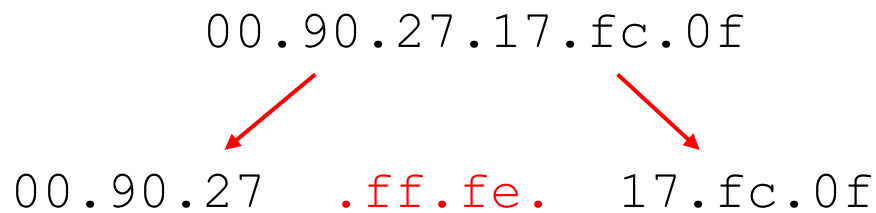


Usando la MAC

- ▶ La dirección MAC en principio es un número de 48 bits único asignado a cada interfaz de hosts y routers
- ▶ Se transforma este número en otro de 64 bits
- ▶ Proceso

00.90.27.17.fc.0f

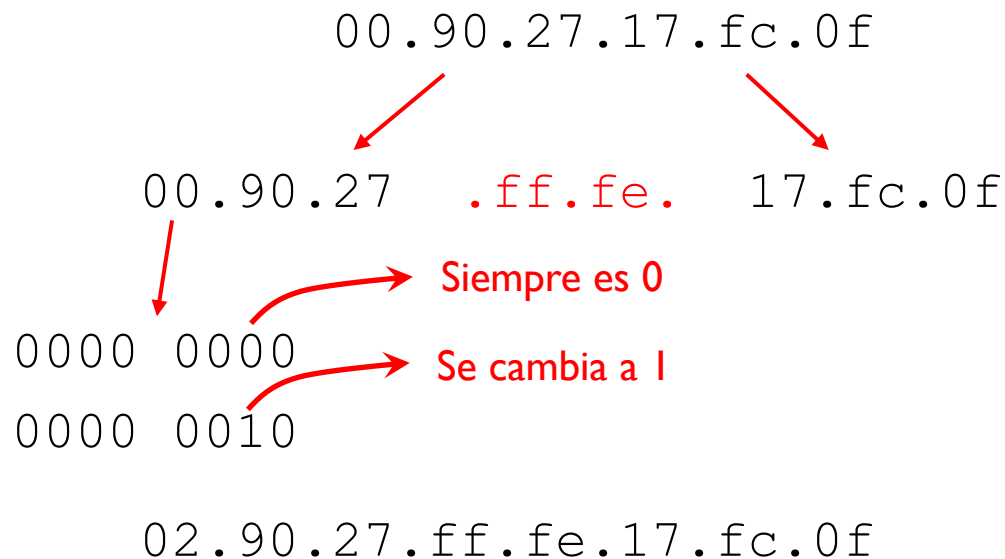
00.90.27 .ff.fe. 17.fc.0f



Se añaden estos 16 bits en el medio para llegar a 64 bits

Usando la MAC

- ▶ La dirección MAC en principio es un número de 48 bits único asignado a cada interfaz de hosts y routers
- ▶ Se transforma este número en otro de 64 bits
- ▶ Proceso



→ En notación IPv6

`290:27ff:fe17:fc0f`

Usando la MAC

► Problemas

1) Los fabricantes han admitido vender tarjetas con la misma MAC

De manera que con este proceso, dos tarjetas lleguen a tener la misma @IPv6

Se ejecuta DAD para descubrir si está duplicada

2) ?



Usando la MAC

► Problemas

1) Los fabricantes han admitido vender tarjetas con la misma MAC

De manera que con este proceso, dos tarjetas lleguen a tener la misma @IPv6

Se ejecuta DAD para descubrir si está duplicada

2) Privacidad?



