

Mecanismos de traducción

4.1. Introducción

Los mecanismos de traducción nos van a permitir que dispositivos IPv6-only sean capaces de comunicarse con dispositivos IPv4-only. De la misma forma, también nos van a permitir que dispositivos IPv4-only sean capaces de comunicarse con dispositivos IPv6-only.

Estos mecanismos no realizan técnicas de tunelado sino que traducen directamente los datagramas IPv6 e IPv4, bien en la capa de red, transporte o aplicación. Así, estos mecanismos dan solución a las situaciones en las que los mecanismos de tunelado no se puedan aplicar.

4.2. Clasificación

En función del nivel dónde se realice la traducción, existen diferentes tipos de traductores:

- Traductores de nivel de red (Figura 4.1):

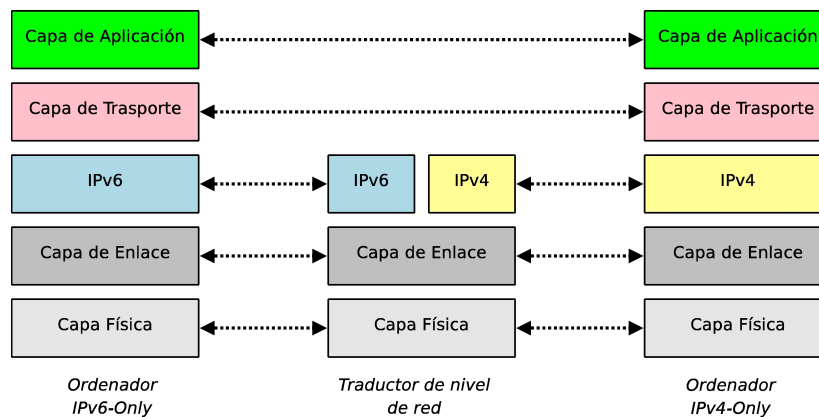


Figura 4.1: Traductor a nivel de red

- SIIT (*Stateless IP/ICMP Translator*), definido en la RFC 2765 [7].
- BIS (*Bump in the Stack*), definido en la RFC 2767 [5].
- Traductores de nivel de transporte (Figura 4.2):
 - TRT (*Transport Relay Translator*), definido en la RFC 3142 [10].

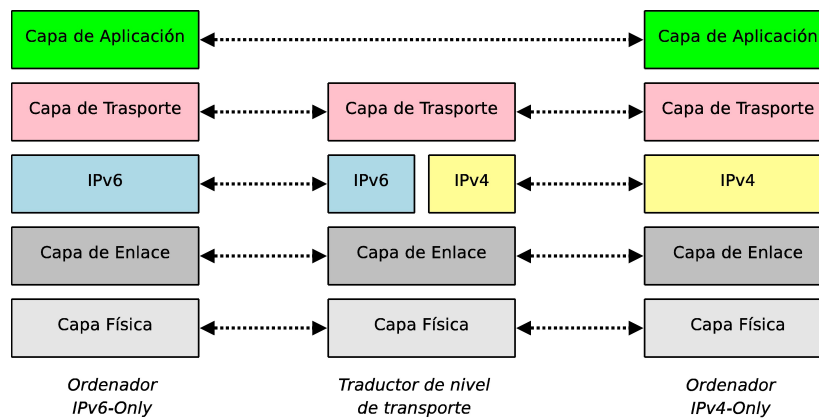


Figura 4.2: Traductor a nivel de transporte

- Traductores de nivel de aplicación (Figura 4.3):

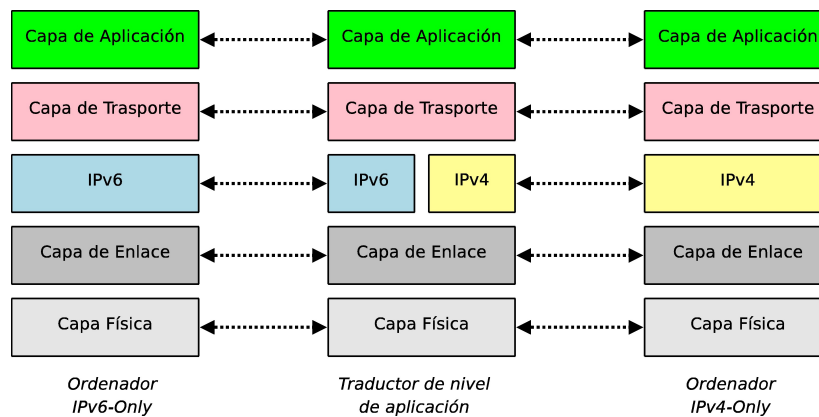


Figura 4.3: Traductor a nivel de aplicación

- SOCKS IPv6/IPv4, definido en la RFC 3089 [12].
- BIA (*Bump in the API*), definido en la RFC 3338 [14].
- ALG (*Application Level Gateways*).

