# Capítulo 3: Mecanismos de Transición a IPv6

Con la creación de un nuevo protocolo para resolver el problema de direccionamiento que presentan actualmente la redes de comunicaciones, es necesario que se piense en un método que permita la migración de IPv4 a IPv6. Este capítulo hace mención de estos métodos que son conocidos como Mecanismos de Transición. Existen tres bloques básicos definidos por la IETF: Pila-dual, Traducción y *Tunnneling*.

## 3.1 Introducción

Como es de notarse, IPv6 es el candidato potencial para la siguiente generación del Protocolo de Internet. Sus creadores no lo consideran en sí como un protocolo revolucionario diseñado para reemplazar a la existente versión 4 (IPv4); sino mas bien, es considerado como la mejora más esperada de este protocolo desde 1981 [RFC 2373]. Gran parte de las mejoras de IPv6 están basadas en problemas y situaciones existentes en el Internet actual.

Como tecnología promete avances, que incluyen:

- Un mayor espacio para direcciones y un esquema flexible de direccionamiento
- Mayor eficiencia en el envío de paquetes
- Soporte para la seguridad en las comunicaciones
- La habilidad de permitir servicios diferenciados
- Mayor soporte a la movilidad
- Fácil manejo

En general, las aplicaciones legadas por IPv4 necesitan reescribirse para dar soporte a IPv6, por ejemplo el Protocolo de Transporte de Archivos (*File Transport Protocol*, FTP) trabaja con direcciones IP en su protocolo, esto requiere cambios en las aplicaciones tanto del cliente como del servidor. Debido a esto Internet tiende a convertirse en un complejo conglomerado de diversos protocolos, ya que IPv4

coexistirá con IPv6 junto con otros protocolos globales estandarizados. Su desarrollo e implementación no está planeado de la noche a la mañana. Para que esto suceda y el Internet evolucione hacia IPv6, al principio existirán "islas" entre sí, pero es necesario utilizar algunos métodos que colaboren con la transición.

La transición a IPv6 no es enteramente transparente a lo que se refiere a las capas superiores de IP. Las direcciones IPv6 son más largas que las direcciones IPv4, lo que requiere un cambio en la aplicación de las estructuras de los datos más que en la asignación de direcciones. Como consecuencia a esto, Interfaces de Programación de Aplicaciones (APIs, *Application Programming Interfaces*) deben ser extendidas para dar soporte tanto a IPv4 como IPv6; así como, la habilidad de seleccionar el protocolo apropiado para cada comunicación entre usuarios [KOT05]. Se debe prever que este proceso de transición evolucionará y tomará la forma de nodos duales (*dual-stacked nodes*) en los cuales cada nodo de Internet puede ser tanto IPv4 como IPv6. Éste es el mecanismo más básico de transición utilizado. Sin embargo, esto puede causar cierta complejidad debido a la exigencia, en funcionalidad, de los sistemas finales.

Las razones principales para que IPv6 crezca para prevalecer como su antecesor IPv4 son dos. Primero, el número de mecanismos de transición propuestos por la IETF brindan a los administradores de red un camino fácil hacia la migración permitiendo a los nodos de una red, más específicamente a las aplicaciones de estos nodos, comunicarse entre sí. Segundo, dominios de aplicaciones especializadas con su respectivo mercado de interés, particularmente el área móvil, demandan características IP que no pueden satisfacerse sólo con IPv4, tal como es el caso de una mayor disponibilidad en el número de direcciones y su fácil configuración.

### 3.1.2 Mecanismos de transición y aproximaciones

El cambio de IPv4 a IPv6 ya ha comenzado. Durante 20 años se espera que convivan ambos protocolos y que la implantación de IPv6 sea paulatina [WAD02]. Existen una serie de métodos que permitirán la convivencia y la migración progresiva tanto de las redes como de los equipos de usuario. En general, éstos se agrupan en tres componentes mayores:

- Pila-dual (*dual-stack*)
- Traducción de encabezado
- *Tunneling*

La pila-dual como su nombre lo sugiere, significa literalmente mantener dos pilas de protocolos que trabajen paralelamente y así permitir al dispositivo trabajar vía ambos protocolos.

El mecanismo de la traducción utiliza, como su nombre lo indica, la traducción de encabezados para lograr la transición y se puede considerar transparente cuando el tráfico es ruteado inherentemente a través de un traductor en la red. Este mecanismo puede ser del tipo *stateless* o *stateful*. Un traductor tipo *stateless* es capaz de procesar cada conversión individualmente sin ninguna referencia de paquetes previamente traducidos; un traductor de tipo *stateful* necesita mantener la forma con respecto al estado de una traducción previa. Se debe mantener alguna relación entre las direcciones IPv4 e IPv6.

El bloque final en la transición es el *tunneling*. Este mecanismo es utilizado para enlazar nodos compatibles a través de una red incompatible. Esto se puede ver de manera técnica como la transferencia de la carga útil de un protocolo entre dos nodos por medio de otro protocolo portador que se encarga de encapsularla. Este encapsulamiento se lleva a cabo en la entrada de un túnel y ese desencapsula a la salida de ese mismo túnel (existe, por lo tanto, una dirección lógica a un túnel dado). Esta asociación lógica entre la entrada y salida de un túnel es lo que define el túnel. Desde diversas perspectivas de transición IPv4/IPv6, *tunneling* es utilizado en la mayoría de los casos para relacionar segmentos IP incompatibles: una carga IPv6 sobre un portador IPv4, o una carga IPv4 sobre un portador IPv6.

La Tabla 3.1 muestra los mecanismos de transición actuales propuestos por la IETF, en tipo de su conectividad, y los elementos que requieren para desarrollarse. La conectividad se refiere a la relación entre el nodo que inicia sesión y su nodo correspondiente. Por ejemplo, 6-a-4 significa: un nodo que trabaja con IPv6 solo es capaz de corresponder a un nodo que soporta solo IPv4. La comunicación por lo general es bi-direccional; sin embargo, el orden 6-a-4 infiere que IPv6 es el que inicia la sesión.