Usando la MAC

► Problemas

1) Los fabricantes han admitido vender tarjetas con la misma MAC

De manera que con este proceso, dos tarjetas lleguen a tener la misma @IPv6

Se ejecuta DAD para descubrir si está duplicada

2) Como la InterfaceID ahora sería siempre la misma independiente del lugar donde se conecta un host móvil (solo cambiaría el routing-prefix), ahora sería posible conocer su localización y sus movimientos



InterfaceID aleatoria

- ▶ Se puede activar la opción de generar un interfaceID aleatorio
- ▶ Luego DAD descubre si es única



Método usado generalmente

- ▶ MAC como InterfacID para conexiones a redes internas (por razones de seguridad y permisos a sistemas y servicios internos)
- ▶ InterfacID aleatoria para conexiones a redes externas



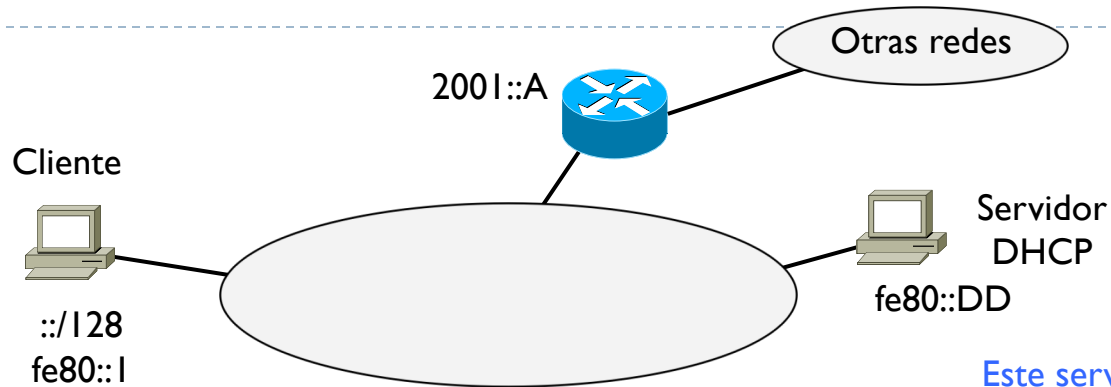
Configuración stateful

- ▶ Manual
 - ▶ Se asigna de forma manual y se usa DAD para reconocer @IPv6 duplicadas
- ▶ DHCPv6



DHCPv6

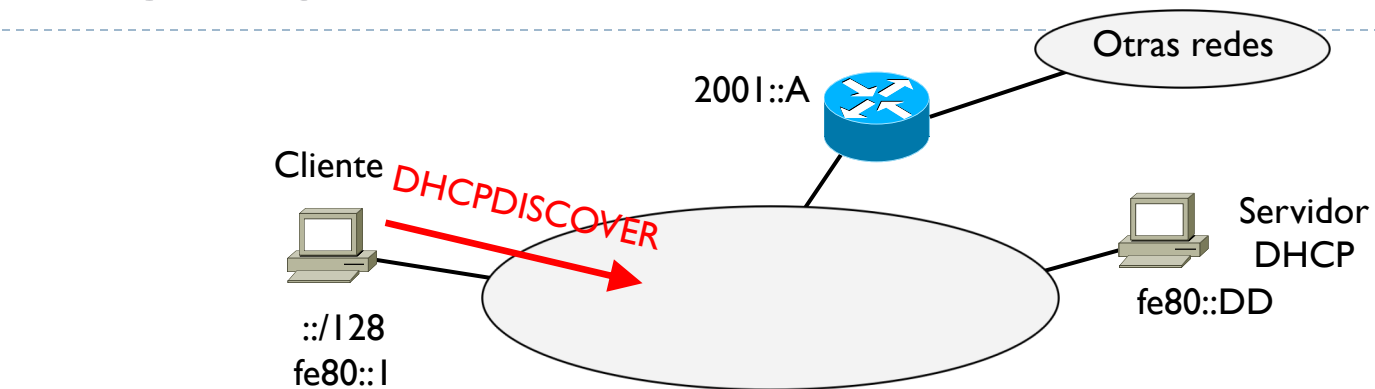
Este host al principio
no tienen ninguna @IP
global
Pero puede tener una
link-local