Otra posible clasificación de los mecanismos de traducción anteriores se realiza en función de la capacidad de almacenar estados de la comunicación. Así, los mecanismos SIIT, BIS y BIA se caracterizan por no almacenar ningún estado y los mecanismos TRT y SOCKS se caracterizan por almacenar el estado de cada una de las comunicaciones.

4.3. Mecanismo SIIT

El mecanismo SIIT realiza una traducción directa de la cabecera de los paquetes IPv4 e IPv6 ya que se trata de un mecanismo de traducción de nivel de red. La Figura 4.4 muestra la pila de protocolos correspondiente a este mecanismo.

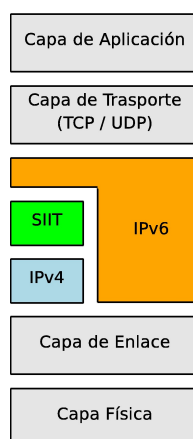


Figura 4.4: Torre de protocolos del mecanismo SIIT

4.3.1. Direcciones “con correspondencia” y “traducidas”

Este mecanismo utiliza dos tipos de direcciones IPv6 especiales: las direcciones IPv6 con correspondencia IPv4 (*IPv4 mapped Address*) y las direcciones IPv6 traducidas de IPv4 (*IPv4 translated address*). El primer tipo se construye tal y como se muestra en la Figura 4.5. Este tipo de direcciones se consideran no válidas para ser el origen de un túnel, por lo que es necesario un segundo tipo de direcciones (Figura 4.6).

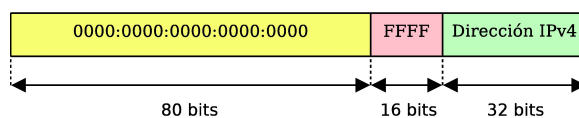


Figura 4.5: Direcciones “con correspondencia”

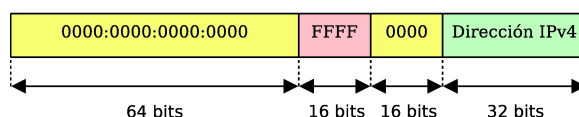


Figura 4.6: Direcciones “traducidas”

4.3.2. Traducción de cabeceras

Partiendo de la estructura de la cabecera de un datagrama IPv4 (Figura 4.7) y de la estructura de la cabecera de un datagrama IPv6 (Figura 4.8), se va a detallar a continuación la correspondencia de cada uno de los campos que se realiza durante la traducción.

4.3.2.1. Traducción de la cabecera IPv4 a IPv6

A partir de los campos de la cabecera IPv4, los campos de la cabecera IPv6 tendrán el siguiente contenido:

- **Version:** 6.
- **Clase de tráfico:** Campo *Tipo de Servicio* de la cabecera IPv4.