*Tabla 3.1: Clasificación de la IETF de los mecanismos de transición [WAD02].*

| Nombre | Conectividad | Tipo |
|---|---|---|
| Pila-dual | 4-a-4 sobre 4, 6-a-6 sobre 6 | Pila Doble |
| SIIT | 6-a-4, 4-a-6 | Traducción |
| *Bump-in-Stack* (BIS) | 4-a-6 | Traducción |
| *Bump-in-API* (BIA) | 4-a-6 | Traducción |
| NAT-PT (*Network Address Translation-Protocol Translation*) | 6-a-4, 4-a-6 | Traducción |
| MTP (*Multicast Translator base don IGMP/MPL Proxying*) | 4-a-6, 4-a-6 (tipo *multicast*) | Traducción |
| TRT (*Transport Relay Translator*) | 6-a-4 | Traducción |
| SOCKS64 | 4-a-6, 4-a-6 | Traducción |
| 6over4 | 6-a-6 sobre 4 | Túnel |
| ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*) | 6-a-6 sobre 4 | Túnel |
| DSTM (*Dual Stack Transition Mechanism*) | 4-a-4 sobre 6 | Túnel |
| IP-en-IP Configurado | 6-a-6 sobre 4, 4-a-4 sobre 6 | Túnel |
| 6to4 | 6-a-6 sobre 4 | Túnel |

# 3.2 Pila Doble

El primer paso en la migración es el desarrollo de sistemas que soporten IPv6. Al principio, cuando menos, será probable que estos sistemas no tengan otros sistemas con los cuales comunicarse, la gran mayoría de los sistemas estará utilizando IPv4. Estos sistemas duales pueden utilizar IPv6 para comunicarse con sistemas iguales, y al mismo tiempo pueden retroceder para comunicarse con sistemas "viejos" que manejen IPv4.

Este mecanismo, como su nombre lo sugiere, refiere al uso de dos pilas, de diferente protocolo, que trabajan paralelamente y permiten al dispositivo trabajar vía ambos protocolos. En el esquema de pila- dual [RFC 2893] el nodo instala pilas IPv4 e Ipv6 en paralelo (ver Figura 3.1) y se conocen como nodos IPv4/IPv6.

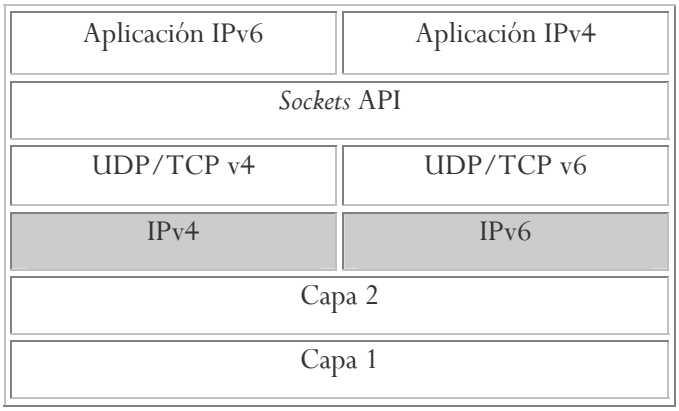


*Figura 3.1: Esquema de pila doble [WAD02].*

Los nodos IPv4/IPv6 procesan las aplicaciones IPv4 utilizando la pila IPv4, mientras que para las aplicaciones IPv6 utilizan la pila IPv6. Hay que notar, que este mecanismo solo es útil para nodos similares; es decir, IPv6-IPv6 e IPv4-IPv4 (ver Tabla 3.1). Las decisiones de flujo se basan en el encabezado de IP, en su campo versión para recibir y en la dirección destino para enviar. Aunque el nodo cuente con ambas pilas para operar, una de estas dos tiene que ser desactivada por razones de operabilidad. De esta forma los nodos IPv6/IPv4 pueden operar en uno de tres modos [RFC 2460]:

- Pila IPv4 habilitada y la pila IPv6 deshabilitada.
- Pila IPv6 habilitada y la pila IPv4 deshabilitada.
- Con ambas pilas habilitadas.

Nodos IPv4/IPv6 con pila IPv6 deshabilitada trabajan como un nodos IPv4 enteramente. Similarmente para los que trabajan con la pila IPv4 deshabilitada, su comportamiento será como el de un nodo IPv6.

Actualmente muchos sistemas comerciales operantes ya cuentan con un protocolo IP de pila-dual. En consecuencia, esto lo hace el mecanismo más utilizado en la solución de transición. La pila dual hace referencia sólo a una solución de nivel IP [RFC 2893]. Se necesita de más para completar una solución que permita la comunicación IPv6-IPv4 e IPv4-IPv6 (la pila dual puede o no utilizarse en conjunto con las técnicas de *tunneling*, sección 3.4.

## 3.2.1 Configuración de Direcciones

Debido a que este la Pila-dual soporta ambos protocolos, cada nodo IPv4/IPv6 debe ser configurado con dos direcciones una de tipo IPv4 y otra IPv6. Para obtener la dirección IPv4 se utilizan mecanismos como DHCP, mientras que para IPv6 se utilizan otros diferentes (e.g. auto-configuración estática de direcciones) o por medio de direcciones compatibles (ver sección 2.4.2).

Un nodo IPv4/IPv6 con una dirección IPv4 compatible, utiliza esa misma dirección como una dirección IPv6 incluyéndola en los últimos 32 bits. Esto se representa con la siguiente sintaxis:

**x:x:x:x:x:x:d.d.d.d**

donde x representa valores hexadecimales de las seis primeras partes más significativas (de 16 bits cada una) y las d, son valores decimales de las 4 partes menos significativas (de 8 bits) que corresponden a cada uno de los octetos de una dirección IPv4 estándar. Por ejemplo, la siguiente Figura muestra lo que se quiere decir:

| 96 bits | 32 bits |
|---|---|
| **x:x:x:x:x:x:<br>0:0:0:0:0:0:** | **d.d.d.d<br>13.1.68.3** |
| Prefijo de red | Identificador de usuario |

*Figura 3.2: Sintaxis de las direcciones IPv4 compatibles.*

El nodo dual debe adquirir su dirección compatible vía cualquier mecanismo de adquisición para obtener su dirección IPv4; después, mapearla en otra compatible tipo IPv6. Una característica de estas direcciones, es su constante prefijo "nulo" de 96 bits: 0:0:0:0:0:0. Este modo de configuración intenta permitir el "acomodamiento" de IPv6 de una manera más sutil. El algoritmo utilizado para la obtención de direcciones IPv4 compatibles (basado en protocolos de configuración), según el RFC 2893, es el siguiente:

1. Obtención de una dirección IPv4 vía :
   - DHCP ( *Dynamic Host Configuration Protocol*)]

- BOOTP (*Bootstrap Protocol*)
- RARP (*Reverse Address Resolution Protocol*)
- Configuración manual
- Algún otro mecanismo el cual incluye en el nodo la ya existente dirección IPv4.