Este servidor tiene el socket
abierto en escuche para
proporcionar servicio DHCP,
puerto 547



DHCPv6

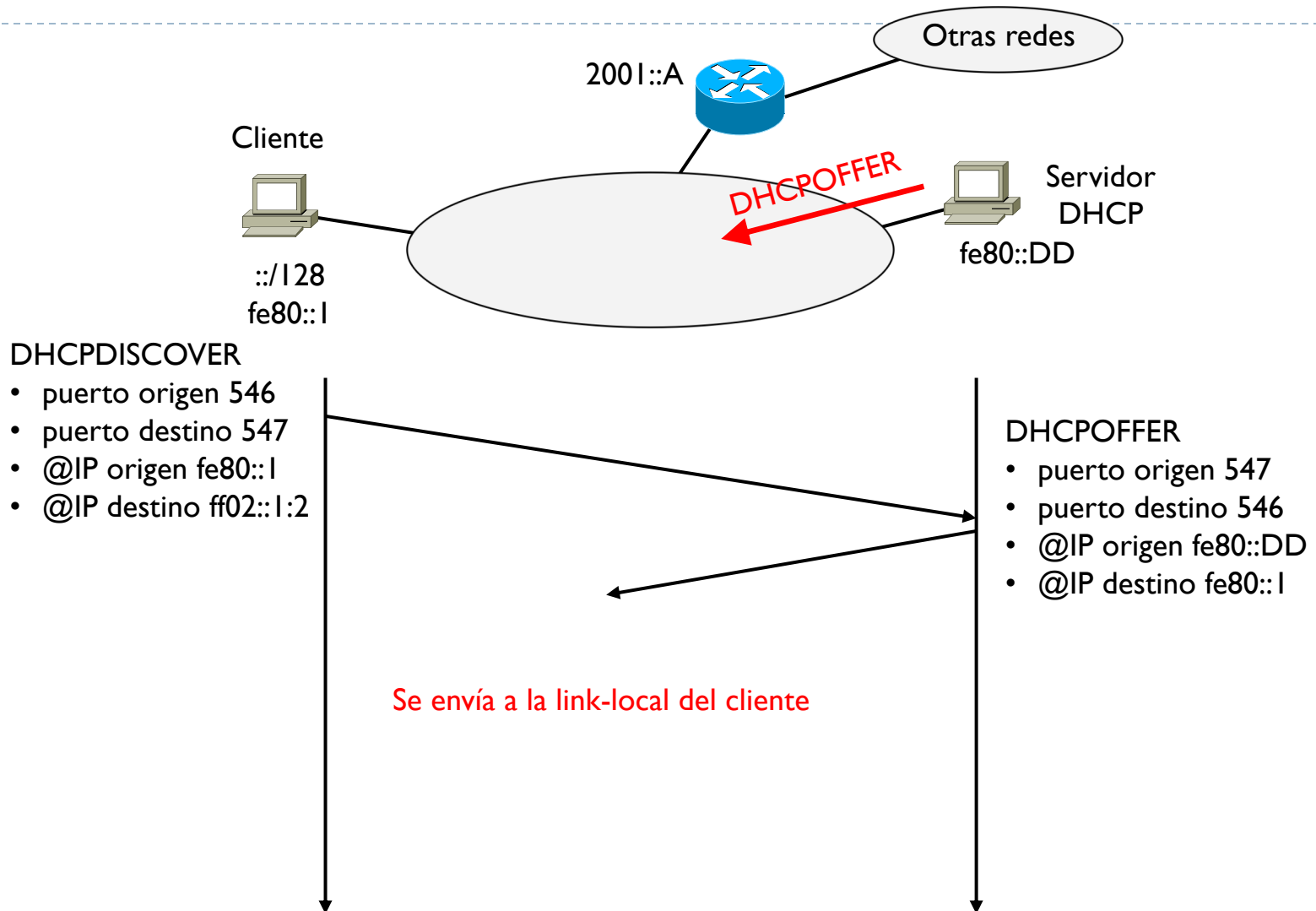


DHCPDISCOVER

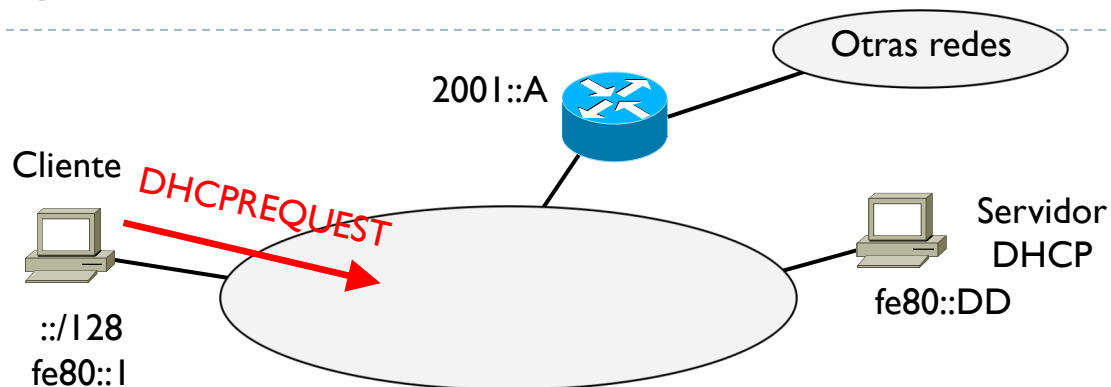
- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

