**Capa de Red - IPv6 Redes y comunicaciones - 2022**

**Práctica 9**

**Capa de Red - IPv6**

**IPv6**

**10, 11, 13**

**1. ¿Qué es IPv6? ¿Por qué es necesaria su implementación?**

IPV6 es un sucesor de IPv4. La principal motivación fue que el espacio de direcciones IPV4 estaba comenzando a agotarse, a causa de las nuevas subredes y nodos IP que estaban conectándose a Internet (a los que se les estaban asignando direcciones IP únicas).

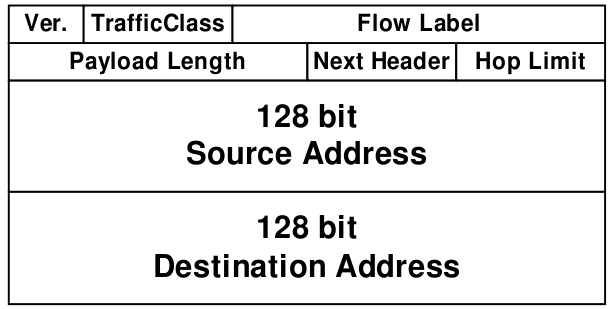
Para responder a esta necesidad de un espacio de direcciones IP más grande, se desarrolló el protocolo IPv6. En este se ajustaron y aumentaron otros aspectos de IPv4, basándose en la experiencia previa.

**Beneficios de IPv6**

* No son versiones del mismo protocolo, IPv4 e IPv6.
* Mayor espacio de direcciones - 128 bits: 340.282.366.920.938.463.463.374.607.431.768.211.456 direcciones.
* Formato de cabecera simplificado.
* Menor overhead de procesamiento.
* Ordenar las tablas de enrutamiento.
* Conectar todo, usar auto-configuración de direcciones (plug and play).
* Arquitectura de red jerárquica para un ruteo eficiente.
* Seguridad a nivel IP (IPSec obligatorio).
* Jumbogramas, size(datagrama) > 64KB 🡪 Puede manejar datagramas más grandes que IP.
* Movilidad y más direcciones de multicast 🡪 Lo mejora de IPV4

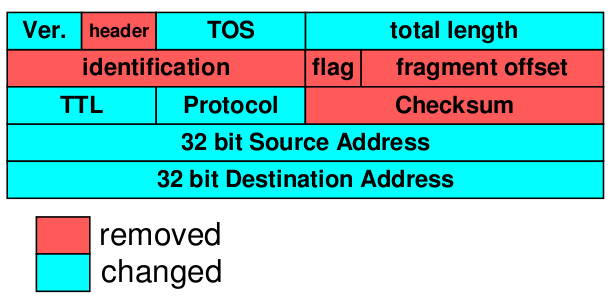
**Cambios en IPv6**

* Direcciones más largas.
* Datagramas de 40 bytes (IPV4: 20 bytes + opt = max 60B)
* Simplifica cabecera:
  + Se saca la fragmentación, se deja solo de extremo a extremo como opción.
  + Se saca checksum de cabecera.
  + Header de tamaño fijo. No existen más las Opciones.
  + Flow Label: identificador de flujo (20 bits).
  + Se renombran los campos: Traffic Class (TOS), Hop Limit (TTL), Next Header (Protocol).
  + Cabeceras de extensión.

**Formato cabecera**

* Ver: Version🡪 6
* Traffic Class: TOS
* Hop Limit: TTL
* Next Header: Protocol
* Payload length: Total Length
* Source address, Destination address: Campos de direcciones.

Cambios en IPv4:



**Cabecera IPv4 vs IPv6**

La 6 es más larga, pero con menos campos y de tamaño fijo, lo que la hace más fácil de leer para el router.



**Servicios Básicos**

Mismas que en IPV4:

o Direccionamiento.

o Ruteo/Forwarding.

o Mux/Demux de protocolos superiores.

o Fragmentación y detección de errores el en header se relegan a capa superior

o Otras: como evitar loops.

Nuevas:

Descubrimiento de Vecinos (NDP):

ND propiamente.