2. Que el nodo utilice esta dirección para identificar la interfase.
3. Se anteponga el prefijo nulo de 96 bits a la dirección IPv4 adquirida.

El resultado final es una dirección IPv4 compatible con IPv6 para que el nodo pueda utilizar esta dirección como una dirección IPv6.

## 3.3 Mecanismo de Traducción o SIIT (*Stateless IP/ICMP Translation algorithm*):

Debido a que el mecanismo de pila doble, mencionado en la sección 3.2, no resuelve enteramente la situación "de la transición IPv4 e IPv6 por sí mismo, se tiene un segundo bloque auxiliar disponible que es la Traducción. El término de Traducción se refiere a la conversión directa de protocolos (entre IPv4 e IPv6) de manera bidireccional y puede incluir una transformación tanto del encabezado como de la carga efectiva del protocolo (ver Figura 3.3).

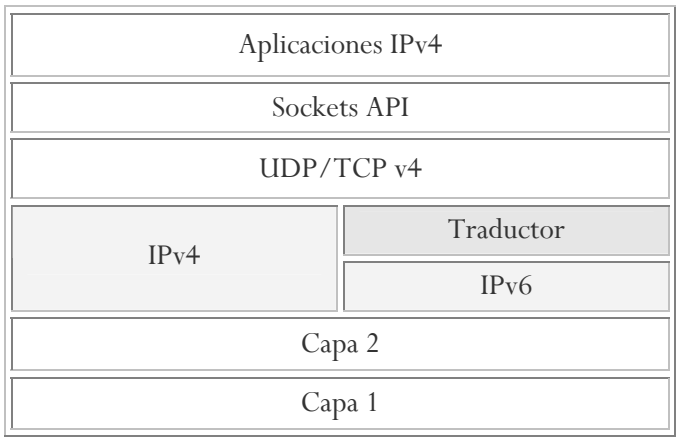


*Figura 3.3: Mecanismo de traducción que trabaja en capa IP [WAD02].*

La traducción puede ocurrir en diferentes capas de la pila, incluyendo la capa de red, transporte y aplicación [WAD02].

Las Figuras siguientes muestran como el algoritmo SIIT puede utilizarse para pequeñas redes y más tarde para sitios que tienen usuarios únicamente IPv6 en una red de pila doble.

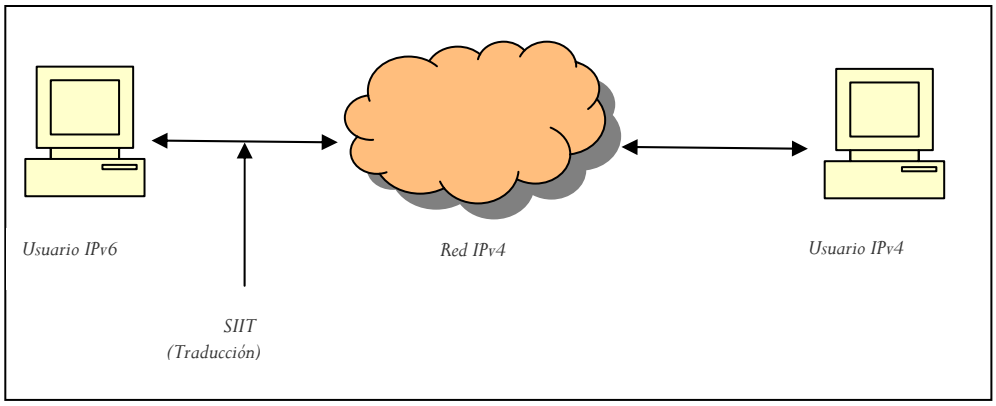


*Figura 3.4: Uso de SIIT para una sola red tipo IPv6 únicamente [RFC 2765].*

Los traductores empleados en los nodos finales pueden resolver los problemas de interoperabilidad. Estos son relativamente fáciles de implementar, pero son más difíciles de manejar a gran escala [RFC 2765]. Los dos más populares propuestos actualmente por la IETF son BIS y BIA (mencionado más adelante). Ambos orientados a permitir que aplicaciones IPv4 operen sobre una red IPv6 [WAD02].

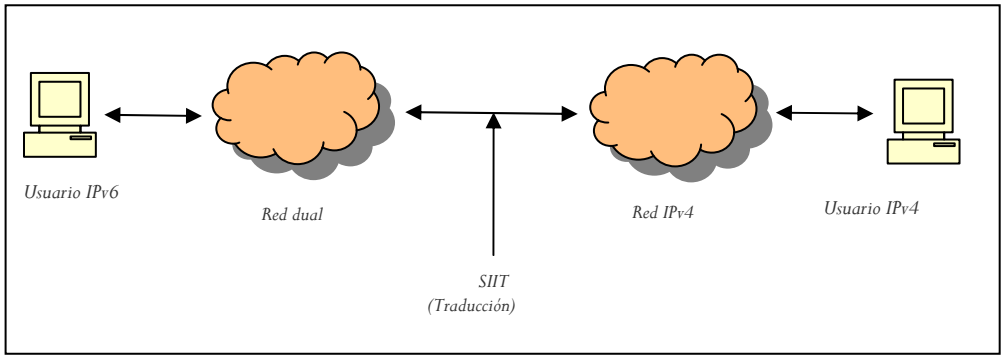


*Figura 3.5: Uso de SIIT para una red IPv6 o de pila-dual que contiene usuarios IPv6 e IPv4 [RFC 2765].*

La traducción puede llevarse a cabo de dos maneras:

- IPv4 a IPv6
- IPv6 a IPv4

A continuación se hablará de estos dos tipos de traducción.

