Protocolo de Internet: IPv6

Eusebio Gomez
Universidad Católica
Paraguay
eusebio.gomez@uc.edu.py

Universidad Católica "Nuestra Señora de la Asunción"
http://universidadcatolica.edu.py

Resumen En este documento se desarrollara contenido acerca del **Protocolo de Internet (IP)** en su versión más reciente, el **IPv6**, tal como las características del **Protocolo de Internet**, sus componentes principales, sus funciones principales, direcciones, prefijos, subredes, fragmentación, y las características adoptadas por el **Protocolo IPv6**. También se abordan temas como el motivo que impulsó la creación del **Protocolo IPv6**, la causa por la que el **Protocolo IPv4** esta volviéndose insuficiente para la demanda actual, las similitudes entre ambos protocolos y sus diferencias. También se discutirá acerca de las controversias que generó la creación del **Protocolo IPv6**, y se hablará acerca de una solución para la implementación del **Protocolo IPv6** para poder coexistir con el **Protocolo IPv4**.

Key words: Protocolo IP, Protocolo de Internet, IPv4, IPv6, direcciones de red, Internet, datagrama, encabezado, router, fragmentación, tunelización, cabecera.

1. INTRODUCCIÓN

El **Protocolo de Internet (IP)** es el principal protocolo utilizado en Internet para la transmisión de datagramas en las redes. Un datagrama is la unidad básica de transferencia asociada a una red de conmutación de paquetes (packet-switching). Los datagramas están compuestos por un encabezado y una sección de carga. La característica principal del **Protocolo IP** es que permite entregar paquetes desde el host (computadora o cualquier elemento conectado a la red) de origen al host destino basado únicamente en las direcciones IP de los encabezados de los paquetes. El **Protocolo IP** también define las estructuras de los paquetes que llevan los datos a entregar. La primera y mayor versión del **Protocolo IP**, el **IPv4**, es la que domina Internet. En este documento nos enfocaremos en la versión más reciente del **Protocolo IP**, el **IPv6**.

El 31 de enero de 2011, las direcciones públicas IPv4 fueron utilizadas en su totalidad, cumpliéndose la primera problemática predicha del **Protocolo IPv4**

que incentivó la creación del **Protocolo IPv6**: la cantidad relativamente pequeña de direcciones posibles (2^{32}) [1]. El **Protocolo IPv6** permite tener hasta 2^{128} direcciones posibles, lo que aproximadamente equivale a 3×10^{38}. Si la Tierra completa, incluidos los océanos, estuvieran cubiertos de computadoras, el **IPv6** permitiría 7×10^{23} direcciones IP por metro cuadrado [2]. Esta cantidad astronómica de direcciones posibles hace que el **Protocolo IPv6** sea el mas apropiado para la alta demanda que Internet esta teniendo en la actualidad.

Pese a la problemática del **Protocolo IPv4**, **IPv6** ha demostrado ser muy difícil de implementar. Es un protocolo de capa de red diferente que en realidad no es compatible internamente con **IPv4**, a pesar de tantas similitudes. El resultado es que **IPv6** se implementa y utiliza sólo en una pequeña fracción de Internet (se estima un 1 %), a pesar de haber sido un estándar de Internet desde 1998 [3][4].

La creación del **Protocolo IPv6** se basa en una serie de requisitos: soportar miles de millones de hosts, incluso con una asignación de direcciones ineficiente; reducir el tamaño de las tablas de enrutamiento; simplificar el protocolo para permitir a los routers procesar los paquetes con más rapidez; proporcionar mayor seguridad; poner más atención en cuanto al tipo de servicio; ayudar a la multidifusión al permitir la especificación de alcances; permitir que un host pueda desplazarse libremente sin tener que cambiar su dirección; permitir que el protocolo evolucione en el futuro; permitir que el protocolo viejo y el nuevo coexistan durante años [5].

En este documento abordaremos más profundamente los detalles del **Protocolo IPv6**, describiendo sus características, componentes, datagrama, direcciones, su problemática para coexistir con el **Protocolo IPv4** y las posibles soluciones a estos problemas.