Router discovery y auto-configuación.

Manejo de Grupos de Multicast.

**2. ¿Por qué no es necesario el campo Header Length en IPv6?**

IPv6 tiene una cabecera de 40 bytes simplificada, que elimina algunos campos de IPV4, y permite un procesamiento más rápido del datagrama IP.

Se eliminó el campo opciones y fue reemplazado por cabeceras adicionales y opcionales generando un procesamiento más flexible de las mismas y permitiendo tener una cabecera de 40 bytes fijos, que por ser de tamaño fijo no necesita incluir el campo “Header length”, que era necesario en IPV4 por que su header era variable, debido a su campo opciones y por eso se debía especificar.

**3. ¿En qué se diferencia el checksum de IPv4 e IPv6? Y en cuánto a los campos checksum de TCP y UDP, ¿sufren alguna modificación en cuanto a su obligatoriedad de cálculo?**

En IPv6 se eliminó el checksum que estaba incluido en IPV4, y se delegó a la capa de transporte. En UDP, con esta modificación, la funcionalidad de comprobación de errores será obligatoria.

Suma de comprobación (checksum) de cabecera. Puesto que los protocolos de transporte (Ej. TCP y

UDP) y de la capa de enlace de datos (ej. Ethernet) en las capas de Internet realizan sumas de comprobación, los diseñadores de IP probablemente pensaron que esta funcionalidad era

redundante aquí y podía eliminarse. El procesamiento rápido de los paquetes IP era lo principal.

IPv6 no implementa protección de integridad ya que asume que esta protección se realiza en los protocolos de transporte de capa superior como UDP o TCP, lo que mejora la performance.

Los campos checksum de TCP y UDP no deben cambiar nada en la protección de integridad checksum.

**4. ¿Qué sucede con el campo Opciones en IPv6? ¿Existe, en IPv6, alguna forma de enviar información**

**opcional?**

En IPv6 se sustituye el campo Opciones que se usaba en IPv4, ya que este método de campos de longitud variable era ineficiente, por lo que se cambió por la Extensión de Encabezados, que consiste en una lista de encabezados (Headers de extensión) enlazados mediante un campo llamado “Siguiente encabezado”.

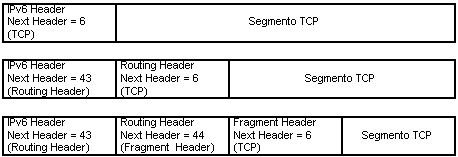
Es posible, enviando los datos en uno de estos headers.

**5. Si quisiese que IPv6 soporte una nueva funcionalidad, ¿cómo lo haría?**

Habría que definir un nuevo tipo de header de extensión, identificado con un valor propio del campo Next Header

**Cabeceras de Extensión (Más servicios)**

* Permite la extensibilidad del protocolo, esto es, más opciones en el encabezado al incorporar los encabezados adicionales. No agregan más trabajo al router por que en general no son para este o es poco lo que tiene que hacer sobre esto. Generalmente solo se ocupa del fijo.
* Mantienen orden (Procesados por nodos intermedios, más cerca del encabezado original, más externo a la pila, más fácil de desacoplar. Más adentro en la pila🡪 Nodos finales).
* Se encuentran a continuación del header.
* En general, son procesadas por los extremos.



1)

2)

3)

*1) Sin cabecera de extensión.*

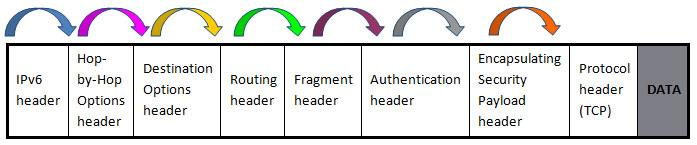
*2) Cabecera de extensión que tiene que ser procesada por el router (Routing header)*

*3) Dos cabeceras de extensión, una para fragmentación que trabaja similar a IPv4, pero no la procesan los routers sino los destinos (Fragment header).*

Orden Cabeceras de Extensión

* Hop-by-hop: procesado por cada router. Datagramas mayores a 64 k.
* Dest Opt: procesado por routers incluidos.
* Routing: procesado por routers, RH0 desaconsejado. Más seguridad.
* Frag, Auth, Sec, Dest. procesado por extremos.