## 3.3.1 Traducción de IPv4 a IPv6

Cuando el traductor recibe un datagrama IPv4 que contiene una dirección destino que está fuera de la red IPv4, entonces traduce el encabezado de ese datagrama por uno IPv6 y lo reenvía basándose en la dirección IPv6 destino. Una descripción básica y rápida de esta traducción consiste en que el encabezado IPv4 del paquete es removido y reemplazado por uno IPv6. La Figura 3.6 muestra como es esto:

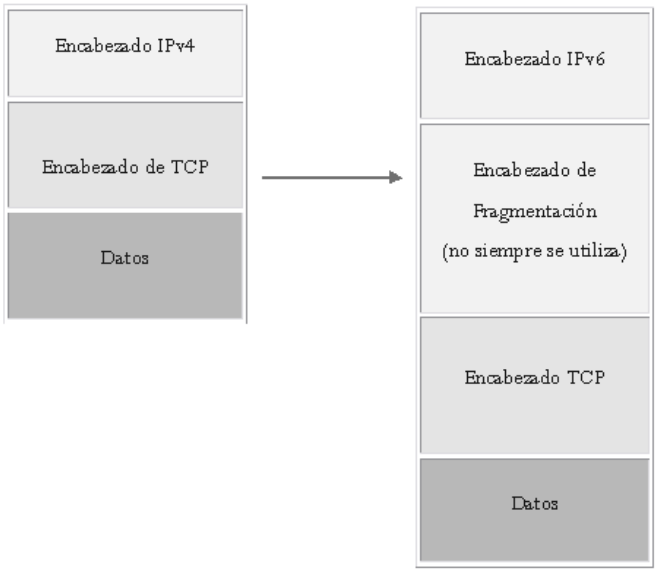


*Figura 3.6: IPv4-a-IPv6 traducción.*

En este proceso de traducción hay que tomar en cuenta que una diferencia entre IPv4 e IPv6 (aparte de las ya mencionadas en el capítulo 2) es que la detección del camino de MTU es mandatario para IPv6 pero opcional para IPv4, lo que lleva a que la fragmentación, si es necesario, se haga desde el nodo fuente y se especifique en el nuevo encabezado de IPv6.

### 3.3.1.1 Traducción de encabezados

De acuerdo con el RFC 2765, para efectuar la traducción de encabezados de IPv4 a IPv6 es necesario considerar ciertas condiciones a priori de realizar la traducción.

*Tabla 3.2: Traducción de encabezados IPv4 a IPv6 [RFC 2765].*

| Campo | IPv6 (*8 campos*) | IPv4 (*12 campos*) |
|---|---|---|
| **Versión** | 6 | 4 |
| **Clase de Tráfico** | Copiada directamente del campo de TOS (los 8 bits se copian iguales) ** | TOS |
| **Etiqueta de Flujo** | Todos los bits en cero | |
| **Longitud Total** | Se establece restando el valor de la Longitud Total del encabezado de IPv4 con el tamaño del encabezado de IPv4 (junto con sus opciones si hay). | Longitud Total |
| **Siguiente Encabezado** | Copiado directamente del campo de Protocolo en el encabezado IPv4. | Protocolo |
| **Límite de salto** | Copiado del valor TTL. *** | TTL |
| **Dirección fuente** | La dirección fuente IPv4 es colocada en los últimos 32 bits, mientras que los 96 bits superiores restantes son sustituidos por un prefijo (::ffff:0:0/96) | |
| **Dirección destino** | Se realiza la misma operación que con la dirección fuente, sólo que en esta vez se utiliza la dirección destino y se utiliza un prefijo diferente (0::ffff:0:0:o/96) | |

*** En algunos ambientes, este campo debe ser utilizado con al antigua semántica del TOS y Precedencia, y la implementación de un traductor debe proveer la habilidad de ignorar el TOS de IPv4 y poner en ceros la Clase de Tráfico.*

**** Como parte del envío el paquete debe decrementar tanto el IPv4 TTL (antes de la traducción) o el IPv6 Límite de Salto (después de la traducción).*

Si la bandera DF (D*on't Fragment*) en el encabezado IPv4 es cero y el tamaño del paquete va a resultar en un paquete mayor a 1280 bytes, el paquete IPv4 debe fragmentarse antes de hacer la traducción. Debido a que los paquetes con DF igual a cero siempre resultarán en un encabezado fragmentado, el paquete IPv4 debe fragmentarse de tal manera que su extensión (excluyendo el encabezado IPv4) debe ser de al menos 1232 bytes (los 48 bytes de diferencia son debido a los 40 bytes correspondientes al encabezado de IPv6 y a 8 del encabezado de Fragmentación). Si el bit de DF está en alto, 1, y el paquete no está fragmentado, y no hay necesidad de agregar un encabezado de fragmentación.

## 3.3.2 Traducción de IPv6 a IPv4

Por el contrario de la sección 3.3.1, cuando el traductor recibe un datagrama IPv6 destinado a una dirección IPv4-mapeada, éste traduce el encabezado IPv6 a un encabezado IPv4. Nuevamente, el encabezado original es removido y sustituido, en este caso, por un encabezado IPv4. La Figura 3.7 muestra como sucede esto:

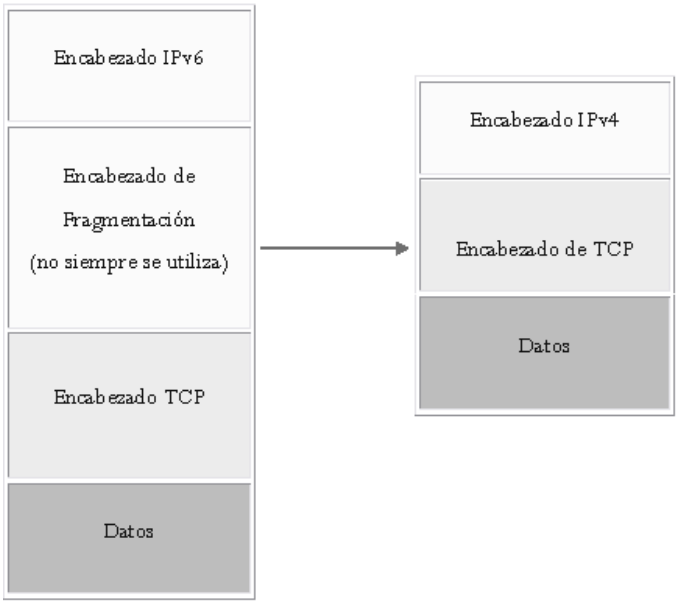


*Figura 3.7: Traducción IPv6-a-IPv4 [RFC 2765].*

Existen algunas diferencias entre la fragmentación IPv6 e IPv4 y el mínimo MTU por enlace que efectúa la traducción. Un enlace IPv6 tiene un MTU mínimo de 1280 bytes. El límite correspondiente para IPv4 es de 68 bytes [RFC 2765].

Así, a menos de que existan medidas especiales, el reconocimiento punto a punto de MTU cuando se incluye un traductor IPv6-a-IPv4 no sería posible, ya que el nodo IPv6 puede recibir un mensaje ICMP "paquete demasiado grande" originado por un ruteador de IPv4 que detecta un MTU menor a 1280. Sin embargo, IPv6 necesita que todos sus nodos puedan manejar tales mensajes ICMP reduciendo el tamaño de MTU del enlace a 1280 e incluyendo un encabezado de fragmentación en cada paquete. Al reducirse el tamaño de MTU en el enlace por debajo de 1280, el nodo IPv6 fuente originará paquetes de 1280 bytes que serán fragmentados por lo ruteadores IPv4 después de haber sido traducidos [RFC 2765].

