

Diseño y administración de redes

Área de Ingeniería Telemática

Grado en Ingeniería Informática
Cuarto curso. Primer Cuatrimestre.



**Departamento de
Ingeniería Electrónica
y Comunicaciones**
Universidad Zaragoza

Bloque 2. Interconexión de redes IP

IPv6

Área de Ingeniería Telemática

Contenidos

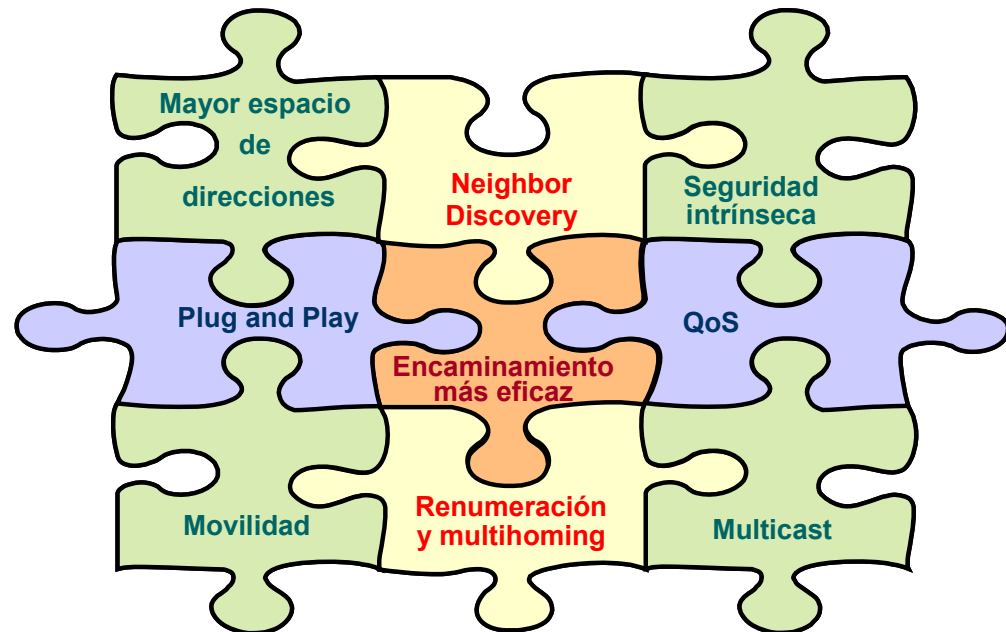
- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- Direccionamiento
- PDU. Cabeceras de extensión
- Autoconfiguración
- Funciones de control
- Encaminamiento

Introducción a IPv6

- Desarrollado fundamentalmente para resolver el problema de escasez de direcciones de IPv4

- <http://www.ipv6forum.com/>
- <http://www.ipv6tf.org/>
- <http://playground.sun.com/pub/ipng/html/ipng-main.html>
- <http://technet.microsoft.com/en-us/network/bb530961.aspx>
- <http://tldp.org/HOWTO/Linux+IPv6-HOWTO/index.html>
- <http://www.rediris.es/conectividad/direccionamientoipv6.html>
- <http://gogonet.gogo6.com/>

- ¿Qué ofrece?:



Introducción a IPv6

- Principales novedades en IPv6
 - Modificación de la cabecera IP.
 - Opciones encadenadas: el campo de opciones se reemplaza por el de Siguierte cabecera; esto simplifica el proceso en cada *router* y da un mecanismo de extensión de opciones.
 - Eliminación de campos redundantes (*checksum*)
 - Se elimina la fragmentación de la red (y los campos en IP)
 - Mínimo MTU es 1280 bytes (28 en IPv4)
 - Posibilidad de envíos *unicast*, *multicast* y *anycast*
 - Autoconfiguración (sin servidor, “*plug-n-play*”) y movilidad
 - Flexible y extensible
 - Seguro

Contenidos

- Introducción a IPv6
- **Coexistencia / Transición IPv4-IPv6**
- Direccionamiento
- PDU. Cabeceras de extensión
- Autoconfiguración
- Funciones de control
- Encaminamiento

Coexistencia / Transición IPv4 – IPv6

- **Dual stack:**

<http://technet.microsoft.com/en-us/library/bb726951.aspx>

- *Dual IP stack*

- Todos los elementos de red funcionan con las pilas IPv4 y IPv6

- *Dual IP layer*

- Capa TCP/UDP compartida

- *DNS resolver* devuelve IPv6, IPv4 o ambas a la aplicación

- **Túneles** (unir “islas”)

- Túneles manuales (configurados)

- Túneles automáticos

- Túneles “6 to 4”
 - Túneles “6 over 4”
 - Teredo

Encapsulado: algún
nodo *dual stack*

- **Traducción** (cambio de protocolo)

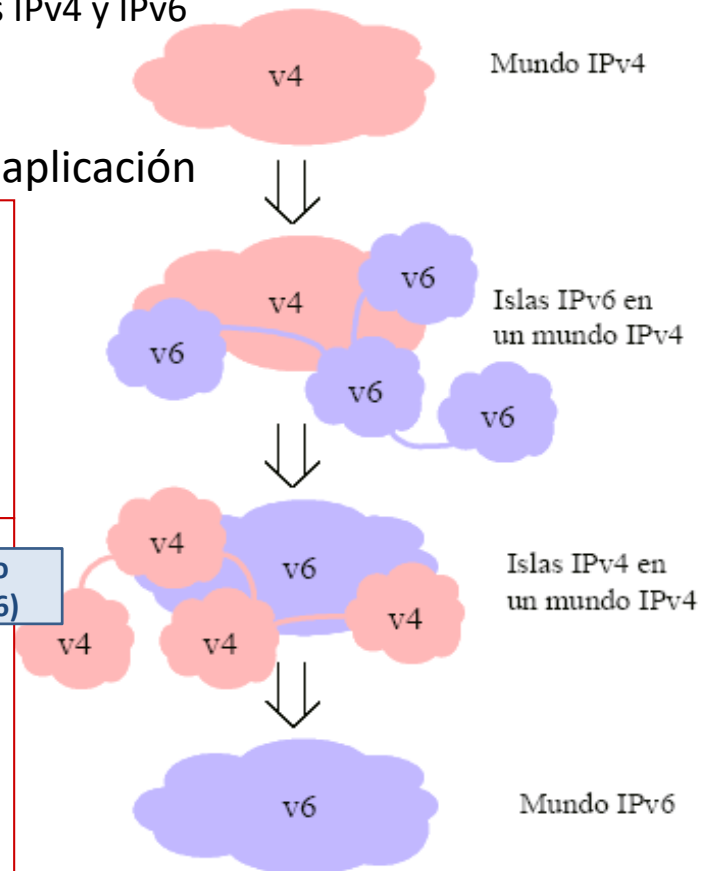
- *NAT-Protocol Translation* (NAT-PT)

- Cambia dirección IP (= NAT IPv4)
 - Cambia la cabecera IP completa (datos intactos)

- *Bump in the stack* (BIS), *Bump in the API* (BIA)

- Traducción IPv4/IPv6 en un nodo que ejecuta una aplicación IPv4 (interna a nivel de *host*)

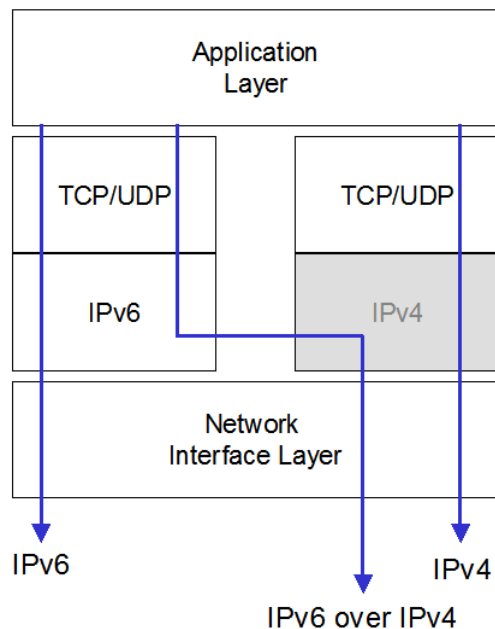
Inevitable comunicación *IPv4-only – IPv6-only*



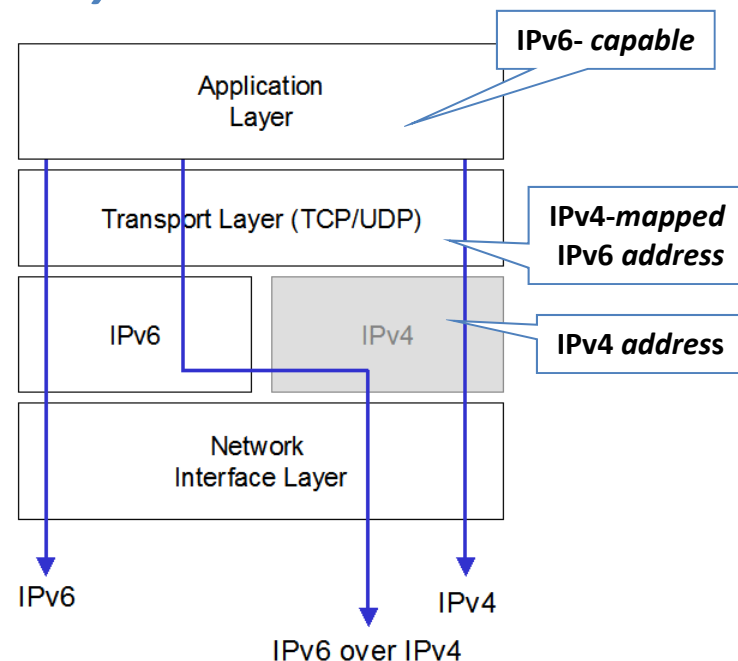
Obsoleto
(RFC 4966)

Coexistencia / Transición IPv4 – IPv6

Dual stack architecture



Dual IP layer architecture



RFC 4038

La mayoría de las implementaciones *dual stack* permiten la interoperatividad de aplicaciones IPv6-only con nodos IPv4 e IPv6

Utilización de direcciones IPv4-mapeadas

IPv6 \Rightarrow **::FFFF:x.y.z.w**

IPv4 \Rightarrow **x.y.z.w.**

Coexistencia / Transición IPv4 – IPv6

- **Túneles manuales (configurados)**
 - Acceso a un ISP IPv6 (6-bone)
 - Configuración “sencilla” (herramientas de ayuda, como *Tunnel Broker*) **RFC 3053**
 - Transparentes para las aplicaciones
- **Túneles automáticos**
 - Permiten a nodos duales comunicarse a través de una isla IPv4 mediante interfaz virtual (crear dirección IPv6 a partir de prefijo conocido y dirección IPv4 - **IPv4-compatible**)
 - **ISATAP** **RFC 5214**
 - Unir islas IPv6 a través de un *backbone* IPv4
 - Interfaz de red: dirección IPv4-compatible (con cualquier prefijo global)
 - Encaminamiento a través de *router* configurados ISATAP
 - **Túneles 6 to 4** **RFC 3056**
 - Unir islas IPv6 a través de un *backbone* IPv4
 - Encaminamiento: prefijo de red IPv6 identifica al *router* IPv4 frontera (@ en la cabecera IPv6)
 - Encaminamiento en IPv4 apoyándose en el encaminamiento de esa red
 - **Túneles 6o4** **RFC 2529**
 - Nodos IPv6 dispersos en redes IPv4 forman una “LAN virtual” IPv4
 - El tráfico se encapsula en IPv4: uso de *multicast* (*unicast* y *multicast* IPv6)
 - Los *router* con *dual-stack* (IPv6) permiten el acceso al 6-bone
 - **Teredo** **RFC 4380, 5991**
 - Túnel que permite la interconexión entre equipos detrás de NAT/Firewall
 - Túnel IPv6 over UDP over IPv4

Coexistencia / Transición IPv4 – IPv6

Ejemplo IPv6 over IPv4

Túnel manual

Mediante herramienta para facilitar la configuración

Tunnel Broker

1.- Petición web IPv4

2.- Respuesta del servidor:
Info para configurar el túnel

Tunnel Broker
(Proveedor)

3.- El servidor configura en el router
el extremo del túnel

Red IPv6

Tunnel server
Dual Stack (IPv6/IPv4)

Red IPv4

Cabecera IPv6	DATOS
---------------	-------

Cabecera IPv4	Cabecera IPv6	DATOS
---------------	---------------	-------

4.- El cliente configura su extremo,
puede empezar la comunicación IPv6



Cliente

Conectado a Internet IPv4
Dual Stack (IPv6/IPv4)

Contenidos

- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- **Direccionamiento**
- PDU. Cabeceras de extensión
- Autoconfiguración
- Funciones de control
- Encaminamiento

Direccionamiento

- Las direcciones IPv6 son identificadores de **128 bits de longitud**.
 - $2^{128} = 3,40 \times 10^{38}$ direcciones posibles
(340.282.366.920.938.463.463.374.607.431.768.211.456)
 - 665.570.793.348.866.943.898.599 IP's por m² de la superficie terrestre.
- Representación **X:X:X:X:X:X:X**, cada X representa el valor hexadecimal de un grupo de 16 bits. Ejemplo:
 - 3FFE:3328:6:0:0:0:0:7890/64
 - 3FFE:3328:6::7890/64
 - 2001:410:0::/48 (red con 2¹⁶ subredes)
 - 2001:410:0:1::/64 (red con 2⁶⁴ hosts)
 - 2001:410:0:1::0:45FF/128 (dirección de un host)
- No hay direcciones reservadas para red y *broadcast*.

Notación simplificada siempre que no exista ambigüedad

Prefijo de red (notación CIDR)
Marca el número de bits destinados a definir una red

Direccionamiento

Teniendo en cuenta que:

64 es el número de bits que identifican una red o subred y 64 es el número de bits que identifican un dispositivo dentro de una red o subred.

Varios dispositivos por persona.

Si la población mundial es: 6.302.309.691

Y las direcciones IPv6 para posibles dispositivos son:

$$2^{64} = 18.446.744.073.709.600.000$$

Las posibles direcciones IPv6 por persona serían: 2.926.981.532

Mientras el total de direcciones IPv4 son: $2^{32} = 4.294.967.296$

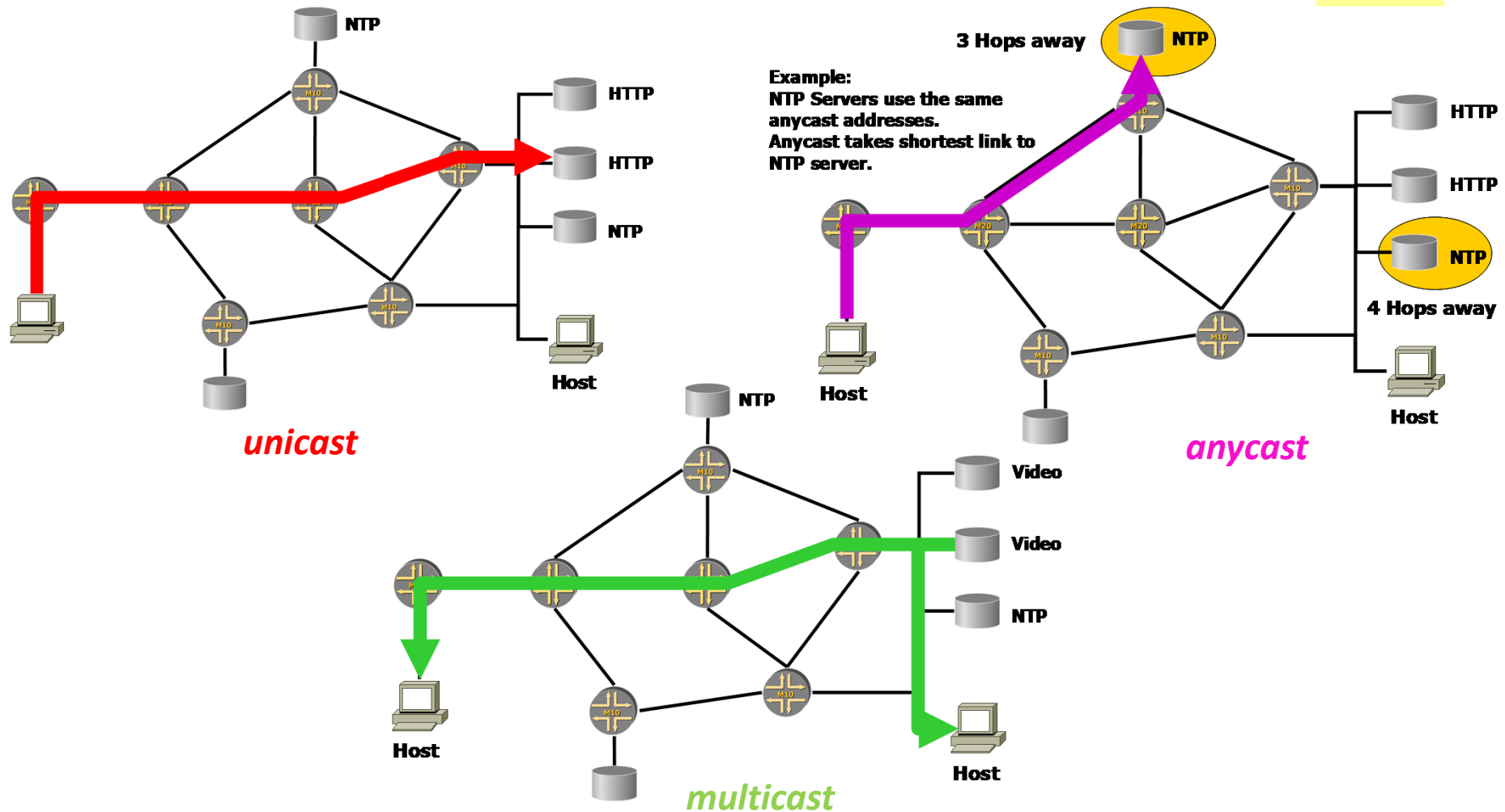
Direccionamiento

RFC 4291

- Clases de direcciones IPv6
 - **Unicast**: Identificador para **una única interfaz**. Un paquete enviado a una dirección *unicast* es entregado sólo a la interfaz identificada con dicha dirección.
 - **Anycast**: Identificador para un **conjunto de interfaces** (típicamente pertenecen a diferentes nodos). Un paquete enviado a una dirección *anycast* es **entregado a una** (cualquiera) de las interfaces identificadas con dicha dirección (**la más próxima**, de acuerdo a las medidas de distancia del protocolo de encaminamiento).
 - **Multicast**: Identificador para un **conjunto de interfaces** (por lo general pertenecientes a diferentes nodos). Un paquete enviado a una dirección *multicast* es **entregado a todas** las interfaces identificadas por dicha dirección.
 - **Broadcast** no es más que un caso particular de *multicast*