2. EL PROTOCOLO DE INTERNET

El Protocolo de Internet está diseñado para su uso en sistemas interconectados de redes de computadoras con conmutación de paquetes (packet-switching) . El Protocolo de Internet permite transmisión de bloques de datos llamados datagramas, de fuentes a destinos, donde las fuentes y los destinos son hosts identificados por direcciones de longitud fija. El Protocolo de Internet también permite la fragmentación y reensamblaje de datagramas largos, si es necesario, para su transmisión.

El Protocolo de Internet implementa dos funciones básicas: direccionamiento y fragmentación. Los componentes de la red utilizan estas direcciones contenidas en los encabezados de los paquetes para transmitir los datagramas de internet hacia su destino. Esto se realiza mediante la selección de una ruta específica, a

través de un método llamado ruteo. El modelo de operación es tal que en en cada host o gateway (router) que se encuentre en la red, se encuentra un módulo de internet. Estos módulos comparten reglas en común para la interpretación de los campos de direcciones para la fragmentación y reensamblaje de los datagramas de internet. Además, estos módulos contenidos en los gateways, tienen procedimientos para tomar decisiones de ruteo.

El Protocolo de Internet trata a cada datagrama como una entidad independiente no relacionada a ningún otro datagrama. No existen conexiones ni circuitos lógicos entre ellos.

El Protocolo de Internet no provee una interfaz de comunicación confiable. No existen acuses de recibo ni punto a punto ni salto por salto. No existe control de errores para los datos, solo una suma de verificación en el encabezado. Tampoco existe control de flujo ni retransmisiones. Los errores detectados son reportados a través del módulo del Protocolo de Control de Mensajes de Internet (ICMP, por sus siglas en inglés) [6].

2.1. Protocolo IPv4

Empezaremos a hablar del Protocolo IPv4 describiendo su modelo de operación. Suponemos que la transmisión involucrara un router intermedio. La aplicación emisora prepara los datos y llama al módulo de internet local para la transmisión de datos en forma de datagrama y pasa la dirección destino y otros parámetros como argumento de la llamada [7].

El módulo de internet prepara el encabezado del datagrama y adjunta el dato a ella. El módulo de internet determina una dirección de red local, en este caso, es la dirección del router [7].

Envía el datagrama y la dirección de red local a la interface de red local. La interface de red local crea un encabezado de red local, y adjunta el datagrama a ella, y luego lo envía a través de la red local. El datagrama llega al router envuelto en el encabezado de red local. La interface de red local remueve el encabezado, y entrega el datagrama al módulo de internet. El módulo de internet determina en base a la dirección de internet que el datagrama debe ser reenviado a otro host en una segunda red. El módulo de internet determina una dirección de red local para el host destino. Llama a la interface de red local para que envíe el datagrama [8].

La interface de red local crea un encabezado de red local y adjunta el datagrama enviando el resultado al host destino. En el host destino, la interfaz de red local remueve el encabezado de red local en el datagrama y es reenviada al módulo de internet. El módulo de internet determina que el datagrama corresponde a una aplicación en este host y luego pasa los datos a la aplicación en respuesta a una llamada de sistema, pasando la dirección emisora y otros

parámetros como parte de la llamada [8].

Descripción de funcionalidad: La función del Protocolo de Internet es mover datagramas a través de un conjunto interconectado de redes. Esto se logra pasando los datagramas de un módulo de internet a otro hasta que se llegue al destino. Los módulos de internet residen en los hosts y routers en el sistema de internet. Los datagramas son enviados de un módulo de internet a otro atravesando redes individuales en base a la interpretación de la dirección de internet. Así, un importante mecanismo en el Protocolo de Internet es la dirección de internet.

Al enviar datos de un módulo de internet a otro, los datagramas pueden necesitar atravesar una red cuyo tamaño de paquete máximo es menor al tamaño del datagrama. Para solucionar este problema, un mecanismo de fragmentación es proveído en el Protocolo de Internet.

Direcciones: El Protocolo de Internet se maneja primeramente con direcciones. Es la tarea de protocolos de mayor nivel hacer el mapeo de direcciones y nombres (URLs. a Direcciones IP). El módulo de internet mapea las direcciones de internet con direcciones de red local. Es la tarea de protocolos de menor nivel hacer el mapeo de direcciones de red locales y rutas [9].