Las tres primeras son pensadas para procesarse por los routers pero en la mayoría de los casos no se usan.



De la practica vieja:

**Tipos de cabeceras de Extensión (RFC 2460):**

■ **De opción Hop-by-Hop (RFC 2460):** Esta cabecera debe ser examinada Salto-a-Salto, es decir, en cada uno de los nodos de la ruta del paquete.

■ **De enrutado (RFC 2460):** Da una lista de uno o más nodos que deben estar en la ruta seguida por un paquete.

■ **De fragmento (RFC 2460):** Un host IPv6 que quiere enviar un paquete a un destino IPv6 utiliza el

llamado “Path MTU discovery” para determinar el tamaño máximo de paquete que se puede utilizar en el path hasta ese destino. Si el paquete que hay que enviar es más grande que el MTU soportado, el host origen fragmenta el paquete. Gracias a esta forma de actuar, la fragmentación se gestiona de extremo a extremo, liberando a los routers del path de este trabajo.

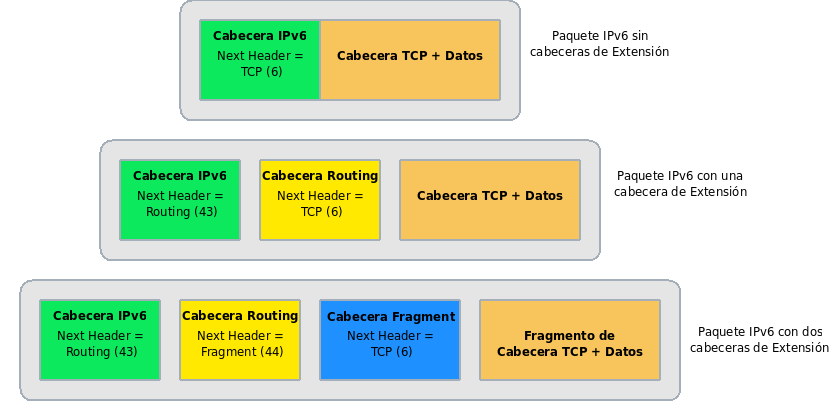
En caso de que el "Path MTU discovery" falle, se usará el valor mínimo de "Path MTU" en IPv6, 1280 bytes. El valor máximo es de 65536 bytes, en los cuales se incluye la carga o payload del datagrama y los 40 bytes de la cabecera.

■ **De opciones de destino (RFC 2460):** Estas cabeceras llevan información que será procesada, exclusivamente, por el nodo de destino.

■ **De autenticación (AH) (RFC 4302):** Proporciona integridad y autenticación (que no confidencialidad) para todos los paquetes de datos IP. Soporta distintos mecanismos de autenticación.

■ **De carga útil de seguridad encriptada (Encrypted Security Payload =ESP) (RFC 4303**): Proporciona integridad, confidencialidad, autenticación de datos y otras funciones para todos los paquetes de datos IP.

Habría que basarse en las cabeceras opcionales (campo Next Header, que indica el tipo del siguiente header) para extender el protocolo IPv6. Pero previamente, habría que desarrollar un RFC y que este se apruebe como estándar (para que lo implementen todos los dispositivos que soportan IPv6). O implementarlo dentro de todos los dispositivos del sistema autónomo (pero solo funcionaría dentro de este).



En IPv6 las opciones se manejan por medio de las llamadas Cabeceras de Extensión (Extension Headers). Estas cabeceras se insertan en el paquete sólo si las opciones son necesarias.

➢ En el **primer paquete** hay una única cabecera IPv6 que precede a los datos de la capa superior de transporte.

➢ En el **segundo paquete** se ha insertado una tercera cabecera entre las dos anteriores. Ahora, la cabecera IPv6 indica que la siguiente cabecera es una cabecera de Extensión del tipo Routing, cuyo código identificativo es el 43 y que se utiliza para dar una lista de uno o más nodos que deben estar en la ruta seguida por un paquete. En el campo Next Header de esa cabecera de Routing se indica ya que a continuación van los datos de TCP.