Direccionamiento

RFC 4291



Direccionamiento

<http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml>

• Asignación de direcciones IPv6

RFC 4291

48 bits		16 bits		64 bits	
Prefijo global		ID de subred		ID de interfaz	
001	Prefijo global	ID de subred		ID de interfaz	
00	...	00	...	00	::/128
00	...	00	...	01	::1/128

Global
(actualmente, IANA)

sin especificar
loopback

00	00	a.a.a.a	::a.a.a.a
	11	11	a.a.a.a	::FFFF:a.a.a.a

IPv6 over IPv4

IPv4 mapeada IPv6
(no enrutable)

Obsoleto – RFC4291

10 bits		64 bits	
111111010	00 ... 00	ID de interfaz	
		FE80::/10	

local de enlace
link-local

unicast

10 bits	16 bits	64 bits
111111011	00 ... 00	ID de subred ID de interfaz
		FEC0::/10

local de sitio
site-local

Obsoleto - RFC 3879

7 bits
1111110	00	...	00	...
1111110 0	00	...	00	...
1111110 1	00	...	00	...
8 bits	40 bits (generados aleatoriamente)	16 bits	64 bits	
		ID de subred	ID de interfaz	
				FC00::/7 FC00::/8 FD00::/8

unique-local
dividido en:
(aún no definido)
(prefijo aleatorio)

Alternativa - RFC 4193

n bits	121 - n bits	7 bits
Prefijo subred	111 ... 111	ID anycast

anycast
(caso particular de unicast)

*Reserved Anycast
Address* **RFC 2526**

8 bits	4 bits	4 bits	
11111111	ORPT	Ámbito	Grupo
			FF00::/8

multicast

Direccionamiento

Direcciones IPv4		Direcciones IPv6
224.0.0.0/24	<i>Multicast</i>	FF00::/8
0.0.0.0	Sin especificar	::
127.0.0.0	<i>loopback</i>	::1
Class A, B, C ⊘ Todo 0's y todo 1's	IP pública	Global <i>unicast</i> agregables: 2000::/3 posible todo 0's y todo 1's
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	Privadas	locales de sitio: FEC0::/10 (obsoleto) locales de sitio: FC00::/7 (RFC 4193) locales de enlace: FE80::/10 Túneles IPv6 sobre IPv4, ::<dir.IPv4> (IPv4 compatible)-obsoleto Direcciones mapeadas desde IPv4, ::FFFF<dir.IPv4> NO ENCAMINABLE

2⁴⁵ posibles redes públicas: 35.184.372.088.832

2³⁸ posibles redes privadas: 274.877.906.944

Direccionamiento

- Direcciones IPv6 requeridas

RFC 4291

- Por un **HOST**

- Sus direcciones **locales de enlace** (privada) para cada interfaz
- Las direcciones **unicast/anycast** (pública o privada) asignadas
- La dirección **loopback**.
- Las direcciones **multicast de todos los nodos**
- Las direcciones **multicast** solicitadas para cada dirección *unicast* o *anycast* asignadas (**Solicited Node Address**)
- Las direcciones **multicast de todos los grupos** a los que dicho *host* pertenece

- Por un **ROUTER**, además...

- La dirección **anycast del router** de la subred(*), para las interfaces en las que está configurado para actuar como *router*
- Todas las direcciones **anycast** con las que el *router* ha sido **configurado**
- Las direcciones **multicast de todos los router**
- Las direcciones **multicast de todos los grupos** a los que el *router* pertenece

(*)subnet-router **anycast** address

n bits	128 – n bits
Prefijo de subred	0000

Dirección anycast del *router* de la subred:

Los paquetes enviados a esta dirección se entregarán a un *router* de la subred.

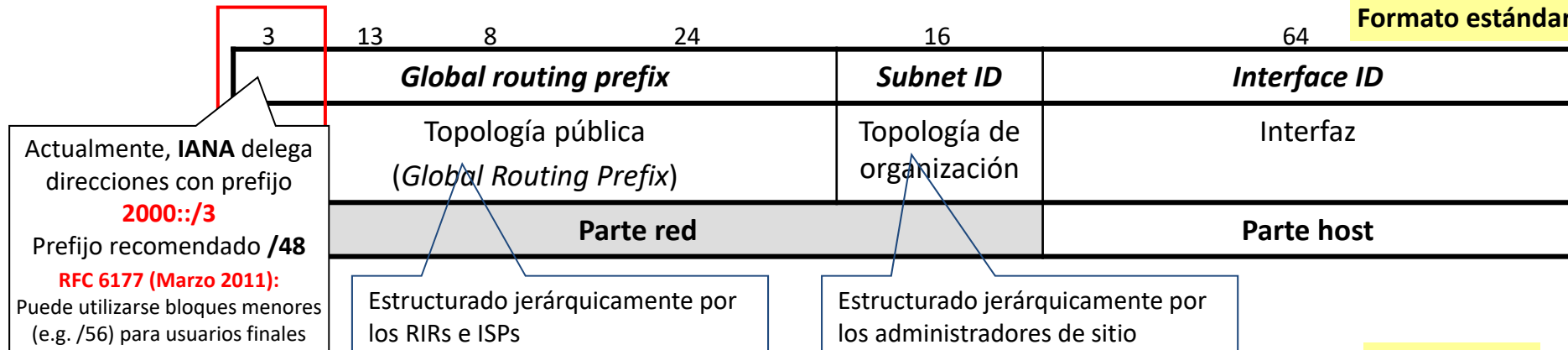
Direccionamiento

Filosofía
CIDR

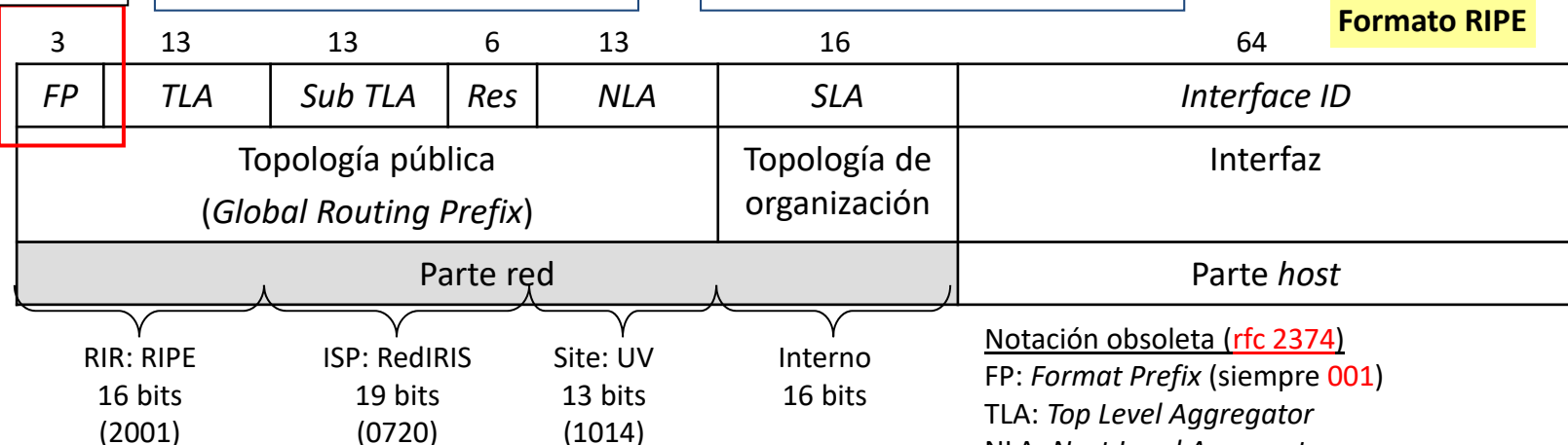
RFC 3587

- Formato de direcciones unicast **agregables** en IPv6

Formato estándar



Formato RIPE



Notación obsoleta (rfc 2374)
 FP: *Format Prefix* (siempre 001)
 TLA: *Top Level Aggregator*
 NLA: *Next Level Aggregator*
 SLA: *Site level Aggregator*

<http://www.rediris.es/conectividad/direccionamientoipv6.html>

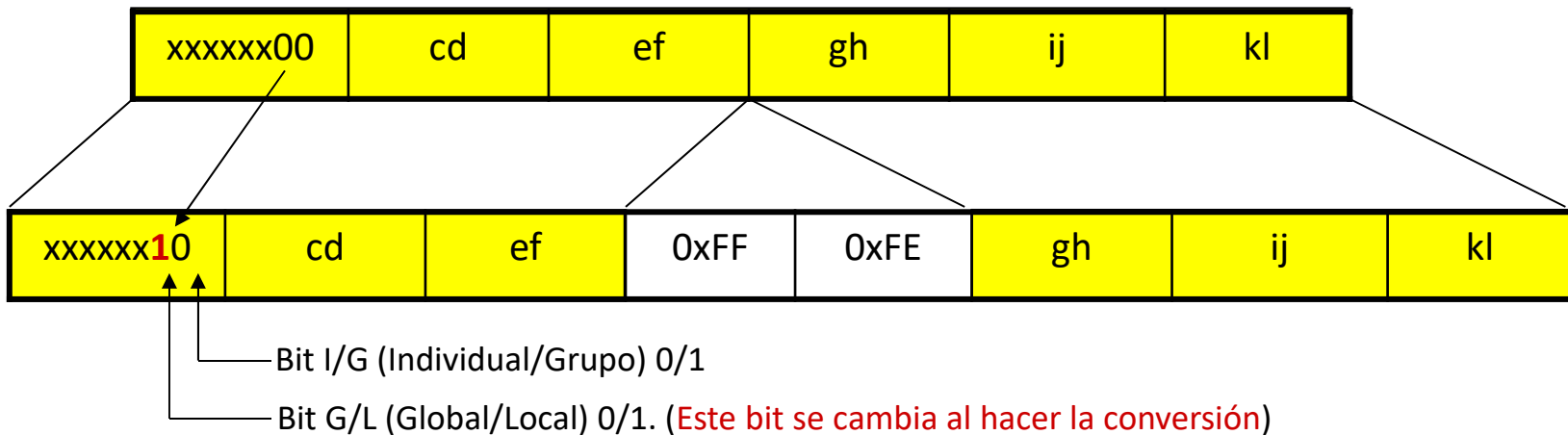
Direccionamiento: Autoconfiguración

- Autoconfiguración de dirección unicast/anycast en IPv6: Host construye su propia dirección a partir de dos partes:
 - la parte red (8 bytes) que le indica el router
 - la parte host (8 bytes)

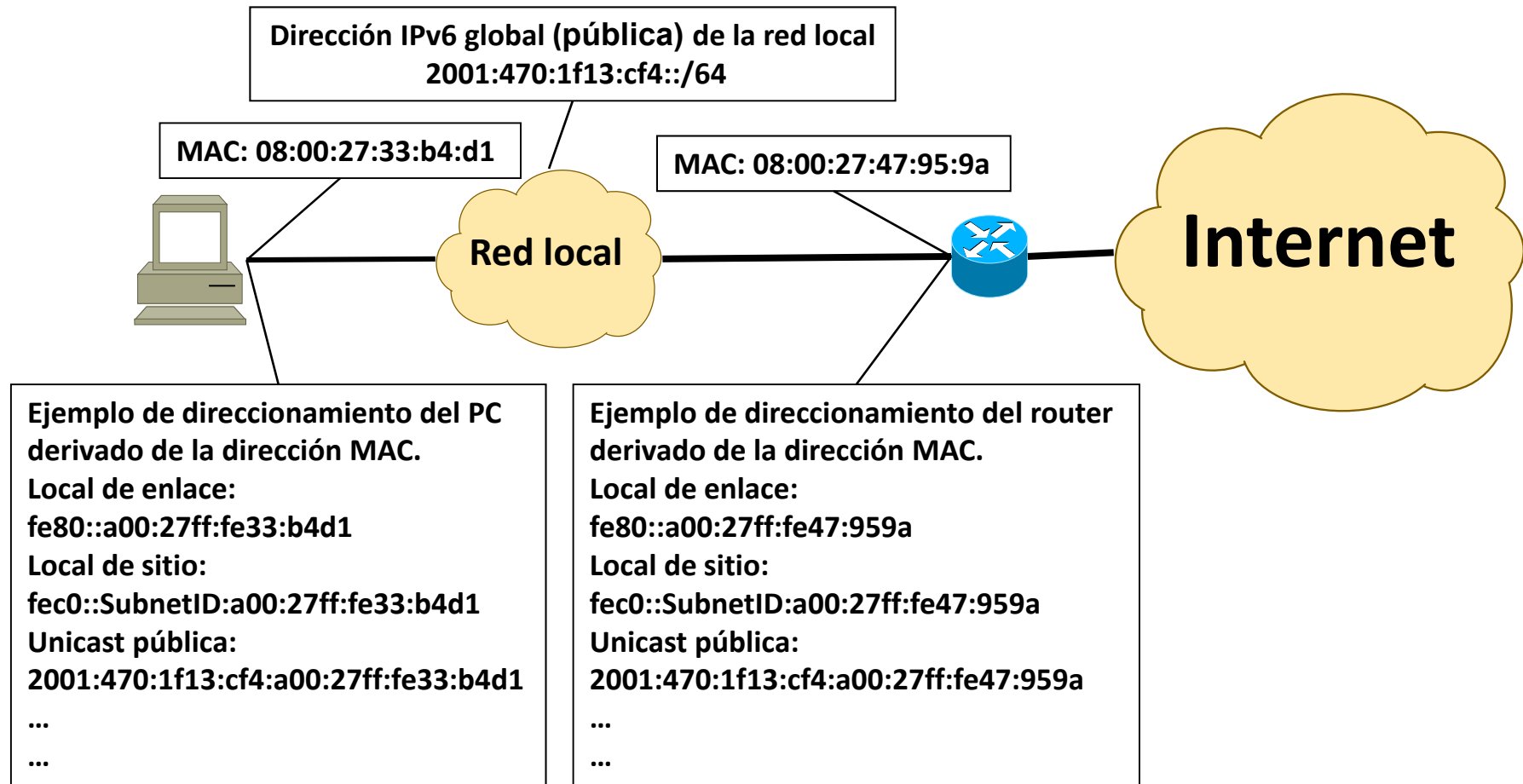
Por defecto,
lo habitual
(veremos)

- **Derivado de la dirección MAC** extendida: *Extended Unique Identifier (EUI)-64 address*. **RFC 4291**
- Número aleatorio cambiante en el tiempo (anónimo) **RFC 4941**
- Asignado durante la configuración *stateful* (ej. DHCPv6) **RFC 3315**
- Configuración manual
- Dirección criptográfica (*Cryptographically Generated Address (CGA)*) **RFC 3972**

dirección IEEE 802 → identificador EUI-64



Direccionamiento



RFC 4291



Direccionamiento

- Formato de direcciones multicast en IPv6

RFC 4291

<i>IPv6 Well-known multicast address</i>	<i>IPv4 Well-known multicast address</i>	<i>Multicast Group</i>
<i>Interface-local scope (loopback)</i>		
FF01:0:0:0:0:0:0:1	224.0.0.1	<i>All-nodes address</i>
FF01:0:0:0:0:0:0:2	224.0.0.2	<i>All-router address</i>
<i>Link-local scope</i>		
FF02:0:0:0:0:0:0:1	224.0.0.1	<i>All-nodes address</i>
FF02:0:0:0:0:0:0:2	224.0.0.2	<i>All-router address</i>
FF02:0:0:0:0:1:FFXX:XXXX	Solicited Node Address XX:XXXX – últimos 24 bits de la IP unicast/anycast origen (uso en mecanismo de detección de direcciones duplicadas: autoconfiguración)	
FF02:0:0:0:0:0:0:9	224.0.0.9	<i>RIP router</i>
FF02:0:0:0:0:0:0:D	224.0.0.13	<i>All PIM router</i>
<i>Site-local scope</i>		
FF05:0:0:0:0:0:0:2	224.0.0.2	<i>All-router address</i>
<i>Any valid scope</i>		
FF0X:0:0:0:0:0:0:101	224.0.1.1	<i>Network time protocol NTP</i>