Prefijos: A diferencia de las direcciones Ethernet, las direcciones IP son jerárquicas. Cada dirección de 32 bits está compuesta de una porción de red de longitud variable en los bits superiores, y de una porción de host en los bits inferiores. La porción de red tiene el mismo valor para todos los hosts en una sola red, como una LAN Ethernet. Esto significa que una red corresponde a un bloque contiguo de espacio de direcciones IP. A este bloque se le llama prefijo [10].

Las direcciones IP se escriben en notación decimal con puntos. En este formato, cada uno de los 4 bytes se escribe en decimal, de 0 a 255. Por ejemplo, la dirección hexadecimal 80D00297 de 32 bits se escribe como 128.208.2.151. Para escribir los prefijos, se proporciona la dirección IP menor en el bloque y el tamaño del mismo. El tamaño se determina mediante el número de bits en la porción de red; el resto de los bits en la porción del host pueden variar. Esto significa que el tamaño debe ser una potencia de dos [10]. Por convención, el prefijo se escribe después de la dirección IP como una barra diagonal seguida de la longitud en bits de la porción de red (Ejemplo: 192.168.0.0/24).

Como la longitud del prefijo no se puede inferir sólo a partir de la dirección IP, los protocolos de enrutamiento deben transportar los prefijos hasta los routers. La longitud del prefijo corresponde a una máscara binaria de 1s en la porción de

red. Cuando se escribe de esta forma, se denomina máscara de subred. Se puede aplicar un AND a la máscara de subred con la dirección IP para extraer sólo la porción de la red [10].

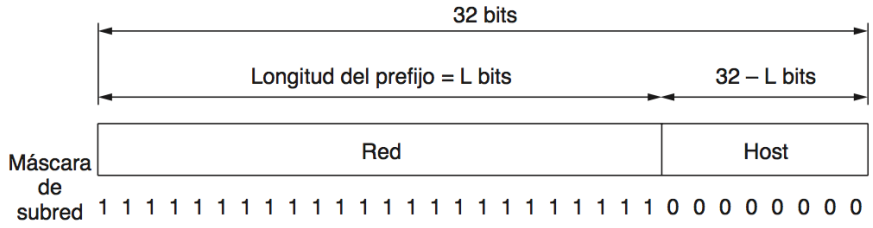

Figura 1. Dirección IPv4 con máscara de subred

Las direcciones jerárquicas tienen ventajas y desventajas considerables. La ventaja clave de los prefijos es que los routers pueden reenviar paquetes con en base en la porción de red de la dirección, siempre y cuando cada una de las redes tenga un bloque de direcciones único. La porción del host no importa a los routers, ya que enviarán en la misma dirección a todos los hosts de la misma red. Sólo hasta que los paquetes llegan a la red de destino es cuando se reenvían al host correcto. Esto hace que las tablas de enrutamiento sean mucho más pequeñas de lo que podrían ser con cualquier otro método. Aunque el uso de una jerarquía permite a Internet escalar, tiene dos desventajas. En primer lugar, la dirección IP de un host depende de su ubicación en la red. Una dirección Ethernet se puede usar en cualquier parte del mundo, pero cada dirección IP pertenece a una red específica, por lo que los routers sólo podrán entregar paquetes destinados a esa dirección en la red. Se necesitan diseños como IP móvil para soportar hosts que se desplacen de una red a otra, pero que deseen mantener las mismas direcciones IP. La segunda desventaja es que la jerarquía desperdicia direcciones a menos que se administre con cuidado. Si se asignan direcciones a las redes en bloques (muy) grandes, habrá (muchas) direcciones que se asignen pero no se utilicen [10].

Subredes: Para evitar conictos, los números de red se administran a través de una corporación sin fines de lucro llamada ICANN (Corporación de Internet para la Asignación de Nombres y Números, del inglés Internet Corporation for Assigned Names and Numbers). Esta corporación ha delegado partes de este espacio de direcciones a varias autoridades regionales, las cuales reparten las direcciones IP a los ISP y otras compañías. éste es el proceso por el cual se asigna un bloque de direcciones IP a una compañía. Sin embargo, este proceso es sólo el principio de la historia, puesto que la asignación de direcciones IP es continua a medida que

crecen las compañías. El ruteo por prefijo requiere que todos los hosts en una red tengan el mismo número de red. Esta propiedad puede provocar problemas a medida que las redes aumentan su tamaño [10].