➢ En el **tercer paquete** se ha insertado una cabecera más. En este caso es una cabecera de Extensión de Fragmento, cuyo código es el 44.

➢ Como se puede ver los campos Next Header de las distintas cabeceras mantienen la lógica explicada.

➢ Podemos enumerar ya algunas cuestiones generales relativas a las cabeceras de extensión:

* En un paquete IPv6 puede haber cero, una o más cabeceras de Extensión.
* Estas cabeceras se sitúan entre la cabecera IPv6 y la cabecera del protocolo de la capa superior (capa de transporte).
* Las cabeceras existentes deben ser procesadas en el orden en que aparecen en la cabecera del paquete.
* Cada cabecera de Extensión es identificada por el campo “Next Header” de la cabecera precedente.
* Las cabeceras de Extensión son examinadas o procesadas solo por el nodo identificado en el campo dirección de Destino de la cabecera IPv6, con una única excepción:
  + Si la cabecera de Extensión es del tipo Opciones Hop-by-Hop, la información que lleva debe ser examinada y procesada por cada uno de los nodos que se encuentran en la ruta del paquete. Este tipo de cabecera debe seguir inmediatamente a la cabecera IPv6 y su valor de "Next Header" es 0.
* Si en el campo "Dirección de destino" hay una dirección multicast, las cabeceras de Extensión serán examinadas y procesadas por todos los nodos que pertenezcan al grupo multicast.
* La longitud de cada cabecera de Extensión es un múltiplo de 8 bytes de forma que, independientemente del número de ellas que se utilicen, siempre quedan alineadas.

La flexibilidad de esta arquitectura permitirá el desarrollo de nuevas cabeceras de Extensión en el futuro, a medida que sean necesarias. Lo bueno de este sistema es que las nuevas cabeceras de Extensión se pueden definir y usar sin cambiar la cabecera IPv6.

**6. ¿Es necesario el protocolo ICMP en IPv6? ¿Cumple las mismas funciones que en IPv4?**

En IPv6 es necesario, en IPv4 no. Los nodos IP pueden gracias al protocolo ICMP, reportar errores e información de diagnóstico y proporcionar información limitada (por ejemplo, la respuesta de eco de un mensaje ping) a un sistema terminal.

En la RFC 4443 se ha definido una nueva versión de ICMP para IPv6. Además de reorganizar las definiciones de tipos y códigos, añade nuevos tipos y códigos requeridos por la nueva funcionalidad de IPv6, como “Paquete demasiado grande” y el código de error “Opciones IPv6 no reconocidas”.

Además, ICMPv6 incluye la funcionalidad del Protocolo de gestión de grupos de Internet (**IGMP**, Internet Group Management Protocol) que se emplea para gestionar el modo en que un host se une a un grupo de multidifusión y lo abandona, anteriormente era un protocolo separado de ICMP en IPv4.

Resuelve: Multicast Listener Discovery (MLD), reemplazo de IGMP. Neighbor Discovery Protocol (NDP), reemplazo de ARP y mensajes Router Discovery, Redirect. Mensajes de control de ICMP: informativos (ping), errores.

Además de cumplir las mismas funciones que en IPv5, fue diseñado para ser fácilmente extendido, usando su framework.

**7. Transforme las siguientes direcciones MACs en Identificadores de Interfaces de 64 bits.**

**00:1b:77:b1:49:a1**

**e8:1c:23:a3:21:f4**

Esto se hace porque la computadora puede calcular su dirección IPv6 local a partir de su MAC (es un identificador de **48 bits** -6 bloques de dos caracteres hexadecimales: 4 bits-) que corresponde de forma única a una tarjeta o dispositivo de red. Se la conoce también como dirección física, y es única para cada dispositivo. Con esa dirección ya **pueden tener una comunicación en la red con los hosts de la misma red (no pueden salir a Internet).**

Una dirección de enlace-local es una dirección IP creada únicamente para comunicaciones dentro de una subred local. Los routers no enrutan paquetes con direcciones de enlace local.