Direccionamiento

- Formato de direccionamiento

IPv6 Well-known multicast address	
FF01:0:0:0:0:0:0:1	
FF01:0:0:0:0:0:0:2	
FF02:0:0:0:0:0:0:1	
FF02:0:0:0:0:0:0:2	
FF02:0:0:0:0:1:FFXX:XXXX	
Solicited Node Address XX:XXXX – últimos 24 bits de la IP unicast/anycast origen (uso en mecanismo de detección de direcciones duplicadas: autoconfiguración)	
FF02:0:0:0:0:0:0:9	224.0.0.9
FF02:0:0:0:0:0:0:D	224.0.0.13
Site-local scope	
FF05:0:0:0:0:0:0:2	224.0.0.2
Any valid scope	
FF0X:0:0:0:0:0:0:101	224.0.1.1
	Network time protocol NTP

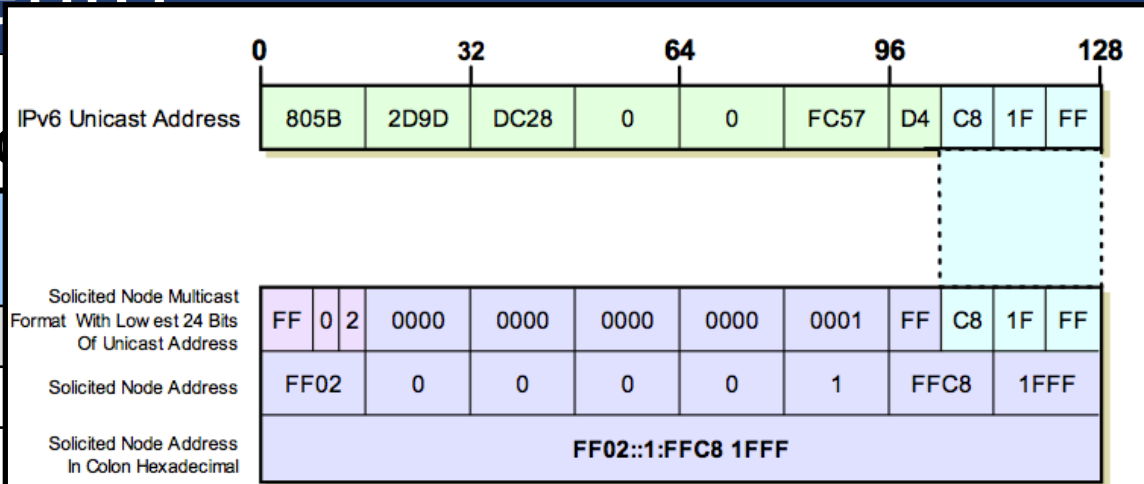
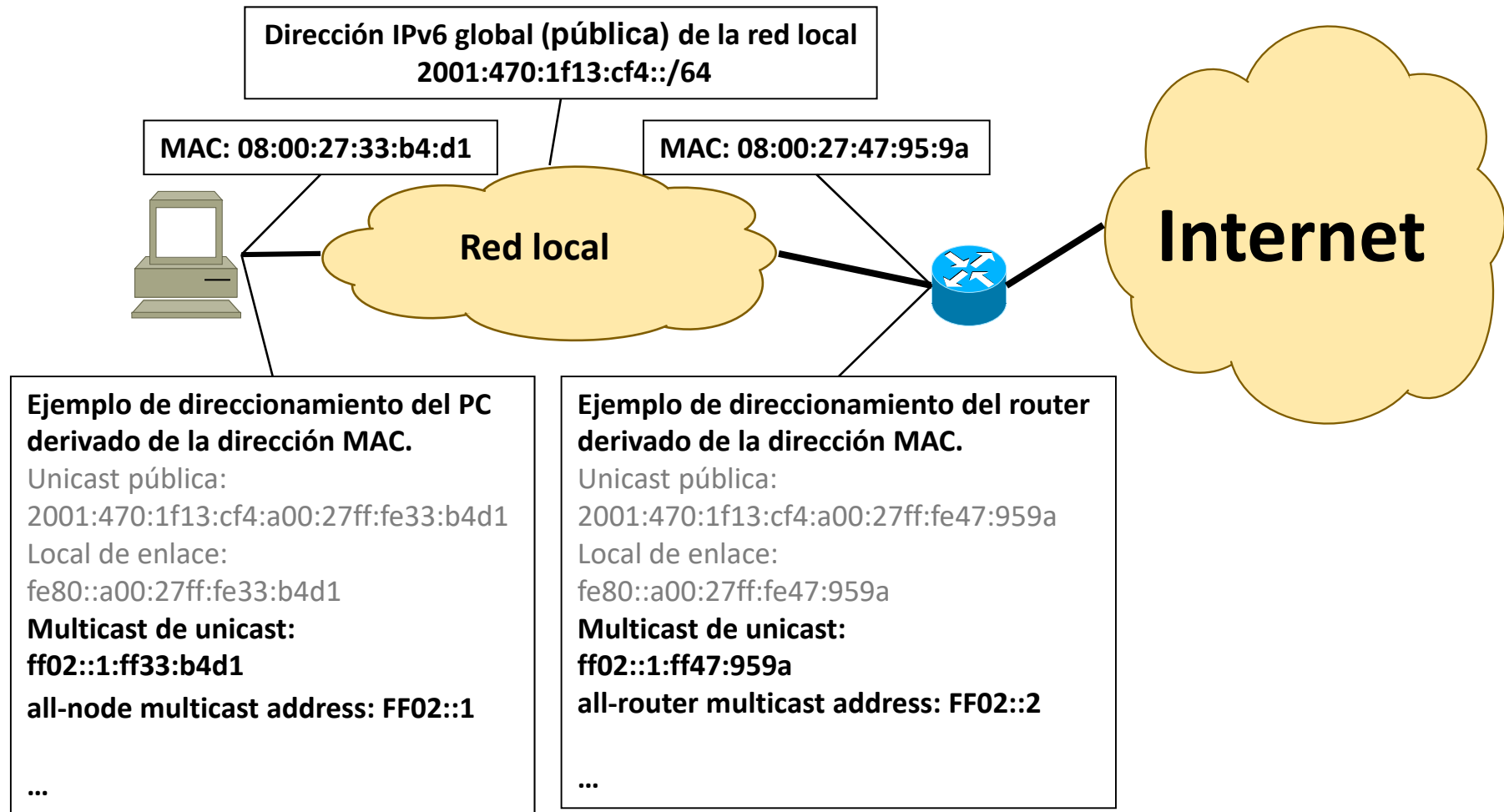


Figure 103: IPv6 Solicited Node Address Calculation

The solicited node multicast address is calculated from a unicast address by taking the last 24 bits of the address and prepending them with the IPv6 partial address "FF02:0:0:0:0:1:FF". This shows the example address from Figure 95 converted to its solicited node address, FF02::1:FFC8:1FFF.

Direccionamiento



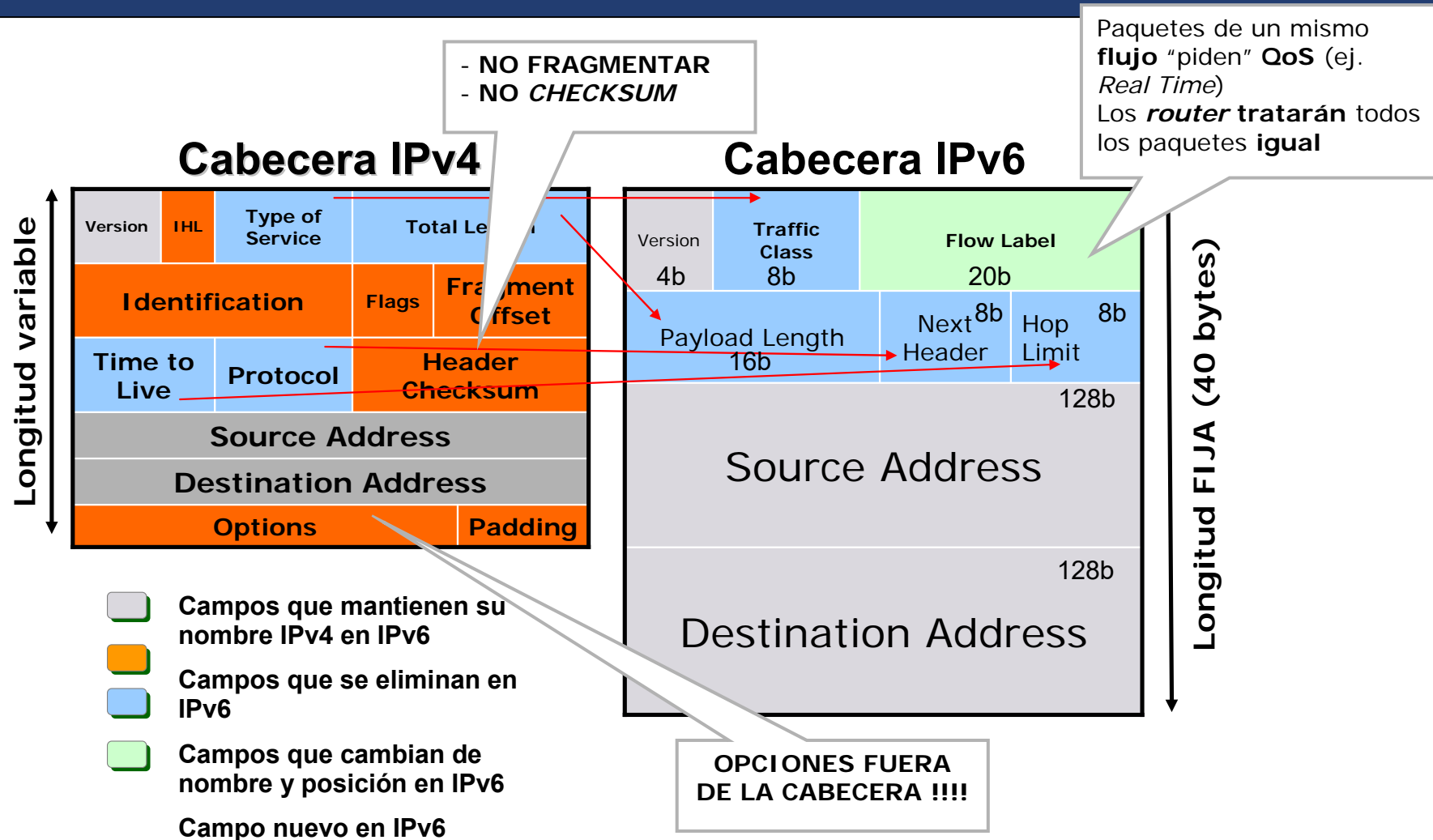
Contenidos

- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- Direccionamiento
- **PDU. Cabeceras de extensión**
- Autoconfiguración
- Funciones de control
- Encaminamiento

PDU. Cabeceras de extensión

- Principales novedades en IPv6
 - Modificación de la cabecera IP.
 - Opciones encadenadas: el campo de opciones se reemplaza por el de Siguiete cabecera; esto simplifica el proceso en cada *router* y da un mecanismo de extensión de opciones.
 - Eliminación de campos redundantes (*checksum*)
 - Se elimina la fragmentación de la red (y los campos en IP)
 - Mínimo MTU es 1280 bytes (28 en IPv4)
 - Posibilidad de envíos *unicast*, *multicast* y *anycast*
 - Autoconfiguración (sin servidor, “*plug-n-play*”) y movilidad
 - Flexible y extensible
 - Seguro

PDU. Cabeceras de extensión



PDU. Cabeceras de extensión

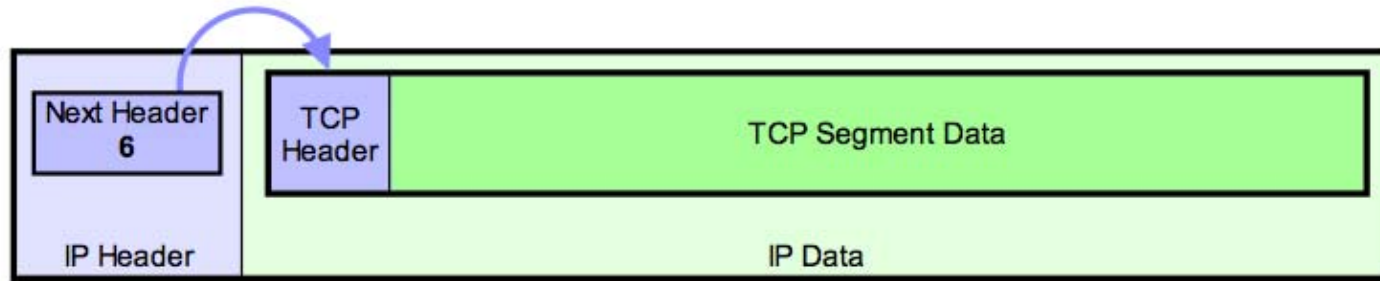
- Cabeceras de extensión **RFC 2460**
 - **Sólo** se añaden las cabeceras adicionales **necesarias**
 - menor carga
 - Ofrecen varios servicios y mejoras.
 - Cabeceras **Hoy-by-Hop, Routing y Destination Options [1]** analizadas salto a salto.
 - Resto de cabeceras **analizadas solo por el extremo final**
 - Campo '*Next Header*' (NH) indica la cabecera siguiente

- **Hop-by-Hop Options (NH=0)**
 - para todos los *router* intermedios
- **Destination Options [1] (NH=60)**
 - para el destinatario (IP dest) y nodos que aparecen en *routing header*
- **Routing (NH=43)**
 - encaminamiento fuente
- **Fragment (NH=44)**
 - sólo en origen
- **Authentication (NH=51)**
 - asegura la identidad del remitente **RFC 2402**
- **Encapsulating Security Payload (NH=50)**
 - cifrado de los datos
- **Destination Options [2] (NH=60)**
 - sólo para el destinatario final **RFC 2406**
- **Upper-layer (NH= 6:TCP, 17:UDP, 58:ICMPv6)**
 - Cabeceras superiores (como IPv4 Protocol)
- **Mobility (NH=135)**
 - Opciones de movilidad

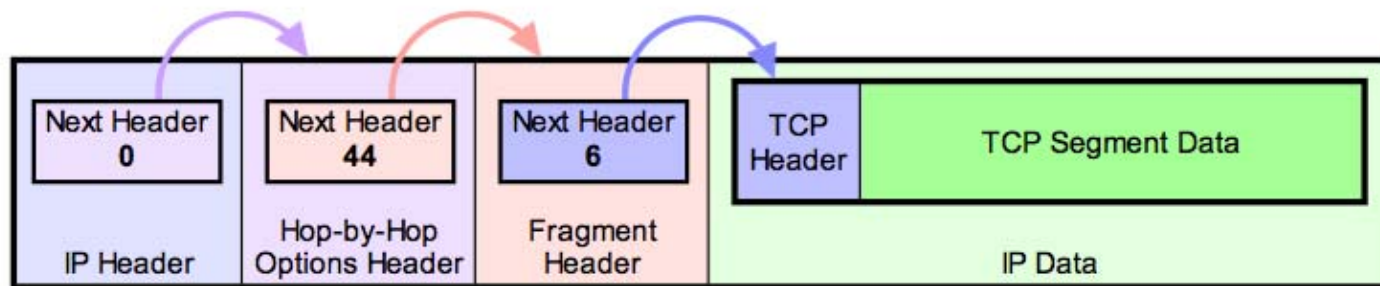
Siempre la primera

Orden recomendado

PDU. Cabeceras de extensión



IPv6 Datagram With No Extension Headers Carrying TCP Segment



IPv6 Datagram With Two Extension Headers Carrying TCP Segment

Figure 106: IPv6 Extension Header Linking Using the *Next Header* Field

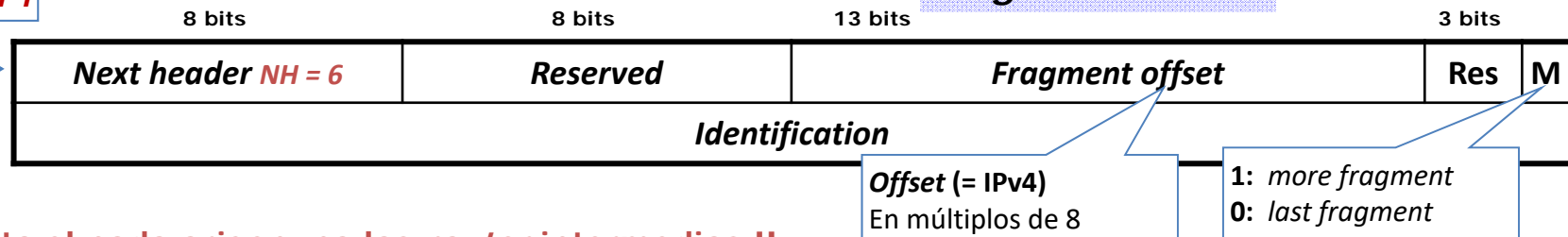
The *Next Header* field allows a device to more easily process the headers in a received IPv6 datagram. When a datagram has no extension headers, the "next header" is actually the header at the start of the IP *Data* field, in this case a TCP header with a value of 6. This is the same way the *Protocol* field is used in IPv4. When extension headers do appear, the *Next Header* value of each header contains a number indicating the type of the following header in the datagram, so they logically "chain together" the headers, as shown above.