Fragmentación: La fragmentación de un datagrama de internet es necesaria cuando el datagrama es originado en una red que permite grandes tamaños de paquetes y tiene que atravesar otras redes que limitan el paquete a un tamaño menor. Un datagrama puede ser marcado como no fragmentar. Cualquier datagrama marcado de esa manera no debe ser fragmentado bajo ninguna circunstancia. En caso de no poder entregarse a destino sin tener que fragmentarse, el paquete debe ser descartado y debidamente reportado al origen.

La fragmentación de internet y el procedimiento de reensamblaje deben poder dividir los datagramas en una cantidad arbitraria de partes que posteriormente pueden ser reensambladas. El receptor de los fragmentos utiliza un identificador en el encabezado del paquete para asegurar que fragmentos de diferentes datagramas no se mezclen. También se utiliza un offset y una longitud por fragmento para luego poder realizar el reensamblaje de los fragmentos en el orden correcto. Se utiliza una bandera que indica en cada paquete si existen más fragmentos del mismo datagrama.

El identificador del datagrama debe ser un número único entre receptor y emisor que pueda servir para identificar a uno y solo un datagrama. Para esto es necesario asegurar que este número identificador sea único mientras el datagrama este activo.

2.2. Formato de un datagrama IPv4 [11][12][13]

Un datagrama IPv4 consiste en dos partes: el encabezado y el cuerpo o carga útil. El encabezado tiene una parte fija de 20 bytes y una parte opcional de longitud variable. El formato del encabezado se muestra en la Figura 2. Los bits se transmiten en orden de izquierda a derecha y de arriba hacia abajo, comenzando por el bit de mayor orden del campo Versión.

Versión: El campo *Versión* lleva el registro de la versión del protocolo al que pertenece el datagrama. La versión 4 es la que domina Internet en la actualidad. Al incluir la versión al inicio de cada datagrama, es posible tener una transición entre versiones a través de un largo periodo de tiempo.

IHL: *Internet Header Length.* Es un campo que indica la longitud del datagrama en términos de palabras de 32 bits dado que la longitud del encabezado no es constante. El valor mínimo es de 5, cifra que se aplica cuando no hay opciones. El valor máximo de este campo es 15, lo que limita el encabezado a 60

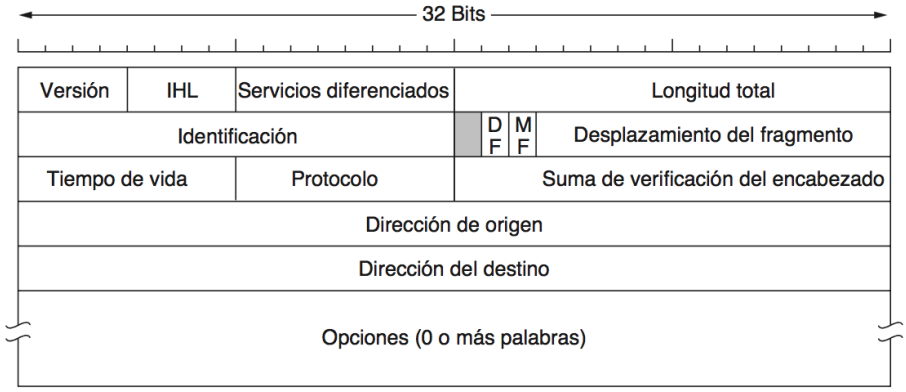

Figura 2. Datagrama IPv4

bytes y por lo tanto, el campo Opciones a 40 bytes.

Servicios diferenciados: Especifica los parámetros de fiabilidad, prioridad, retardo y rendimiento. Este campo se utiliza muy raramente; su interpretación ha sido sustituida recientemente. Los primeros 6 bits del campo son denominados ahora campo de servicios diferenciados (DS, Dierentiated Services). Los 2 bits restantes están reservados para un campo de notificación explícita de congestión (ECN), actualmente en fase de estandarización.

Longitud total: El campo *Longitud total* incluye todo en el datagrama: tanto el encabezado como los datos. La longitud máxima es de 65 535 bytes.