Las direcciones de enlace local se asignan usando los procedimientos de stateless address autoconfiguration para Internet Protocol versión 4 (IPv4) e IPv6. En IPv6, las direcciones de enlace local son necesarias para el funcionamiento interno de varios componentes del protocolo.

Transformo **00:1b:77**:**b1:49:a1**

* Paso 1 → Dividir la dirección MAC entre el OUI y el identificador de dispositivo

OUI → **00:1b:77** → **00000000:00011011:01110111**

ID → **b1:49:a1** → **10110001:01001001:10100001**

* Paso 2 → Insertar el valor hexadecimal FFFE, que en formato binario es: **1111 1111 1111 1110**

**00000000:00011011:01110111**:**1111:1111:1111:1110**:**10110001:01001001:10100001**

* Paso 3 → Convertir los primeros dos valores hexadecimales del OUI a binario e invertir el séptimo (bit U/L)

**0000 0010:0001 1011:0111 0111**:**1111:1111:1111:1110**:**1011 0001:0100 1001:1010 0001**

* Resultado final en hexadecimal: FE80::21b:77 ff:fe b1:49a1/64, que creo que se puede escribir como: FE80:0000:0000:0000:21B:77FF:FEB1:49A1

Transformo **e8:1c:23**:a3:21:f4

En IPV6 las direcciones de enlace local tienen la forma FE80::X/64

* Paso 1 → Dividir la dirección MAC entre el OUI y el identificador de dispositivo

OUI → **e8:1c:23** → **11101000:00011100:00100011**

ID → a3:21:f4 → 10100011:00100001:11110100

* Paso 2 → Insertar el valor hexadecimal FFFE, que en formato binario es: **1111 1111 1111 1110**

**11101000:00011100:00100011**:**1111:1111:1111:1110**:**10100011:00100001:11110100**

* Paso 3 → Convertir los primeros dos valores hexadecimales del OUI a binario e invertir el bit U/L (séptimo bit)

**11101010:00011100:00100011**:**1111:1111:1111:1110**:**10100011:00100001:11110100**

* Resultado final en hexadecimal: FE80::ea1c:23ff:fea3:21f4/64

Otra forma:

MAC: **00**:1b:77:b1:49:a1

* Tomo el primer octeto: **0x00** 🡪 0x indica hexadecimal
* Lo convierto a binario: **0b00000000** 🡪 0b indica binario
* Invierto el segundo bit menos significativo: **0b00000010**
* Lo convierto a hexadecimal: **0x02**
* Reemplazo el primer octeto con el resultado: **02**:1b:77:b1:49:a1
* Inserto **ff:fe** en el medio: **02:1b**:77:**ff:fe**:b1:**49:a1**
* Convierto la dirección al formato IPv6: **021b**:77ff:feb1:**49a1** 🡪 Fusionando de a dos
* Agrego el prefijo Link-Local: **fe80**::021b:77ff:feb1:49a1
* Elimino los 0s más significativos: fe80::21b:77ff:feb1:49a1

MAC: **e8**:1c:23:a3:21:f4

* Tomo el primer octeto: **0xe8**
* Lo convierto a binario: **0b11101000**
* Invierto el segundo bit menos significativo: **0b11101010**
* Lo convierto a hexadecimal: **0xea**
* Reemplazo el primer octeto con el resultado: **ea**:1c:23:a3:21:f4
* Inserto **ff:fe** en el medio: **ea:1c**:23:**ff:fe**:a3:**21:f4**
* Convierto la dirección al formato IPv6: **ea1c**:23ff:fea3:**21f4** 🡪 Fusionando de a dos
* Agrego el prefijo Link-Local: **fe80**::ea1c:23ff:fea3:21f4