PDU. Cabeceras de extensión

- Cabeceras de extensión - ejemplo

RFC 2460
(upd: RFC 5722)

NH = 44



Fragmenta el nodo origen, no los *router* intermedios !!

Parte no fragment.	Parte fragmentable
--------------------	--------------------

- Parte no fragmentable: Cabecera IPv6 y toda extensión tratable en ruta (*hop-by-hop* y *routing header*)
- Parte fragmentable: resto de cabeceras de extensión y *payload*

Parte no fragment.	Fragment header	First fragment
Parte no fragment.	Fragment header	Second fragment

...

Parte no fragment.	Fragment header	Last fragment
--------------------	-----------------	---------------

PDU. Cabeceras de extensión

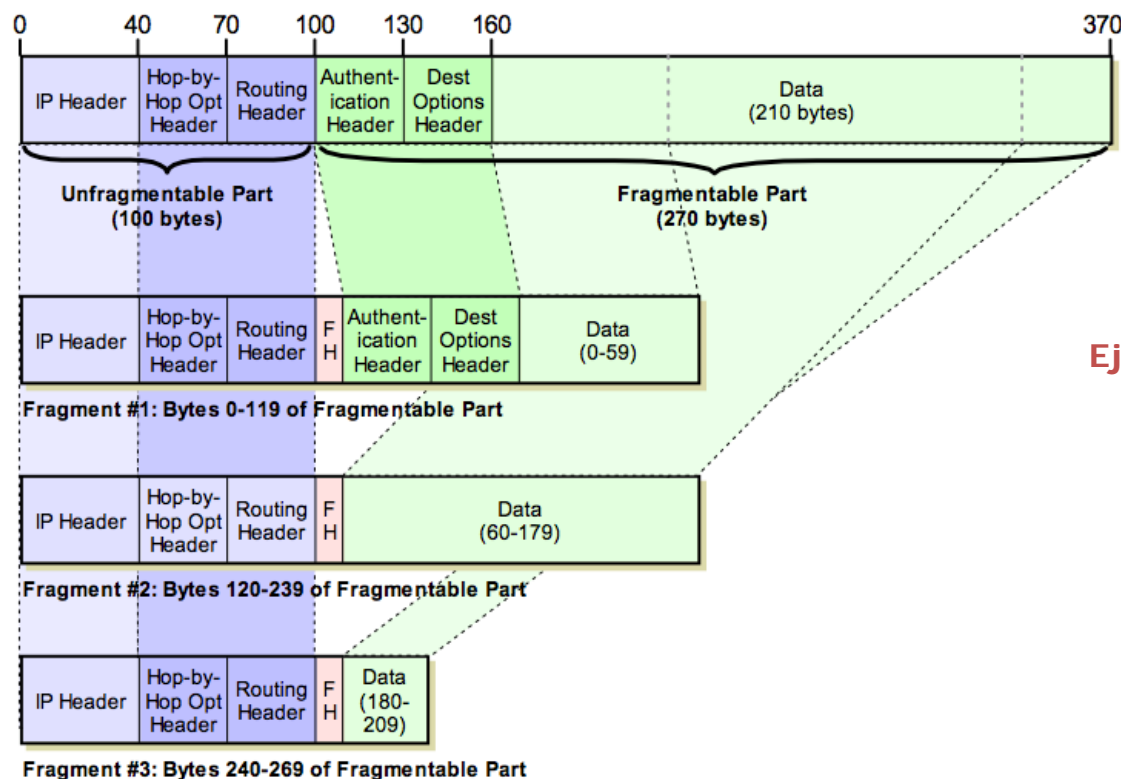


Figure 110: IPv6 Datagram Fragmentation

In this illustration, a 370-byte IPv6 datagram, containing four 30-byte extension headers, is broken into three fragments. The sizes of the fields are shown to scale. The *Unfragmentable Part*, shown in blue, begins each fragment, followed by the *Fragment* header (abbreviated as "FH" in the figure). Then, portions of the *Fragmentable Part* are placed into each fragment in sequence. The *Authentication* and *Destination Options* extension headers are part of the *Fragmentable Part* so they appear as part of the first fragment.

Contenidos

- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- Direccionamiento
- PDU. Cabeceras de extensión
- **Autoconfiguración**
- Funciones de control
- Encaminamiento

Autoconfiguración

- Proceso mediante el cual un host decide cómo autoconfigurar sus interfaces (*“Plug&Play”*)
- Existen dos formas de realizarse:
 - ***Stateless (StateLess Address Auto Configuration – SLAAC)*** **RFC 4862**
 - Un host se autoconfigura con la información local que tiene y la anunciada por los *router* (usada cuando no importa la configuración exacta de un *host*)
 - Se puede complementar (resto de parámetros de configuración) con *stateful* (indicado por los *router*).
 - ***Stateful (como IPv4)*** **RFC 3315**
 - Un host se autoconfigura mediante el uso de un servidor que mantiene una base de datos con las direcciones que deben ser asignadas a cada host:
DHCPv6: modelo cliente/servidor sobre UDP
 - Servidores envían en el puerto 546
 - Clientes envían en el puerto 547

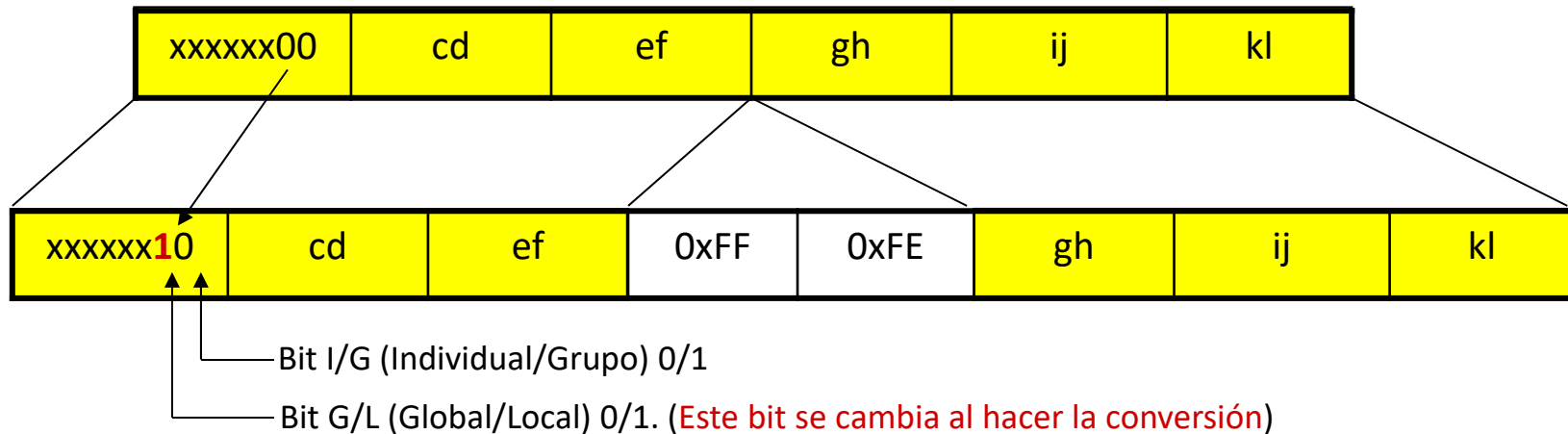
Autoconfiguración

- Autoconfiguración en IPv6: Host construye su propia dirección a partir de dos partes:
 - la parte red (8 bytes) que le indica el router
 - la parte host (8 bytes)

Por defecto,
lo habitual
(veremos)

- **Derivado de la dirección MAC** extendida: *Extended Unique Identifier (EUI)-64 address*. **RFC 4291**
- Número aleatorio cambiante en el tiempo (anónimo) **RFC 4941**
- Asignado durante la configuración *stateful* (ej. DHCPv6) **RFC 3315**
- Configuración manual
- Dirección criptográfica (*Cryptographically Generated Address (CGA)*) **RFC 3972**

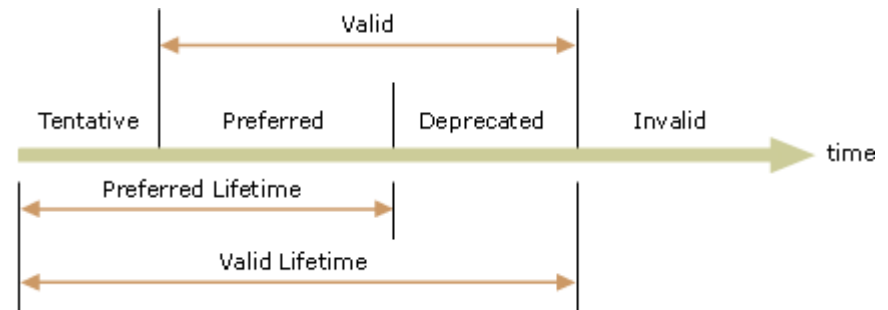
dirección IEEE 802 → identificador EUI-64



Autoconfiguración

- Direcciones IPv6
 - *Tentative*: en proceso de verificar si es única (no se considera asignada)
 - Duplicate Address Detection (DAD) similar al ARP gratuito. Se rechaza el resto de paquetes recibidos.
 - *Preferred*: Verificado que **es única** (mediante DAD) y utilizable sin restricciones (tanto fuente como destino)
 - *Deprecated*: Sigue válida pero su uso se deja de considerar para la comunicación.
 - Sigue pudiendo recibirse tráfico (IPdest)
 - Sólo se envía tráfico si lo requiere la aplicación (IPsrc), por ej, si ya había una comunicación en curso
 - Debería evitarse iniciar comunicaciones con dicha dirección (IPsrc)
 - *Invalid*: ya no puede usarse (cuando expira el tiempo de vida de una dirección válida)
 - El sistema de encaminamiento no lo puede entregar
 - Un receptor no sabe responder

- Tiempo de vida
 - *Preferred LifeTime*
 - *Valid LifeTime*



Autoconfiguración stateless

Requiere el uso de direcciones multicast (all-nodes y solicited-node): es necesario unirse a los grupos (protocolo MLD)

2. Router Advertisement (RA)

@S (*link-local*)

@D (*multicast todos los nodos IPv6*)

“El prefijo es **2001:720:1014:2**”

(y parámetros adicionales...)

- *Lifetime, MTU, stateful*

1. Router Solicitation (RS)

@S (*link-local*)

@D (*multicast a todos los router IPv6*)

“¿Me podéis decir el prefijo de esta red?”

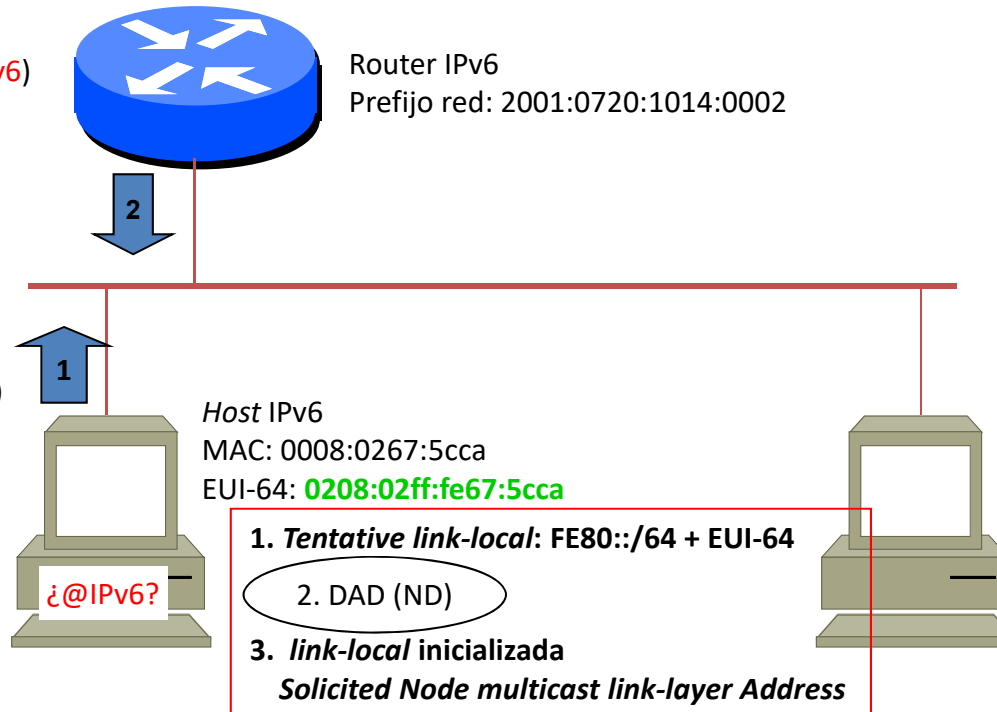
3. Autoconfiguración:

2001:720:1014:2:208:2ff:fe67:5cca

4. DAD (ND)

5. @IPv6 OK

Si el RA especificó “*stateful*”, el resto de parámetros (DNS, *router* por defecto...) se piden al servidor.



Capacidad *multicast* para autoconfiguración *stateless*

Router Advertisement	1	0
Configuración de dirección (M-bit)	<i>stateful</i> (DHCPv6)	<i>stateless</i>
Configuración de otros parámetros (O-bit)	<i>stateful</i>	<i>stateless</i>

Autoconfiguración **stateful**

Requiere el uso de direcciones multicast (all-nodes y solicited-node): es necesario unirse a los grupos (protocolo MLD)

2. Router Advertisement (RA)

@S (*link-local*)

@D (*multicast todos los nodos IPv6*)

"El prefijo es **2001:720:1014:2**"

(y parámetros adicionales...)

- *Lifetime, MTU, stateful*

1. Router Solicitation (RS)

@S (*link-local*)

@D (*multicast a todos los router IPv6*)

"¿Me podéis decir el prefijo de esta red?"

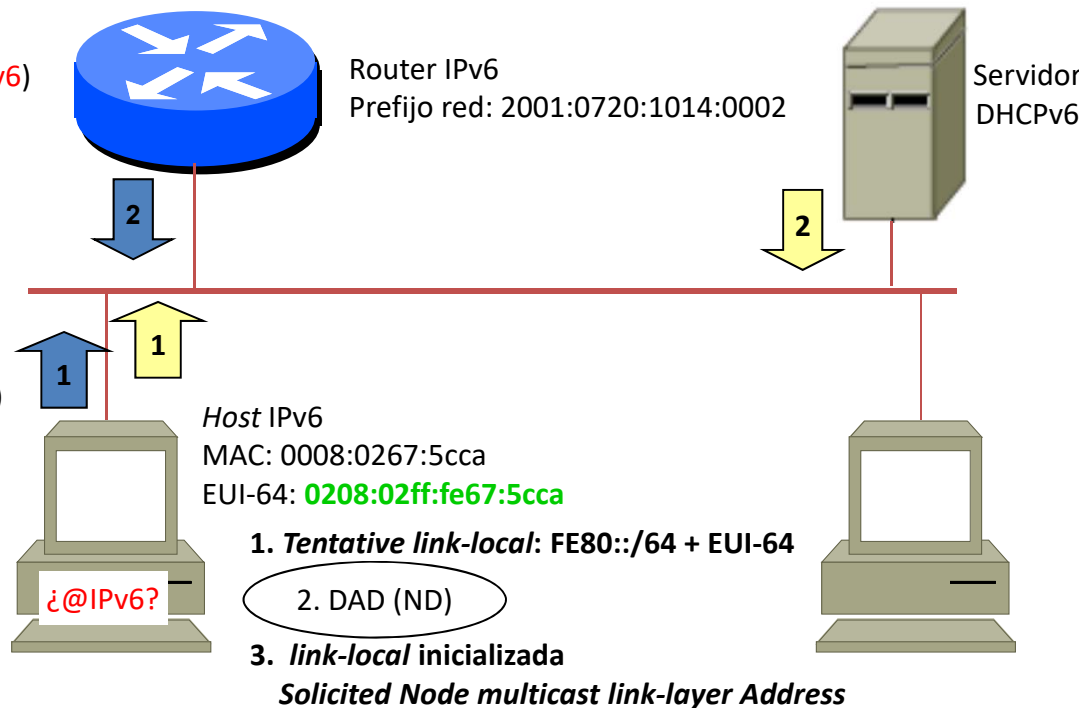
3. Autoconfiguración:

2001:720:1014:2:208:2ff:fe67:5cca

4. DAD (ND)

5. @IPv6 OK

Si el RA especificó "*stateful*", el resto de parámetros (DNS, *router* por defecto...) se piden al servidor.