Identificación: El campo *Identificación* es necesario para que el host de destino determine a qué paquete pertenece un fragmento recién llegado. Todos los fragmentos de un paquete contienen el mismo valor de Identificación.

DF: *Dont Fragment*. Es una orden para que los routers no fragmenten el paquete.

MF: *More Fragments*. Todos los fragmentos excepto el último tienen establecido este bit, que es necesario para saber cuándo han llegado todos los fragmentos de un datagrama.

Desplazamiento del fragmento: El *Desplazamiento del fragmento* indica a qué parte del paquete actual pertenece este fragmento. Todos los fragmentos excepto el último del datagrama deben ser un múltiplo de 8 bytes, que es la unidad de fragmentos elemental. Dado que se proporcionan 13 bits, puede haber un máximo de 8.192 fragmentos por datagrama, para soportar una longitud máxima de paquete de hasta el límite del campo Longitud total. En conjunto,

los campos *Identificación*, *MF* y **Desplazamiento del fragmento** se utilizan para implementar la fragmentación.

Tiempo de vida: El campo *TTL* (Tiempo de vida) es un contador que se utiliza para limitar el tiempo de vida de un paquete. En un principio se suponía que iba a contar el tiempo en segundos, lo cual permitía un periodo de vida máximo de 255 seg. Hay que decrementarlo en cada salto (cada vez que el paquete atraviesa un router) y se supone que se decrementa muchas veces cuando un paquete se pone en cola durante un largo tiempo en un router. En la práctica, simplemente cuenta los saltos. Cuando el contador llega a cero, el paquete se descarta y se envía de regreso un paquete de aviso al host de origen. Esta característica evita que los paquetes anden vagando eternamente.

Protocolo: Una vez que la capa de red ha ensamblado un paquete completo, necesita saber qué hacer con él. El campo *Protocolo* le indica a cuál proceso de transporte debe entregar el paquete. TCP es una posibilidad, pero también están UDP y otros más. La numeración de los protocolos es global en toda la Internet [14].

Suma de verificación del encabezado: puesto que el encabezado transporta información vital, estima su propia suma de verificación por protección. El algoritmo suma todas las medias palabras de 16 bits del encabezado a medida que vayan llegando, mediante el uso de la aritmética de complemento a uno, y después obtiene el complemento a uno del resultado. Para los fines de este algoritmo, se supone que la Suma de verificación del encabezado es cero al momento de la llegada. Dicha suma de verificación es útil para detectar errores mientras el paquete viaja por la red. Tenga en cuenta que se debe recalcular en cada salto, ya que por lo menos hay un campo que siempre cambia (el campo *Tiempo de vida*).

Dirección origen: dirección IP del host fuente.

Dirección destino: dirección IP del host destino.

Opciones: El campo *Opciones* se diseñó para proporcionar un recurso que permitiera que las versiones subsiguientes del protocolo incluyeran información que no estuviera presente en el diseño original, para que los experimentadores puedan probar ideas nuevas y evitar la asignación de bits de encabezado a la información que se necesite muy poco. Las opciones son de longitud variable. Cada una empieza con un código de 1 byte que identifica la opción. Algunas opciones van seguidas de un campo de longitud de la opción de 1 byte, y luego de uno o más bytes de datos. El campo Opciones se rellena para completar múltiplos de 4 bytes.

3. IPv6

3.1. Antecedentes

El Protocolo IPv4 ha demostrado ser robusto, fácil de implementar e interoperable, y ser capaz de escalarse a nivel global como es hoy día el caso de Internet.

De todos modos, el diseño inicial no anticipaba las siguientes condiciones a la par del crecimiento abrupto de Internet:

El espacio de direcciones IPv4 no es suficiente: Como es sabido, las direcciones IPv4 consisten en 32 bits, lo que supone un numero de 2^{32} direcciones posibles. Pero debido a la alta demanda, hoy día ya no existen direcciones IPv4 disponibles para su adquisición [1].

Los sistemas de routers de backbone mantienen tablas de ruteo excesivamente largas: Debido al incorrecto planeamiento del desarrollo inicial del IPv4, muchos bloques de direcciones IPv4 están ubicados de manera discontinua, así las rutas no convergen de manera efectiva.