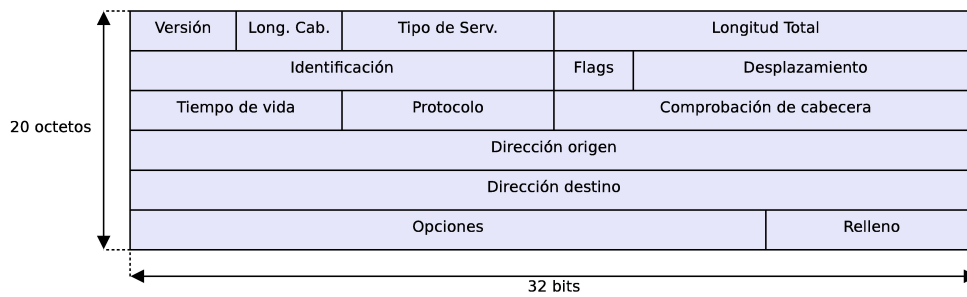


Figura 4.7: Estructura de la cabecera de un datagrama IPv4

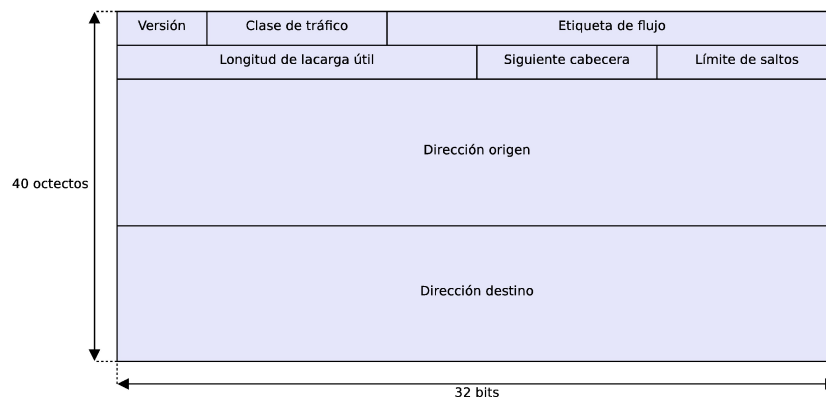


Figura 4.8: Estructura de la cabecera de un datagrama IPv6

- **Etiqueta de flujo:** 0.
- **Longitud de la carga útil:** Valor del campo *Longitud Total* de la cabecera IPv4 menos la suma de la longitud de la cabecera IPv4 y la longitud de las opciones de la cabecera IPv4.
- **Siguiete cabecera:** Valor del campo *Protocolo* de la cabecera IPv4.
- **Límite de saltos:** Valor del campo *Tiempo de Vida* de la cabecera IPv4 menos uno.
- **Dirección origen:** 0:0:0:0:FFFF::/80 concatenado con el valor del campo *Dirección Origen* de la cabecera IPv4.
- **Dirección destino:** 0:0:0:0:0:FFFF::/96 concatenado con el valor del campo *Dirección Destino* de la cabecera IPv4.

4.3.2.2. Traducción de la cabecera IPv6 a IPv4

A partir de los campos de la cabecera IPv6, los campos de la cabecera IPv4 tendrán el siguiente contenido:

- **Version:** 4.
- **Longitud de la cabecera:** 5.
- **Tipo de Servicio:** Valor del campo *Clase de Servicio* de la cabecera IPv6.
- **Longitud total:** Valor del campo *Longitud de la carga útil* de la cabecera IPv6 más la longitud de la cabecera IPv4.

- **Identificador:** 0.
- **Flags:** Si no se fragmenta, el valor es 0. En el caso que existan más fragmentos, el valor es 1.
- **Desplazamiento:** 0.
- **Tiempo de vida:** Valor del campo *Límite de saltos* de la cabecera IPv6 menos uno.
- **Protocolo:** Valor del campo *Siguiente cabecera* de la cabecera IPv6.
- **Comprobación de cabecera:** Se calcula a partir de la cabecera IPv4.
- **Dirección origen:** Los últimos 32 bits del campo *Dirección origen* de la cabecera IPv6 (*IPv4 translated address*).
- **Dirección destino:** Los últimos 32 bits del campo *Dirección destino* de la cabecera IPv6 (*IPv4 mapped address*).
- **Opciones:** Ninguna.

4.3.3. Funcionamiento

Una vez instalado el mecanismo en un ordenador IPv6, ya podemos comunicarnos tanto a través del protocolo IPv6 como del protocolo IPv4. En el caso de que el ordenador IPv6 desee comunicarse con un ordenador IPv4, éste empleará una dirección IPv6 creada a partir de la dirección IPv4 (*IPv4 mapped address*). Así, el mecanismo SIIT reconoce este tipo de direcciones y realiza la traducción correspondiente, tras lo cual envía el datagrama IPv4 a través de la red. En el caso contrario, cuando reciba un datagrama IPv4, el mecanismo se encargará de realizar la traducción correspondiente y enviar los datos a la capa superior.

Para los aspectos de resolución de nombres, este mecanismo se suele utilizar junto a alguno de estos dos mecanismos:

- *Bump-in-the-Stack* (BIS).
- *Bump-in-the-API* (BIA).