Como no se conoce la @IP del servidor se envía un multicast específico para DHCPv6

DHCPv6



DHCPv6



DHCPDISCOVER

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

DHCPOFFER

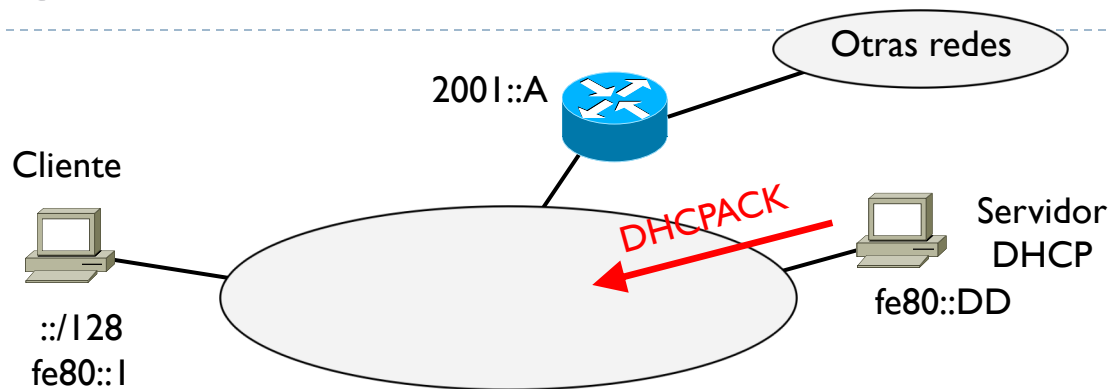
- puerto origen 547
- puerto destino 546
- @IP origen fe80::DD
- @IP destino fe80::1