La auto-configuración y la reubicación de direcciones no son sencillas: Debido a que las direcciones IPv4 usan solo 32 bits y la asignación de direcciones no es pareja, las reubicación de direcciones IPv4 es usualmente necesaria cuando una red está en proceso de crecimiento. Por lo tanto, la auto-configuración y la reubicación son necesarias para reducir el trabajo de mantenimiento.

NAT (Network Address Translation) no siempre puede resolver el problema de escasez de direcciones: NAT adopta el método de mapeo entre direcciones internas privadas y direcciones externas públicas mediante puertos para resolver el problema de escasez, pero la relación del mapeo entre direcciones internas privadas y direcciones externas públicas (puertos) debe ser grande para que NAT sea efectivo. De todas formas, cuando hay muchos hosts dentro de la red usando el mismo puerto, el mismo protocolo no puede utilizarse para el mismo puerto externo utilizando NAT.

Dificultad para expandir la capacidad de la red: Diferentes redes pueden usar el mismo espacio de direcciones IPv4 privadas. Así, los conictos entre espacios de direcciones ocurren cuando estas redes se mezclan o interconectan. En esos casos, la reasignación de direcciones o NAT son necesarios para solucionar el conicto, pero incrementa la complejidad del manejo de la red.

3.2. Definición

El IPv6 es la nueva versión del Protocolo IP diseñada para reemplazar el IPv4. IPv6 nace como resultado de una serie de recomendaciones para una nueva versión del Protocolo IPv4. Estas recomendaciones fueron presentadas en 1995 como parte de lo que sería el IPng (IP Next Generation Protocol) [15].

Las recomendaciones para este protocolo incluyen un encabezado simplificado, con una estructura jerárquica de direcciones que permita agregación rigurosa de rutas, así como también suficiente direcciones como para cubrir la creciente demanda. El protocolo también debe incluir autenticación a nivel de paquete y encriptación junto con autoconfiguración. Este diseño cambia la manera en que las opciones del encabezado IP son codificadas para aumentar la exibilidad de introducir nuevas opciones en el futuro mientras se mejora la eficiencia. También debe incluir la habilidad de etiquetar ujos de tráfico.

3.3. Características

Como se ha mencionado, IPv6 fue diseñado como una versión mejorada del Protocolo IPv4. Es decir, todo lo que funcionaba perfectamente en IPv4 se ha mantenido, lo que no funcionaba se ha eliminado, y se ha tratado de añadir nuevas funciones manteniendo la compatibilidad entre ambos protocolos.

Las características principales de IPv6 son: [16]

- Mayor espacio de direcciones.
- Optimización del direccionamiento multicast y aparición del direccionamiento anycast.
- Autoconfiguración de los nodos.
- Encabezados más sencillos y eficientes
- Seguridad intrínseca en el núcleo del protocolo.
- Calidad de servicio y clases de servicios.
- Paquetes eficientes y extensibles.
- Ruteamiento más eficiente en la red troncal.
- Renumeración y multihoming, que facilita el cambio de proveedor de servicios.
- Características de movilidad.

3.4. Funciones [17]

La principal innovación de IPv6 es el uso de direcciones más extensas que con IPv4. Están codificadas con 16 bytes y esto permite que se resuelva el problema que hizo que IPv6 esté a la orden del día: brindar un conjunto prácticamente ilimitado de direcciones de Internet.

La mejora más importante de IPv6 es la simplificación de los encabezados de los datagramas. El encabezado del datagrama IPv6 básico contiene sólo 7 campos (a diferencia de los 14 de IPv4). Este cambio permite que los routers procesen datagramas de manera más rápida y mejore la velocidad en general.

La tercera mejora consiste en ofrecer mayor exibilidad respecto de las opciones. Este cambio es esencial en el nuevo encabezado, ya que los campos obligatorios de la versión anterior ahora son opcionales.

Además, la manera en la que las opciones están representadas es distinta, dado que permite que los routers simplemente ignoren las opciones que no están destinadas a ellos. Esta función agiliza los tiempos de procesamiento de datagramas.

Además, IPv6 brinda más seguridad. La autenticación y confidencialidad constituyen las funciones de seguridad más importantes del protocolo IPv6.

