

Máster en Ingeniería Informática

Redes de Nueva Generación

Profesor:

Dr. Juan Carlos Fabero Jiménez (UCM)



Contenidos

- Tema 0: Entorno de trabajo. User-Mode-Linux (UML)
- Tema 1: IP de nueva generación: IPv6
- Tema 2: Encaminamiento en Sistemas Autónomos. OSPFv2 y OSPFv3
- Tema 3: Encaminamiento entre Sistemas Autónomos: BGPv4
- Tema 4: Encaminamiento Troncal: MPLS
- Tema 5: Redes Definidas por Software: SDN
- Tema 6: Multicast

IPv6

- **¿Por qué IPv6?**
- **Direccionamiento IPv6**
- **Datagrama IPv6**
- **ICMPv6**
 - **Descubrimiento de vecinos**
 - **Autoconfiguración**
- **Mecanismos de Transición**
 - **Dual Stack**
 - **Túneles**
 - **Túneles automáticos**
 - **Túneles configurados**
 - **Traducción de cabeceras**

IP de nueva generación: IPv6



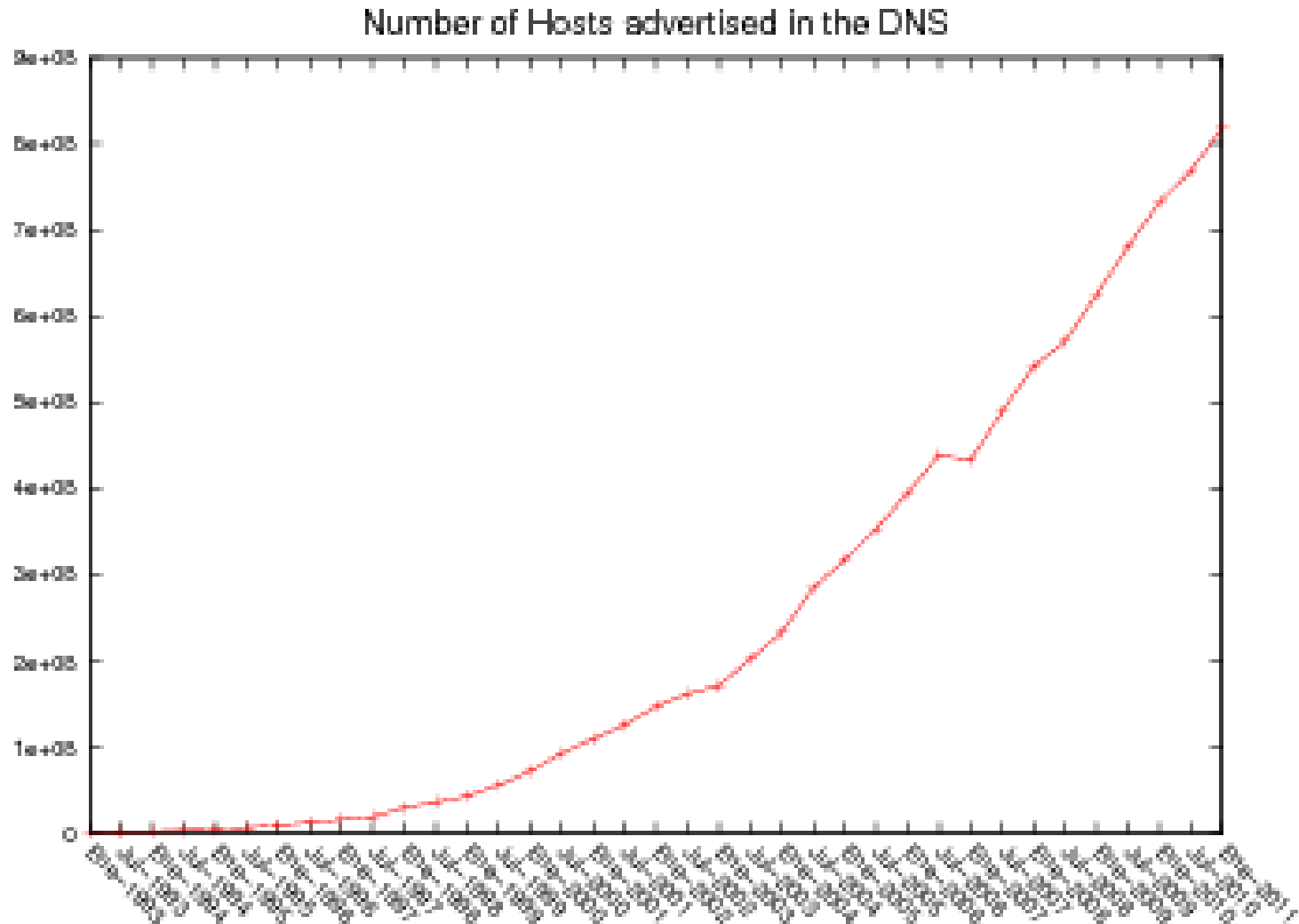
Introducción

¿Por qué IPv6?

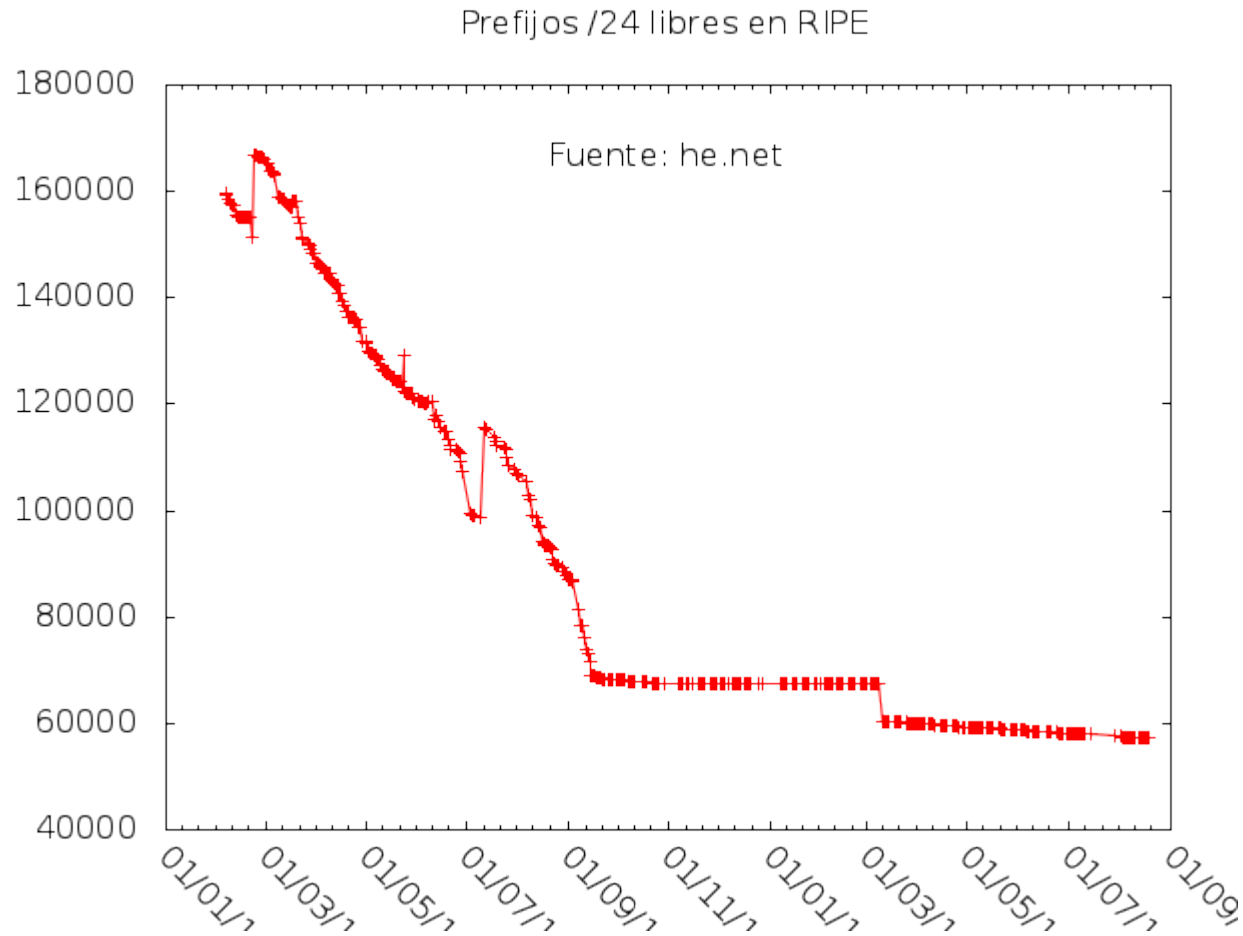
■ Limitaciones de IPv4:

- Direcciones de 32 bits:
 - Inconveniente: pocas direcciones
 - Solución: Utilizar intranets con direcciones privadas y NAT
- Organización en clases:
 - Inconveniente: se desperdician muchas direcciones
 - Solución: CIDR
- Formato de datagrama:
 - Longitud de cabecera variable
 - Fragmentación en los encaminadores
- Seguridad:
 - No tiene prevista ninguna opción de seguridad. Solución: IPSec.
- Multicast:
 - Opcional en IPv4. No se ha llegado a utilizar de manera eficaz.

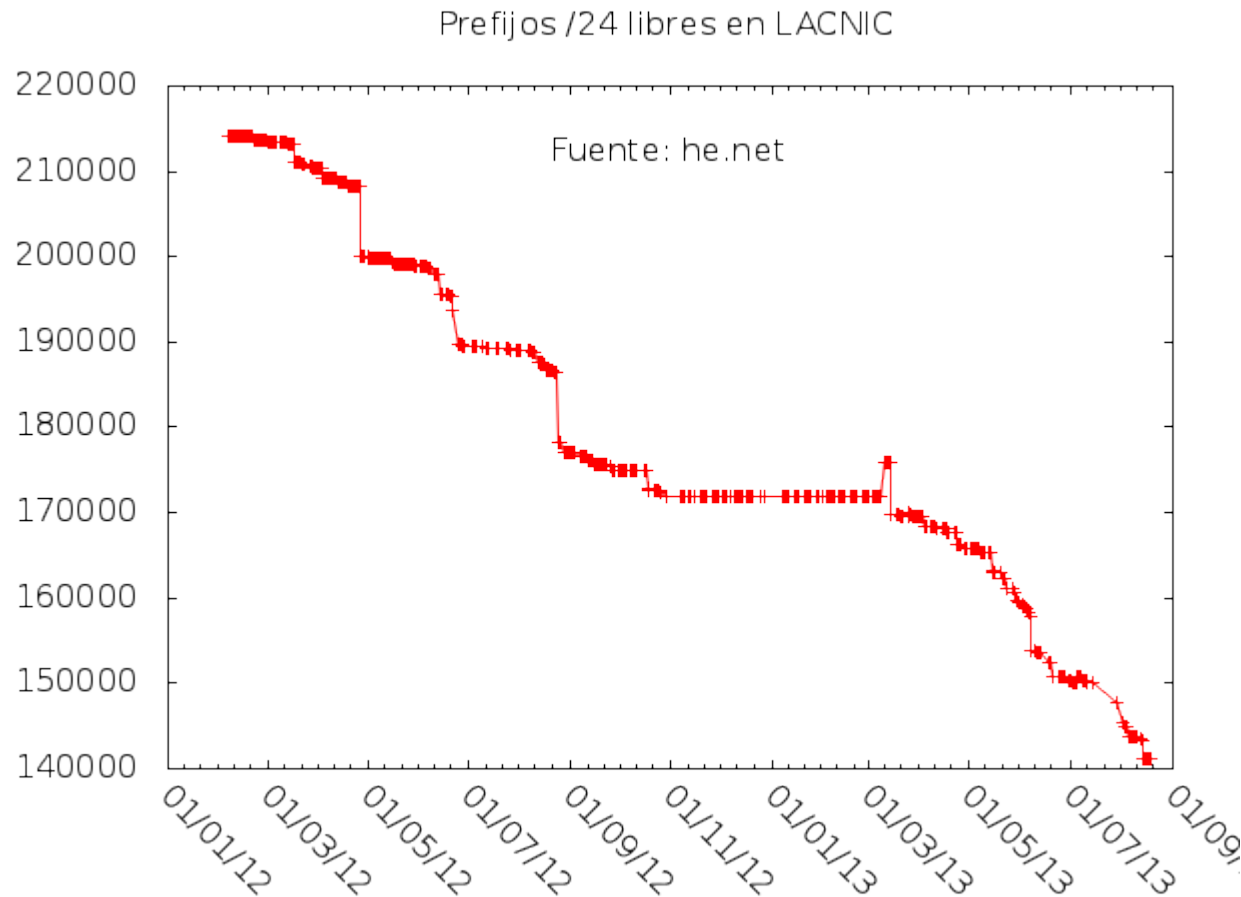
¿Por qué IPv6?



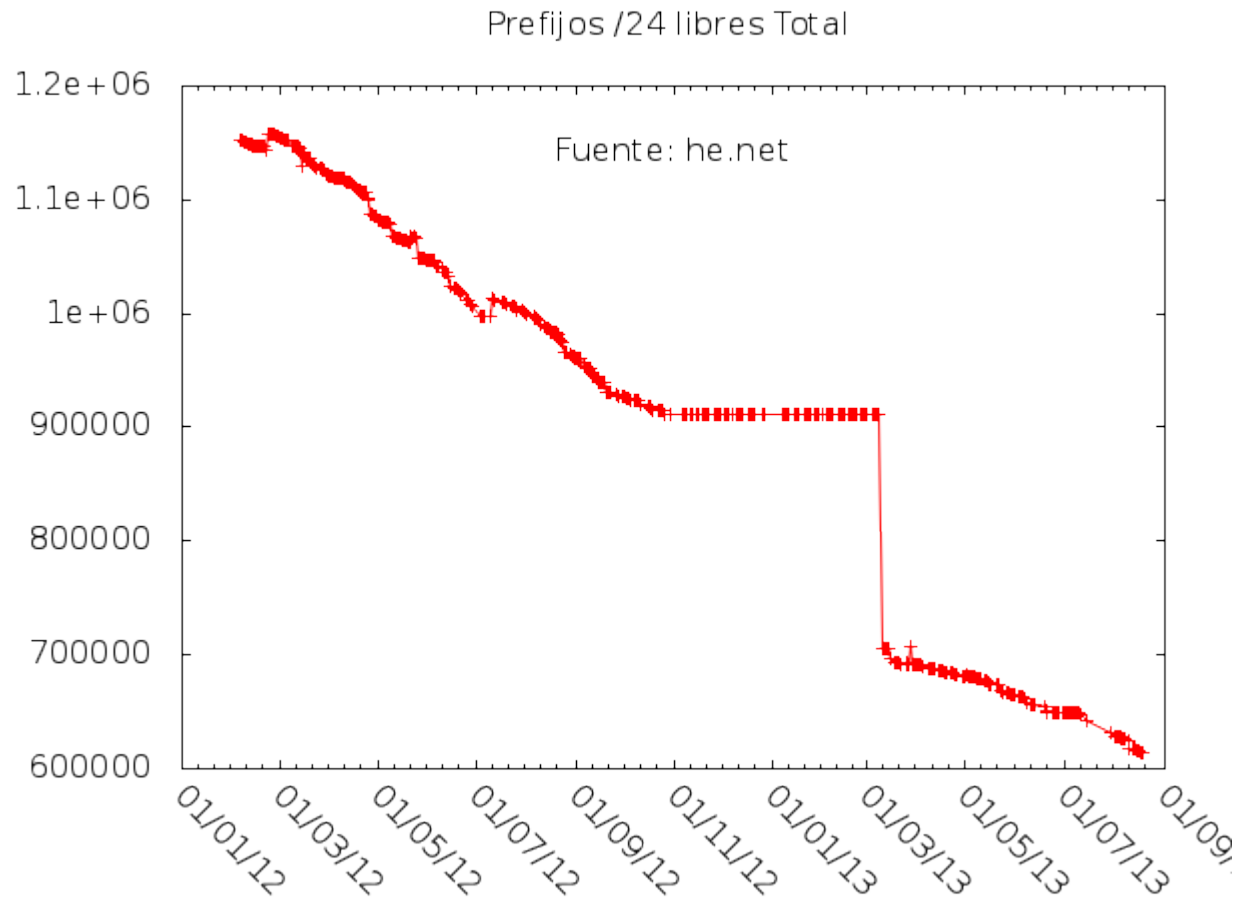
¿Por qué IPv6?



¿Por qué IPv6?



¿Por qué IPv6?



¿Por qué IPv6?