DHCPREQUEST

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

Se envía en multicast para informar todos los posibles servidores DHCPv6

DHCPv6



DHCPDISCOVER

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

DHCPREQUEST

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

DHCPOFFER

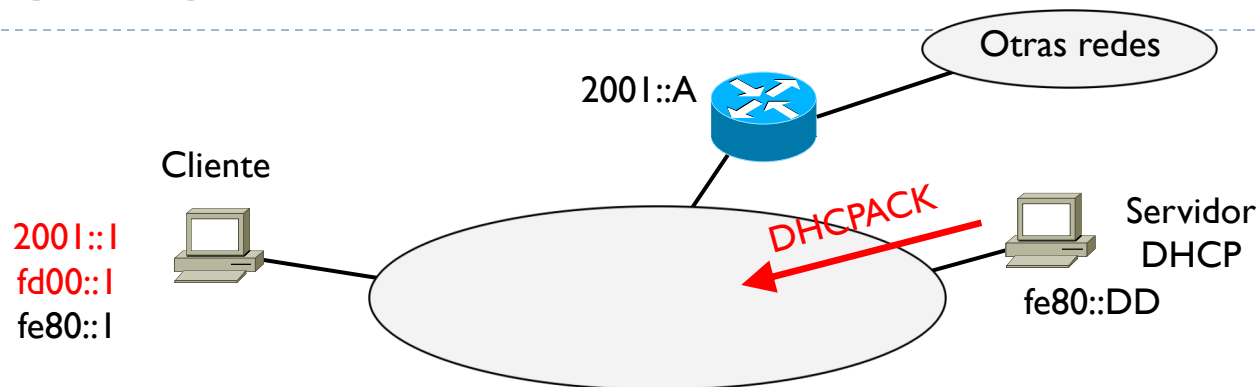
- puerto origen 547
- puerto destino 546
- @IP origen fe80::DD
- @IP destino fe80::1

DHCPACK

- puerto origen 547
- puerto destino 546
- @IP origen fe80::DD
- @IP destino fe80::1

Confirmación final que el cliente
se puede configurar según lo
ofrecido con DHCPOFFER

DHCPv6



DHCPDISCOVER

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

DHCPREQUEST

- puerto origen 546
- puerto destino 547
- @IP origen fe80::1
- @IP destino ff02::1:2

El cliente se configura

DHCPOFFER

- puerto origen 547
- puerto destino 546
- @IP origen fe80::DD
- @IP destino fe80::1

DHCPACK

- puerto origen 547
- puerto destino 546
- @IP origen fe80::DD
- @IP destino fe80::1

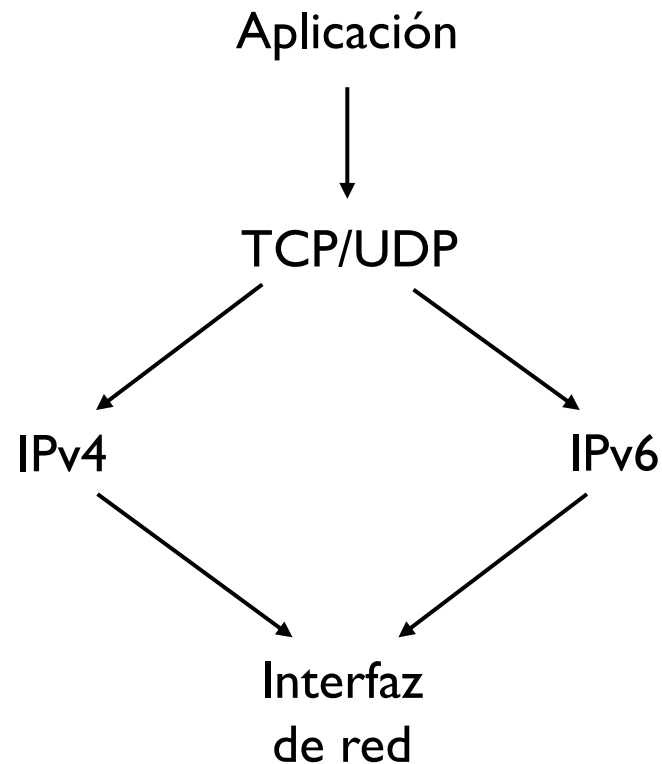
Compatibilidad IPv4 – IPv6

- ▶ **Diferentes métodos según el escenario**
 - ▶ Dual stack
 - ▶ IPv4-mapped
 - ▶ Tunneling
 - ▶ NAT64
 - ▶ etc.