Finalmente, se ha prestado más atención que antes a los tipos de servicios. Si bien el campo Type of services (Tipo de servicios) en el datagrama IPv4 se utiliza pocas veces, el esperado aumento del tráfico multimedia en el futuro demanda que se le otorgue mayor importancia.

3.5. Estructura [18]

Una unidad de datos del protocolo de IPv6 (conocida como paquete) tiene el formato general siguiente:

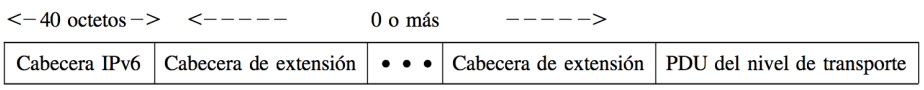

Figura 3. Formato de paquete IPv6

La única cabecera (o encabezado) que se requiere se denomina simplemente cabecera IPv6. ésta tiene una longitud fija de 40 octetos, comparados con los 20 octetos de la parte obligatoria de la cabecera IPv4.

Se han definido las siguientes cabeceras de extensión:

Cabecera de opciones salto a salto: define opciones especiales que requieren procesamiento en cada salto.

Cabecera de ruteo: proporciona un encaminamiento ampliado, similar al ruteo en el origen de IPv4.

Cabecera de fragmentación: contiene información de fragmentación y reensamblado.

Cabecera de autenticación: proporciona la integridad del paquete y la autenticación.

Cabecera de encapsulamiento de la carga de seguridad: proporciona privacidad.

Cabecera de las opciones para el destino: contiene información opcional para que sea examinada en el host destino.

El estándar IPv6 recomienda que, en el caso de que se usen varias cabeceras de extensión, las cabeceras IPv6 aparezcan en el siguiente orden:

- Cabecera IPv6: obligatoria, debe aparecer siempre primero.
- Cabecera de las opciones salto a salto.
- Cabecera de las opciones para el destino: para opciones a procesar por el primer destino que aparece en el campo dirección IPv6 de destino y por los destinos subsecuentes indicados en la cabecera de ruteo.
- Cabecera de ruteo.
- Cabecera de fragmentación.
- Cabecera de autenticación.
- Cabecera de encapsulado de la carga de seguridad.
- Cabecera de opciones para el destino: para opciones a procesar por el destino final del paquete.

La Figura 4 muestra un ejemplo de un paquete IPv6 que incluye un ejemplar de cada cabecera, excepto aquellas relacionadas con la seguridad. Obsérvese que la cabecera IPv6 y cada cabecera de extensión incluyen el campo cabecera siguiente. Este campo identifica el tipo de cabecera que viene a continuación. Si la siguiente cabecera es de extensión, entonces este campo contiene el identificador del tipo de esa cabecera. En caso contrario, este campo contiene el identificador del protocolo de la capa superior que está usando a IPv6 (normalmente un protocolo de la capa de transporte), utilizando el mismo valor que el campo protocolo de IPv4. En la Figura 4, el protocolo de la capa superior es TCP; por tanto, los datos de la capa superior transportados por el paquete IPv6 constan de una cabecera TCP seguida por un bloque de datos de aplicación.

3.6. Encabezado

Versión: El campo *Versión* siempre es 6 para IPv6 (y 4 para IPv4). Durante el periodo de transición de IPv4, que ya se ha tardado más de una década, los routers podrán examinar este campo para saber qué tipo de paquete tienen. Como observación adicional, para hacer esta prueba se desperdician unas cuantas instrucciones en la ruta crítica, dado que el encabezado de enlace de datos indica