Router Advertisement	1	0
Configuración de dirección (M-bit)	<i>stateful</i> (DHCPv6)	<i>stateless</i>
Configuración de otros parámetros (O-bit)	<i>stateful</i>	<i>stateless</i>

Contenidos

- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- Direccionamiento
- PDU. Cabeceras de extensión
- Autoconfiguración
- **Funciones de control**
- Encaminamiento

Funciones de control

- ICMPv6 = ICMP + IGMP + ARP:

NH = 58

- Como ICMPv4 (control de errores)

RFC 4443

- Destination unreachable, Packet too big, Time exceeded, Parameter problem, Echo request, Echo reply*

- Procedimiento **PMTU (Path Maximum Transfer Unit) Discovery**

RFC 1981

- » A través de mensajes ICMPv6 *Packet Too Big*

- MLD (*Multicast Listener Discovery*) (MLDv2)

RFC 3810

- Gestiona la pertenencia a un grupo *multicast* (IGMP en IPv4)

- ND (*Neighbor Discovery*)

RFC 4861

- Gestiona la comunicación nodo a nodo en un mismo enlace

- ARP en IPv4

- Descubrimiento de vecinos, alcanzabilidad

- Autoconfiguración

- ICMP *Redirect*

Funciones de control

- **ICMPv6**, como ICMPv4

Formato de mensaje ICMPv6

7	15	31
Tipo	Código	Checksum
Cuerpo del mensaje		

En caso de error, incluye **la mayor parte posible** (según min. MTU IPv6) del mensaje que lo provocó

Mensajes INFORMATIVOS ICMPv6

Tipo	Código	Descripción
128		<i>Echo Request</i> (Solicitud de eco)
129		<i>Echo Reply</i> (Respuesta de eco)

Mensajes de ERROR ICMPv6

Tipo	Código	Descripción
1	<i>Destination Unreachable</i> (Destino inalcanzable)	
	0	Sin ruta hacia el destino
	1	Comunicación prohibida administrativamente
	2	Sin asignar
	3	Dirección no alcanzable
2	4	Puerto no alcanzable
2	<i>Packet too Big</i> (Paquete demasiado grande)	
3	<i>Time Exceeded</i> (Tiempo excedido)	
	0	Límite de saltos excedido
4	1	Tiempo de desfragmentación excedido
	<i>Parameter Problems</i> (Problema de parámetros)	
	0	Campo erróneo de cabecera
	1	Tipo de cabecera siguiente desconocida
	2	Opción IPv6 desconocida

Funciones de control

Formatos de mensajes básicos (control de errores y diagnóstico)

Mensajes adicionales para funciones ND, MLD

- **ICMPv6, como ICMPv4**

Destination Unreachable

Type = 1	Code	Checksum
Unused		
Parte del paquete que provocó el error (lo máximo posible dado el mínimo MTU de IPv6)		

Packet too big

Type = 2	Code	Checksum
MTU		
Parte del paquete que provocó el error (lo máximo posible dado el mínimo MTU de IPv6)		

Time exceeded

Type = 3	Code	Checksum
Unused		
Parte del paquete que provocó el error (lo máximo posible dado el mínimo MTU de IPv6)		

Parameter problem

Type = 4	Code	Checksum
Pointer		
Parte del paquete que provocó el error (lo máximo posible dado el mínimo MTU de IPv6)		

Pointer: offset del paquete donde se encontró el error

Echo request (E.RQ)

Type = 128	Code	Checksum
Identifier		Sequence number
Data...		

Echo reply (E.RP)

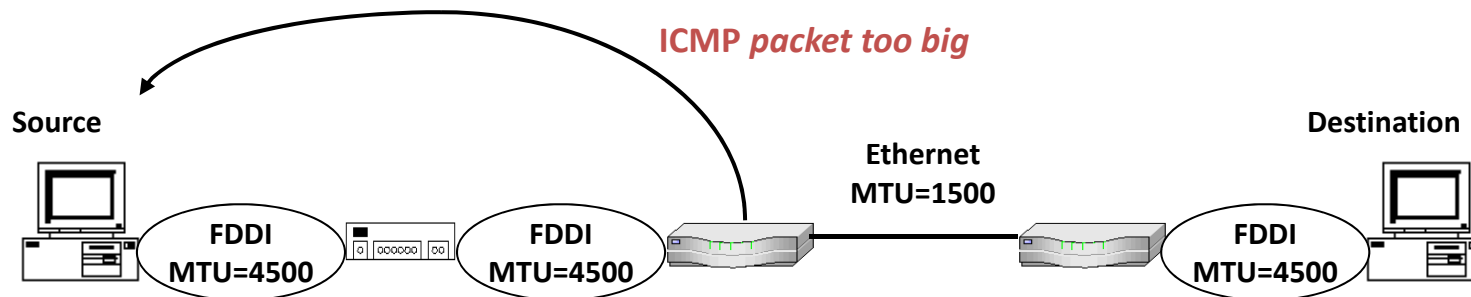
Type = 129	Code	Checksum
Identifier (= E.RQ)		Sequence num. (= E.RQ)
Data...		

Funciones de control

ICMPv6 como ICMPv4 - *PMTU (Path Maximum Transfer Unit) Discovery*

Fragmentación sólo extremo a extremo:

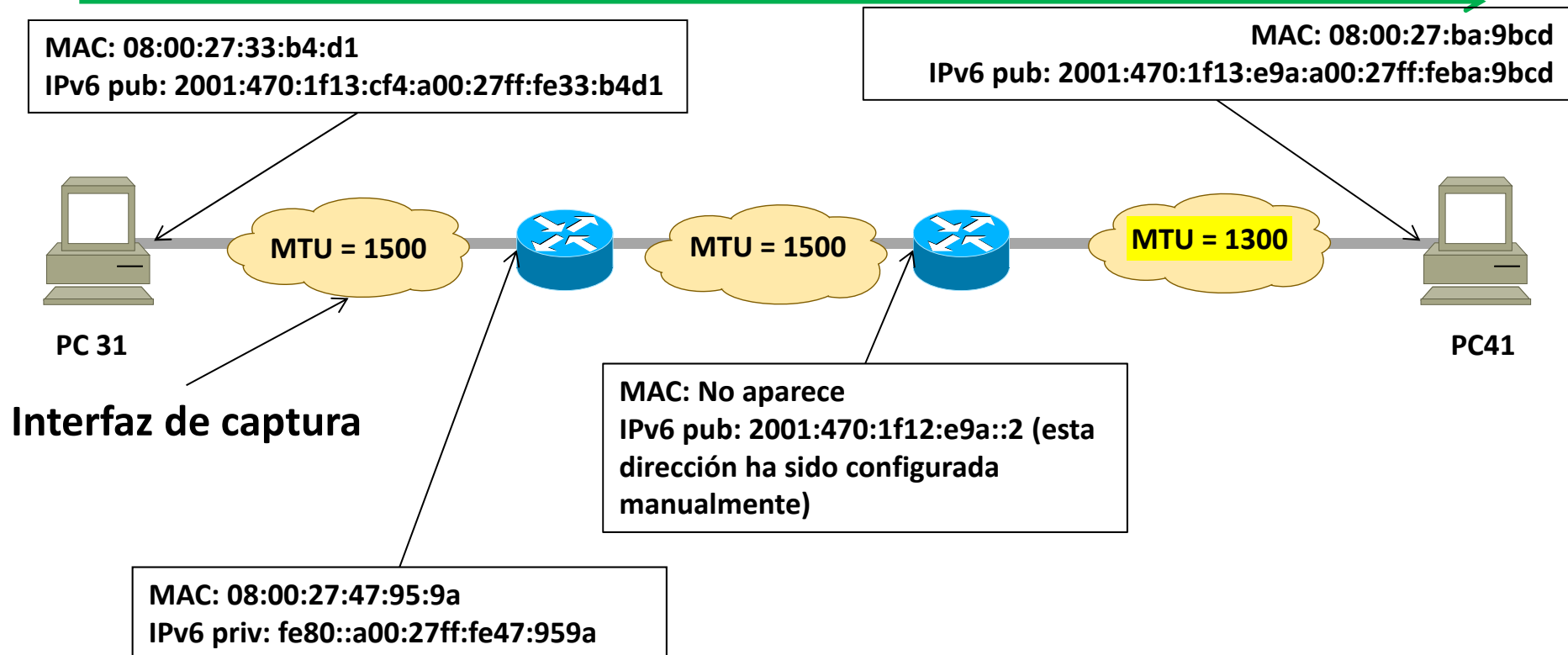
- a) usar el tamaño mínimo (1280 bytes)
 - b) descubrir el MTU cuando los paquetes son mayores
 1. Empezar asumiendo el MTU = el del primer salto
 2. Router intermedio no fragmenta: si el paquete no “cabe”, envía **ICMPv6 “packet too big”** a la fuente (con el nuevo tamaño de MTU)
 - Mejores prestaciones: reducción procesamiento en los router
 3. La fuente fragmenta de acuerdo al nuevo tamaño MTU: volver a 1.
- IPv6 informa al protocolo superior (ej. TCP) del tamaño MTU hacia ese destino



Funciones de control – Ejemplo PMTU

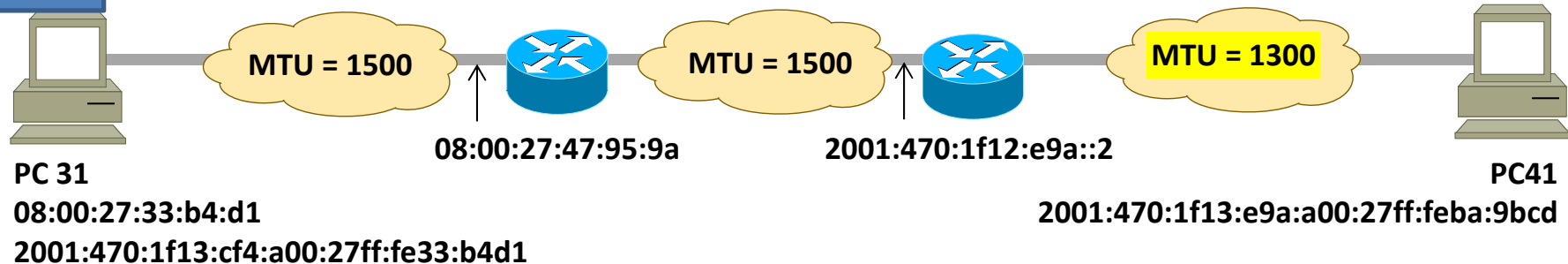
Caso 1

PC31 hace un ping a PC41 (1400 bytes de datos)



Funciones de control – Ejemplo PMTU

Caso 1



ICMP Echo Request (datos = 1400)

Source
2001:470:1f13:cf4:a00:27ff:fe33:b4d1

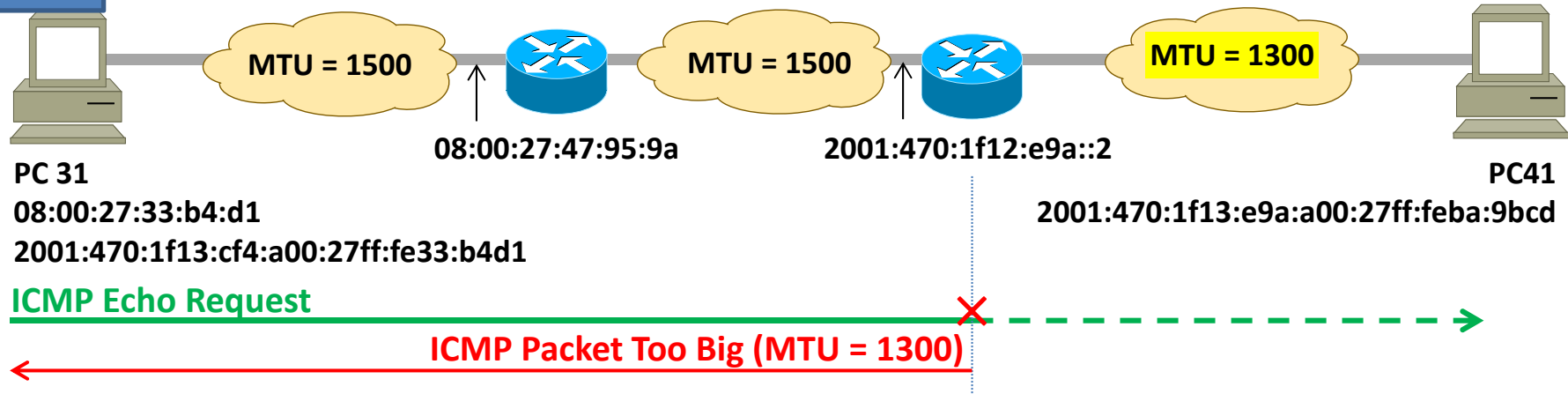
Destination
2001:470:1f13:e9a:a00:27ff:feba:9bcd

Protocol Info
ICMPv6 Echo request

Ethernet II, Src: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1), Dst: 08:00:27:47:95:9a (08:00:27:47:95:9a)
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 1408
Next header: ICMPv6 (0x3a)
Hop limit: 64
Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)
Destination: 2001:470:1f13:e9a:a00:27ff:feba:9bcd (2001:470:1f13:e9a:a00:27ff:feba:9bcd)
Internet Control Message Protocol v6
Type: 128 (Echo request)
Code: 0
Checksum: 0x9e98 [correct]
ID: 0x8313
Sequence: 0x0001
Data (1400 bytes)

Funciones de control – Ejemplo PMTU

Caso 1



Funciones de control – Ejemplo PMTU

Caso 1

← ICMP Packet Too Big (MTU = 1300)

Source: 2001:470:1f12:e9a::2 Destination: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 Protocol Info: ICMPv6 Too big

Ethernet II, Src: 08:00:27:47:95:9a (08:00:27:47:95:9a), Dst: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1)
Type: IPv6 (0x86dd)

Internet Protocol Version 6

Payload length: 1240
Next header: ICMPv6 (0x3a)
Hop limit: 62
Source: 2001:470:1f12:e9a::2 (2001:470:1f12:e9a::2)
Destination: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)

Internet Control Message Protocol v6

Type: 2 (Too big)
Code: 0
Checksum: 0xab7d [correct]
MTU: 1300

Internet Protocol Version 6
Payload length: 1408
Next header: ICMPv6 (0x3a)
Hop limit: 62
Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)
Destination: 2001:470:1f13:e9a:a00:27ff:feba:9bcd (2001:470:1f13:e9a:a00:27ff:feba:9bcd)

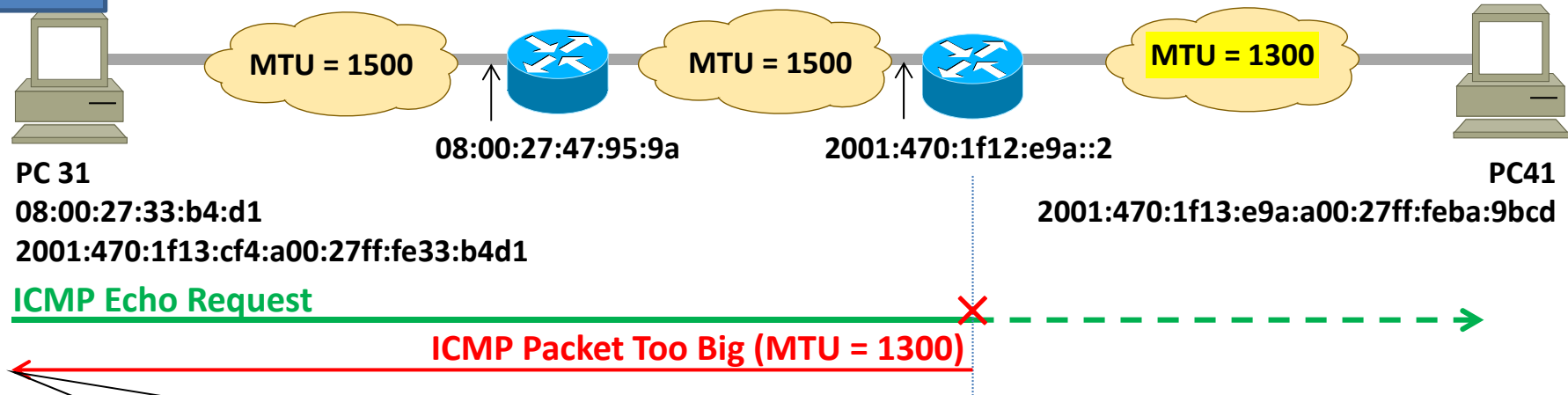
Internet Control Message Protocol v6

Type: 128 (Echo request)
Code: 0
Checksum: 0x9e98
ID: 0x8313
Sequence: 0x0001
Data (1184 bytes)

1184 + 8(ICMP) + 40(IP) +
8(ICMP) + 40(IP) = 1280??