Dual stack

- ▶ Generalmente un host funciona con dual stack y se adapta según la red y el destino



IPv4-mapped

- ▶ Si una aplicación solo funciona con @IPv6, se usa IPv4-mapped
- ▶ Rango reservado `::ffff:0:0/96`

La @IPv4 se mapea en los últimos 32 bits del
`::ffff:0:0/96`

La aplicación de esta forma trabaja con @IPv6

`::ffff:147.83.0.1`

IPv4-mapped

IPv4

`147.83.0.1`

Aplicación IPv6

TCP/UDP

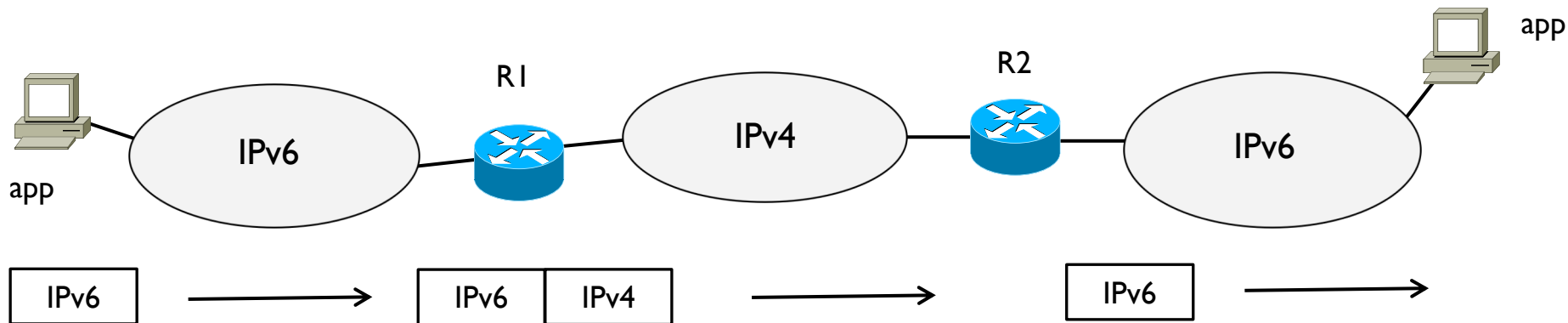
IPv6

Interfaz
de red



Tunneling

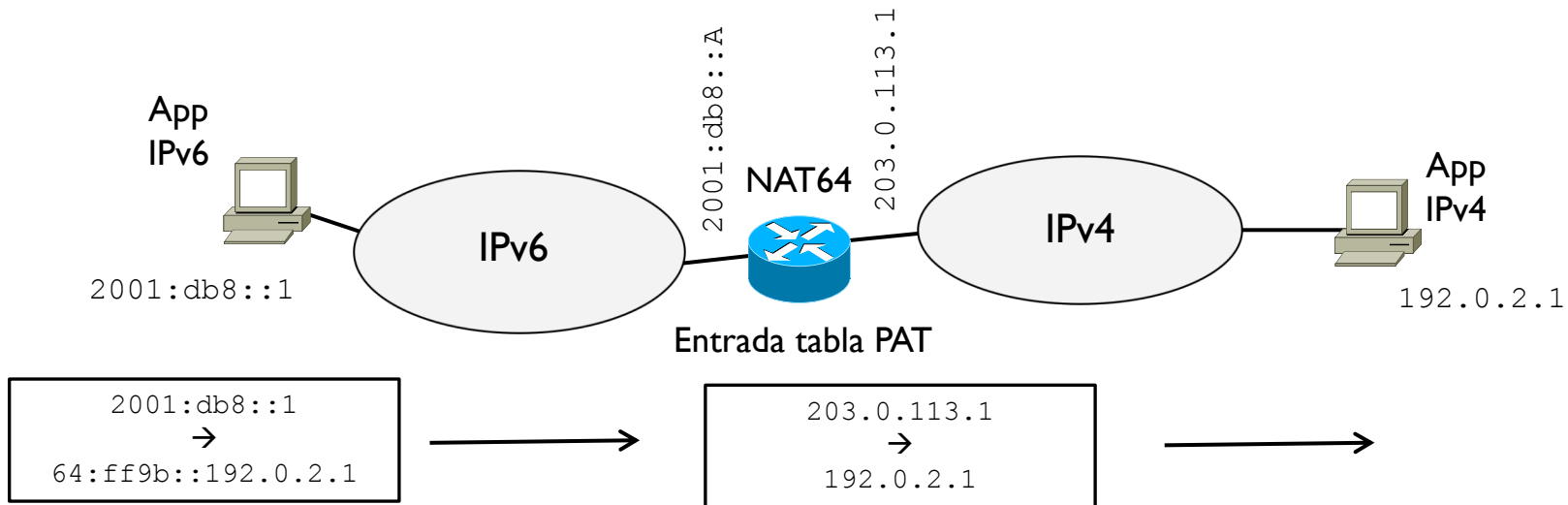
- ▶ Cuando dos aplicaciones pueden usar IPv6 pero en el camino hay IPv4