Por el contrario, si el tamaño de MTU en un enlace IPv4 es menor, entonces el nodo IPv6 fuente no recibirá ningún mensaje ICMP de error y no podrá ajustar el tamaño de fragmentos que estará mandando. Para una mayor referencia y explicación de cómo se maneja esto consultar el RFC 2765.

### 3.3.2.1 Traducción de encabezados

Si no está presente un encabezado IPv6 de Fragmentación, el encabezado IPv4 queda de la siguiente manera:

Si el campo de Fragmentos, diferente de cero, está presente, el paquete no debe ser traducido y a su vez un mensaje ICMPv6 "*parameter problem/ erroneous header field encountered*" debe ser enviado.

*** En algunos ambientes, este campo debe ser utilizado con al antigua semántica del Tipo de Servicio (TOS) y Precedencia, y la implementación de un traductor debe proveer la habilidad de ignorar el TOS de IPv4 y poner en ceros la Clase de Tráfico.*

**** Debido a que el traductor es un ruteador, como parte del envío de paquetes que realiza, necesita decrementar o el límite de salto (antes de la traducción) o el IPv4 TTL (después de la traducción).*

***** Si no se tiene una dirección IPv4-mapeada, la dirección destino es escrita como 0.0.0.0, con la finalidad de evitar completamente que se descarte el paquete o que algún error sea marcado.*

*Tabla 3.3: Traducción de encabezados IPv6 a IPv4 de acuerdo con en RFC 2765.*

| Campo | IPv4 (*12 campos*) | IPv6 (*8 campos*) |
|---|---|---|
| **Versión** | 4 | 6 |
| **IHL** | 5 (no IPv4 opciones) | |
| **TOS** | Copiado directamente del encabezado de Clase de Tráfico (los 8 bits completos). La semántica utilizada para IPv4 e IPv6 es la misma. ** | Clase de Tráfico |
| **Longitud Total** | Este valor se obtiene sumando la Longitud total del encabezado IPv6 más el tamaño del encabezado de IPv4. | Longitud Total |
| **Identificación** | Todos los bits en cero | |
| **Banderas** | M (*More Fragment*) =0<br>DF (*Don't Fragment*)=1 | |
| **Offset** | Todos los bits en cero | |
| **TTL** | Copiado del campo IPv6 Límite de Salto *** | Límite de Salto |
| **Protocolo** | Copiado del valor del Encabezado Siguiente. | Encabezado Siguiente |
| **Comprobación** | Calculado una vez que el encabezado IPv4 ha sido creado | |
| **Dirección fuente** | Si la dirección origen es una dirección IPv4 traducida, entonces se toman lo últimos 32 bits de la dirección IPv6 fuente y se copian directamente. **** | |
| **Dirección destino** | Los paquetes IPv6 que son traducidos tienen una dirección IPv4-mapeada por dirección destino. Así, los últimos 32 bits son copiados directamente en el campo de dirección destino. | |

Si el paquete IPv6 contiene un encabezado de Fragmentación, la traducción (arriba mencionada) se realiza de la misma manera sólo con algunas excepciones:

**Longitud total:** Longitud total del encabezado IPv6 menos 8 (correspondiente al encabezado de Fragmentación), más el tamaño del encabezado de IPv4.

**Identificación:** Se copian los últimos 16 bits del campo de Identificación del encabezado de Fragmentación.

**Banderas:** La bandera de Más Fragmentos toma el valor de la bandera M en IPv4. La bandera DF se asigna a cero, permitiendo de esta manera que los que paquetes puedan ser fragmentados por IPv4.

***Offset*:** Se copia del campo de *Offset* de Fragmento del encabezado de Fragmentación.

Para comprender mejor los campos pertenecientes a los encabezados Extras, se puede consultar el RFC 2460.

## 3.4 *Tunneling*

El uso de la Pila-dual limita el aprovechamiento de todo el campo de direccionamiento que ofrece IPv6, ya que para hacer "compatibles" estos protocolos se utilizan direcciones IPv4 compatibles (ver sección 3.2.2). El mecanismo de traducción propuesto seguidamente, suena a una solución atractiva, ya que no presenta hay un límite en cuanto a direcciones se refiere. A decir verdad, es quizás este mecanismo uno de los más agresivos utilizados durante la migración. Con agresivo nos referimos a que, si se observa bien el proceso y se analizan los encabezados generados y los eliminados, podemos ver como al realizar la traducción, ciertos campos son perdidos totalmente en este proceso.

Ahora bien, no es sólo el direccionamiento lo que en realidad tacha a los métodos anteriores como "inefectivos". Como se muestra en la Tabla 3.1 y en la Figura 3.5, el tipo de conectividad (el soporte que brinda) es lo que interesa. La pila doble solo se realiza para ambientes compatibles, por ejemplo IPv4 a IPv4 sobre un ambiente versión 4 (Figura 3.8a) o IPv6 a IPv6 sobre un ambiente IPv6 (Figura 3.8b), pero nunca para un ambiente mixto como IPv4 a IPv4 sobre IPv6.