4.4. Mecanismo BIS

El mecanismo *Bump-in-the-Stack* (BIS) permite que las aplicaciones, que utilizan IPv4, sean capaces de comunicarse a través de una red IPv6. Para ello traduce los paquetes IPv4 en IPv6 y viceversa, a nivel de red.

4.4.1. Componentes

Este mecanismo está compuesto de los siguientes elementos (Figura 4.10):

- **Traductor:**
Este elemento se encarga de traducir las cabeceras de los datagramas IPv4 e IPv6 de acuerdo a lo mostrado en el mecanismo SIIT.
- **Resolución de nombres:**
Este elemento se encarga de interceptar las peticiones de resolución de nombres, solicitar tanto la dirección IPv4 como la dirección IPv6 y procesar la respuesta que recibe del servidor y la que va a mandar a la capa superior.
- **Correspondencia de direcciones:**
Este elemento se encarga de almacenar las diferentes correspondencias entre un nombre y sus direcciones IPv4 e IPv6.

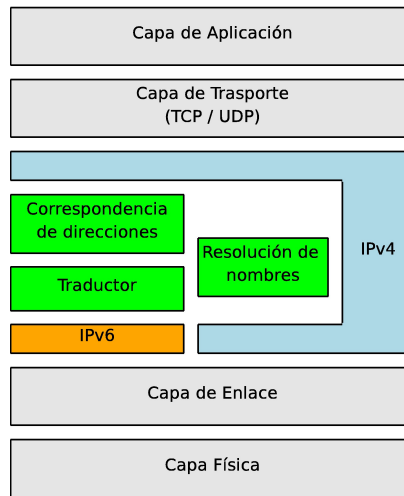


Figura 4.9: Componentes del mecanismo BIS

4.4.2. Funcionamiento

Cuando una aplicación realiza una petición de resolución de nombre a un servidor DNS, el componente *Resolución de nombres* se encarga de procesar la respuesta del servidor para actuar en función de las direcciones recibidas. Así, se pueden dar las siguientes situaciones:

- El nombre sólo tiene una dirección IPv4 asociada:
La aplicación utilizará la dirección IPv4 directamente a través de la red IPv4, por lo que el componente *Correspondencia de direcciones* almacenará el nombre y la dirección IPv4 correspondiente.
- El nombre tiene tanto una dirección IPv4 como una dirección IPv6 asociadas:
Se almacenará una nueva entrada en el componente *Correspondencia de direcciones*, donde se indique el nombre y las direcciones IPv4 e IPv6 correspondientes, devolviendo a la aplicación solamente la dirección IPv4 como respuesta del servidor DNS.
- El nombre sólo tiene una dirección IPv6 asociada:
El componente *Correspondencia de direcciones* asignará una dirección IPv4, perteneciente a un conjunto de direcciones predefinidas, al nombre y a la dirección IPv6 devueltas por el servidor DNS. Así el componente *Resolución de nombres* devolverá a la aplicación solamente la dirección IPv4 almacenada en el componente *Correspondencia de direcciones*.

De esta forma, cualquier datagrama IPv4 cuya dirección destino pertenezca al conjunto de direcciones predefinidas será traducido y enviado a través de la red IPv6. En el caso de recibir un datagrama IPv6 con una dirección origen que no está almacenada en el componente *Correspondencia de direcciones*, éste creará una nueva correspondencia entre la dirección IPv6 y la dirección IPv4 asignada, permitiendo su traducción.

4.5. Mecanismo BIA

El mecanismo *Bump-in-the-API* (BIA) permite que las aplicaciones, que utilizan IPv4, sean capaces de comunicarse a través de una red IPv6. Para ello traduce la interfaz de aplicación (API, *Application Programming Interface*) de tal forma que no sea necesario traducir los propios datagramas. Es por ello que este mecanismo de traducción, pertenece a los que se aplican a nivel de aplicación.