Funciones de control – Ejemplo PMTU

Caso 1

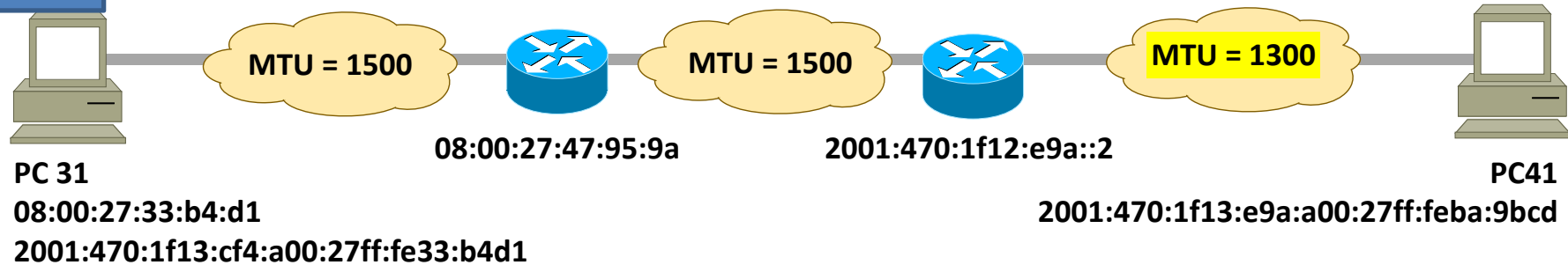


Modifica la entrada en la Destination Cache

```
2001:470:1f13:e9a:a00:27ff:feba:9bcd via fe80::a00:27ff:fe47:959a dev eth0 metric 0  
cache expires 0sec mtu 1300 advmss 1440 hoplimit 64
```


Funciones de control – Ejemplo PMTU

Caso 1



ICMP Echo Request (Fragment 1)

Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1
Destination: 2001:470:1f13:e9a:a00:27ff:feba:9bcd
Protocol Info: IPv6 fragment

Ethernet II, Src: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1), Dst: 08:00:27:47:95:9a (08:00:27:47:95:9a)

Type: IPv6 (0x86dd)

Internet Protocol Version 6

Payload length: 1256

Next header: IPv6 fragment (0x2c)

Hop limit: 64

Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)

Destination: 2001:470:1f13:e9a:a00:27ff:feba:9bcd (2001:470:1f13:e9a:a00:27ff:feba:9bcd)

Fragmentation Header

Next header: ICMPv6 (0x3a)

0000 0000 0000 0... = Offset: 0 (0x0000)

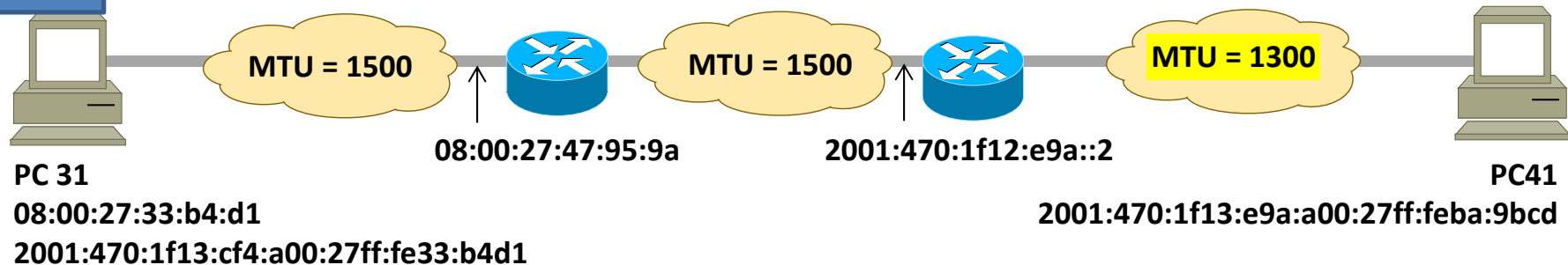
.... 1 = More Fragment: Yes

Identification: 0x00000001

Data (1248 bytes)

Funciones de control – Ejemplo PMTU

Caso 1



ICMP Echo Request (Fragment 1)

ICMP Echo Request (Fragment 2)

Source	Destination	Protocol Info
2001:470:1f13:cf4:a00:27ff:fe33:b4d1	2001:470:1f13:e9a:a00:27ff:feba:9bcd	IPv6 fragment

Ethernet II, Src: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1), Dst: 08:00:27:47:95:9a (08:00:27:47:95:9a)
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 168
Next header: IPv6 fragment (0x2c)
Hop limit: 64
Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)
Destination: 2001:470:1f13:e9a:a00:27ff:feba:9bcd (2001:470:1f13:e9a:a00:27ff:feba:9bcd)

Fragmentation Header
Next header: ICMPv6 (0x3a)
0000 0100 1110 0... = Offset: 156 (0x009c)
.... 0 = More Fragment: No
Identification: 0x00000001

1300 (MTU) – 40 (IPv6) – 8 (FH) = 1252
Múltiplo de 8 próximo: 1248 = **156** * 8

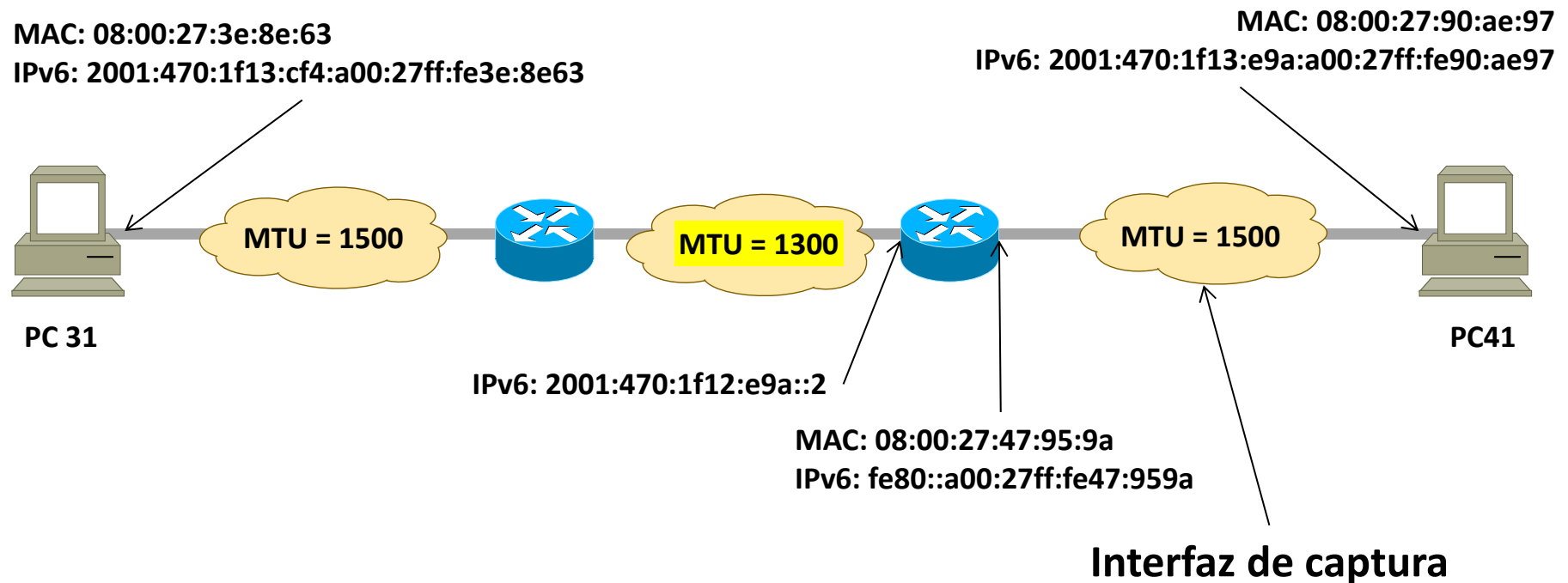
1408-1248=160

Data (160 bytes)

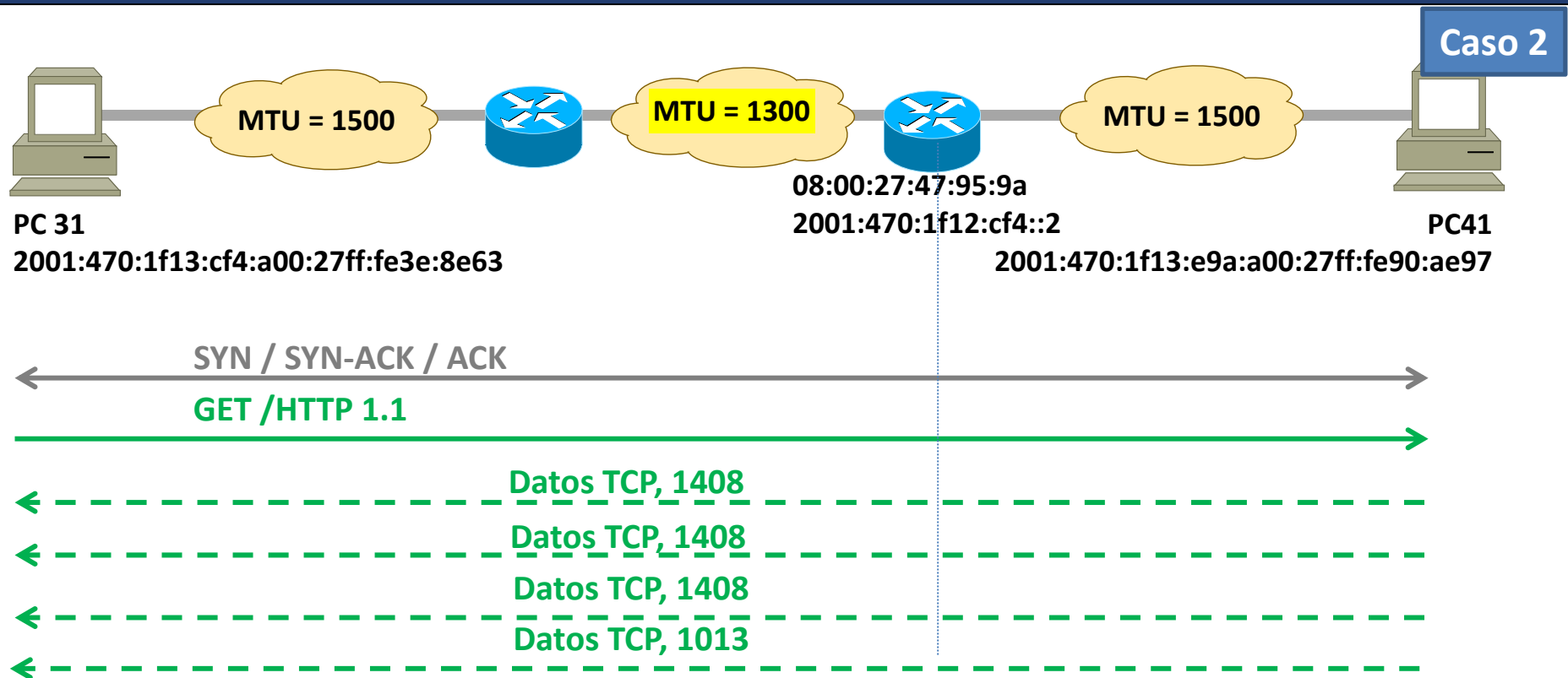
Funciones de control – Ejemplo PMTU

Caso 2

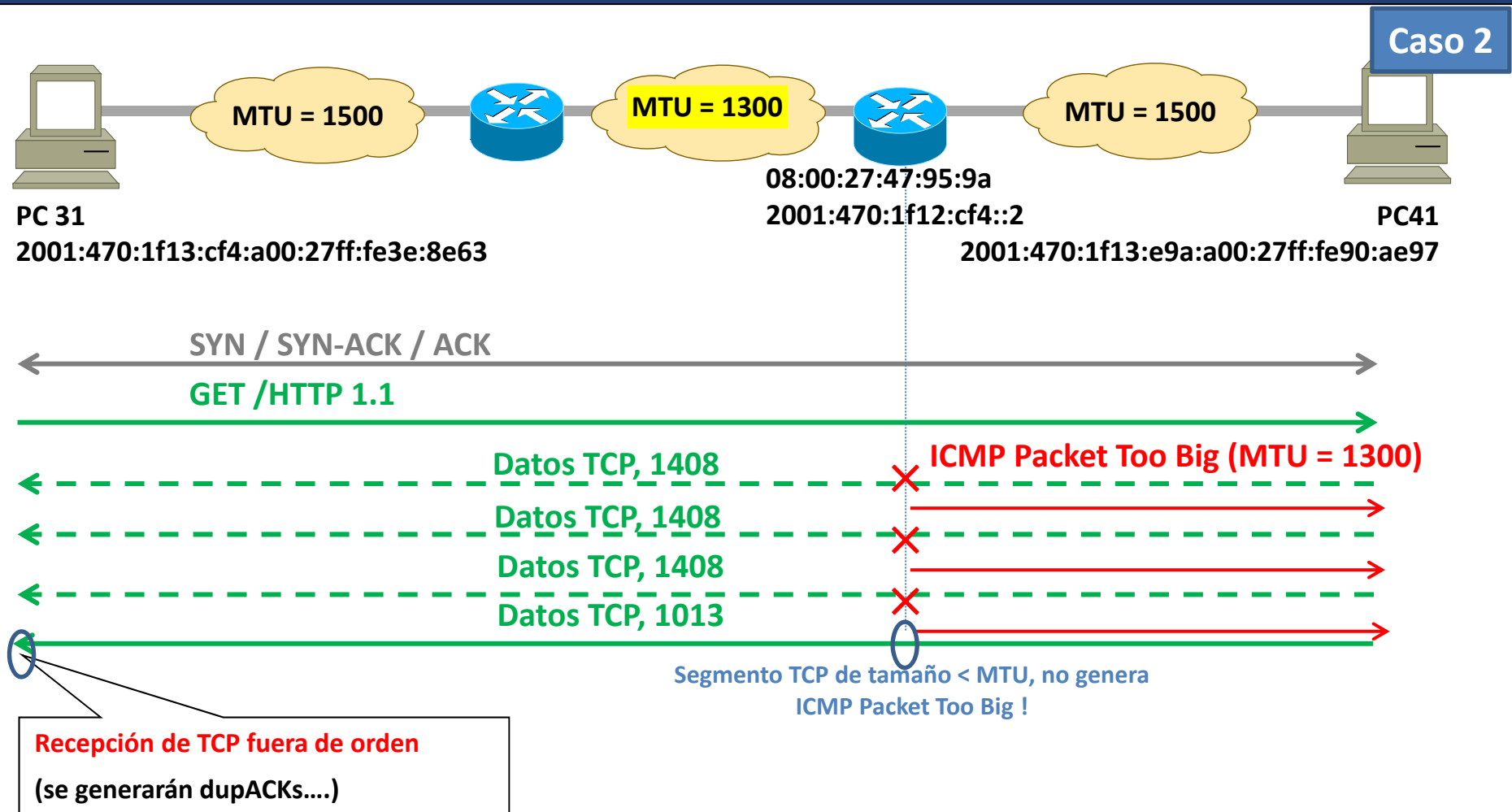
PC 31 accede a un servidor web (HTTP) de PC41