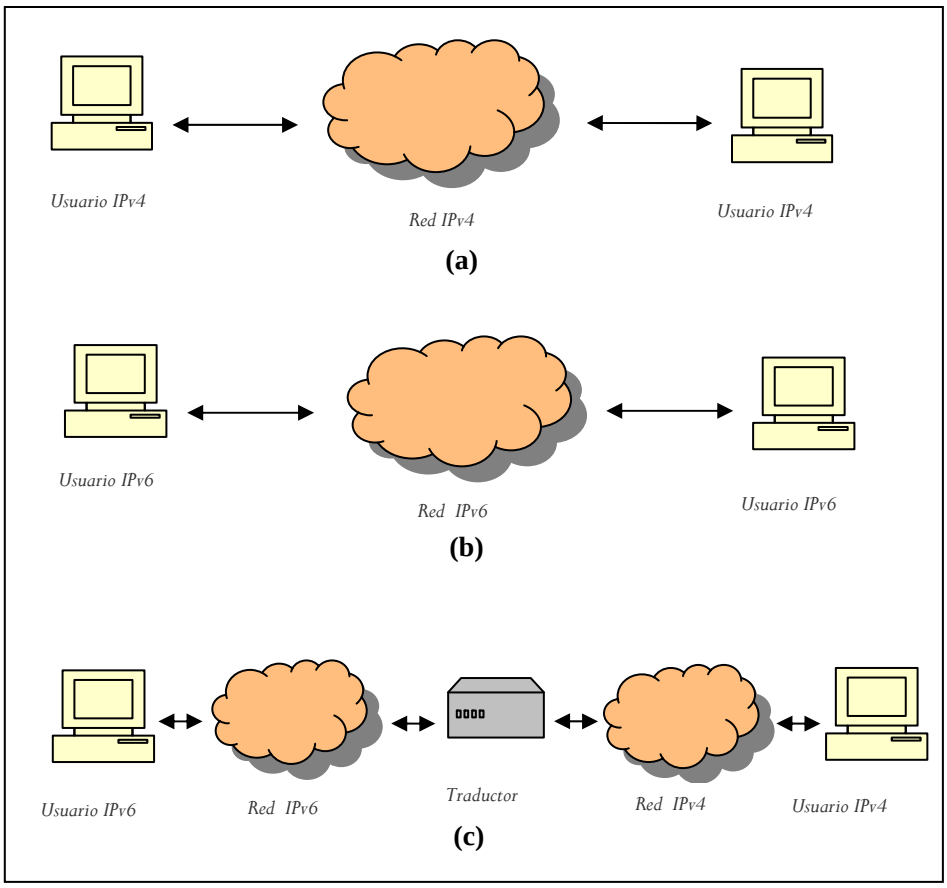


*Figura 3.8: Pila doble (a) Ipv4-a-IPv4 sobre un ambiente IPv4, (b) IPv6-a-IPv6 sobre un ambiente IPv6, (c) Traducción.*

Con respecto a la traducción, se puede decir que trabaja de manera similar a lo que es la pila doble; ya que como se puede ver en la Figura 3.8c, el traductor acepta paquetes de cualquiera de los protocolos (como si fuera un nodo IPv4/IPv6) permitiendo que la comunicación sea como la de la pila doble IPv4-a-IPv4 o IPv6-a-IPv6 y los traduce según las necesidades de envío. Este mecanismo de transición permite crear un "puente" entre redes incompatibles y se desarrolla típicamente de manera secuencial o punto-a-punto y es llamado *Tunneling* (por su falta de traducción al español, se manejará el término anglosajón para referirse a él). El RFC 2473 define al IPv6 *Tunneling*, como una técnica para establecer "enlaces virtuales" entre dos nodos IPv6 (ver Figura 3.9). Desde el punto de vista de los nodos, este "enlace virtual" es lo que se conoce como túneles IPv6 y que actúan como un enlace punto a punto.

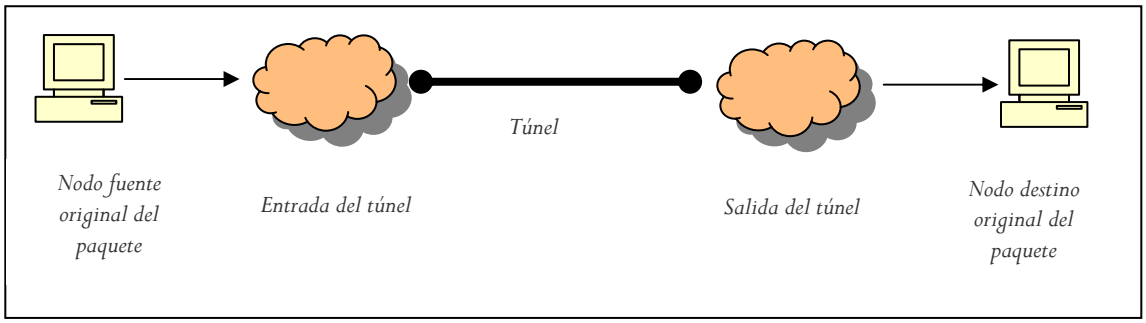


*Figura 3.9: Proceso de tunneling[RFC 2473].*

Los nodos mostrados en la Figura anterior, juegan un papel específico. Uno de ellos se dedica a encapsular los paquetes recibidos de otros nodos y los reenvía por un túnel. El otro nodo desencapsula el paquete que llega del túnel y reenvía el paquete original recién desencapsulado a su destino final. El nodo encapsulador es llamado nodo de entrada o entrada del túnel, mientras que el nodo desencapsulador, es llamado nodo de salida o destino del túnel.

Estos "túneles" creados, permiten conectar dos redes IPv6 extremas "saltando" por encima de redes IPv4. Éstos trabajan encapsulando los paquetes IPv6 en paquetes IPv4 teniendo como siguiente capa la de IP (número 41). De esta manera los paquetes IPv6 pueden ser enviados sobre una infraestructura IPv4 (o viceversa) pero sólo en una dirección. Si se desea un mecanismo bidireccional, es necesario juntar dos procesos unidireccionales, esto es: configurar dos túneles (ver Figura 3.10). Hay muchas tecnologías de túneles disponibles. La principal diferencia se encuentre en el método que utilizan los nodos encapsuladores (los que encapsulan al principio del túnel) para determinar la dirección a la salida del túnel.

Los escenarios más comunes para habilitar el *Tunneling* son:

- Permitir a los usuarios finales de un sistema, utilizar dispositivos de transición (como ruteadores de pila-dual) que estén fuera del enlace en una red de transición escasamente distribuida.
- Habilitar dispositivos de borde de red para que se puedan interconectar con redes incompatibles.