- ▶ 6to4 tunnel → creación del túnel automático con 2002::/16 RFC 3056
- ▶ Teredo tunnel → si hay un NAT IPv4 RFC 4380
- ▶ Tunnel bróker → configuración a través de un server (bróker) RFC 3053

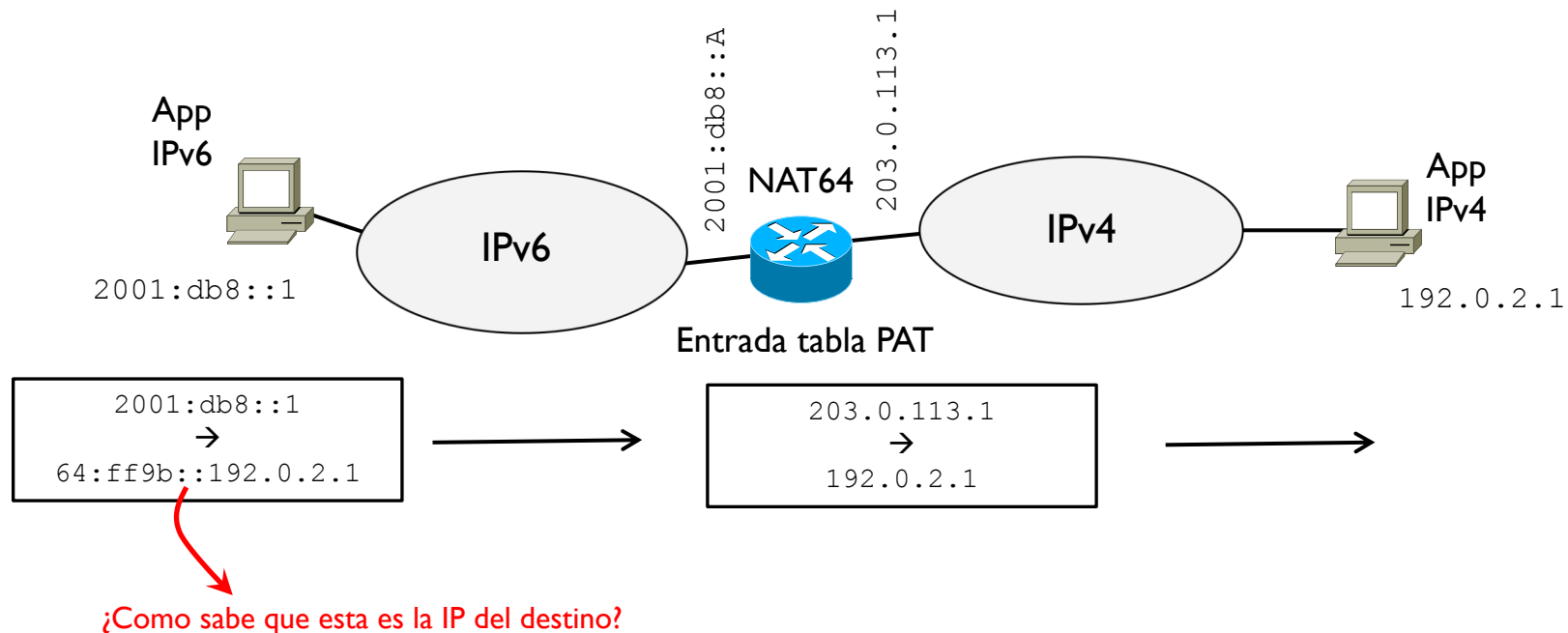
NAT64

- ▶ Cuando una aplicación IPv6 comunica con una IPv4
- ▶ Rango reservado: `64:ff9b::/96`



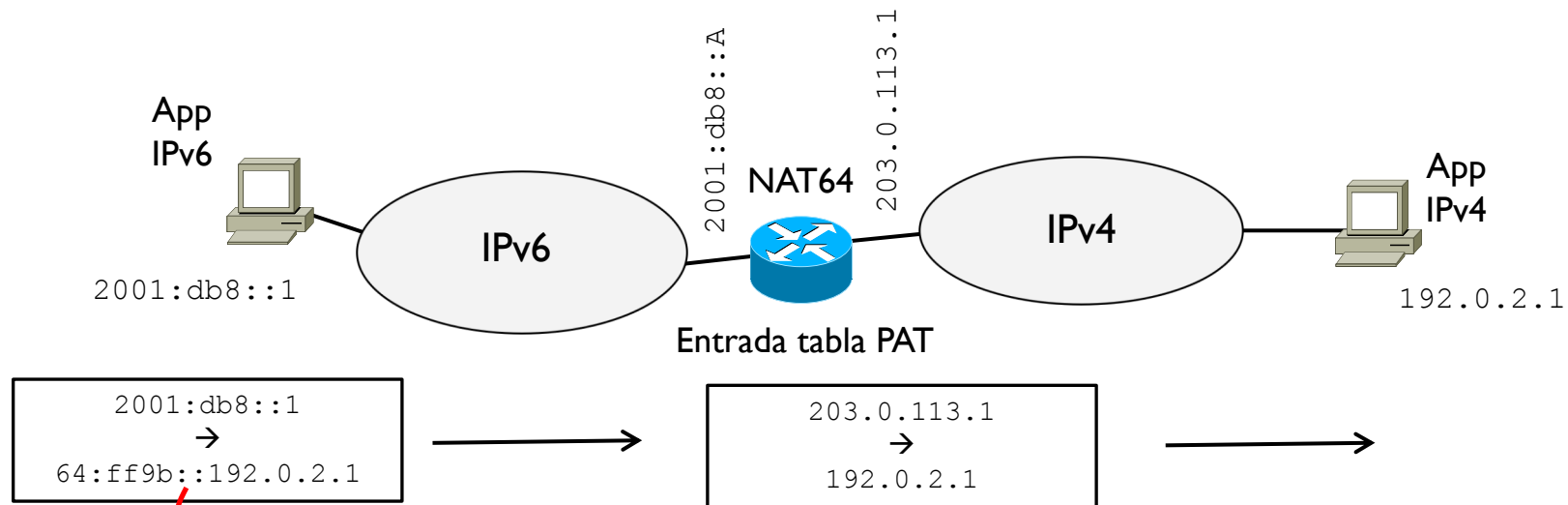
NAT64

- ▶ Cuando una aplicación IPv6 comunica con una IPv4
- ▶ Rango reservado: `64:ff9b::/96`



NAT64

- ▶ Cuando una aplicación IPv6 comunica con una IPv4
- ▶ Rango reservado: `64:ff9b::/96`



¿Como sabe que esta es la IP del destino?
Se lo dice el DNS!

Otros cambios

- ▶ RIP → RIPng RFC 2080
- ▶ OSPF → OSPFv6 RFC 5340
- ▶ BGP → BGP4+ RFC 2545

- ▶ DNS → nuevos RR para asociar IPv6 y nombres



Xarxes de computadors II