■ La solución: IPv6

- Direcciones: 128 bits (más de $3.4 * 10^{38}$ direcciones)
- Formato simplificado de cabecera (mayor velocidad de procesamiento)
- Encaminamiento jerárquico
- Autoconfiguración de los interfaces de red (*plug-and-play*)
- Soporte para tráfico en tiempo real (VoIP, Vídeo bajo demanda...)
- Opciones de seguridad

IP de nueva generación: IPv6



Direccionamiento IPv6

Direccionamiento IPv6

■ Tipos de direcciones.

- IPv6 soporta los siguientes tipos de direcciones:
 - **Unicast:** se refieren a una sola interfaz en Internet. Un datagrama dirigido a una dirección unicast se entrega sólo a la interfaz con esa dirección.
 - **Multicast:** identifican a un grupo de interfaces. Un datagrama dirigido a una dirección multicast se entrega a todas las interfaces que tienen esa dirección.
 - **Anycast:** identifican a un grupo de interfaces. Un datagrama dirigido a una dirección anycast se entrega sólo a la interfaz más cercana con esa dirección.
 - No hay direcciones de **broadcast**.

Direccionamiento IPv6

■ Representación de las direcciones

- Las direcciones se representan por 8 grupos de 16 bits cada uno expresados con caracteres hexadecimales. Los grupos están separados por el carácter ":".
- Ejemplos:
 - FEC0:BAC8:934F:0234:5678:12AB:CF23:0987
 - 2001:0DB8:0000:0000:0000:0001:0000:0056
- Los 0 al comienzo de un campo se pueden omitir:
 - FEC0:BAC8:934F:234:5678:12AB:CF23:987
 - 2001:db8:0:0:0:1:0:56
- Uno o varios grupos :0: contiguos se pueden resumir por ::
 - 2001:db8::1:0:56
- Sólo puede aparecer una vez ::
 - 2001:db8::1::56 → es ambigua

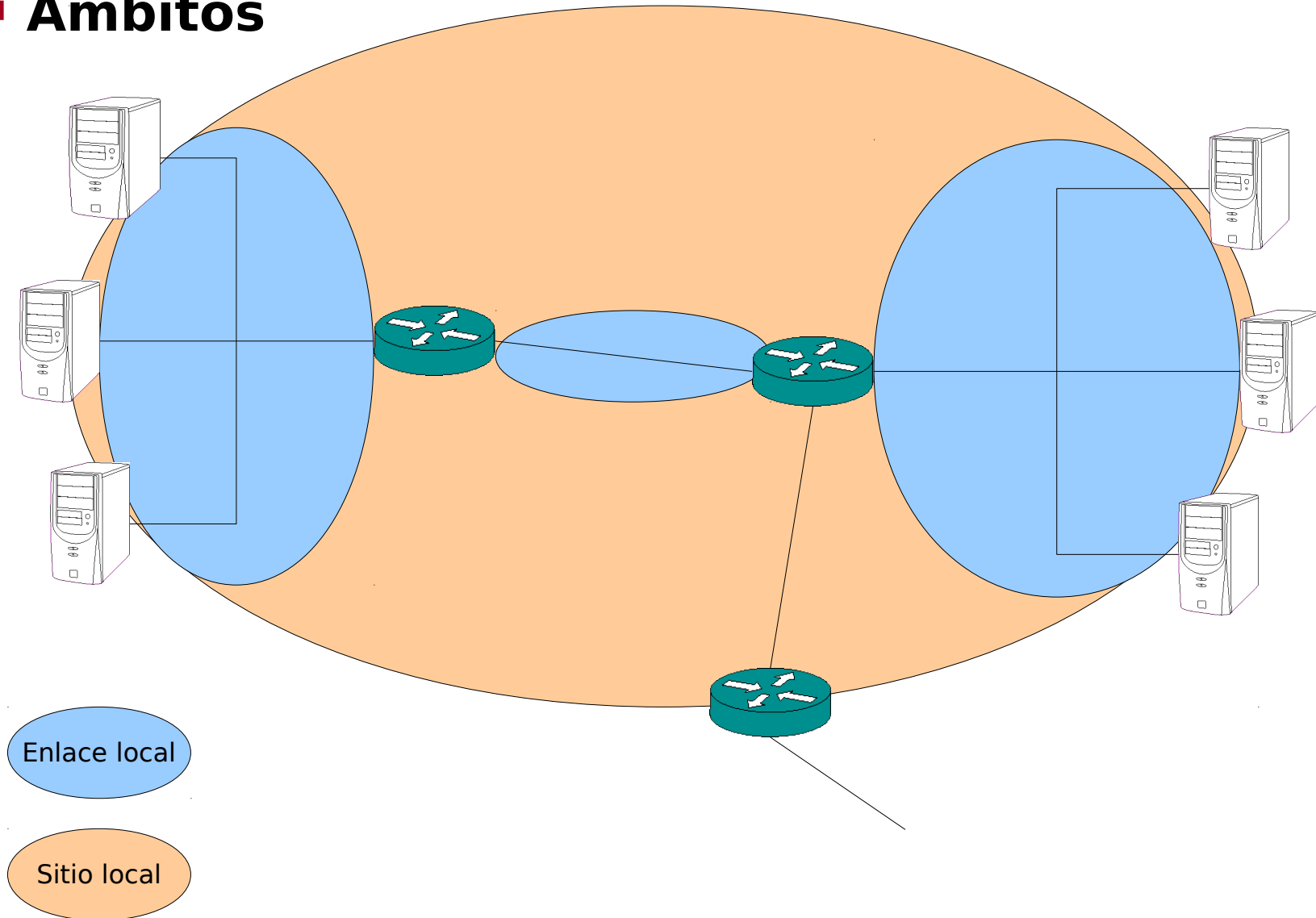
Direccionamiento IPv6

■ Ámbitos

- El ámbito de una dirección define dónde es válida dicha dirección, es decir, dónde puede ser utilizada como identificador único de un interfaz (*unicast*) o grupo de interfaces (*multicast*).
- Las direcciones *unicast* tienen definidos 3 ámbitos:
 - Enlace local: la dirección sólo es válida dentro del enlace donde está conectado el interfaz de red (p.e. una red Ethernet).
 - Sitio local: la dirección sólo es válida dentro del “sitio” o red de la organización (p.e. un campus universitario).
 - Global: la dirección es válida en toda Internet.
- Las direcciones *multicast* definen su ámbito mediante un campo de 4 bits:
 - 2 = enlace local
 - 5 = sitio local
 - 8 = organización local
 - E = global

Direccionamiento IPv6

■ Ámbitos



Direccionamiento IPv6

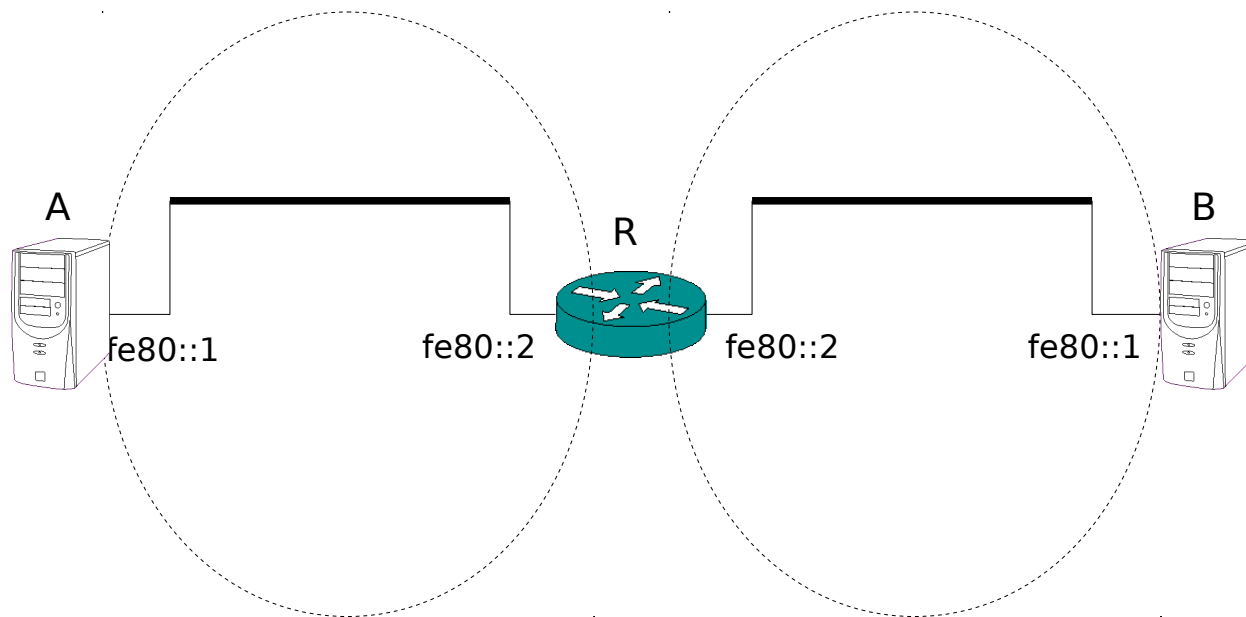
■ Zonas de ámbito

- Una región conexas de determinado ámbito se denomina *zona de ámbito*.
 - Ejemplo: una zona de enlace local está formada por un enlace y todos los interfaces conectados a él.
- Cada dirección IPv6 corresponde con una y sólo una zona, según el ámbito de la dirección.
- La unicidad de una dirección sólo se garantiza en su zona.
- Un datagrama con direcciones origen y/o destino con determinado ámbito nunca se redirige a una zona distinta de donde fue originado.
 - Ejemplo: un datagrama con dirección origen fe80::244:2ff:fe23:4 (enlace local) y dirigido a la dirección fec0:1::2 (sitio local) no será reenviado, sino descartado.
- Las zonas pertenecientes a distintos ámbitos están fuertemente ordenadas.
 - Si un interfaz *i* pertenece a las zonas **X** e **Y**, con el ámbito de **Y** mayor que el de **X**, entonces $X \subset Y$

Direccionamiento IPv6

■ Identificadores de zonas

- Una dirección con ámbito inferior a global es ambigua en los límites de zona.
 - Ejemplo:
 - Las máquinas A y B pueden tener la misma dirección, pues pertenecen a zonas distintas.
 - Incluso el encaminador R puede tener la misma dirección en todos sus interfaces.
 - En R, no basta con especificar `fe80::1` para saber a qué máquina nos referimos.

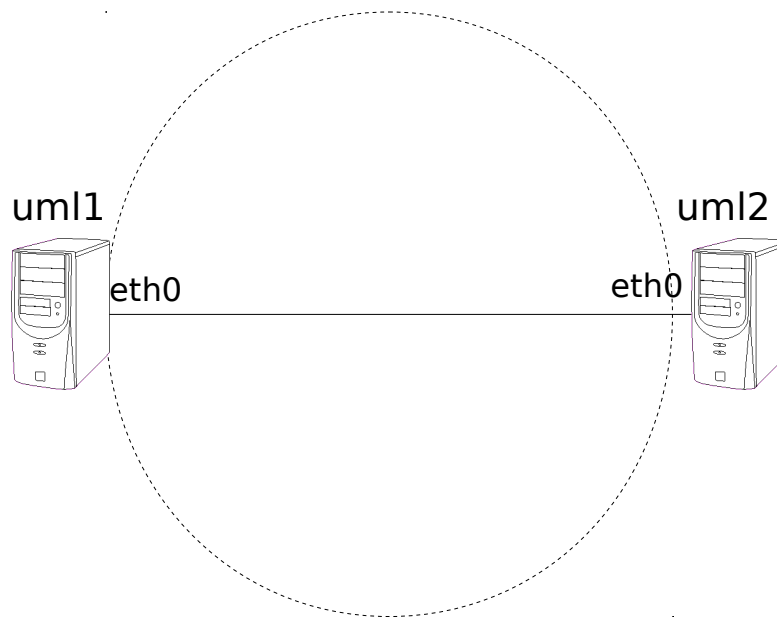


Direccionamiento IPv6

■ Identificadores de zonas

■ Ejercicio:

- Configurar dos máquinas virtuales como en la figura.
- Levantar los interfaces eth0 de ambas, e intentar efectuar un ping6 entre ambas, empleando las direcciones de enlace local (fe80::/10).



Direccionamiento IPv6

■ Estructura de las direcciones

- IPv4 sólo permite un nivel jerárquico: netid y hostid.
 - ¡Son necesarias más de 2 millones de entradas para las redes de clase C!
- Para facilitar las tareas de encaminamiento, IPv6 permite más niveles de jerarquía.
 - Es lo que se conoce como *agregado de direcciones*.
- Cada dirección IPv6 comienza por un prefijo que indica qué tipo de dirección es (prefijo de formato):

Tipo de dirección	FP (binario)	FP (hexadecimal)
Reservada	0000 0000	00
Unicast Global Agregable	001	2 ó 3
Unicast Enlace Local	1111 1110 10	FE8
Unicast Sitio Local	1111 1110 11	FEC
ULA	1111 1101	FD
Multicast	1111 1111	FF

Direccionamiento IPv6

■ Direcciones Unicast Globales Agregables

- Son las que utiliza una máquina conectada a Internet.
- Permiten la autoconfiguración:
 - Una máquina puede obtener el prefijo de red desde el encaminador de su red.
 - El identificador de interfaz se puede construir a partir de la dirección MAC, o bien se puede fijar en la configuración de la máquina.

3	13	8	24	16	64
001	TLA	Res	NLA	SLA	Interfaz

TLA: Top-Level Aggregation.

Res: Reservado; permitirá ampliar TLA o NLA en el futuro.

NLA: Next-Level Aggregation.

SLA: Site-Level Aggregation.

(RFC 2374)

Direccionamiento IPv6

■ Direcciones Unicast Globales Agregables

- El RFC 3587 (2003) reemplaza al RFC 2374 y define una nueva estructura para las direcciones unicast globales agregables.



- Se recomienda asignar a las organizaciones o usuarios finales un prefijo /48.
 - Cada organización dispone, entonces, de 65536 subredes.
- El prefijo de formato 2000::/3 ya no es tal, aunque de momento todas las direcciones asignadas por IANA empiezan así. Esto puede cambiar en el futuro, si fuese necesario.

Direccionamiento IPv6

■ Direcciones Unicast de Enlace Local

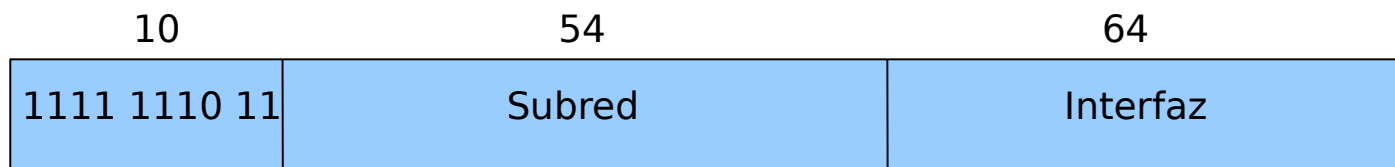
- Son direcciones privadas que pueden utilizarse en intranets no jerárquicas (planas).
- Nunca se encaminan hacia el exterior (pertenecen a una zona de enlace local).
- Permiten realizar las funciones de “*descubrimiento de vecino*”.
- En GNU/Linux todo interfaz de red se autoconfigura con una dirección unicast de enlace local.
- Ejemplo:
 - fe80::20f:b0ff:fea5:6e



Direccionamiento IPv6

■ Direcciones Unicast de Sitio Local

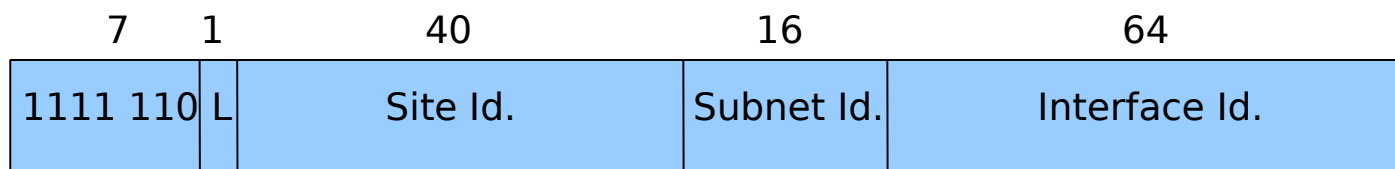
- Son direcciones privadas que pueden utilizarse en intranets jerárquicas.
- Nunca se encaminan hacia el exterior (pertenecen a una zona de sitio local).
- Están obsoletas.
- Ejemplos:
 - fec0:50:16::1
 - fec0:50:16:0:20f:b0ff:fea5:6e



Direccionamiento IPv6

■ Direcciones Unique Local IPv6 Unicast Addresses (ULA)