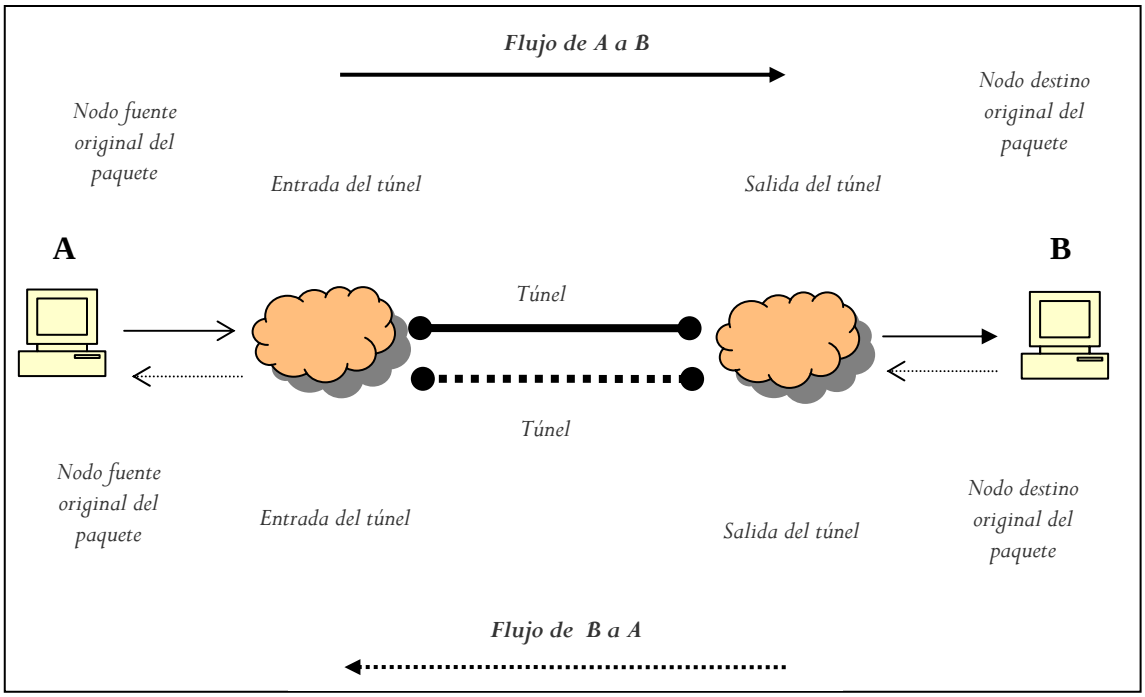


*Figura 3.10: Tunneling bi-direccional [RFC 2473].*

El problema principal en el despliegue del túnel es la configuración de los puntos finales, la definición de dónde se llevará a cabo el encapsulamiento y a qué paquetes debe ser aplicado. Las direcciones de los puntos finales del túnel generalmente son obtenidas por:

- Por asignación manual
- A través de servicios tales como DNS o DHCP
- Añadiendo información en las direcciones IP (prefijos)
- Utilizando direcciones tipo *anycast*

Tres mecanismos básicos túneles se proponen por la IETF: 6over4, 6to4, DSTM (*Dual Snack Transition Mechanism*) [WAD02]. Los dos primero se tratarán a detalle en las secciones subsecuentes, el restante está referido en el Apéndice B.

## 3.4.1 Tipos de *Tunneling*

El proceso de migración toma algo de tiempo para que se concrete en su totalidad y que toda la infraestructura de ruteo IPv6 se completada. Existen algunos dispositivos IPv4 que pueden ser

utilizados para transportar tráfico IPv6 y *tunneling* les permite realizar esto. *Tunneling* puede ser utilizado en una variedad de formas:

- Ruteador-a-ruteador.
- Usuario-a-ruteador.
- Usuario-a-usuario.
- Ruteador-a-usuario.

Lo anterior hace referencia a cómo son establecidos los túneles y se clasifican de acuerdo al mecanismo por el cual el nodo encapsulador determina la dirección final del túnel. En las dos primeras técnicas, arriba listadas, el paquete tiene por destino ruteador (el cual tiene como tarea el desencapsular el paquete y entregarlo a su destino). Debido a que el destino final del túnel es un ruteador, la dirección del paquete IPv6 es diferente a la dirección que corresponde al final del túnel. Esta dirección se determina de la información de configuración del nodo encapsulador, a este termino se le conoce como "*tunneling* configurado" [RFC 2893]. En los últimos dos métodos, el paquete IPv6 es enviado a su dirección destino. En este caso, tanto la dirección destino del paquete como la del túnel es exactamente la misma ya que se utilizan direcciones IPv4 compatibles para que automáticamente determinan el nodo final. Esto elimina la tarea de configurar el nodo final del túnel y es conocido como "*tunneling* automático" [RFC 2893].

Estas dos técnicas difieren principalmente en como es determinado el final del túnel, por lo demás se comportan de la misma manera –encapsulando el paquete al inicio del túnel, transmitiéndolo a través de éste, desencapsular el paquete al final del túnel, reensamblar si es necesario, retirar el encabezado IPv4 y procesar el paquee IPv6 recibido.

### 3.4.1.1 6over4

Definido en el RFC 2529, este mecanismo utiliza direcciones IPv4 empotradas en direcciones de capa de enlace IPv6 (ver capítulo 2).6over4 realiza el *tunneling* dentro de una sola organización o sitio (se comporta efectivamente como una LAN virtual [WAD02]), esto lo logra tratando a una red como una funcional subred IPv6 y permitiendo la auto-configuración de direcciones. Desafortunadamente 6over4

requiere que la infraestructura IPv4 soporte ruteo *multicast*. De modo que la mayoría de las redes IPv4 no soportan este tipo de ruteo 6over4 no ha sido del todo desarrollado. Un punto positivo acerca de este mecanismo es que conserva todas las características que ofrece IPv6, incluyendo la seguridad punto-a-punto y configuración estática [WAD02].

#### 3.4.1.2 6to4 (*Tunneling* Automático)

Como se mencionaba en la sección 3.4, el *tunneling* automático se refiere a que las configuraciones de túnel se realizan sin la necesidad de un manejo explícito. Este mecanismo es el más ampliamente utilizado y crea túneles IPv6 para tráfico del mismo tipo sobre redes IPv4 entre redes aisladas de 6to4. Debido a que la migración a IPv6 sigue en proceso, aún existen nodos más nodos IPv6 aislados entre sí que IPv4 (lo que se pretende pase en la fase final de la transición). Cada red 6to4 asume un prefijo que se agrega a la dirección IPv4 del 6toa *gateway* (2002:V4ADDR::/48). Esto significa que la dirección de fin de túnel se obtiene más fácilmente [WAD02].