**8. . ¿Cuál de las siguientes direcciones IPv6 no son válidas?**

**2001:0:1019:afde::1** Válida, equivale a 2001:0000:1019:afde:0000:0000:0000:0001

**2001::1871::4** Inválida, el :: sólo puede usarse una vez

**3ff**g**:8712:0:1:0000:aede:aaaa:1211** Inválida, los dígitos hexadecimales son de 0 a 9, y A a F

**3::1** Válida, equivale a 0003:0000:0000:0000:0000:0000:0000:1

**::** Válida, equivale a 0000:0000:0000:0000:0000:0000:0000:0000

**2001::** Válida, equivale a 2001:0000:0000:0000:0000:0000:0000:0000

**3ffe:1080:1212:56ed:75da:43ff:fe90:affe** Válida, es la dirección completa de 128 bits

**3ffe:1080:1212:56ed:75da:43ff:fe90:affe:1001** Inválida, no puede tener más de 32 dígitos (128 bits)

**9. ¿Cuál sería una abreviatura correcta de 3f80:0000:0000:0a00:0000:0000:0000:0845?**

**3f80::a00::845** No, :: sólo puede usarse una vez

**3f80::a:845** No se representan :0000:0000:0000:. Además, se sacan 0 a la derecha de a, lo cual no representa al mismo número.

**3f80::a00:0:0:0:845:4567** No es, la dirección original no termina en :4567, son bits agregados

**3f80:0:0:a00::845**

**3f8:0:0:a00::845** No es, se saca un cero en el primer grupo. Abreviaría 03f8:0000:0000:0a00:0000:0000:0000:0845

**10. Indique si las siguientes direcciones son de link-local, global-address, multicast, etc.**

**fe80::1/64**

**3ffe:4543:2:100:4398::1/64**

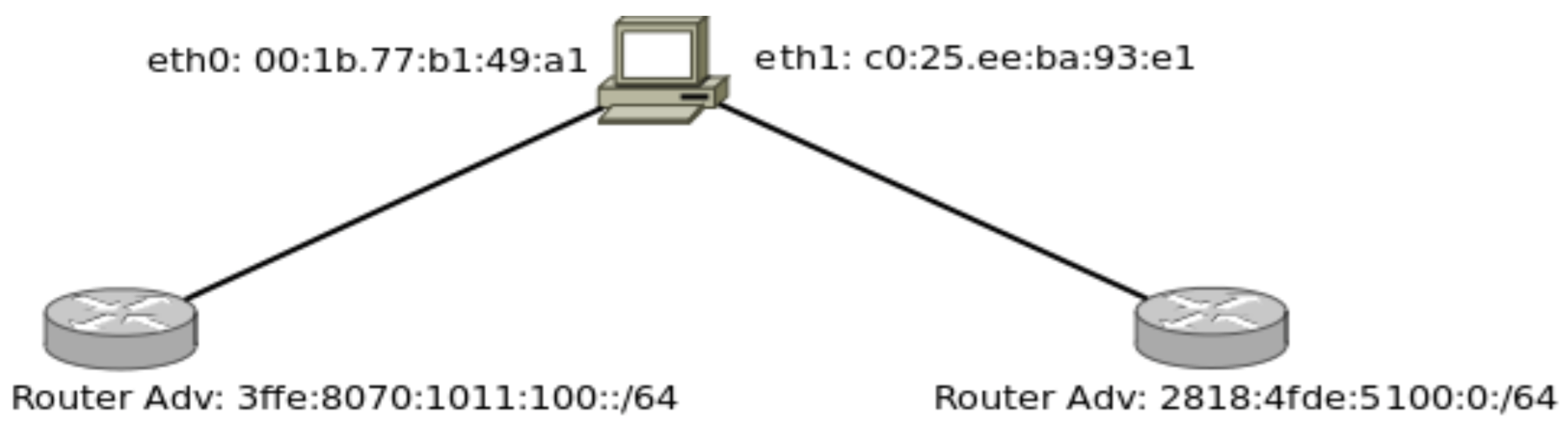
**::**

**::1**

**ff02::2**

**2818:edbc:43e1::8721:122**

**ff02::9**

**11. Dado el siguiente diagrama, ¿qué direcciones IPv6 será capaz de autoconfigurar el nodo A en cada una de sus interfaces?**

La idea sería que el nodo A genere una dir local usando su MAC, luego como el router manda cada tanto sus redes (una por cada interfaz, en general son varias, acá manda una sola), concateno el prefijo de red a lo anterior.