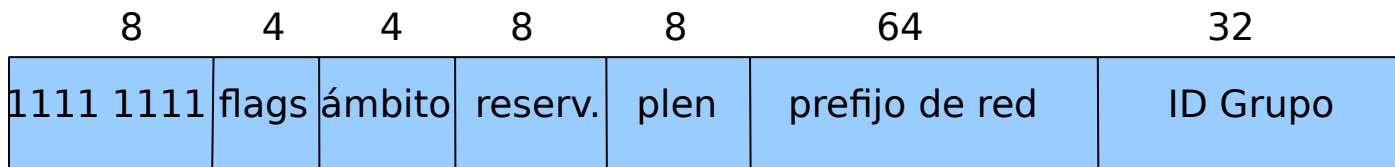
- Son direcciones privadas que pueden utilizarse en intranets jerárquicas.
- Nunca se encaminan hacia el exterior.
- Sustituyen a las direcciones únicas de sitio local (fec0::/10) (RFC 3879).
- Su prefijo de formato es fc00::/7, más un bit L que indica si la gestión es local o global. Este bit debe valer 1.
- Ejemplos:
 - fd00:50:16::1 (no es buena práctica, demasiado previsible y propenso a colisiones cuando se unen varios dominios)
 - fd12:761f:e8e1:28ea:20f:b0ff:fea5:6e



Direccionamiento IPv6

■ Direcciones Multicast

- Son direcciones asignadas a grupos de máquinas.
- Se caracterizan por su **ámbito**:
 - Nodo local (0001) (0x1)
 - Enlace local (0010) (0x2)
 - Sitio local (0101) (0x5)
 - Organización (1000) (0x8)
 - Global (1110) (0xE)



T=0, permanente; T=1, temporal

P=0, no basada en prefijo; P=1, basada en prefijo

R=0, no incluye RP; R=1, dirección de RP incluida

Direccionamiento IPv6

■ Direcciones Multicast

- El protocolo de descubrimiento de vecino tiene direcciones reservadas:
 - Desde FF02::1:FF00:0000 hasta FF02::1:FFFF:FFFF
 - Ejemplo:
 - Se necesita averiguar la dirección MAC asociada con la dirección IPv6 2001::1:800:200E:8C6C
 - Se envía un mensaje ICMP Neighbour Discovery a la dirección FF02::1:FF0E:8C6C
- Direcciones multicast de los encaminadores:
 - FF01::2 → encaminadores del nodo local.
 - FF02::2 → encaminadores del enlace local.
 - FF05::2 → encaminadores del sitio local.
 - FF02::9 → encaminadores RIP del enlace local.
- Direcciones multicast de los computadores:
 - FF01::1 → computador del nodo local (todos los interfaces del nodo local).
 - FF02::1 → computadores del enlace local.
 - Ejemplo: ping6 -c 2 ff02::1%eth0

Direccionamiento IPv6

■ Direcciones multicast

- Los identificadores de grupo se dividen en tres rangos:
 - 0x00000001 a 0x3fffffff: identificadores para direcciones multicast permanentes (“bien conocidas”)
 - 0x40000000 a 0x7fffffff: identificadores para direcciones multicast permanentes (“bien conocidas”) basadas en prefijos de red.
 - 0x80000000 a 0xffffffff: identificadores para direcciones multicast dinámicas.
- Ejemplos:
 - ff02:0:0:0:0:0:0:101 → Servidores NTP en un enlace.
 - ff35:40:2001:db8:1111:0:4040:4040 → grupo de servidores en un “sitio” con prefijo de red 2001:db8:1111::/64

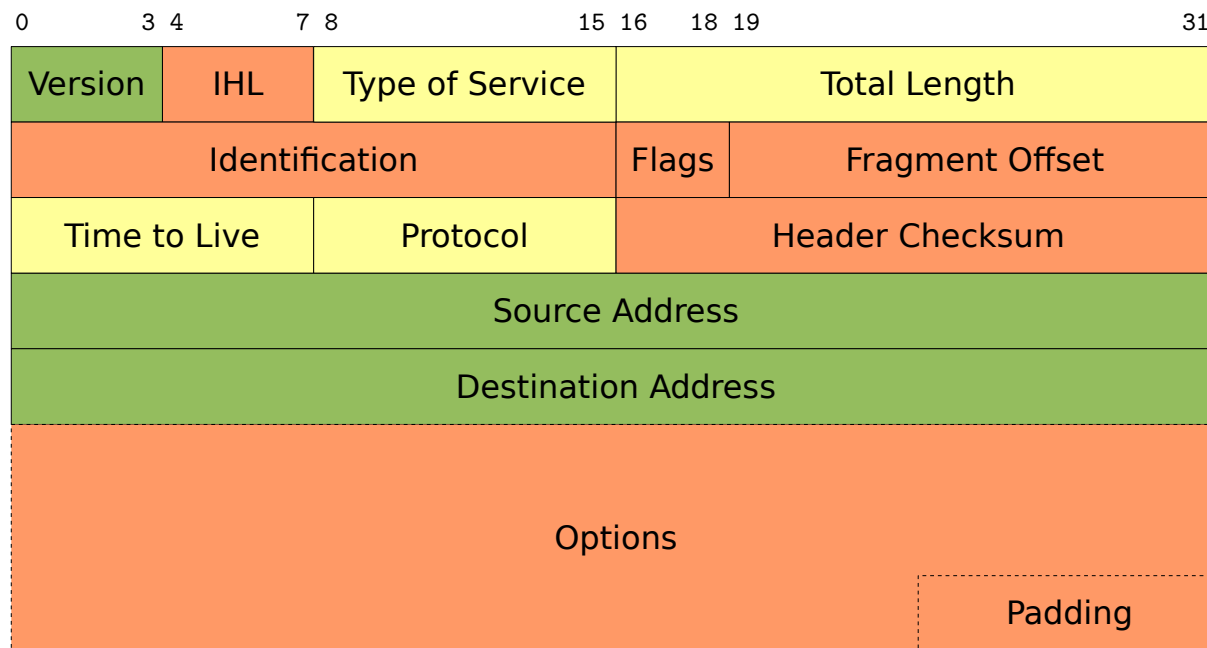
IP de nueva generación: IPv6



Datagrama IPv6

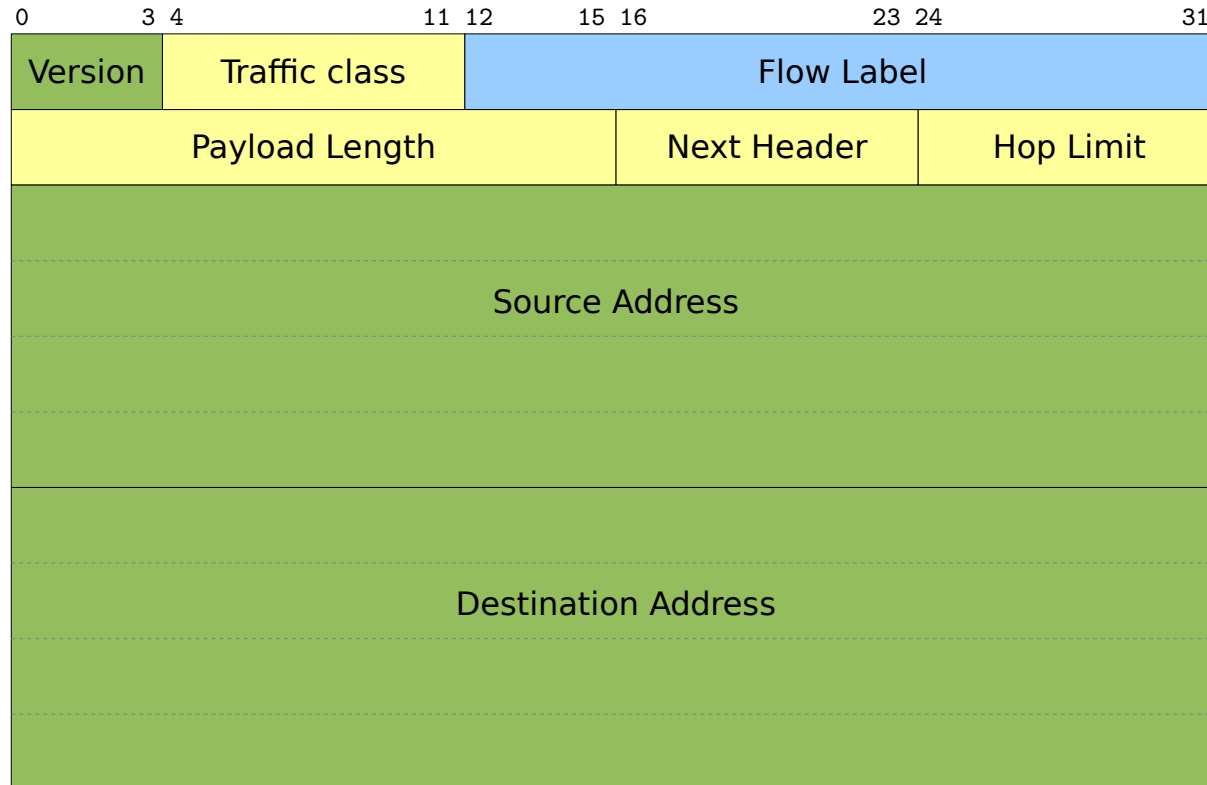
IPv6 vs. IPv4

■ Formato de la cabecera IPv4



IPv6 vs. IPv4

■ Formato de la cabecera IPv6



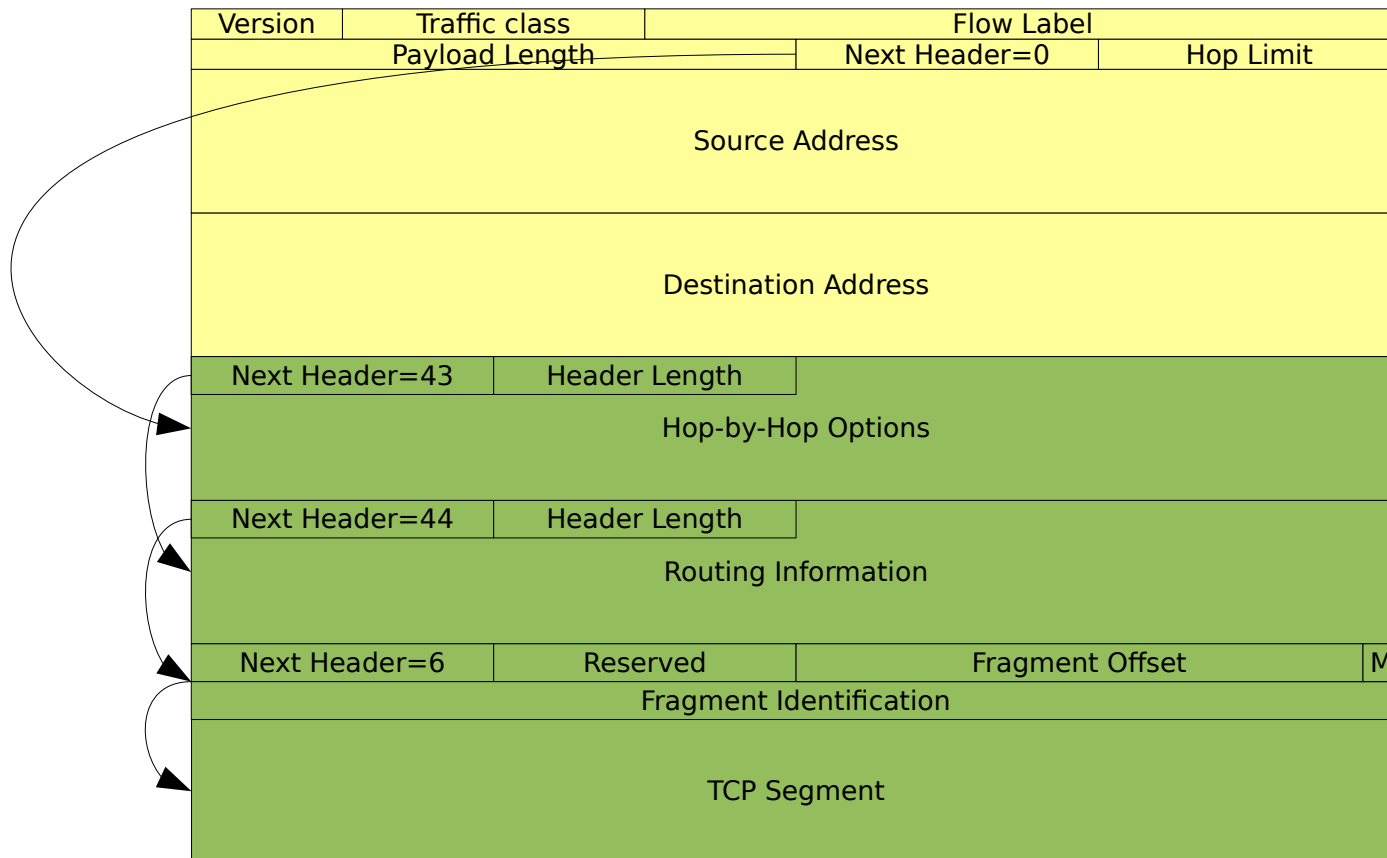
IPv6 vs. IPv4

■ Formato de cabecera IPv6

- **Version**: distingue entre las versiones 4 y 6.
- **Traffic Class**: clase de tráfico. 0-7: tráfico normal; 8-15: tráfico en tiempo real.
- **Flow Label**: 0 para tráfico normal; permite a los encaminadores distinguir entre datagramas que pertenecen al mismo flujo de datos.
- **Payload Length**: longitud del datagrama, excluyendo esta cabecera.
- **Next Header**: indica el tipo de la siguiente cabecera de extensión (si la hay), o el protocolo de capa superior en el campo de datos.
- **Hop Limit**: similar al TTL de IPv4, pero ahora se mide en saltos, no en segundos.
- **Source & Destination Addresses**: 128 bits cada una.

Datagrama IPv6

■ Cabeceras de extensión



Datagrama IPv6

■ Cabeceras de extensión

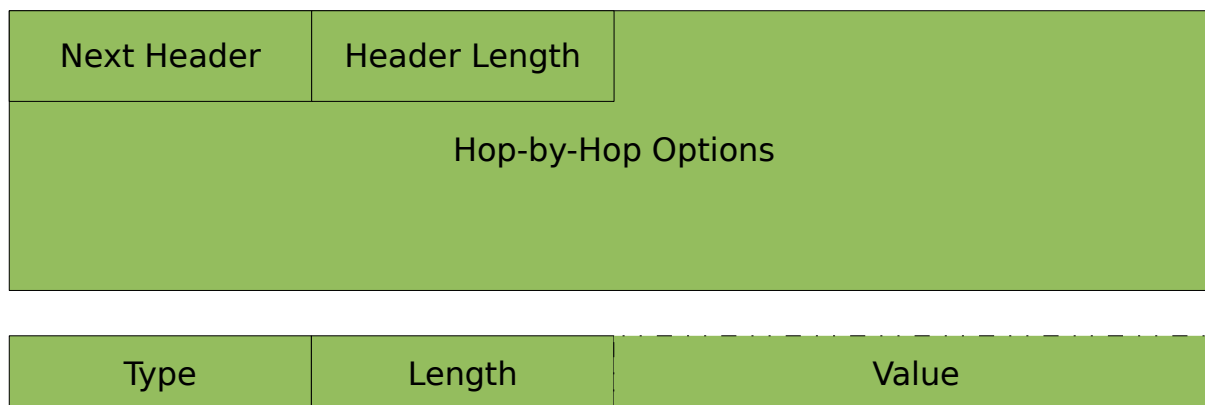
- La cabecera IPv6 tiene siempre la misma longitud. En el campo Next Header se especifica cómo interpretar los datos que siguen.
 - 41: IPv6 Header
 - 45: Interdomain Routing Protocol
 - 46: Resource Reservation Protocol
 - 58: ICMPv6 Packet
 - 0: Hop-by-hop Options Header
 - 43: IPv6 Routing Header
 - 44: IPv6 Fragment Header
 - 50: Encapsulating Security Payload
 - 51: IPv6 Authentication Header
 - 59: No Next Header
 - 60: Destination Options Header

Cabeceras de extensión

Datagrama IPv6

■ Cabeceras de extensión

- Hop-by-hop (0): debe ser examinada por todos los nodos en la ruta, incluidos el origen y el destino.



Type: xxyzzzzz

xx: qué hacer cuando la opción no es reconocida:

0: ignorar la opción.

1: descartar el paquete discretamente.

2: descartar el paquete y enviar al origen un ICMPv6 “Unrecognized Type”.