## 3.4.2 Encapsulado

Como se ha venido manejando constantemente, el proceso de encapsulado es con el que *tunneling* comienza su operación. El encapsulado por su parte realiza otras tareas más complejas como el de determinar cuando fragmentar o enviar un mensaje de error ICMP. [RFC 2893]. La Figura 3.3 muestra a grandes rasgos como el encapsulado se lleva a cabo:

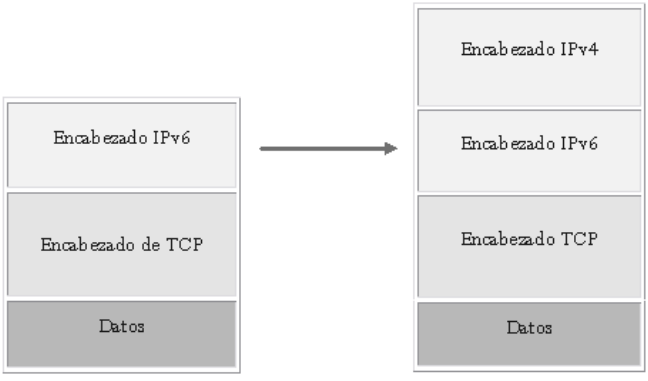


*Figura 3.11: Encapsulado de IPv6 en IPv4.*

Al igual que en la traducción (sección 3.3) se "construye" un encabezado nuevo, para *tunneling* sucede lo mismo. Al encapsular lo único que se hace es agregar un encabezado extra al datagrama ya

existente sin eliminar el original (de ahí una gran ventaja de este mecanismo), y sus campos se establecen de la siguiente forma en la Tabla 3.4:

*Tabla 3.4: Construcción de encabezado para tunneling [RFC 2893].*

| Campo | Valor |
|---|---|
| *Versión* | 4 |
| *IHL* | 5 (no opciones) |
| *Longitud total* | Longitud total del paquete IPv6 más encabezados IPv6 e IPv4. |
| *Identificación* | Generada durante la transmisión |
| *Banderas* | DF: definidas según sea el caso por RFC 2893<br>M: Como se necesiten según la fragmentación |
| *Fragmento* | Según se necesite |
| *TTL* | (ver RFC 2893) |
| *Protocolo* | 41 (el valor correspondiente para IPv6) |
| *Comprobación* | Calculado después del paquete resultante |
| *Dirección fuente* | Dirección IPv4 del origen del túnel |
| *Dirección destino* | Dirección IPv4 del fin de túnel |

## 3.4.3 Desencapsulado

Cuando el nodo IPv4/IPv6 recibe una dirección que le pertenece y el valor del campo Protocolo es 41, el paquete se reensambla (si es que viene fragmentado) y desencapsula el paquete removiendo el encabezado de IPv4. El desencapsulado se muestra en la siguiente Figura:

La diferencia con un paquete "ordinario" es que una vez desencapsulado no deber ser reenviado a menos de que el nodo esté configurado para realizar esto (túnel configurado).

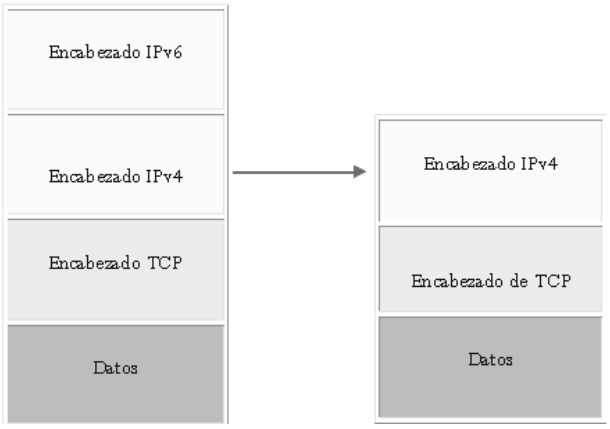


*Figura 3.12: Desencapsulando IP4 a IPv6 [RFC 2893].*

# 3.7 Resumen

Pila doble, Traducción y T*unneling* sólo son las tres categorías en las que se pueden agrupar los mecanismos de transición especificados por la IETF. Como es sabido, al intentar migrar la nueva versión del protocolo IP un método que ayude y soporte este proceso es necesario. La Tabla 3.5, muestra de manera resumida los tres bloques básicos de transición, sus algoritmos más representativos, sus ventajas y desventajas.

Actualmente el protocolo IPv6 es soportado en la mayoría de los sistemas operativos modernos, en algunos casos como una opción de instalación. Linux, Solaris, Mac OS, OpenBSD, FreeBSD, Windows (2k, CE) y Symbian (dispositivos móviles) son sólo algunos de los sistemas operativos que pueden funcionar con IPv6

*Tabla 3.5: Resumen de los tres mecanismos de transición y sus algoritmos más representativos.*

| Nombre | Tipo de mecanismo | Conectividad | Descripción | Ventajas | Desventajas |
|---|---|---|---|---|---|
| Pila-dual | Pila doble | Sólo ente sistemas del mismo tipo (IPv4-IPv4, IPv6-IPv6) | •Trabaja con ambos protocolos (IPv4 e IPv6)<br>•Procesa sólo los encabezados IP<br>•Uno de los más populares dentro de su tipo<br>•Se basa en DHCP y direcciones compatibles para asignación de direcciones | • Fácil de implementar<br>• Una solución inmediata y accesible<br>• Permite a los nuevos dispositivos IPv6 relacionarse rápidamente con el resto de los dispositivos | •No trabaja en ambientes mixtos (IPv4 sobre IPv6 y viceversa)<br>•Si la red no es IPv6, no se ve beneficiada de las características de esta versión |
| SIIT (*Stateless IP/ICMP Translator*) | Traducción | De IPv6-a-IPv4 y de IPv4-a-IPv6 | •Para hacer dos protocolos "compatibles" realiza la traducción de encabezados<br>•Se necesita de un traductor que lleve a cabo la tarea de traducción | •Permite a nodos IPv4 comunicarse con nodos IPv6<br>•Fácil de soportar por un dispositivo<br>•No se afecta el checksum de capa de transporte<br>•Puede manejar paquetes encriptados, ya que no modifica capas superiores | •Al realizar la traducción IPv6 a IPv4 se pierden muchos campos, y con éstos, beneficios de IPv6<br>•Se ignoran la mayoría de los encabezados de extensión<br>•Ya que se manejan dos protocolos, se necesita de utilizar dos tablas de ruteo diferentes<br>•Al trabajar con direcciones IPv4 compatibles, se reduce el campo de direccionamiento<br>•Se reduce el tamaño del MTU lo que resulta en fragmentación |
| 6over4 | *Tunneling* | IPv6-a-IPv6 sobre IPv4 | •Se comporta como una red "virtual" | • Permite la auto configuración<br>• Conserva todas las características de IPv6 | • Necesita soporte de ruteo *multicast* (IPv4 raramente cuenta con este soporte) |
| 6to4 | *Tunneling* | IPv6-a-IPv6 sobre IPv4 | •Crea túneles automáticamente<br>•Algoritmo más popular dentro de su clase | • Ayuda a conectar redes IPv6 aisladas entre sí | |