Eth0: **00**:1b:77:b1:49:a1

* Tomo el primer octeto: **0x00** 🡪 0x indica hexadecimal
* Lo convierto a binario: **0b00000000** 🡪 0b indica binario
* Invierto el segundo bit menos significativo: **0b00000010**
* Lo convierto a hexadecimal: **0x02**
* Reemplazo el primer octeto con el resultado: **02**:1b:77:b1:49:a1
* Inserto **ff:fe** en el medio: **02:1b**:77:**ff:fe**:b1:**49:a1**
* Convierto la dirección al formato IPv6: **021b**:77ff:feb1:**49a1** 🡪 Fusionando de a dos
* Agrego el prefijo pedido al router (Prefijo de red): **3ffe**::021b:77ff:feb1:49a1
* Elimino los 0s más significativos: 3ffe::21b:77ff:feb1:49a1

- eth0) Primero genera la fe80::21b:77ff:feb1:49a1 /64. Recibe los prefijos de red y arma: 3ffe:8070:1011:100:21b:77ff:feb1:49a1 /64

eth1: **c0**:25:ee:ba:93:e1

* Tomo el primer octeto: **0xc0**
* Lo convierto a binario: **0b11000000**
* Invierto el segundo bit menos significativo: **0b11000010**
* Lo convierto a hexadecimal: **0xc2**
* Reemplazo el primer octeto con el resultado: **c2**:25:ee:ba:93:e1
* Inserto **ff:fe** en el medio: **c2:25**:ee:**ff:fe**:ba:**93:e1**
* Convierto la dirección al formato IPv6: **c225**:eeff:feba:**93e1**
* Agrego el prefijo pedido al router (Prefijo de red): **2818**::c225:eeff:feba:93e1

- eth1) Primero genera la fe80::c225:eeff:feba:93e1 /64. Recibe los prefijos de red y arma:

2818:4fde:5100::c225:eeff:feba:93e1 /64

**12. Al autogenerarse una dirección IPv6 sus últimos 64 bits en muchas ocasiones no se deducen de la**

**dirección MAC, se generan de forma random, ¿por qué sucede esto? ¿Qué es lo que se intenta evitar?**

**(Ver direcciones temporarias, RFC 8981)**

Esto sucede a propósito, siguiendo la RFC 8981, con la finalidad de aumentar la privacidad de los usuarios, al dificultar el seguimiento del tráfico que produce en la red. Además, como uno podría obtener su IP mientras está navegando, hacer que cambie limita el tiempo en el que uno podría aprovechar conocerla

**13. Utilizando la máquina virtual abrir la topología llamada 3-ruteo-OSPF.imn para realizar las siguientes pruebas:**

**a. Habilitar la vista de las direcciones IPv6 en la topología (View ->show ->IPv6 Addresses).**

**b. Esperar a que la red converja. Verificar, mediante ping6, la comunicación entre n6 y n7.**

**c. Observar la configuración IPv6:**

**i. De la PC n6.**

**ii. De la PC n7.**

**iii. Del router n1.**

**iv. La tabla de rutas tanto de las PCs como de los routers.**

**d. Responda:**

**i. ¿Cuántas direcciones IPv6 se observan tanto en la PC n6 como en la PC n7?**

**ii. ¿Es posible desde la PC n7 hacer un ping6 a cada una de las direcciones IPv6 de la PC n6?**

**¿Por qué?**

**e. Cuando se quiere hacer ping6 a una dirección link-local es necesario especificar la interfaz que se**

**quiere utilizar (ping6 -I eth0 <IPv6-address>) ¿Por qué?**

**f. Deshabilite la configuración de IPv6 en la PC n7 mediante el comando:**

**sysctl -w net.ipv6.conf.all.disable\_ipv6=1**

**i. Verifique las IPs configuradas en la PC.**

**ii. Luego de deshabilitarse IPv6, ¿puede comunicarse con la PC n6? ¿Cómo?**