3: como 2, salvo en el caso de que la dirección destino sea multicast.

y: si está activado, la opción puede cambiar en ruta.

zzzzz: 0 Pad1

1 PadN

5 Router Alert

194 Jumbo Payload Length

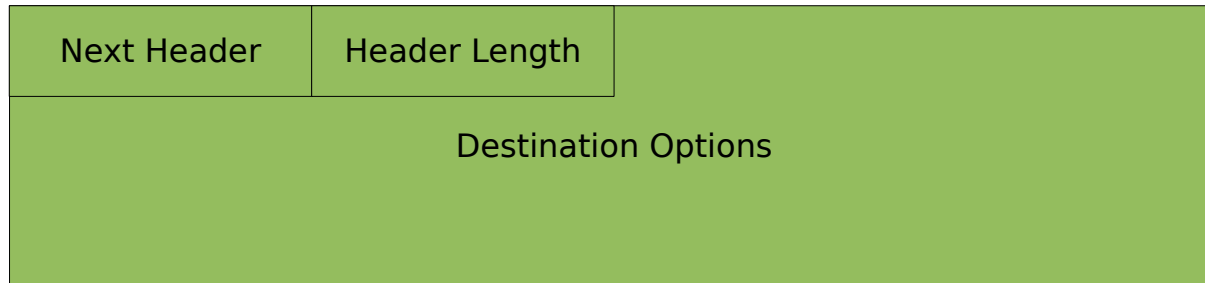
■ Cabeceras de extensión

- Hop-by-hop (0): debe ser examinada por todos los nodos en la ruta, incluido el destino.
 - Tipos soportados:
 - Pad1: se emplea para alinear opciones añadiendo un byte.
 - PadN: alinea opciones añadiendo N bytes.
 - Router Alert: el datagrama contiene datos relevantes para el encaminador (p.e. RSVP). Definido en RFC2711 (1999) y actualizado en RFC6398 (2011).
 - Jumbo Payload Length: cuando un paquete tiene una longitud superior a 65535 bytes, se utiliza esta cabecera para indicar la longitud total del paquete, excepto los 40 bytes de la cabecera estándar. En este caso, el campo Payload Length de la cabecera estándar debe valer 0.

Datagrama IPv6

■ Cabeceras de extensión

- Destination Option Header (60): debe ser examinada por el destino del datagrama. Sin embargo, si esta cabecera precede a la Routing Header, debe ser examinada por todos los nodos listados en la Routing Header.



Datagrama IPv6

■ Cabeceras de extensión

- Routing Header (43): lista una serie de nodos que deben ser atravesados en la ruta hacia el destino final.

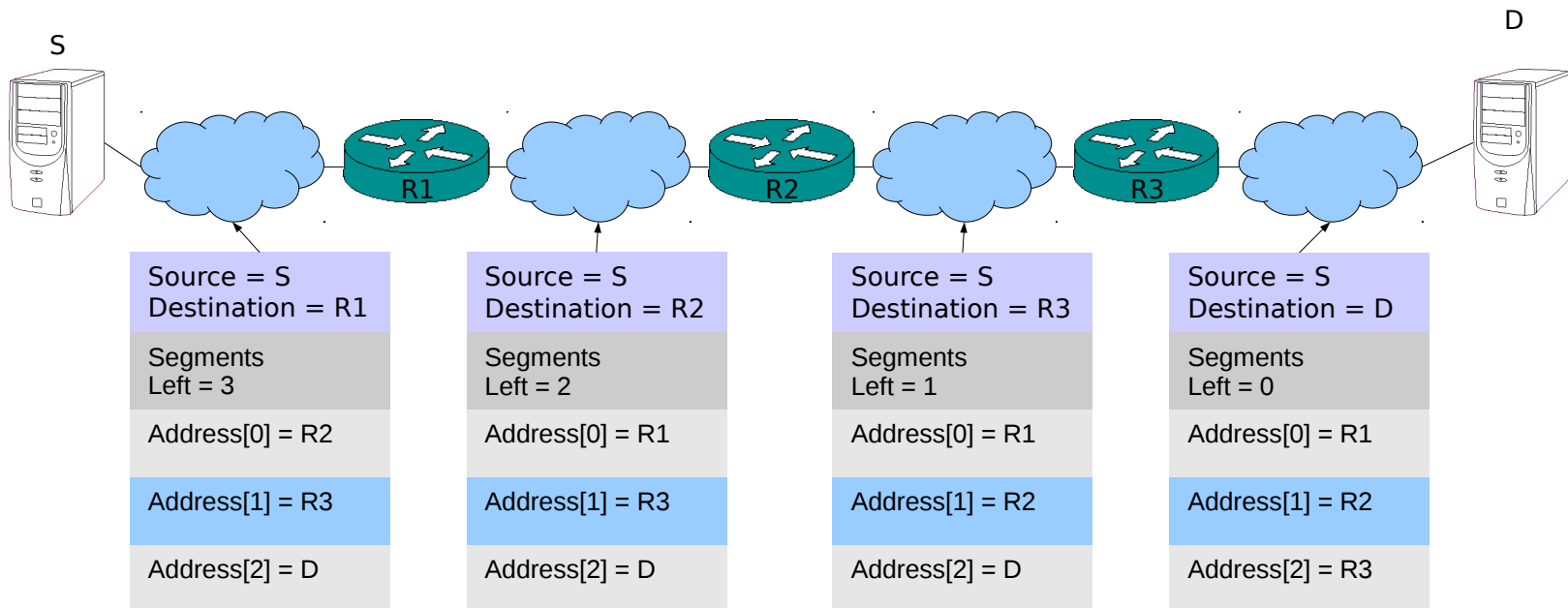
Next Header	Header Length	Type	Segments Left
Reserved			
Address[1]			
Address[2]			
...			
Address[n]			

Datagrama IPv6

■ Cabeceras de extensión

■ Routing Header (43)

- Routing Type: debe valer 0 (strict/loose routing).
- Segments Left: el número de nodos intermedios que faltan por visitar.
- Address[]: direcciones de los nodos intermedios.
- Se desaconseja su uso en RFC5095 (2007).
- Ejemplo:



Datagrama IPv6

■ Cabeceras de extensión

- Fragment Header (44): la fragmentación siempre se realiza en origen, nunca en encaminadores intermedios (Path MTU Discovery). Además, siempre $MTU \geq 1280$.

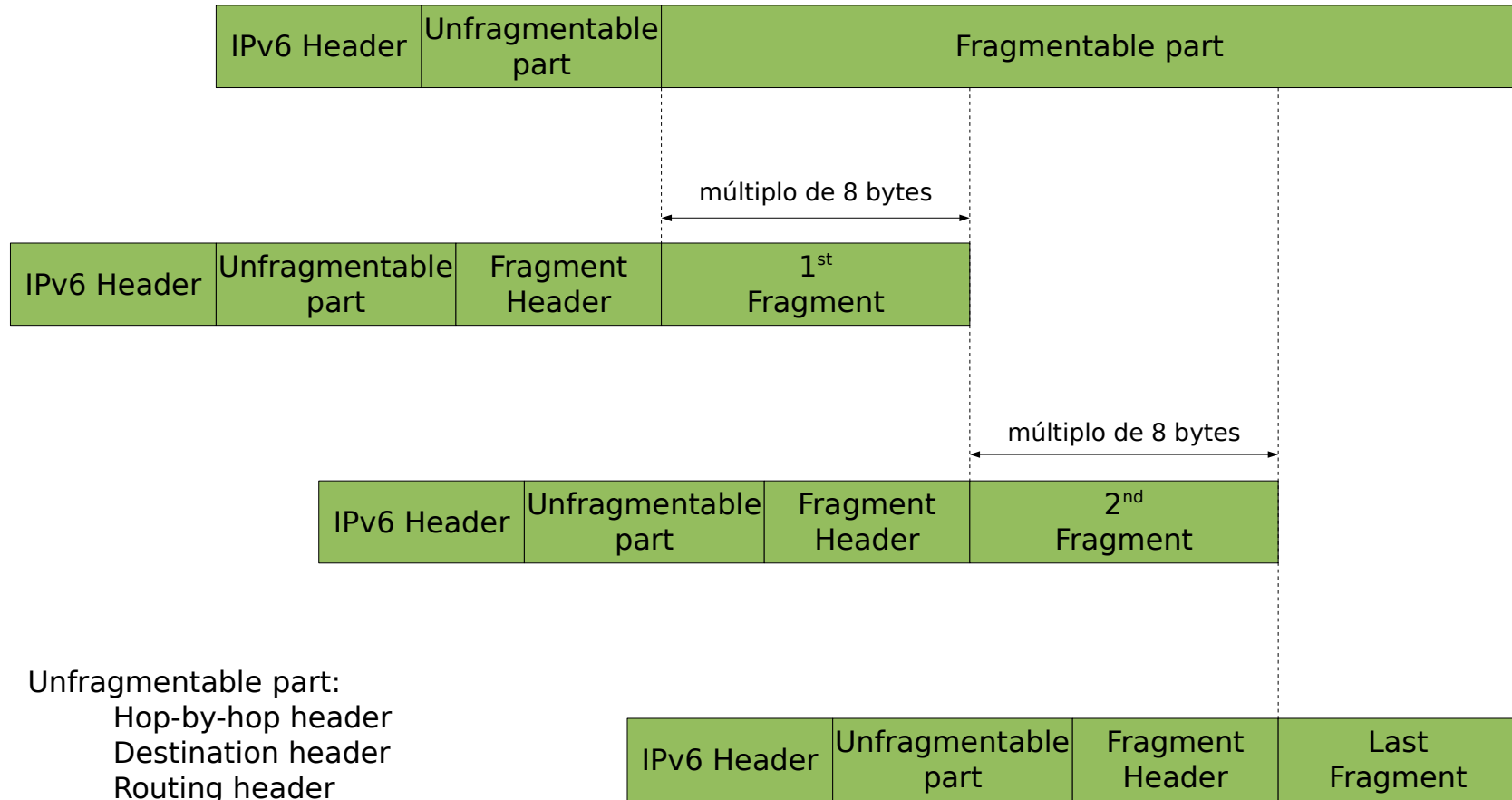
Next Header	Reserved	Fragment Offset	Res.	M
Identificación				

Reserved:	(8 bits)	debe valer 0.
Fragment Offset:	(13 bits)	indica el desplazamiento de los datos que siguen a la cabecera de fragmentación dentro del datagrama original.
Res:	(2 bits)	debe valer 0.
M:	(1 bit)	indica si hay más fragmentos o no.
Identification:	(32 bits)	permite al receptor identificar a los fragmentos pertenecientes al mismo datagrama.

Datagrama IPv6

■ Cabeceras de extensión

- Fragment Header (44): proceso de fragmentación.



Datagrama IPv6

■ Cabeceras de extensión

- Authentication Header (51): se emplea para garantizar que el datagrama no ha sido alterado en tránsito (integridad) y que procede del emisor que aparece en la dirección origen (autenticación).

Next Header	Payload Length	Reserved
Security Parameters Index (SPI)		
Sequence Number (SN)		
Integrity Check Value (ICV)		

Payload Length: (8 bits) especifica la longitud de AH, en múltiplos de 8 bytes, menos 2.

Reserved: (16 bits) debe valer 0.

SPI: (32 bits) identificador de la SA (Security Association).

SN: (32 bits) contador incremental.

ICV: (n*32 bits) resultado de aplicar el algoritmo de verificación de integridad elegido para la SA al datagrama.

IP de nueva generación: IPv6



The diagram illustrates the ICMPv6 protocol layer. It features a thick red horizontal bar at the top, followed by a thin red line. Below these is a large white rectangular box with a black border. The text 'ICMPv6' is centered within this box.

ICMPv6

■ ICMP asume nuevas funciones (RFC 4443 y 4884):

- Información sobre pertenencia a grupos *multicast*.
 - Era una tarea de IGMP.
- Descubrimiento de direcciones.
 - Sustituye al protocolo ARP.
- Autoconfiguración.
 - Permite descubrir encaminadores presentes en el segmento.
 - No es necesario DHCP (aunque existe DHCPv6).

ICMPv6

■ Formato de los mensajes

- Todos los mensajes ICMPv6 tienen un formato genérico como el mostrado:

Type	Code	Checksum
Cuerpo del mensaje ICMP		

IP de nueva generación: IPv6

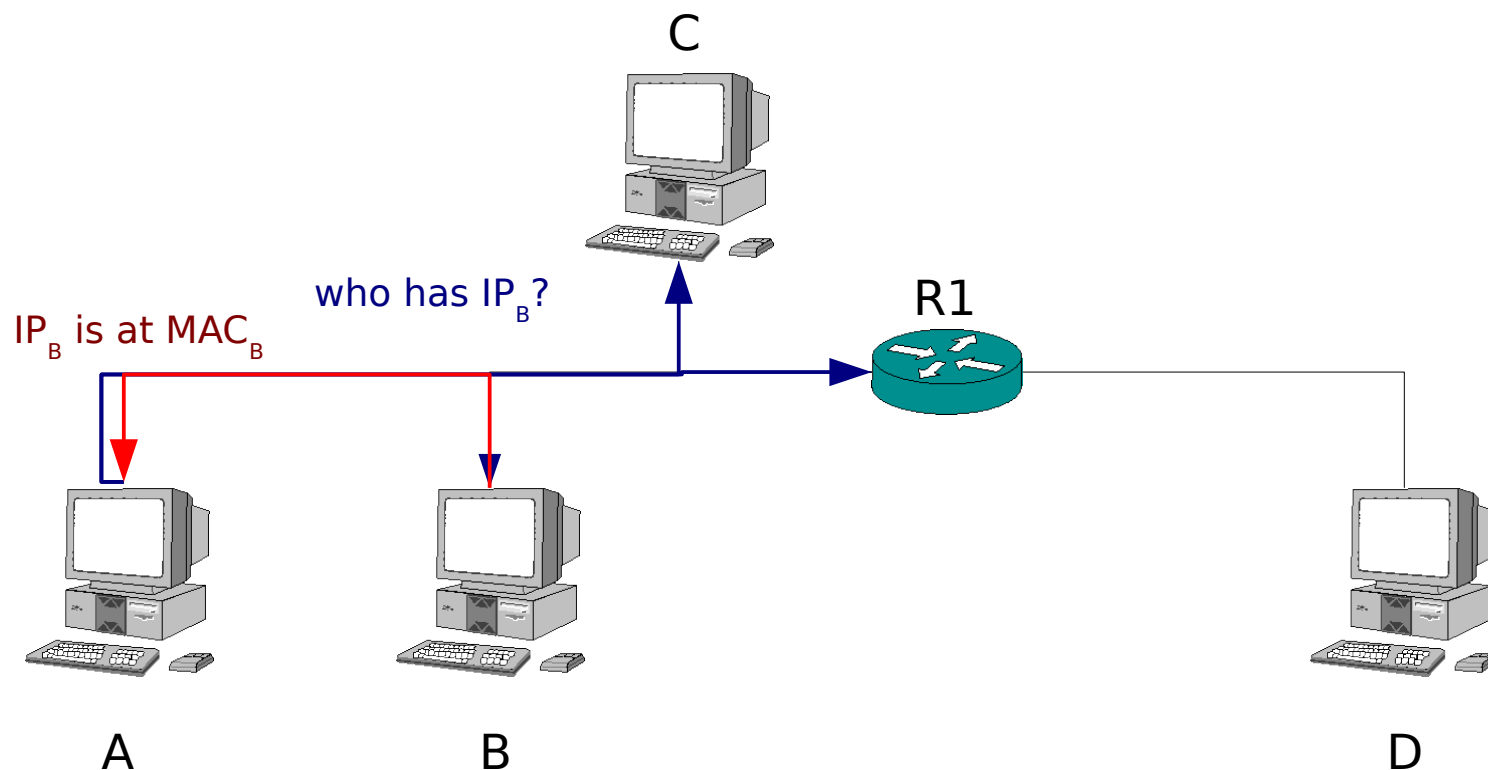


Descubrimiento de vecinos

Descubrimiento de vecinos

■ ARP (Address Resolution Protocol)