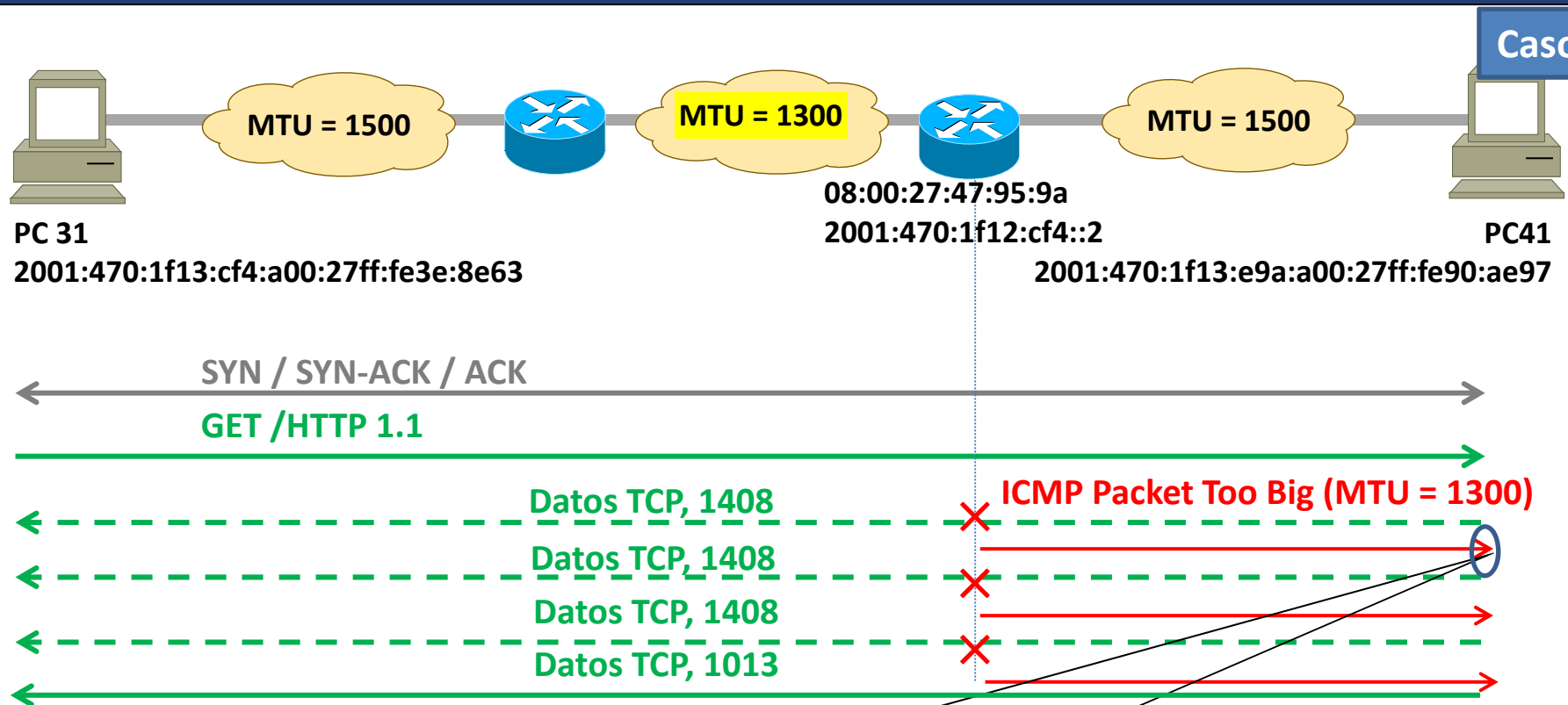
Funciones de control – Ejemplo PMTU



Funciones de control – Ejemplo PMTU



Funciones de control – Ejemplo PMTU

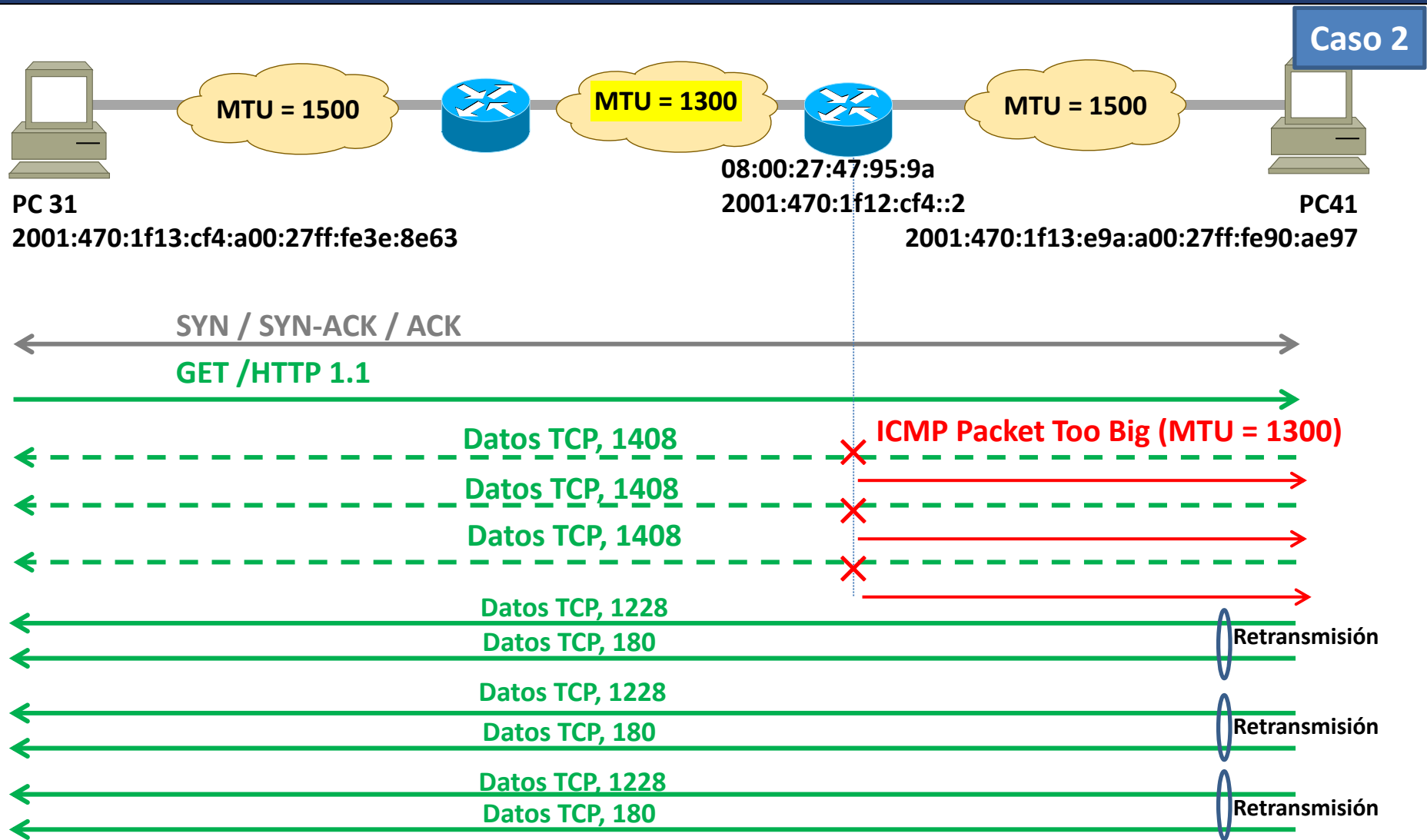


Modifica la entrada en la Destination Cache

2001:470:1f13:e9a:a00:27ff:fe3e:8e63 via fe80::a00:27ff:fe47:959a dev eth0 metric 0
cache expires 0sec **mtu 1300** advmss 1440 hoplimit 64

Informa a TCP del MTU, así que $MSS = MTU - h_IP - h_TCP = 1300 - 40 - 32 = 1228$

Funciones de control – Ejemplo PMTU



IPv6: ejemplo de Path MTU Discovery (TCP)

Caso 2

No.	Time	Source	Destination	Protocol Info
05	6.6349470	fe80::a00:27ff:fe47:959a	ff02::1	ICMPv6 Router advertisement
12	11.399134	fe80::a00:27ff:fe47:959a	ff02::1	ICMPv6 Router advertisement
13	11.419216	fe80::a00:27ff:fe47:959a	ff02::1:ff90:ae97	ICMPv6 Neighbor solicitation
14	11.420101	2001:470:1f13:cf4:a00:27ff:fe90:ae97	fe80::a00:27ff:fe47:959a	ICMPv6 Neighbor advertisement
15	11.420114	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [SYN] Seq=0 Win=5760 Len=0 MSS=1440
16	11.421094	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP 80 > 48804 [SYN, ACK] Seq=0 Ack=1 Win=5712 Len=0 MSS=1440
17	11.524185	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=1 Ack=1 Win=5760 Len=0
18	11.525563	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	HTTP GET / HTTP/1.1
19	11.526659	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP 80 > 48804 [ACK] Seq=1 Ack=411 Win=6784 Len=0
20	11.530456	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP segment of a reassembled PDU] TCP seg. data=1408
21	11.530496	2001:470:1f12:cf4::2	2001:470:1f13:cf4:a00:27ff:fe90:ae97	ICMPv6 Too big

Actualiza la Destination cache con la información recibida

2001:470:1f13:e9a:a00:27ff:fe3e:8e63 via fe80::a00:27ff:fe47:959a dev eth0 metric 0 cache expires 0sec mtu 1300 advmss 1440 hoplimit 64

Informa a TCP del MTU, así que que $MSS = MTU - h_IP - h_TCP = 1300 - 40 - 32 = 1228$

22	11.531621	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP segment of a reassembled PDU] TCP seg. data=1408
23	11.531640	2001:470:1f12:cf4::2	2001:470:1f13:cf4:a00:27ff:fe90:ae97	ICMPv6 Too big
24	11.533018	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP segment of a reassembled PDU] TCP seg. data=1408
25	11.533035	2001:470:1f12:cf4::2	2001:470:1f13:cf4:a00:27ff:fe90:ae97	ICMPv6 Too big
26	11.533931	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP segment of a reassembled PDU] TCP seg. data=1013
27	11.538741	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=1228
28	11.538830	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=180
29	11.639733	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP [TCP Dup ACK 18#1] 48804 > 80 [ACK] Seq=411 Ack=1 Win=5760 Len=0
30	11.641465	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=1228
31	11.644313	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=1229 Win=8224 Len=0
32	11.644350	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=1409 Win=10688 Len=0
33	11.645126	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=180
34	11.646181	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=1228
35	11.646591	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP [TCP Ret.] [TCP seg. of a reassembled PDU] TCP seg. data=180
36	11.747105	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=2637 Win=13152 Len=0
37	11.747752	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=2817 Win=15584 Len=0
38	11.751507	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=4045 Win=18048 Len=0
39	11.752202	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [ACK] Seq=411 Ack=5239 Win=20512 Len=0
40	11.752649	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	2001:470:1f13:cf4:a00:27ff:fe90:ae97	TCP 48804 > 80 [FIN, ACK] Seq=411 Ack=5239 Win=20512 Len=0
41	11.753273	2001:470:1f13:cf4:a00:27ff:fe90:ae97	2001:470:1f13:e9a:a00:27ff:fe3e:8e63	TCP 80 > 48804 [ACK] Seq=5239 Ack=412 Win=6784 Len=0

Funciones de control

- **ICMPv6 - ND (Neighbor Discovery)**

- Reemplaza procedimientos de IPv4 y proporciona funcionalidad adicional
- Define 5 tipos diferentes de paquetes ICMP:
 - *Router Solicitation / Router Advertisement* (R.S. / R.A.)
 - *Neighbor Solicitation / Neighbor Advertisement* (N.S / N.A.)
 - *Redirect*
- **Host** lo usan para:
 - Descubrimiento de *router* vecinos (poder determinar *router* por defecto)
 - Descubrimiento de direcciones, prefijos y otros parámetros de configuración
- **Router** lo usan para:
 - Anunciar su presencia, parámetros de configuración del *host*, prefijos del enlace (*on-link*)
 - Informar a los *host* de una mejor dirección de siguiente salto para el encaminamiento hacia un destino específico (*redirect*)
- **Nodos (genérico)** lo usan para:
 - Resolver cuál es la dirección de enlace de un vecino para el que enviar un paquete IPv6 (ARP)
 - Determinar si los paquetes IPv6 pueden enviarse y ser recibidos por un vecino (alcanzabilidad)
- Requiere el uso de **direcciones multicast de ámbito local**
 - *all-node multicast address*: todos los nodos (**FF02::1**)
 - *all-router multicast address*: todos los router (**FF02::2**)
 - *solicited-node multicast address* (**FF02:0:0:0:0:1:FFXX:XXXX**) (X:EUI-64)
- Requiere el uso de **direcciones unicast de ámbito local**
 - *link-local*: ámbito local no fuera del mismo (**FE80::XXXX:XXXX:XXXX:XXXX**) (X:EUI-64)
 - Dirección sin especificar: dirección fuente mientras se desconoce la dirección (::)

Funciones de control

- **ICMPv6 - ND (*Neighbor Discovery*)**

Procedimiento ND	Descripción
Router Discovery	Como ICMPv4 Router Discovery Descubre <i>router</i> locales en su red y configura el <i>router</i> por defecto (para destinos ' <i>off-link</i> ')
Prefix Discovery	Descubrimiento de prefijos para los destinos locales (determinar quién está ' <i>on-link</i> ')
Parameter Discovery	Descubrimiento de parámetros adicionales, como MTU y límite de saltos por defecto
Router Advertisement (R.A.) @dest = FF02::1 (todos los nodos) Router Solicitation (R.S.) @dest = FF02::2 (todos los router)	
Address autoconfiguration	Configuración automática de la dirección IP para las interfaces locales (con/sin servidores de configuración <i>stateful</i>)
Address Resolution	Como ARP en IPv4 (tabla ARP) Resolución de la dirección de enlace (caché de vecinos: <i>neighbors cache</i>) @dest = Solicited Node Address
Next-hop determination	Resolución del siguiente salto para determinada dirección destino: será dicho destino o el siguiente <i>router</i> . (Tabla de destinos: <i>destination cache</i>)
Neighbor Unreachability Detection (NUD)	Mecanismo para determinar si un vecino ha dejado de ser alcanzable
Duplicate Address Detection (DAD)	Equivalente a ARP gratuito en IPv4 Mecanismo para determinar si la dirección considerada (tentativa) ya está en uso. Uso temporal de la dirección @src = link-local
Redirect function	Equivalente a IPV4 ICMP redirect <i>Router</i> informa a un host de otra dirección IPv6 como primer salto para alcanzar el destino

Se analiza quién es siguiente salto, se almacena la información y el procedimiento no se hace continuamente

Funciones de control

RFC 4861
RFC 5942

Detalles completos

- ICMPv6 - ND (*Neighbor Discovery*)

- Periódicos (*unsolicited*)
- En respuesta a R.S.

ROUTER SOLICITATION (R.S.)

...			
			Hop = 255
IP src (IP, o ::)			
IP dst (FF02::2)			
133	0	Checksum	
Reserved			
Options...			

Options:

- **Source link-layer address**
(dirección de enlace del nodo fuente – IP src ≠ ::)

Si el nodo (*router/host*) informa en su anuncio de la dirección de enlace, se evita la resolución de dirección (~ARP)

ROUTER ADVERTISEMENT (R.A.)

...			
			Hop = 255
IP src (link-local del router)			
IP dst (IP src del R.S. o FF02::1)			
134	0	Checksum	
Cur.hop.lim	M O Reserv	Router Life Time	
Reachable time			
Retransm. Time			
Options...			

Cur. Hop.limit = límite de saltos

Flags:

M: autoconfiguración ⇒ IP add via DHCPv6

O: autoconfiguración ⇒ otros parámetros via DHCPv6

Options:

- **Source link-layer address**
- MTU
- Prefix information (ver sig...)

Funciones de control

- ICMPv6 - ND (*Neighbor Discovery*)

RFC 4861
RFC 5942

Detalles completos

Opciones...

Type	Length	...
...

Tipo	Opción	datos
1	Source link-layer address	Dirección de enlace (quien origina el mensaje)
2	Target link-layer address	Dirección de enlace (Target address que aparece en el mensaje)
3	Prefix information	Información de prefijos →
4	Redirect header	Paquete que causó el ICMP redirect (cabecera y datos), según MTU
5	MTU	MTU del enlace en el que se envía el R.A.

PREFIX INFORMATION

Type = 3	Len = 4	Pref. len	L A Reserv
Valid lifetime			
Preferred lifetime			
Reserved 2			
Prefix			

Datos útiles del prefijo, según la longitud indicada

L: prefijo para determinar si destino es 'on-link'
Valid lifetime: duración para 'on-link'

A: prefijo para determinar @IP en autoconf.
Preferred lifetime: duración para @IP (auto.)

Funciones de control

RFC 4861

RFC 5942

Detalles completos

- ICMPv6 - ND (*Neighbor Discovery*)

NEIGHBOR SOLICITATION (N.S.)

...			
			Hop = 255
IP src (IP o :: - DAD -)			
IP dst = Solicited Node Address			
135	0	Checksum	
Reserved			
Target Address (NO multicast)			
Options...			

*

NEIGHBOR ADVERTISEMENT (N.A.)

...			
			Hop = 255
IP src (IP)			
IP dst (IP src de N.S. o FF02::1)			
136	0	Checksum	
RS0	Reserved		
Target Address (NO multicast)			
Options...			

*

Options:

- Source link-layer address

Flags:

- **R**: es un *router*
- **S**: respuesta a solicitud de un N.S. (uso en NUD)
- **O**: anular (*override*) una entrada de caché y actualizar la caché de direcciones de enlace

Options:

- Target link-layer address
(dirección de enlace de la 'target address')

Unsolicited N.A. (IP dst = FF02::1) ~ Proxy ARP
D.A.D. N.S. con IP src = ::, IP dst = 'tentative' ~ Gratuitous ARP (detectar IP duplicada)

Funciones de control

RFC 4861

RFC 5942

Detalles completos

- ICMPv6 - ND (*Neighbor Discovery*)

REDIRECT

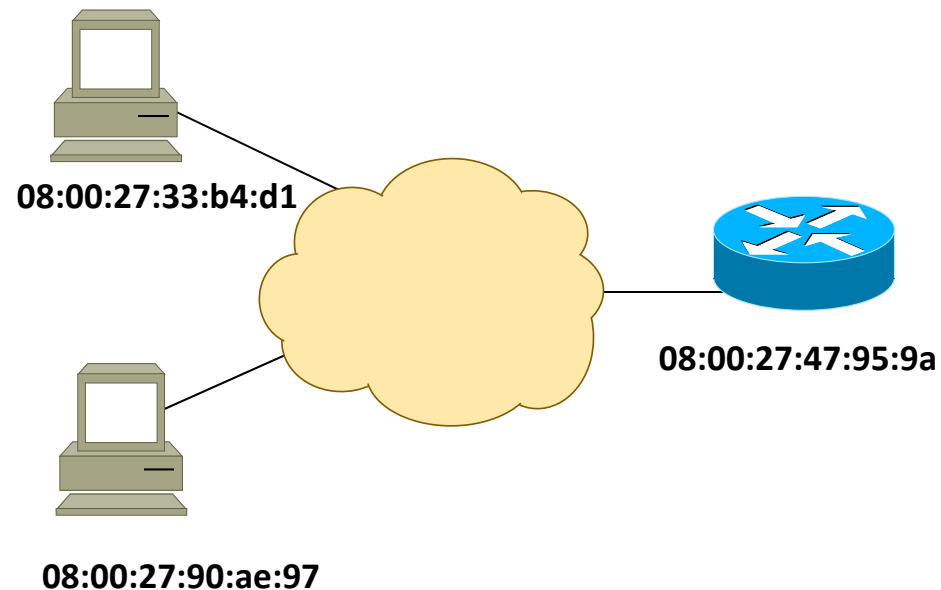
...			
			Hop = 255
IP src (link local del router por defecto)			
IP dst (IP src del paquete original)			
137	0	Checksum	
Reserved			
Target Address (siguiente salto sugerido)			
Destination Address (destino original)			
Options...			