4.5.1. Componentes

Este mecanismo está compuesto de los siguientes elementos (Figura 4.10):

- **Traductor de la interfaz de aplicación:**
Este elemento se encarga de traducir las interfaz de aplicación de la versión 4 de IP a la versión 6.
- **Resolución de nombres:**
Este elemento se encarga de interceptar las peticiones de resolución de nombres, solicitar tanto la dirección IPv4 como la dirección IPv6 y procesar la respuesta que recibe del servidor y la que va a mandar a la capa superior.
- **Correspondencia de direcciones:**
Este elemento se encarga de almacenar las diferentes correspondencias entre un nombre y sus dirección IPv4 e IPv6.
- **Correspondencia de funciones:**
Este elemento se encarga de interceptar las llamadas a las funciones de la API de IPv4 y realizar la correspondencia con las funciones de la API de IPv6.

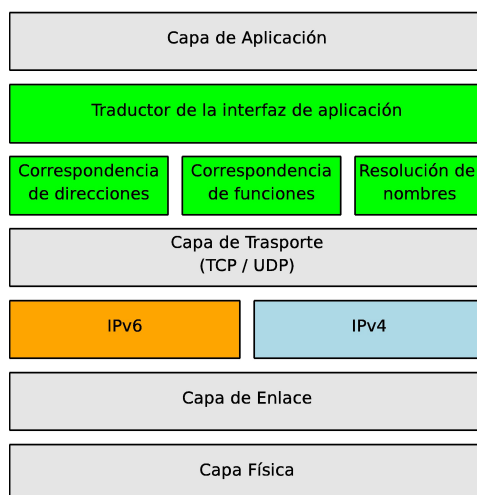


Figura 4.10: Componentes del mecanismo BIA

4.5.2. Funcionamiento

Cuando una aplicación realiza una petición de resolución de nombre a un servidor DNS, el componente *Resolución de nombres*, se encarga de procesar la respuesta del servidor para actuar en función de las direcciones recibidas. Así, se pueden dar las siguientes situaciones:

- El nombre sólo tiene una dirección IPv4 asociada:
La aplicación utilizará la dirección IPv4 directamente a través de la red IPv4, por lo que el componente *Correspondencia de direcciones* almacenará el nombre y la dirección IPv4 correspondiente.
- El nombre tiene tanto una dirección IPv4 como una dirección IPv6 asociadas:
Se almacenará una nueva entrada en el componente *Correspondencia de direcciones*, donde se indique el nombre y las direcciones IPv4 e IPv6 correspondientes, devolviendo a la aplicación solamente la dirección IPv4 como respuesta del servidor DNS.

- El nombre sólo tiene una dirección IPv6 asociada:
El componente *Correspondencia de direcciones* asignará una dirección IPv4, perteneciente a un conjunto de direcciones predefinidas, al nombre y a la dirección IPv6 devueltas por el servidor DNS. Así el componente *Resolución de nombres* devolverá a la aplicación solamente la dirección IPv4 almacenada en el componente *Correspondencia de direcciones*.

De esta forma, cualquier llamada a una función de la API de IPv4 que tenga como dirección IPv4 destino alguna perteneciente al conjunto de direcciones definidas, ésta se traduce directamente a una llamada a la función correspondiente en la API de IPv6. En el caso de recibir un datagrama IPv6 con una dirección origen que no está almacenada en el componente *Correspondencia de direcciones*, éste creará una nueva correspondencia entre la dirección IPv6 y la dirección IPv4 asignada, permitiendo su acceso a través de las funciones de la API de IPv4 traduciendo las funciones de la API de IPv6.

4.6. Mecanismo TRT

El mecanismo *Transport Relay Translator* (TRT) se basa en la interconexión de dos conexiones diferentes, cada una de ellas de una red distinta, con el objetivo de que ordenadores IPv6 se puedan comunicar con ordenadores IPv4. Así, la conexión TCP o los datagramas UDP del ordenador IPv4 terminan en el TRT, encargándose éste de crear la conexión TCP o enviar los datagramas UDP al ordenador IPv6 (Figura 4.11). Para el caso contrario, el TRT funcionaría de la misma forma. Para realizar esta interconexión, es necesario que se almacene el estado de cada una de las conexiones.