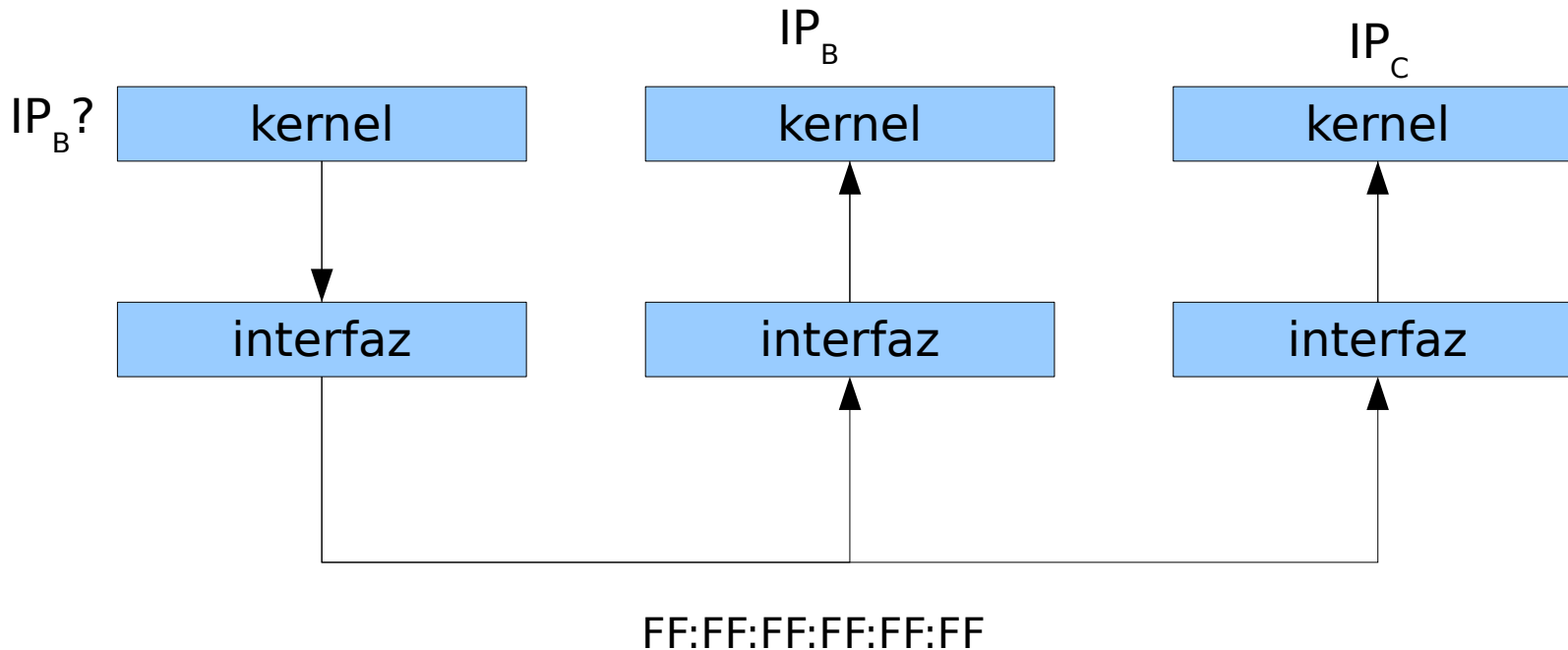
- Se emplea para asociar direcciones IPv4 con direcciones MAC en redes de difusión.
- Ejemplo sobre Ethernet:



Descubrimiento de vecinos

■ Ejemplo ARP

- La pregunta ARP viaja en una trama de difusión
 - Es aceptada por todos los interfaces.
 - Todos los interfaces copian la pregunta ARP al espacio del kernel.
 - Sólo la máquina con dirección IP_B contesta a la pregunta ARP.



Descubrimiento de vecinos

■ Neighbour Discovery (RFC 4861)

- Entre otras cosas, sustituye al protocolo ARP de IPv4
- Se emplea el protocolo ICMPv6 (Tipos 135 y 136)
- Ejemplo sobre Ethernet:

IPv6 2001:0db8:0100:0103:020f:b0ff:fea5:006e

Solicited-node prefix ff02:0000:0000:0000:0000:0001:ff00:0000/104

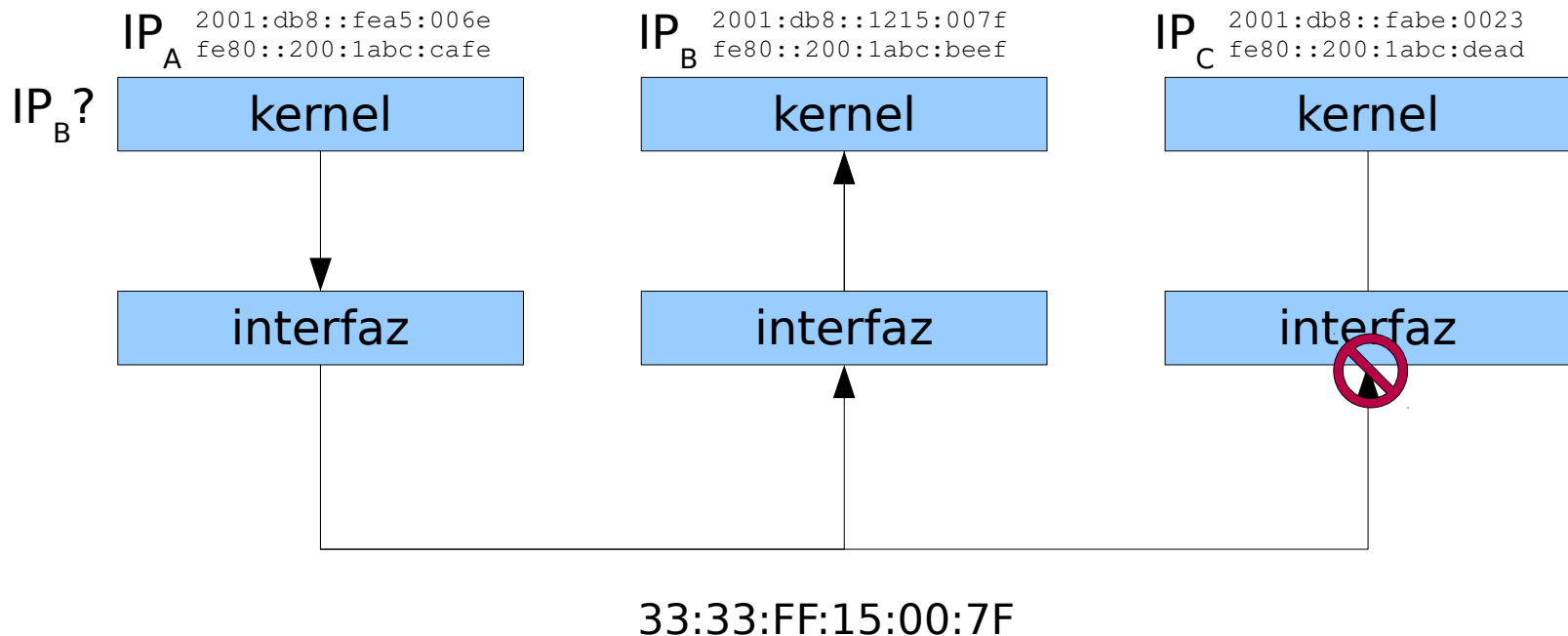
Solicited-node ff02:0000:0000:0000:0000:0001:ffa5:006e

ethernet 33:33:ff:a5:00:6e

Descubrimiento de vecinos

■ Ejemplo ND

- La pregunta ND viaja en una trama de multidifusión
 - Es aceptada sólo por los interfaces interesados (posiblemente, sólo uno)
 - Sólo la máquina con dirección IP_B contesta al *neighbour solicitation*.



Descubrimiento de vecinos

■ Neighbour Solicitation

- Una estación A con dirección 2001:db8::fea5:6e quiere enviar un datagrama a la estación B con dirección 2001:db8::1215:7f

6	Traffic Class	Flow Label	
Payload = 32		Next = 58	Hops = 255
Source Address = 2001:db8::fea5:6e			
Destination Address = ff02::1:ff15:7f			
Type = 135	Code = 0	Checksum	
Reserved = 0			
Target Address = 2001:db8::1215:7f			
Opt Code = 1	Opt Len = 1		
Source Link Layer Address = 0045678923f4			

Neighbour Solicitation

6	Traffic Class	Flow Label	
Payload = 32		Next = 58	Hops = 255
Source Address = 2001:db8::1215:7f			
Destination Address = 2001:db8::fea5:6e			
Type = 136		Code = 0	Checksum
R	S	O	Reserved = 0
Target Address = 2001:db8::1215:7f			
Opt Code = 2		Opt Len = 1	
Target Link Layer Address = 0012345678			

Neighbour Advertisement

R = Router
S = Solicited
O = Override

■ Descubrimiento de encaminadores

- Cuando una estación se activa, intenta autoconfigurar sus interfaces.
- Envía una solicitud de encaminador.

6	Traffic Class	Flow Label	
Payload = 16		Next = 58	Hops = 255
Source Address = fe80::200:1abc:cafe			
Destination Address = ff02::2			
Type = 133	Code = 0	Checksum	
Reserved = 0		Opt Code = 1	Opt Len = 1
Source Link Layer Address = 0045678923f4			

Router Solicitation

■ Descubrimiento de encaminadores

- Si existe algún encaminador configurado en el enlace, responde con un anuncio de encaminador.

6	Traffic Class		Flow Label		
Payload = 16			Next = 58		Hops = 255
Source Address = fe80::feed:beef					
Destination Address = ff02::1					
Type = 134		Code = 0		Checksum	
Hop Limit		M	O	Rsvd Router Lifetime	
Reachable Time					
Retrasmission Timer					
Opt Code = 1		Opt Len = 1			
Source Link Address					
Opt Code = 5		Opt Len = 1		Reserved = 0	
MTU					
Opt Code = 3		Opt Len = 4		Prefix Len	L A Rsvd
Valid Lifetime					
Preferred Lifetime					
Reserved					
Prefix					

M=Managed address
O=Other stateful config.

L=Link
A=Autonomous address

Router Advertisement

ICMPv6

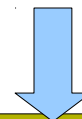
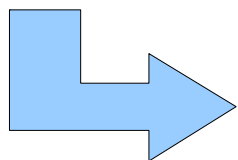
■ Autoconfiguración

- Con IPv6 no es necesario el protocolo DHCP para que una máquina obtenga su dirección en la red (aunque existe DHCPv6)
- Basta con que un encaminador en la red anuncie su prefijo (de 64 bits)
- La máquina recibe el prefijo y construye la dirección basándose en la dirección del interfaz.
- Ejemplo:

MAC: 00:18:84:67:06:1D

Prefijo: 2001:DB8:100:103::/64

0218:84FF:FE67:061D



2001:DB8:100:103:0218:84FF:FE67:061D/64

ICMPv6

■ Autoconfiguración

- La autoconfiguración plantea un problema de **privacidad**.
 - La dirección MAC está asociada a un equipo.
 - El equipo puede estar, o no, asociado a un individuo.
- Sería posible rastrear la posición global de un individuo concreto.
- Ejemplo:



MAC: 00:18:84:67:06:1D

Prefijo:2001:DB8:100:103::/64
(casa)

Prefijo:2001:DB8:1::/64
(oficina)

log del servidor web en la empresa:

Connection from 2001:db8:100:103:**0218:84FF:FE67:061D** at 09:05

Connection from 2001:db8:1::**0218:84FF:FE67:061D** at 12:00

Connection from 2001:db8:100:103:**0218:84FF:FE67:061D** at 13:12

ICMPv6

■ Autoconfiguración

- Extensiones de privacidad (RFC 4941)
 - El identificador de interfaz se genera de forma pseudoaleatoria.
 - Combinación de: MAC, semilla, algoritmo de hash (MD5).
 - En Linux se controla mediante
`net.ipv6.conf.iface.use_tempaddr={0,1,2}`.

MAC: 00:18:84:67:06:1D



Prefijo:2001:DB8:100:103::/64
(casa)

Prefijo:2001:DB8:1::/64
(oficina)

log del servidor web en la empresa:

Connection from 2001:db8:100:103:**3b22:fd33:4530:13d2** at 09:05

Connection from 2001:db8:1::**f432:de23:2129:3f87** at 12:00

Connection from 2001:db8:100:103:**32d2:edf4:2319:57a6** at 13:12

■ Autoconfiguración

- Identificador de interfaz estable y opaco (RFC 7217)
 - Las extensiones de privacidad del RFC 4941 provocan direcciones cambiantes
 - Difícil la gestión de sucesos en la red
 - No adecuado para servidores
 - Sería conveniente tener direcciones estables pero que no permitan el seguimiento ni ofrezcan información sobre el hardware subyacente
 - Solución: RFC7217 (abril 2014)
 - $RID = F(\text{Prefix}, \text{Net_Iface}, \text{Network_ID}, \text{DAD_Counter}, \text{secret_key})$
 - $F()$ puede ser cualquier función criptográfica apropiada, como SHA-1 o SHA-256
 - El RID generado es pseudoaleatorio, pero estable, lo que facilita la gestión de la red al tiempo que respeta la privacidad.

IP de nueva generación: IPv6



Secure Neighbour Discovery

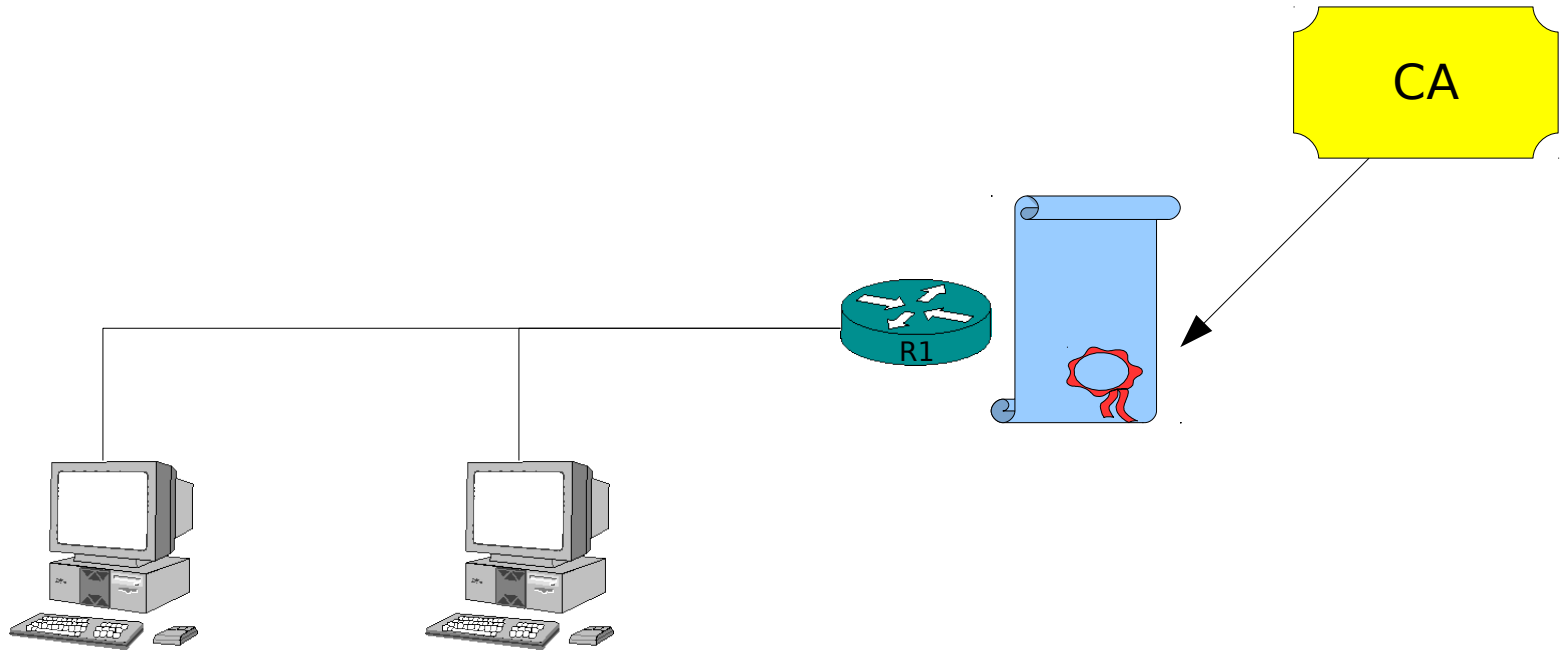
Secure Neighbour Discovery

- **RFC 3971, 6494, 6495 y 6980**
 - NDP es vulnerable.
 - Denegación de servicio.
 - Suplantación de identidad (nodos y encaminadores)
 - Secure Neighbour Discovery (SeND) utiliza Direcciones Generadas Criptográficamente (CGA, *Cryptographically Generated Addresses*): RFC 3972, 4581, 4982.
 - Se asegura de que quien dice poseer una dirección IPv6, realmente la posee (tanto encaminadores como máquinas)
 - Se emplean parejas de claves pública-privada.
 - Despliegue difícil.

Secure Neighbour Discovery

■ Funcionamiento

- Hace uso de PKI (Public Key Infrastructure)
 - Cada encaminador debe tener un certificado válido.
 - Cada nodo debe tener una lista de autoridades reconocidas.
 - Los mensajes de descubrimiento y de anuncio van firmados.