Options:

- **Target link-layer address** (dirección de enlace de la 'target address')
 - IP dst es vecino \Rightarrow target = IP dst (original)
 - IP off-link \Rightarrow link-local address del router
- Redirect header
(lo que "quepa" en la MTU mínima, del paquete que generó este mensaje)

Ejemplo de autoconfiguración *stateless*

- Escenario



Ejemplo de autoconfiguración *stateless*

1. Host IPv6. Tarjeta de red → dirección MAC 08:00:27:90:ae:97

1.1 Obtener EUI-64 → a00:27ff:fe90:ae97

1.2 Obtener *Tentative link-local* → fe80::a00:27ff:fe90:ae97

1.3 Obtener *Solicited Node Address (Multicast)* → ff02::1:ff90:ae97

2 Realizar *Duplicated Address Detection (DAD)* para link-local address

2.1 Ethernet destino → MAC *multicast* equivalencia SNA → 33:33:ff:90:ae:97

2.2 Se envía el NS

Source	Destination	Protocol Info
::	ff02::1:ff90:ae97	ICMPv6 Neighbor solicitation

Ethernet II, Src: 08:00:27:90:ae:97, Dst: 33:33:ff:90:ae:97

Type: IPv6 (0x86dd)

Internet Protocol Version 6

Payload length: 24

Next header: ICMPv6 (0x3a)

Hop limit: 255

Source: ::

Destination: ff02::1:ff90:ae97

Internet Control Message Protocol v6

Type: 135 (Neighbor solicitation)

Code: 0

Checksum: 0xedd7 [correct]

Target: fe80::a00:27ff:fe90:ae97

Ejemplo de autoconfiguración *stateless*

2.3 Casos posibles

2.3.1 Se recibe un **NA** en respuesta al **NS** → error (abortar)

```
Source      Destination      Protocol Info
fe80::a00:27ff:fe90:ae97  ff02::1          ICMPv6 Neighbor advertisement

Ethernet II, Src: 08:00:27:90:ae:97, Dst: 33:33:00:00:00:01
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 32
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: fe80::a00:27ff:fe90:ae97
Destination: ff02::1
Internet Control Message Protocol v6
Type: 136 (Neighbor advertisement)
Code: 0
Checksum: 0xbd26 [correct]
Flags: 0x20000000
  0... .. = Not router
  .0.. .. = Not adverted
  ..1. ... = Override
Target: fe80::a00:27ff:fe90:ae97 (fe80::a00:27ff:fe90:ae97)
ICMPv6 Option (Target link-layer address)
Type: Target link-layer address (2)
Length: 8
Link-layer address: 08:00:27:90:ae:97
```

Ejemplo de autoconfiguración *stateless*

2.3 Casos posibles

2.3.2 Se recibe un **NS** igual al enviado por el host → error (abortar)

```
Source      Destination      Protocol Info
::          ff02::1:ff90:ae97 ICMPv6 Neighbor solicitation

Ethernet II, Src: 08:00:27:90:ae:97, Dst: 33:33:ff:90:ae:97
Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Payload length: 24
  Next header: ICMPv6 (0x3a)
  Hop limit: 255
  Source: ::
  Destination: ff02::1:ff90:ae97 → SNA de fe80::a00:27ff:fe90:ae97 (link local)
Internet Control Message Protocol v6
  Type: 135 (Neighbor solicitation)
  Code: 0
  Checksum: 0xedd7 [correct]
  Target: fe80::a00:27ff:fe90:ae97 ←
```

2.3.3 No se recibe nada

2.3.3.1 Se asigna la dirección al interface y pasa a ser *Preferred*

2.3.3.1 Escucho paquetes destinados a esa dirección (*unicast* y *multicast*)

```
eth0  Link encap:Ethernet HWaddr 08:00:27:90:AE:97
      inet6 addr: fe80::a00:27ff:fe90:ae97/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

Ejemplo de autoconfiguración *stateless*

3. Envía **RS**

Source	Destination	Protocol Info
fe80::a00:27ff:fe90:ae97	ff02::2	ICMPv6 Router solicitation

Ethernet II, Src: 08:00:27:90:ae:97, Dst: 33:33:00:00:00:02
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 16
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: fe80::a00:27ff:fe90:ae97
Destination: ff02::2
Internet Control Message Protocol v6
Type: 133 (Router solicitation)
Code: 0
Checksum: 0xb324 [correct]
ICMPv6 Option (Source link-layer address)
Type: Source link-layer address (1)
Length: 8
Link-layer address: 08:00:27:90:ae:97

3.1 El *router* que lo recibe actualiza su *Neighbor Cache* (RFC 4861 sec. 6.2.6)

root:~ # ip -6 neighbor show (comando que muestra la Neighbor cache)

fe80::a00:27ff:fe90:ae97 dev eth1 lladdr 08:00:27:90:ae:97 notrouter STALE

Ejemplo de autoconfiguración *stateless*

3. Recibe RA

Source	Destination	Protocol Info
fe80::a00:27ff:fe47:959a	ff02::1	ICMPv6 Router advertisement

Ethernet II, Src: 08:00:27:47:95:9a, Dst: 33:33:00:00:00:01
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 56
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: fe80::a00:27ff:fe47:959a
Destination: ff02::1 (podría ser también la local de enlace del RS)

Internet Control Message Protocol v6
Type: 134 (Router advertisement)
Code: 0
Checksum: 0x9119 [correct]
Cur hop limit: 64
Flags: 0x00
0... = Not managed (**Flag M**)
.0.. = Not other (**Flag O**)
..0. = Not Home Agent
...0 0... = Router preference: Medium
Router lifetime: 300
Reachable time: 0
Retrans timer: 0

ICMPv6 Option (Prefix information)
Type: Prefix information (3)
Length: 32
Prefix length: 64
Flags: 0xc0
1... = Onlink (**Flag L**)
.1.. = Auto (**Flag A**)
..0. = Not router address
...0 = Not site prefix
Valid lifetime: 86400
Preferred lifetime: 14400
Prefix: 2001:470:1f13:cf4::

ICMPv6 Option (Source link-layer address)
Type: Source link-layer address (1)
Length: 8
Link-layer address: 08:00:27:47:95:9a

Ejemplo de autoconfiguración *stateless*

3. Recibe **RA** (RFC 4861, sección 6.3.4): acciones

3.1 *Default Router List* → escribir entrada

default via fe80::a00:27ff:fe47:959a dev eth0 proto kernel metric 1024 expires 0sec mtu 1500 advmss 1440 hoplimit 64

3.2 *Neighbor Cache* → escribir entrada

fe80::a00:27ff:fe47:959a dev eth1 lladdr 08:00:27:47:95:9a router STALE

3.3 *Prefix List* (Flag L puesto a 1) → escribir entrada

fe80::/64 dev eth0 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0 (*ya estaba*)

2001:470:1f13:cf4::/64 dev eth0 proto kernel metric 256 expires 0sec mtu 1500 advmss 1440 hoplimit 0

Ejemplo de autoconfiguración *stateless*

3.4. Realizar DAD para *Global Address* (Flags M = 0, A = 1)

3.4.1 Ethernet destino → MAC *multicast* equivalencia SNA → 33:33:ff:90:ae:97

3.4.2 Envío del NS

Source	Destination	Protocol Info
::	ff02::1:ff90:ae97	ICMPv6 Neighbor solicitation

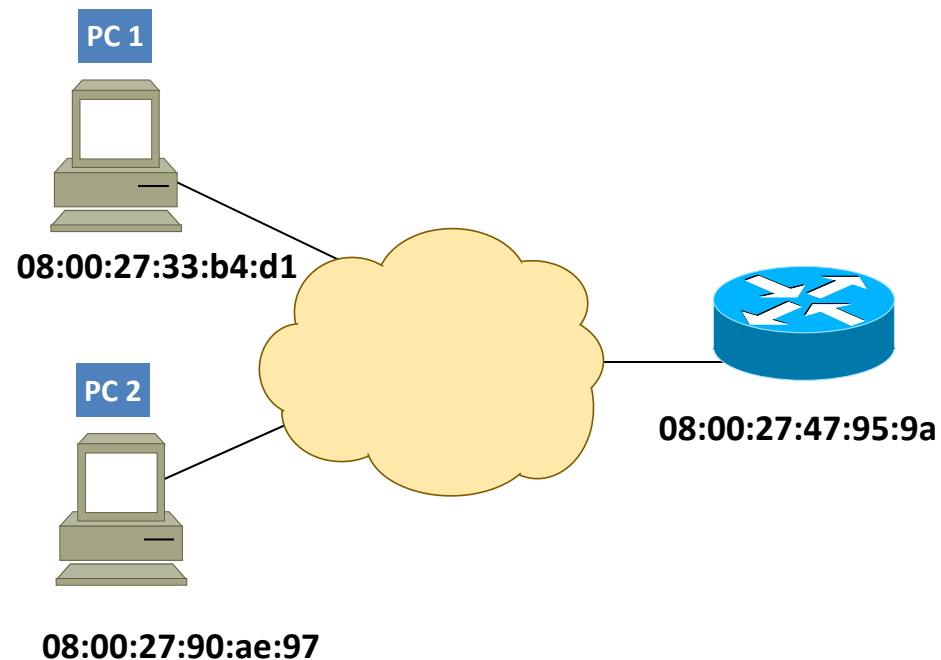
Ethernet II, Src: 08:00:27:90:ae:97, Dst: 33:33:ff:90:ae:97
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 24
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: ::
Destination: ff02::1:ff90:ae97 → SNA de 2001:470:1f13:cf4:a00:27ff:fe90:ae97 (global)
Internet Control Message Protocol v6
Type: 135 (Neighbor solicitation)
Code: 0
Checksum: 0xedd7 [correct]
Target: 2001:470:1f13:cf4:a00:27ff:fe90:ae97 ←

3.4.3 Casos posibles: igual que en el anterior DAD. Si todo OK →

eth0 Link encap:Ethernet HWaddr 08:00:27:90:AE:97
inet6 addr: 2001:470:1f13:cf4:a00:27ff:fe90:ae97/64 Scope:Global
inet6 addr: fe80::a00:27ff:fe90:ae97/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Ejemplo de comunicación

- Escenario



Ejemplo de comunicación

Escenario de comunicación

PC 1

PC 2

Envío ping 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 → 2001:470:1f13:cf4:a00:27ff:fe90:ae97

PC 1

-Determinar el siguiente salto: Consulta Destination Cache.

-Si no está: Next-Hop Determination

-Longest Prefix match en la Prefix List (determina si destino on/off link)

on-link → siguiente salto = dirección destino

off-link → siguiente salto = router por defecto (Default Router List)

-Crea una entrada en la Destination Cache (on-link)

2001:470:1f13:cf4:a00:27ff:fe90:ae97 via 2001:470:1f13:cf4:a00:27ff:fe90:ae97 dev eth0
metric 0 cache expires 0sec mtu 1500 advmss 1440 hoplimit 64

-Resolución de direcciones (sig.salto): Consulta Neighbor Cache

-Si no existe:

-Crea entrada:

2001:470:1f13:cf4:a00:27ff:fe90:ae97 dev eth0 INCOMPLETE (*)

-Realiza Address Resolution: envío de Neighbor Solicitation (NS)

Ejemplo de comunicación

- Realiza **Address Resolution:** envío de *Neighbor Solicitation (NS)*

Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 Destination: ff02::1:ff90:ae97 Protocol Info: ICMPv6 Neighbor solicitation

Ethernet II, Src: 08:00:27:33:b4:d1, Dst: 33:33:ff:90:ae:97
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 32
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1
Destination: ff02::1:ff90:ae97 → **Solicited Node Address** (de 2001:470:1f13:cf4:a00:27ff:fe90:ae97)

Internet Control Message Protocol v6
Type: 135 (Neighbor solicitation)
Code: 0
Checksum: 0x8155 [correct]
Target: **2001:470:1f13:cf4:a00:27ff:fe90:ae97** ←
ICMPv6 Option (Source link-layer address)
 Type: Source link-layer address (1)
 Length: 8
 Link-layer address: 08:00:27:33:b4:d1

- El host que lo recibe actualiza su *Neighbor Cache* (RFC 4861 sec. 7.2.3)

2001:470:1f13:cf4:a00:27ff:fe33:b4d1 dev eth0 lladdr 08:00:27:33:b4:d1
notrouter **STALE**

(paso de STALE a REACHABLE RFC 4861 sección 7.3.3)



Ejemplo de comunicación

- Realiza **Address Resolution**: Recibe *Neighbor Advertisement* (NA)
(RFC 4561 Sección 7.2.5)

Source 2001:470:1f13:cf4:a00:27ff:fe90:ae97	Destination 2001:470:1f13:cf4:a00:27ff:fe33:b4d1	Protocol Info ICMPv6 Neighbor advertisement
--	---	--

Ethernet II, Src: 08:00:27:90:ae:97 (08:00:27:90:ae:97), Dst: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1)
Type: IPv6 (0x86dd)
Internet Protocol Version 6
Payload length: 32
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: 2001:470:1f13:cf4:a00:27ff:fe90:ae97
Destination: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1
Internet Control Message Protocol v6
Type: 136 (Neighbor advertisement)
Code: 0
Checksum: 0xa2be [correct]
Flags: 0x60000000
0... .. = Not router
.1... .. = Solicited
..1... .. = Override
Target: 2001:470:1f13:cf4:a00:27ff:fe90:ae97
ICMPv6 Option (Target link-layer address)
Type: Target link-layer address (2)
Length: 8
Link-layer address: 08:00:27:90:ae:97

- Completa la entrada en la *Neighbor Cache* (el NA era solicited)

PC 1



2001:470:1f13:cf4:a00:27ff:fe90:ae97 dev eth0 lladdr 08:00:27:90:ae:97 notrouter **REACHABLE (*)**

Ejemplo de comunicación

- Escenario de comunicación

Envío ping 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 → 2001:470:1f13:cf4:a00:27ff:fe90:ae97

Source	Destination	Protocol Info
2001:470:1f13:cf4:a00:27ff:fe33:b4d1	2001:470:1f13:cf4:a00:27ff:fe90:ae97	ICMPv6 Echo request

Ethernet II, Src: 08:00:27:33:b4:d1 (08:00:27:33:b4:d1), Dst: 08:00:27:90:ae:97 (08:00:27:90:ae:97)
Type: IPv6 (0x86dd)
Internet Protocol Version 6
0110 = Version: 6
[0110 = This field makes the filter "ip.version == 6" possible: 6]
.... 0000 0000 = Traffic class: 0x00000000
.... 0000 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
Payload length: 64
Next header: ICMPv6 (0x3a)
Hop limit: 64
Source: 2001:470:1f13:cf4:a00:27ff:fe33:b4d1 (2001:470:1f13:cf4:a00:27ff:fe33:b4d1)
Destination: 2001:470:1f13:cf4:a00:27ff:fe90:ae97 (2001:470:1f13:cf4:a00:27ff:fe90:ae97)
Internet Control Message Protocol v6
Type: 128 (Echo request)
Code: 0
Checksum: 0xc48c [correct]
ID: 0x3c13
Sequence: 0x0001
Data (56 bytes)

Contenidos

- Introducción a IPv6
- Coexistencia / Transición IPv4-IPv6
- Direccionamiento
- PDU. Cabeceras de extensión
- Autoconfiguración
- Funciones de control
- Encaminamiento

Encaminamiento

2000::/3 actualmente asignable, pero teóricamente, sin limitación futura en el prefijo global

- Similar a CIDR (pero @s de 128 bits):
unicast globales agregables (2000::/3)
 - **Completamente jerárquico** (las direcciones dependen estrictamente de la topología de la red)
 - Prefijo global: Asignación por parte de RIRs a los ISPs
 - Prefijo de sitio: División en subredes (administrador del sitio)
 - Identificador de interfaz: Identificador único asignado a cada interfaz conectado a Internet.
- Cabecera de extensión para encaminamiento (R2, en movilidad)
- **Pequeñas modificaciones sobre los protocolos** de encaminamiento dinámicos
 - Adaptación al nuevo formato de direcciones
 - Encaminamiento integrado (soporte simultáneo para IPv4 e IPv6)
 - Modificaciones sobre RIP, OSPF y BGP:
 - RIPng (**RFC 2080**), OSPF para IPv6 (**RFC 5340**), BGP4+ (**RFC 4760**)

RFC 3587