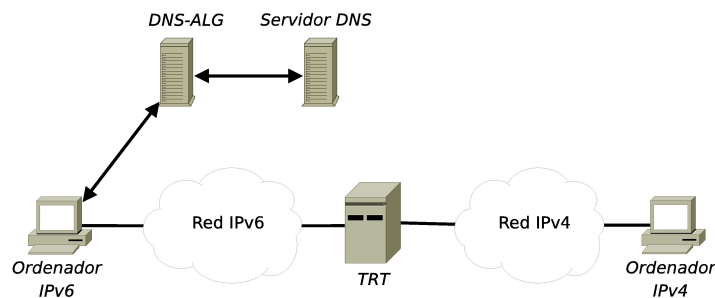


Figura 4.11: Estructura general del mecanismo TRT

4.6.1. DNS-ALG

El elemento *DNS-Application Layer Gateway* (DNS-ALG) tiene la función de proxy DNS. De esta forma, cualquier petición de resolución de un nombre es supervisada por este elemento. El ordenador IPv6 realizará peticiones de resolución de nombres en direcciones IPv6 solamente, puesto que éste sólo entiende dicha versión. Así, en función de las direcciones que devuelva el servidor DNS se pueden dar las siguientes situaciones:

- El nombre no tiene dirección IPv6 asociada:
El elemento DNS-ALG realizará una petición para la resolución de un nombre en una dirección IPv4. Cuando el servidor DNS le devuelva la dirección IPv4, éste creará una dirección IPv6 según se muestra en la Figura 4.12 (el formato no está definido en el IANA¹), y se la devolverá al ordenador IPv6. Dado que el prefijo no está definido, éste será local por lo que cualquier comunicación con dicha dirección IPv6 tendrá que pasar por el TRT.

¹Internet Assigned Number Authority, encargada de la asignación de direcciones en Internet.

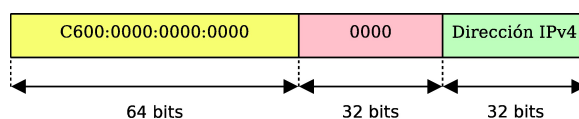


Figura 4.12: Direcciones creadas en el DNS-ALG

- El nombre sólo tiene una dirección IPv6 asociada:
La respuesta del servidor DNS es la que el elemento DNS-ALG devuelve al ordenador IPv6.

4.7. Mecanismo SOCKS IPv6/IPv4

En la RFC 1928 [3] se define el concepto básico de SOCKS que se aplica en el mecanismo SOCKS IPv6/IPv4 para la traducción de IPv6 a IPv4 de las comunicaciones entre el ordenador IPv6 y el ordenador IPv4 (Figura 4.13), a nivel de aplicación. Para el caso contrario, el ordenador IPv4 ha de tener instalado un cliente SOCKS.

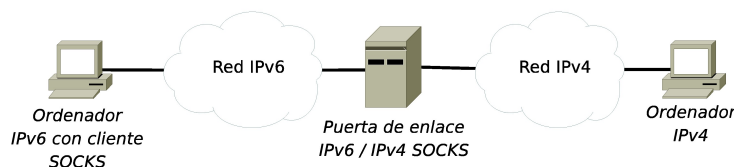


Figura 4.13: Estructura general del mecanismo SOCKS IPv6/IPv4

4.7.1. Elementos

Para el correcto funcionamiento de este mecanismo es necesario dos elementos:

- Un **servidor SOCKS**, por lo que se sitúa en la capa de aplicación.
- Un ordenador con la **librería cliente de SOCKS** (cliente “*socksified*”) instalada, situada entre la capa de aplicación y la capa de transporte.

4.7.2. Funcionamiento

Cuando un cliente SOCKS desea establecer una comunicación con un servidor, el primer paso es establecer un conexión con el proxy definido previamente en el cliente. Este cliente comunica al proxy la dirección y el puerto del servidor con quien desea comunicarse. Entonces el proxy establece la conexión con el servidor indicado. Una vez establecida la conexión entre el proxy y el servidor, el proxy reenvía los paquetes correspondientes, que ha enviado el servidor, al cliente.

4.8. Mecanismo ALG

El mecanismo ALG realiza las funciones de traducción directamente en la capa de aplicación de forma análoga al funcionamiento de un proxy HTTP. En este caso no es necesario instalar ningún elemento en el cliente, ya que el proxy actúa como si fuese directamente el servidor, ocultando cualquier detalle de la comunicación, entre el proxy y el servidor, al cliente. Por tanto, este mecanismo puede resultar muy útil para el acceso web de ordenadores IPv6-Only, conectados a una red IPv6, a servidores IPv4 conectados a una red IPv4.