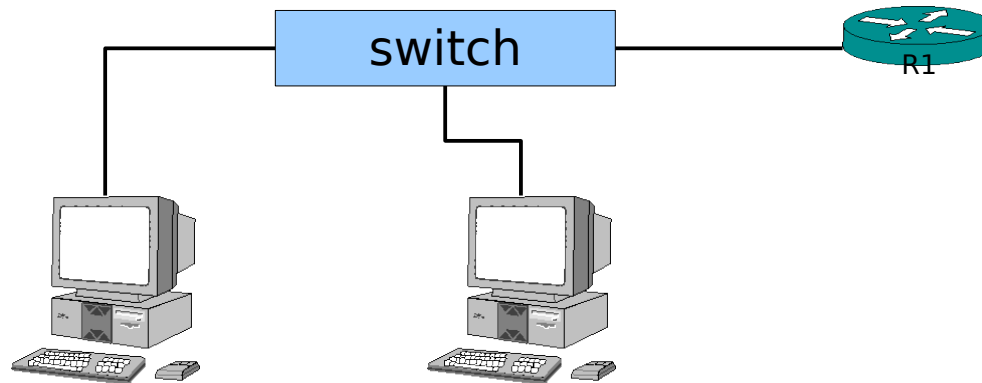
Secure Neighbour Discovery

■ Dificultades

- El despliegue de SEND es difícil.

■ Posibles soluciones

- RA-Guard (RFC 6105)
 - Elementos L2 (switches) se encargan de permitir los RA legítimos y bloquear los falsos.
 - Pueden hacerlo por:
 - Dirección origen
 - Puerto
 - Estado SEND, etc.
- Actualizado: RFC 7113



IP de nueva generación: IPv6



Configuración en GNU/Linux

Configuración

■ Configuración IPv6 en GNU/Linux

- El *kernel* ya incorpora soporte para IPv6.
- Comprobar la configuración:

```
# ifconfig eth0
```

```
eth0      Link encap:Ethernet  HWaddr 00:0F:B0:A5:00:6E  
          inet addr:147.96.80.26  Bcast:147.96.81.255 Mask:255.255.254.0  
          inet6 addr: fe80::20f:b0ff:fea5:6e/64 Scope:Link  
          inet6 addr: fd00::50:10/64 Scope:Site  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Configuración

■ Configuración en GNU/Linux

- Desde la línea de órdenes:

```
# ifconfig eth0 add 2001:db8::50:10/64
# ip addr add 2001:db8::50:10/64 dev eth0          (preferible)
# ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:0F:B0:A5:00:6E
      inet addr:169.96.80.26  Bcast:169.96.81.255  Mask:255.255.254.0
      inet6 addr: fe80::20f:b0ff:fea5:6e/64 Scope:Link
      inet6 addr: 2001:db8::50:10/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1396375 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1626 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:171897115 (163.9 MiB)  TX bytes:316161 (308.7 KiB)
      Interrupt:225 Base address:0x5000
```

Configuración

■ Configuración permanente en GNU/Linux

- En el archivo /etc/network/interfaces :

```
auto eth0
iface eth0 inet static
    address 192.168.1.5
    netmask 255.255.254.0
    network 192.168.0.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-servers 127.0.0.1
```

```
iface eth0 inet6 static
    address 2001:db8::50:10
    netmask 64
    gateway 2001:db8::1
```


Configuración

- **Control de la autoconfiguración en GNU/Linux**
 - El ejemplo anterior proporciona una dirección estática al interfaz.
 - Adicionalmente, si se recibe un anuncio de prefijo por parte de un encaminador, se añadirá la dirección correspondiente por autoconfiguración.
 - El proceso de autoconfiguración se puede desactivar mediante dos variables de kernel (en `/etc/sysctl.conf`)
 - `net.ipv6.conf.[all|default|iface].accept_ra=0`
 - `net.ipv6.conf.[all|default|iface].autoconf=0`

Configuración

■ Configuración permanente mediante zebra

- Se debe tener activado el demonio **zebra**:
 - Asegurarse de que está zebra=yes en /etc/quagga/daemons
 - Reiniciar el servicio si es necesario: service quagga restart
- Conectarse a zebra mediante **vttysh**:

```
Hello, this is Quagga (version 0.99.23.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
uml1# configure terminal  
uml1(config)# interface eth0  
uml1(config-if)# ipv6 address 2001:db8::50:10/64  
uml1(config-if)# exit  
uml1(config)# ipv6 route ::/0 2001:db8::1  
uml1(config)# end  
uml1# write  
Building Configuration...  
Configuration saved to /etc/quagga/zebra.conf  
[OK]  
uml1# quit
```

Configuración

■ Anuncio de prefijos en un encaminador Linux

- Mediante el demonio radvd:
 - Demonio ligero, apto para dispositivos con poca memoria (p.e. encaminadores ADSL o puntos de acceso).
- Ejemplo de configuración en GNU/Linux (/etc/radvd.conf)

```
interface eth0 {  
    AdvSendAdvert on;  
    MinRtrAdvInterval 3;  
    MaxRtrAdvInterval 10;  
    prefix 2001:db8:100:103::/64 {  
        AdvOnLink on;  
        AdvAutonomous on;  
        AdvRouterAddr on;  
    };  
};
```

- Ejemplo de configuración en OpenWRT (/etc/config/radvd)

```
config prefix  
    option interface      'lan'  
    # If not specified, a non-link-local prefix of the interface is used  
    list prefix           '2001:db8:100:103::/64'  
    option AdvOnLink      1  
    option AdvAutonomous  1  
    option AdvRouterAddr  1  
    option ignore         0
```

Configuración

■ Anuncio de prefijos en un encaminador Linux

- Activar el demonio **zebra** (del paquete **quagga**):
 - En /etc/quagga/daemons añadir la entrada: zebra=yes
 - En la consola (como usuario root):

```
root@uml1:~# vtysh
```

```
Hello, this is Quagga (version 0.99.23.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
uml1# configure terminal  
uml1(config)# interface eth0  
uml1(config-if)# no ipv6 nd suppress-ra  
uml1(config-if)# ipv6 nd prefix 2001:db8:100:103::/64  
uml1(config-if)# end  
uml1# write  
Building Configuration...  
Configuration saved to /etc/quagga/zebra.conf  
[OK]  
uml1#
```

Configuración

- **Anuncio de prefijos en un encaminador Linux**
 - Se ha guardado en /etc/quagga/zebra.conf:

```
! Zebra configuration saved from vty
!  
hostname zebra  
password zebra  
enable password zebra  
log file /var/log/zebra/zebra.log  
!  
interface eth0  
    no ipv6 nd suppress-ra  
    ipv6 nd prefix 2001:db8:100:103::/64
```

Configuración

■ Configuración automática en Windows 7

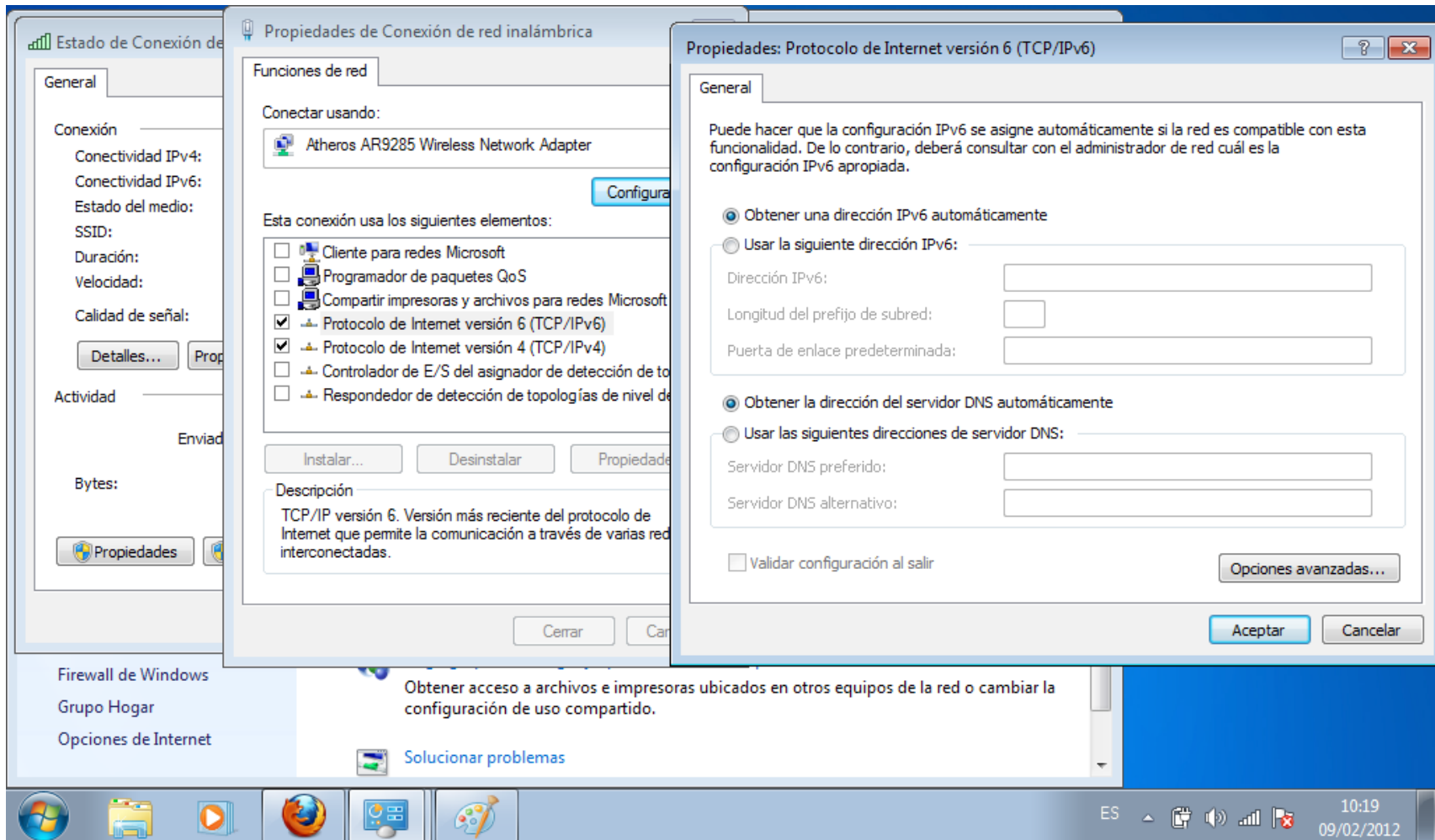
- Desactivar los túneles Teredo
 - Abrir una consola de órdenes con permisos de administrador

```
netsh interface teredo set state disabled
```

Configuración

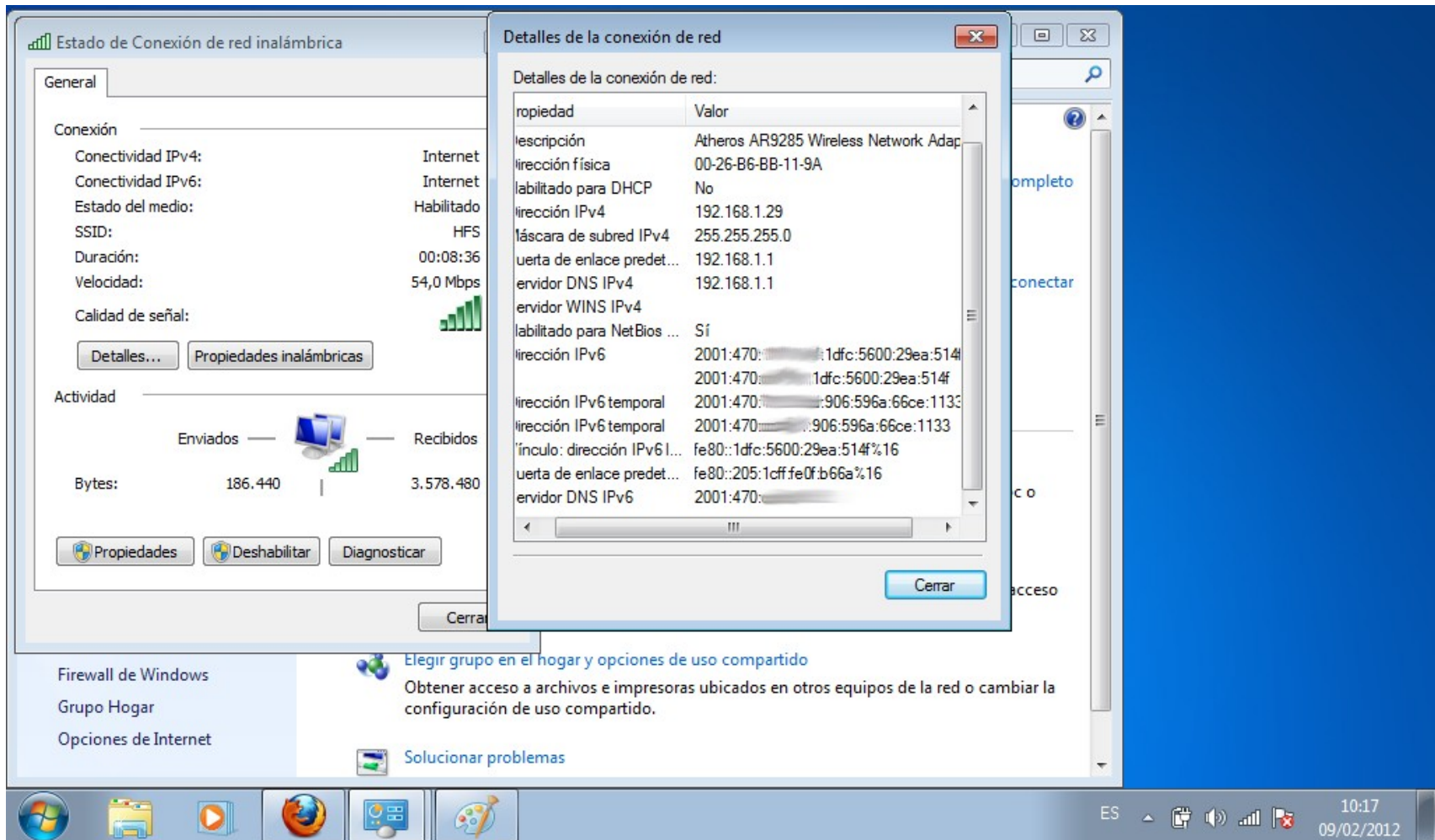
■ Configuración automática en Windows 7

- Configurar las propiedades del adaptador



Configuración

- Configuración automática en Windows 7
 - Configurar las propiedades del adaptador



Configuración

■ Configuración automática en Windows 7

- Comprobar la conectividad

Firefox ▾ Hurricane Electric Internet Services - I... Consulintel - The IPv6 Company - Pá... TEST Test your IPv6.

test-ipv6.com

Test IPv6 FAQ World IPv6 Launch Local Times Mirrors Stats

Test your IPv6 connectivity.

Summary Tests Run Technical Info Share Results / Contact

- Information Your IPv4 address on the public Internet appears to be 147.96.80.███
- Information Your IPv6 address on the public Internet appears to be 2001:470:███906:596a:66ce:1133
- ✓ The [World IPv6 Launch](#) day is June 6th, 2012. **Good news!** Your current browser, on this computer and at this location, are expected to keep working after the Launch. [\[more info\]](#)
- ✓ Congratulations! You appear to have both IPv4 and IPv6 Internet working. If a publisher publishes to IPv6, your browser will connect using IPv6. Your browser prefers IPv6 over IPv4 when given the choice (this is the expected outcome).
- ✓ Your DNS server (possibly run by your ISP) appears to have IPv6 Internet access.

Your readiness scores

10/10 for your IPv4 stability and readiness, when publishers offer both IPv4 and IPv6

10/10 for your IPv6 stability and readiness, when publishers are forced to go IPv6 only

Click to see [test data](#)

Configuración

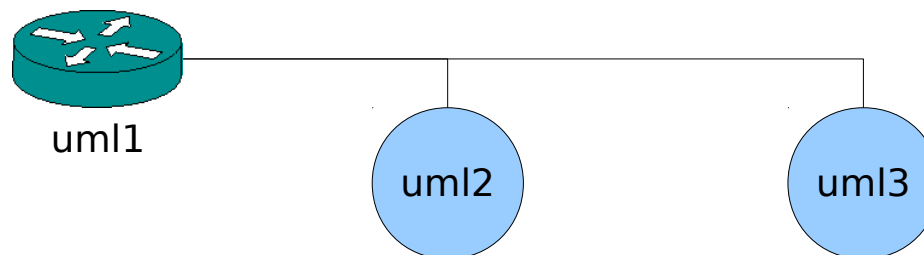
■ Práctica:

- Iniciar 3 máquinas virtuales.
- La máquina uml1 deberá configurarse para que anuncie el prefijo de red 2001:db8:0:1::/64 por su interfaz eth0
- Al levantar el interfaz eth0 de las máquinas uml2 y uml3, deberán obtener el prefijo de red desde uml1 y configurar automáticamente sus direcciones IPv6.

ifconfig eth0 up

ifconfig eth0

- Añadir en uml1 el anuncio del prefijo 2001:db8:101:1::/64



Configuración

■ En uml1:

- Editar el archivo /etc/quagga/daemons
nano /etc/quagga/daemons
- Cambiar la línea zebra=no por zebra=yes
- Conectar a zebra por línea de órdenes:

```
root@uml1:/etc/quagga# vtysh
```

```
uml1# configure terminal
```

```
uml1(config)# interface eth0
```

```
uml1(config-if)# ip address 192.168.1.1/24
```

```
uml1(config-if)# ipv6 address 2001:db8:0:1::1/64
```

```
uml1(config-if)# no ipv6 nd suppress-ra
```

```
uml1(config-if)# ipv6 nd prefix 2001:db8:101:1::/64
```

```
uml1(config-if)# ipv6 nd prefix 2001:db8:0:1::/64
```

```
uml1(config-if)# end
```

```
uml1# write
```

```
Building Configuration...
```

```
Configuration saved to /etc/quagga/zebra.conf
```

```
[OK]
```

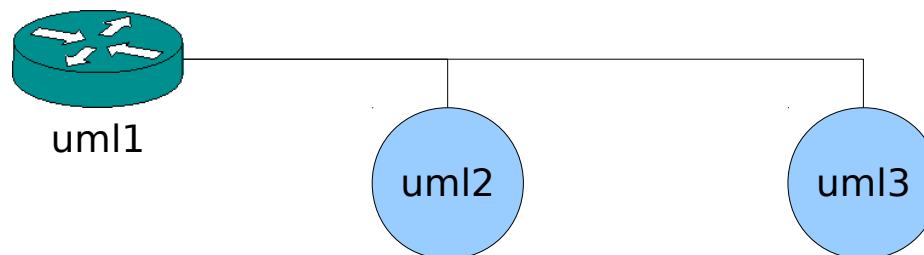
Se pueden
anunciar varios
prefijos

Configuración

■ En uml2 y uml3:

- Basta levantar el interfaz para que se autoconfigure

```
uml2# configure terminal
uml2(config)# interface eth0
uml2(config-if)# no shutdown
uml2(config-if)# end
uml2# show interface eth0
Interface eth0 is up, line protocol detection is
disabled
  index 2 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 02:00:00:00:01:f0
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet6 fe80::ff:fe00:1f0/64
  inet6 2001:db8:101:1::ff:fe00:1f0/64
uml2#
```



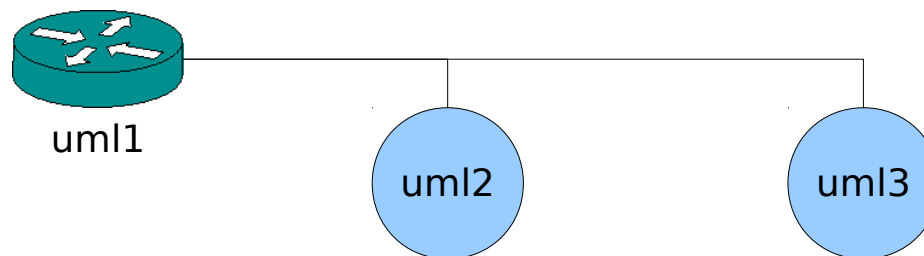
Configuración

■ En uml2:

- Basta levantar el interfaz para que se autoconfigure

```
uml2# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

K>* ::/0 via fe80::ff:fe00:2f0, eth0
C>* ::1/128 is directly connected, lo
C>* 2001:db8:101:1::/64 is directly connected, eth0
C>* fe80::/64 is directly connected, eth0
uml2#
```



IP de nueva generación: IPv6



Mecanismos de transición

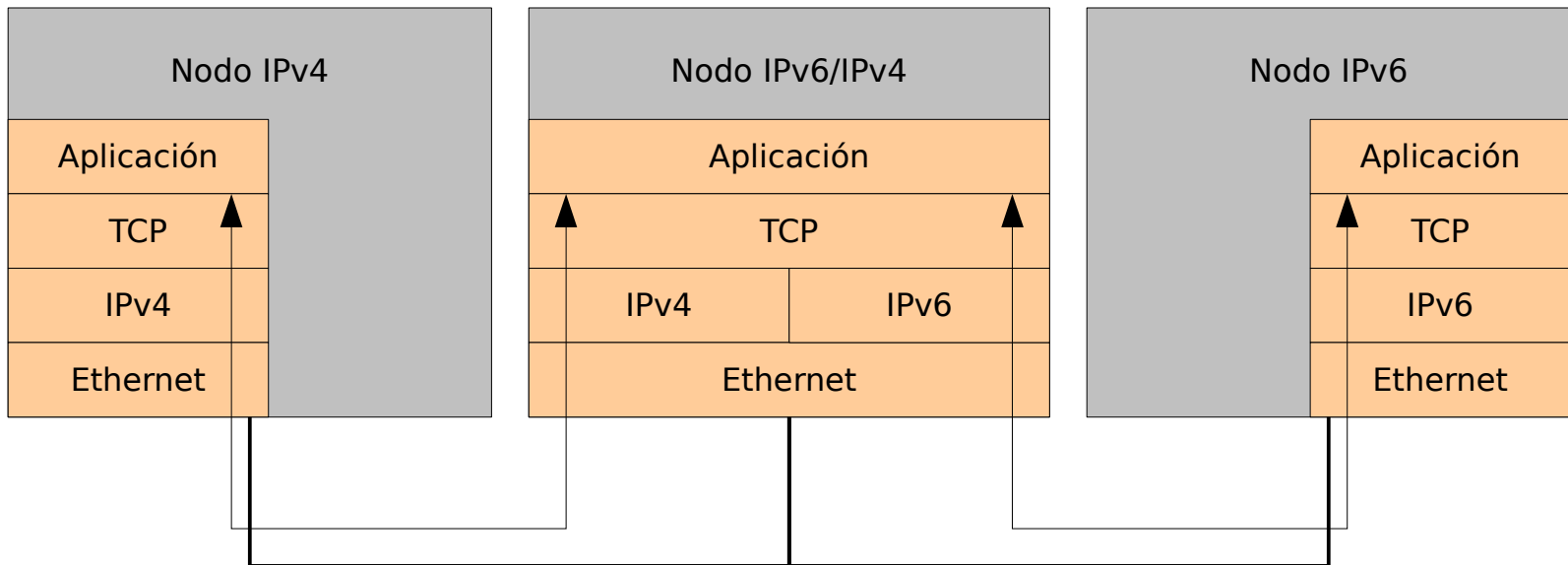
Mecanismos de transición

- Hasta el despliegue total de IPv6 convivirán ambos protocolos, IPv4 e IPv6.
- Es necesario garantizar una transición “suave”, durante la cual la compatibilidad entre sistemas sea la máxima posible.
- Varios escenarios:
 - Islas IPv6 conectadas a través de troncales IPv4.
 - Coexistencia de ambos protocolos.
 - Islas IPv4 conectadas a través de troncales IPv6.
- Soluciones propuestas:
 - Dual Stack.
 - Túneles (automáticos y manuales).
 - Traducción de cabeceras.
 - ...

Mecanismos de transición

■ Dual Stack

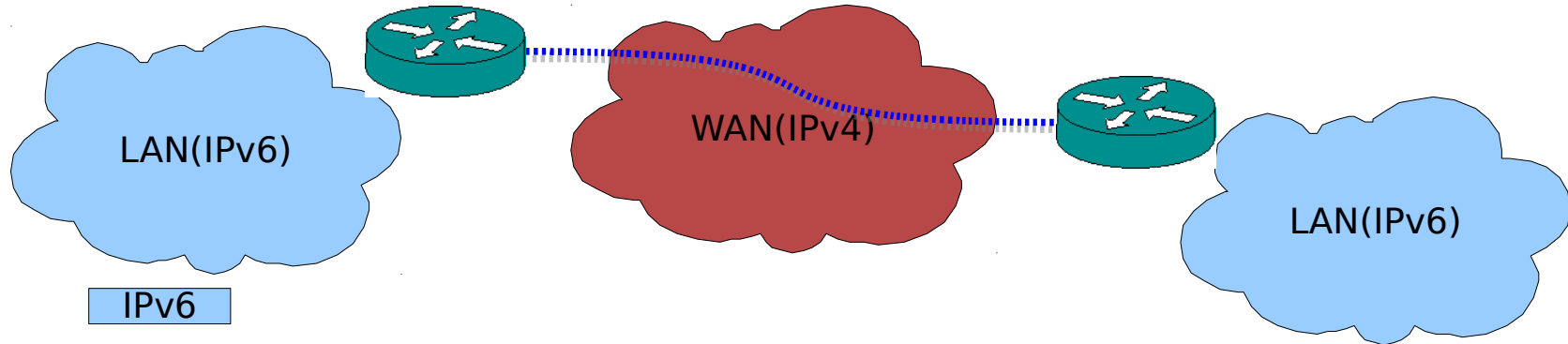
- Un nodo implementa ambos protocolos.
- Es la manera más simple de garantizar la compatibilidad entre sistemas.
- No resuelve el problema de la escasez de direcciones IPv4.



Mecanismos de transición

■ Túneles IPv6-sobre-IPv4

- Útiles para conectar “islas” IPv6 a través de redes antiguas.



Mecanismos de transición

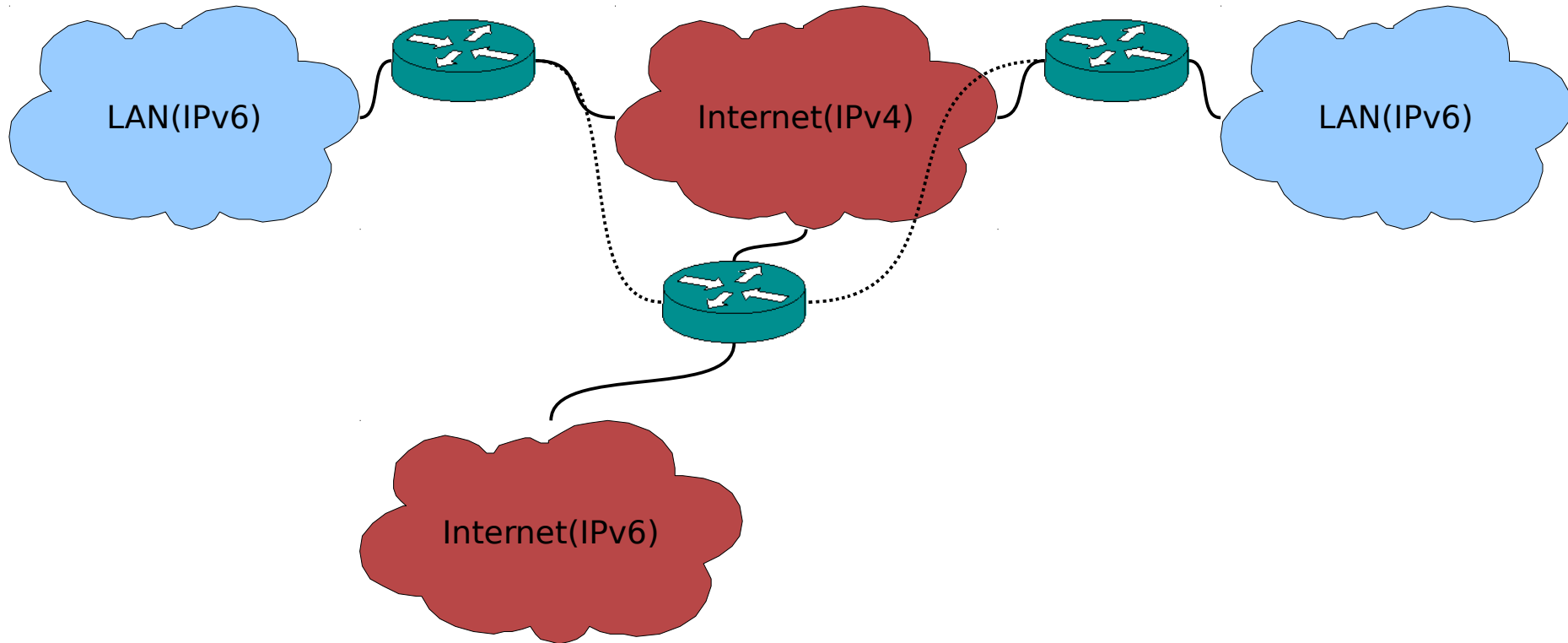
■ Túneles automáticos

- Se pensaron como un medio rápido para facilitar la comunicación entre nodos IPv6 a través de redes IPv4
- Se ha observado un bajo rendimiento, además de problemas de seguridad.
- Se desaconseja su uso.

Mecanismos de transición

■ Túneles manuales 6in4 (RFC 4213)

- Necesitan de un *tunnel broker* para conectarse con Internet6.
- Han de configurarse explícitamente por parte del administrador de la red.



Mecanismos de transición

■ Túneles manuales 6in4 (RFC 4213)

■ tunnelbroker.net

The screenshot shows a web browser window with the URL `tunnelbroker.net/tunnel_detail.php?tid=`. The page is titled "Tunnel Details" and features a sidebar with "Account Menu" and "User Functions". The main content area displays details for an "IPv6 Tunnel", including fields for Tunnel ID, Creation Date (Jul 3, 2010), and Description. It also lists "IPv6 Tunnel Endpoints" with Server and Client IPv4 and IPv6 addresses, "Available DNS Resolvers", "Routed IPv6 Prefixes", and "rDNS Delegations". A "Quick Links" sidebar on the right provides navigation to various resources like Certification, Tunnelbroker, Free DNS, Code, BGP Toolkit, Forums, FAQ, Video Presentations, IPv6 Blog Posts, Usage Statistics, Tunnel Server Status, Network Map, Looking Glass (v4/v6), Route Server (telnet), Global IPv6 Report, and IPv6 BGP View. A "Services" sidebar at the bottom right lists Transit, Colocation, and Dedicated Servers, along with a "v4 Exhaustion" section and a "IPv4 & IPv6 Statistics" button.

Account Menu

- Main Page
- Account Info
- Logout

User Functions

- Combine Tunnels
- Create Regular Tunnel
- Create BGP Tunnel
- IPv6 Portscan

Tunnel Details

IPv6 Tunnel **Example Configurations**

Tunnel ID: [redacted] [Delete Tunnel](#)

Creation Date: Jul 3, 2010

Description: [redacted]

IPv6 Tunnel Endpoints

Server IPv4 Address: 209.51.161.58

Server IPv6 Address: 2001:470:[redacted]:1/64

Client IPv4 Address: **88.20.16.249**

Client IPv6 Address: 2001:470:[redacted]:2/64

Available DNS Resolvers

Anycasted IPv6 Caching Nameserver: 2001:470:20::2

Anycasted IPv4 Caching Nameserver: 74.82.42.42

Routed IPv6 Prefixes

Routed /64: 2001:470:[redacted]:/64

Routed /48: [Assign /48](#)

rDNS Delegations [Edit](#)

rDNS Delegated NS1:

rDNS Delegated NS2:

rDNS Delegated NS3:

Quick Links

- [Certification](#)
- [Tunnelbroker](#)
- [Free DNS](#)
- [Code](#)
- [BGP Toolkit](#)
- [Forums](#)
- [FAQ](#)
- [Video Presentations](#)
- [IPv6 Blog Posts](#)
- [Usage Statistics](#)
- [Tunnel Server Status](#)
- [Network Map](#)
- [Looking Glass \(v4/v6\)](#)
- [Route Server \(telnet\)](#)
- [Global IPv6 Report](#)
- [IPv6 BGP View](#)

Services

- [Transit](#)
- [Colocation](#)
- [Dedicated Servers](#)

v4 Exhaustion

IPv4 & IPv6 Statistics

Mecanismos de transición

■ Túneles 6in4

- Configuración en GNU/Linux:
 - Desde la shell:

```
# iptunnel add sit1 remote 194.179.25.50
# ifconfig sit1 up mtu 1400
# ifconfig sit1 add 2001:db8:40:261c::2/64
# route --inet6 add default gw 2001:db8:40:261c::1
```

```
# ip tunnel add sit1 [mode sit] remote 194.179.25.50
# ip link set dev sit1 up mtu 1400
# ip addr add 2001:db8:40:261c::2/64 dev sit1
# ip -6 route add default via 2001:db8:40:261c::1
    o bien
# ip -6 route add default dev sit1
```

Mecanismos de transición

■ Túneles 6in4

- Configuración en GNU/Linux:
 - Desde /etc/network/interfaces:

```
auto tunel
iface tunel inet6 v4tunnel
    address    2001:db8:40:261c::2
    netmask    64
    endpoint    194.179.25.50
    gateway    2001:db8:40:261c::1
    mtu        1400
```

Mecanismos de transición

■ Túneles 6in4

- Configuración en openwrt (túneles dinámicos con he.net)
 - En /etc/config/network/:

```
config 'interface' 'henet'
    option 'proto' '6in4'
    option 'peeraddr' '216.66.80.30'
    option 'ip6addr' '2001:470:1fb2:23a::2/64'
    option 'defaultrouter' '1'
    option 'ttl' '255'
    option 'mtu' '1400'
    option 'tunnelid' 'id_del_tunel'
    option 'username' 'id_del_usuario (md5)'
    option 'password' 'passwd (md5)'
```

tunnelid: el identificador del túnel en el *tunnelbroker*.

username: el identificador del usuario. Es un hash md5, por ejemplo 'a5bf751e332a8a977b608b8b8fa019cf'. Aparece en la configuración del túnel.

password: el hash md5 de la contraseña en el tunnelbroker. Para generarlo:

```
echo -n "contraseña" | md5sum
484ac397cb407ab7aad776f0663f8c85 -
```

Mecanismos de transición

■ Túneles 6in4

- Configuración en Windows 7 (túneles con he.net)
 - En una ventana de órdenes con permisos de administrador:

```
netsh interface teredo set state disabled
netsh interface ipv6 add v6v4tunnel IP6Tunnel ip4_local 209.51.161.58
netsh interface ipv6 add address IP6Tunnel 2001:db8:6:d32::2
netsh interface ipv6 add route ::/0 IP6Tunnel 2001:db8:6:d32::1
```

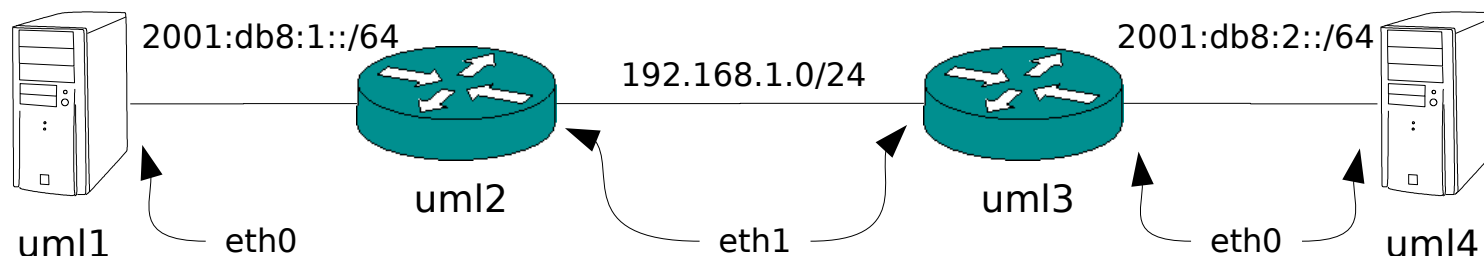
ip4_local: dirección IPv4 del interfaz. Si estamos detrás de NAT, será una dirección privada; si no, será una dirección pública.

Mecanismos de transición

■ Túneles 6in4

■ Práctica:

- Crear la siguiente configuración con máquinas UML:



■ Notas:

- añadir a uml1 y uml4 la ruta 2000::/3 a través de su encaminador respectivo.
`ip route add default via 2001:db8:1::2 (uml1)`
`ip route add default via 2001:db8:2::3 (uml4)`
- Activar el encaminamiento en uml2 y uml3:
`sysctl -w net.ipv6.conf.all.forwarding=1`
- Añadir las rutas necesarias a uml2 y uml3:
`ip route add 2001:db8:2::/64 dev tn (uml2)`
`ip route add 2001:db8:1::/64 dev tn (uml3)`

Mecanismos de transición

■ Túneles 6in4

The image shows a Wireshark capture window titled "umlbridge: Capturing - Wireshark". The packet list shows several ICMPv6 Echo request and reply packets between fec0:1::1 and fec0:2::4. Packet 18 is selected, showing its details in the packet pane. The details pane shows the Ethernet II header, the Internet Protocol header (highlighted with a red box), and the Internet Protocol Version 6 details. The IP header shows Source: 192.168.1.3 and Destination: 192.168.1.2. The IP version details show Version: 6, Traffic class: 0x00000000, Flowlabel: 0x00000000, Payload length: 64, Next header: ICMPv6 (0x3a), Hop limit: 63, Source: fec0:2::4, and Destination: fec0:1::1. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
16	1.014201	fec0:1::1	fec0:2::4	ICMPv6	Echo reply
17	2.016490	fec0:2::4	fec0:1::1	ICMPv6	Echo request
18	2.016677	fec0:2::4	fec0:1::1	ICMPv6	Echo request
19	2.016822	fec0:2::4	fec0:1::1	ICMPv6	Echo request
20	2.016936	fec0:1::1	fec0:2::4	ICMPv6	Echo reply

Frame 18 (138 bytes on wire, 138 bytes captured)

Ethernet II, Src: 02:00:00:00:03:01 (02:00:00:00:03:01), Dst: 02:00:00:00:02:01 (02:00:00:00:02:01)

Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.2 (192.168.1.2)

Internet Protocol Version 6

- 0110 = Version: 6
- 0000 0000 = Traffic class: 0x00000000
- 0000 0000 0000 0000 = Flowlabel: 0x00000000
- Payload length: 64
- Next header: ICMPv6 (0x3a)
- Hop limit: 63
- Source: fec0:2::4 (fec0:2::4)
- Destination: fec0:1::1 (fec0:1::1)

Internet Control Message Protocol v6

0000 02 00 00 00 02 01 02 00 00 00 03 01 08 00 45 00E.

0010 00 7c 00 00 40 00 3f 29 b8 03 c0 a8 01 03 c0 a8 |..@?)

0020 01 02 60 00 00 00 00 40 3a 3f fe c0 00 02 00 00 ..`.@ :?.....

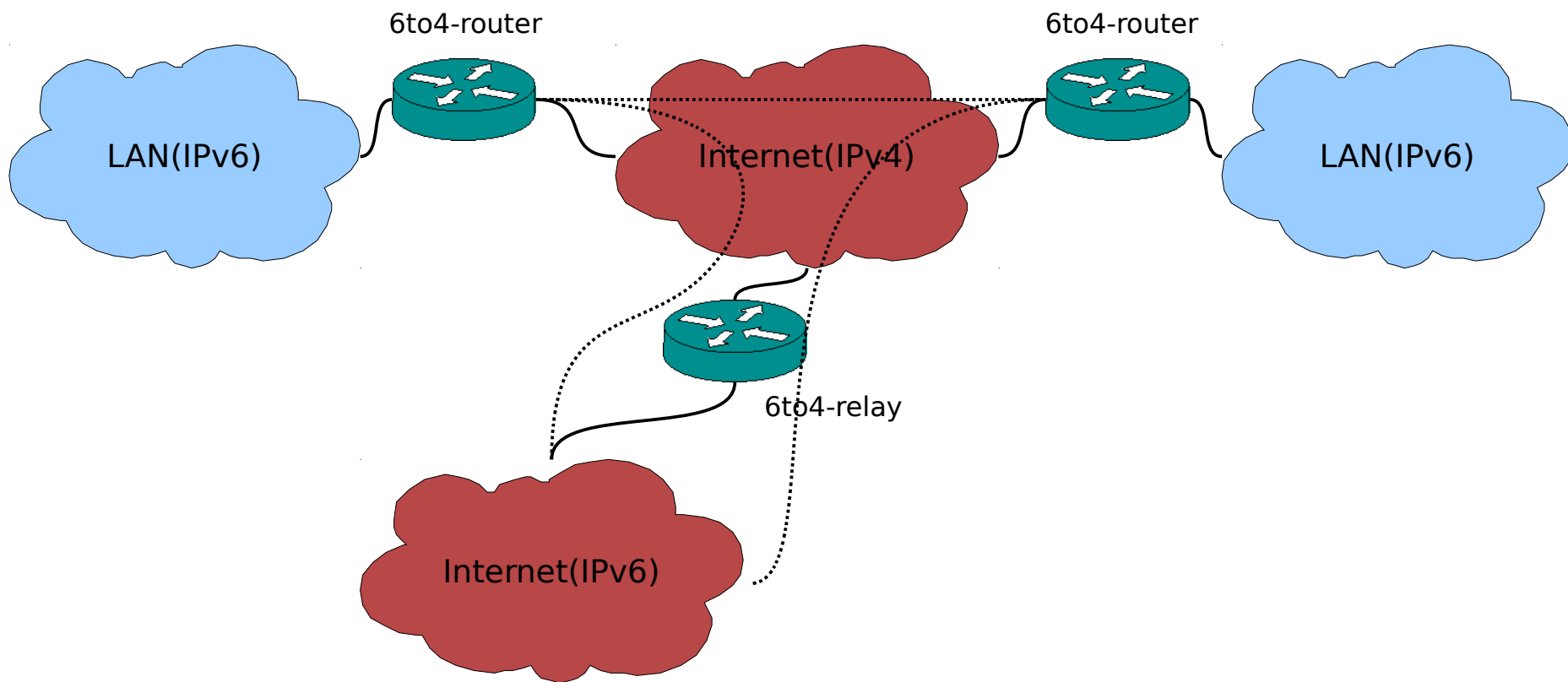
0030 00 00 00 00 00 00 00 00 00 04 fe c0 00 01 00 00E.

Frame (frame), 138 bytes Packets: 36 Displayed: 36 Marked: 0 Profile: Default

Mecanismos de transición

■ Túneles automáticos 6to4 (RFC 3056)

- Utilizan 6to4-relays, con dirección *anycast* 192.88.99.1.
- Rango de direcciones reservadas: 2002::/16
- Prefijo para el sitio: 2002:V4ADDR::/48



Mecanismos de transición

■ Túneles automáticos 6to4

- Configuración en GNU/Linux:
 - Desde la shell:

```
# ip tunnel add 6to4 mode sit remote 192.88.99.1
# ip link set dev 6to4 up mtu 1400
# ip addr add 2002:c0a8:101:17ce:f3b7:a8f8:d203:7411/128 dev 6to4
# ip -6 route add default dev 6to4
```

Nota: si estamos tras un NAT, la dirección V4ADDR es la pública

Mecanismos de transición

■ Túneles automáticos 6to4

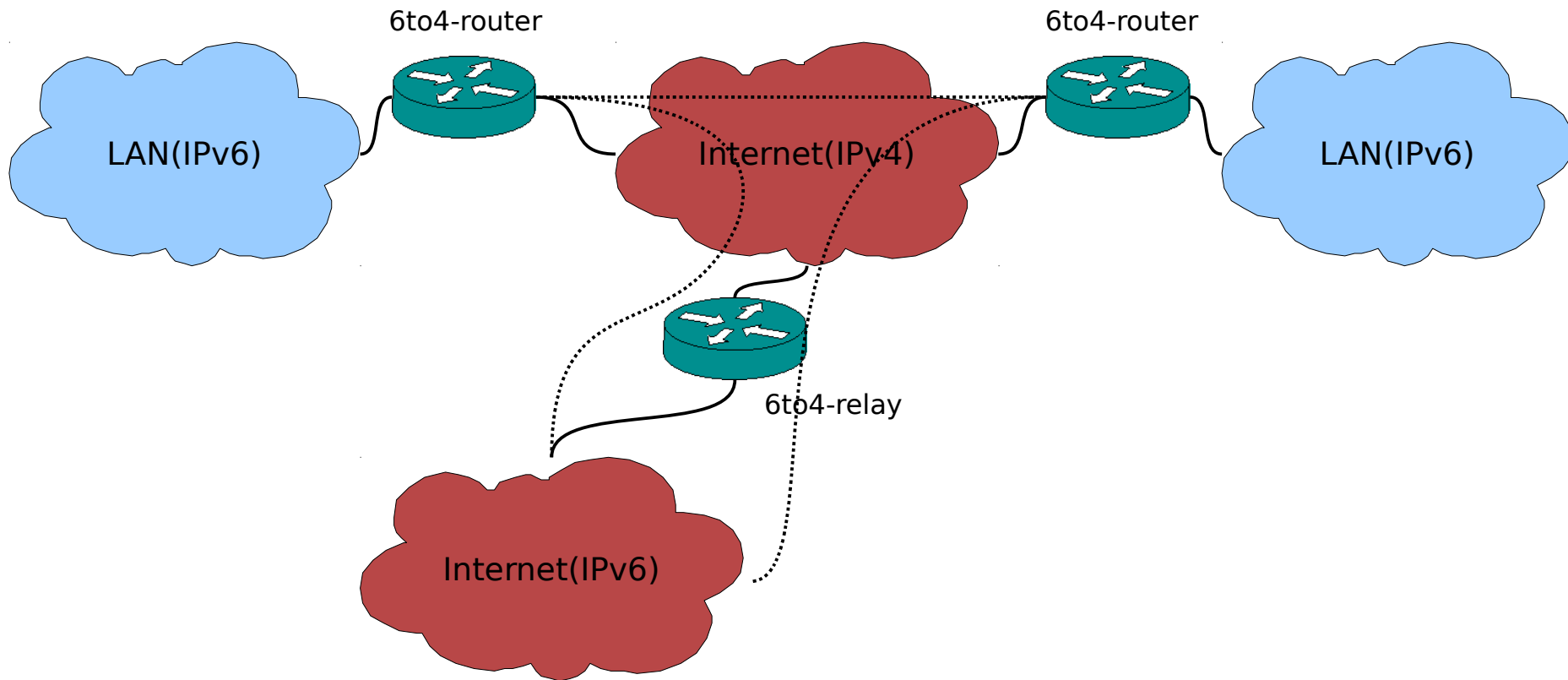
- Configuración en GNU/Linux:
 - Desde /etc/network/interfaces:

```
auto tunel
iface tunel inet6 v4tunnel
    address    2002:c0a8:101:17ce:f3b7:a8f8:d203:7411
    netmask    128
    endpoint    192.88.99.1
    gateway    ::
    mtu        1400
```

Mecanismos de transición

■ Túneles automáticos 6rd (RFC xxxx)

- Utilizan 6to4-relays, con dirección *anycast* 192.88.99.1.
- Rango de direcciones reservadas: 2002::/16
- Prefijo para el sitio: 2002:V4ADDR::/48



IP de nueva generación: IPv6



Plan de despliegue

Plan de numeración

■ Consideraciones:

- Todo enlace debe tener asignado un prefijo /64 (incluidos los enlaces p-a-p)
 - En algunos casos puede bastar con la dirección de enlace local
- El mínimo prefijo recomendado para un usuario final (residencial) es /48
 - Posibilita el uso de 65536 redes internas
- Es preciso cambiar la mentalidad respecto a IPv4
 - No hay que ahorrar direcciones IP, hay que facilitar el agregado de direcciones

Plan de numeración

■ Ejemplo:

- Supongamos una organización con 20 dependencias distribuidas en un solo sitio geográfico
 - Podría solicitar algo más grande, pero supongamos que obtiene un /48
 - Entonces tiene 16 bits para subredes. Puede hacer una distribución entre sus sedes, previendo además un posible aumento en el número de las mismas

Sedes		Redes
2	srrr rrrr rrrr rrrr	32768
4	ssrr rrrr rrrr rrrr	16384
8	sssr rrrr rrrr rrrr	8192
16	ssss rrrr rrrr rrrr	4096
32	ssss srrr rrrr rrrr	2048
64	ssss ssrr rrrr rrrr	1024
128	ssss sssr rrrr rrrr	512

Plan de numeración

■ Ejemplo:

- Supongamos que optamos por reservar espacio para 128 sedes, con 512 subredes cada una
 - Cada sede tiene asignado un prefijo /55
- ¿Cómo se asigna la numeración a cada sede?
 - Consecutivamente:
 - 2001:db8:0::/55
 - 2001:db8:1::/55
 - ...
 - Problema: no hay posibilidad de crecimiento
 - Mediante un árbol binario:
 - 2001:db8:0::/55
 - 2001:db8:ffe::/55
 - 2001:db8:8000::/55
 - ...
 - Cuando es necesario incluir una nueva sede, se hace entre las dos ya asignadas más separadas
 - Inconveniente: las direcciones asignadas a las sedes no son intuitivas

Plan de numeración

■ Numeración de servidores

- Se desaconseja el uso de direcciones consecutivas
- Es preferible el empleo de direcciones pseudoaleatorias
 - El servicio DNS es imprescindible
 - Se dificulta la exploración de direcciones por fuerza bruta
 - Pueden emplearse directamente las direcciones de autoconfiguración, o alguna variante pseudoaleatoria basada en la MAC