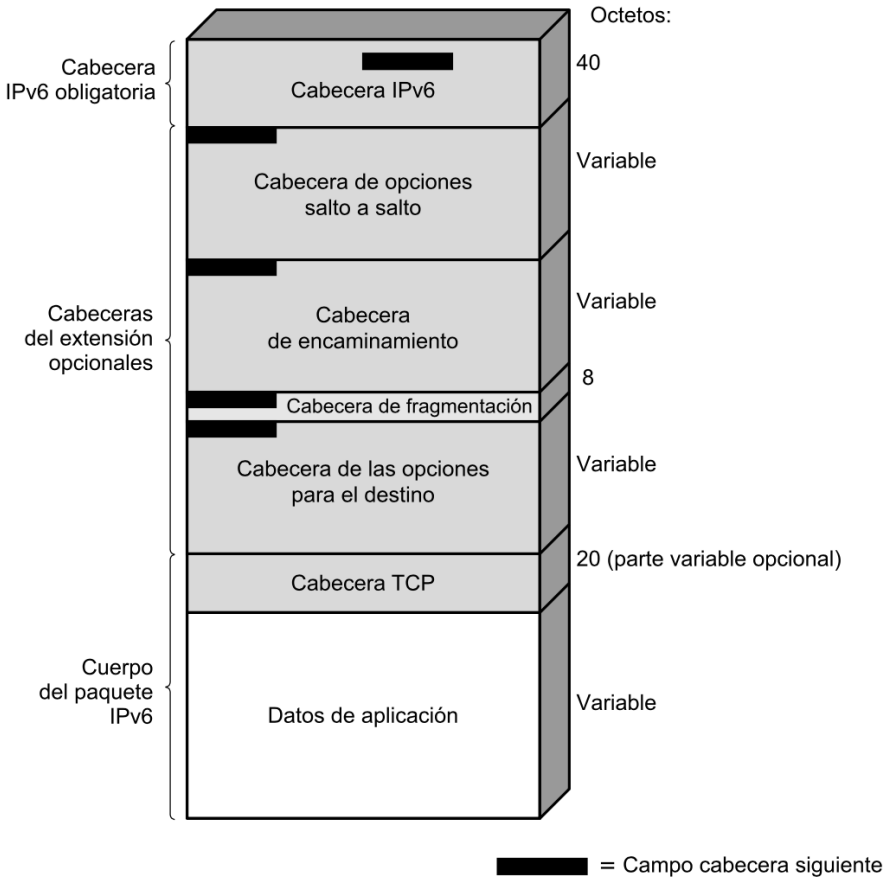

Figura 4. Paquete IPv6

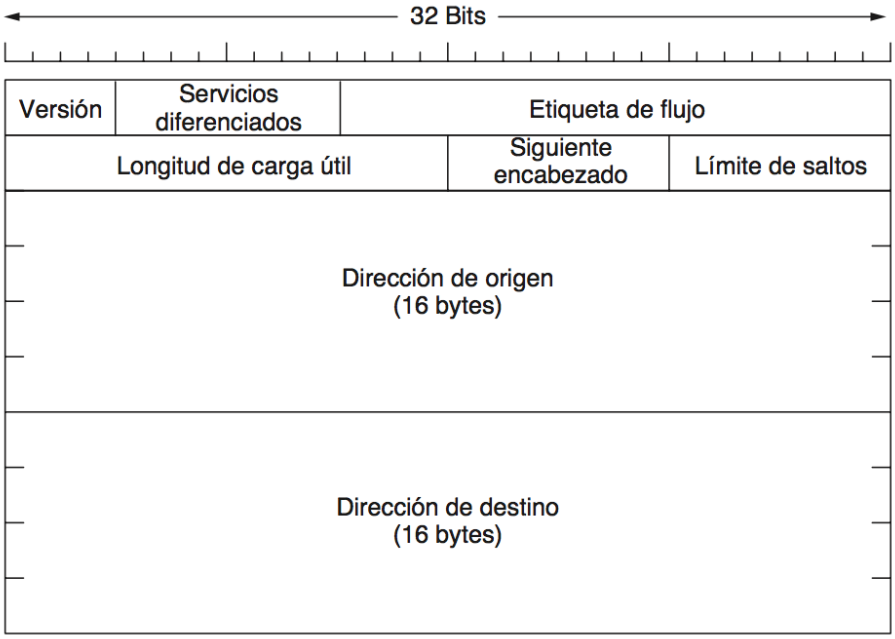

Figura 5. Encabezado IPv6

por lo general el protocolo de red para demultiplexar, por lo que tal vez algunos routers omitan la verificación [19].

Servicios diferenciados: El campo *Servicios diferenciados* (originalmente conocido como Clase de tráfico) se utiliza para distinguir la clase de servicio para los paquetes con distintos requerimientos de entrega en tiempo real. Se utiliza con la arquitectura de servicio diferenciado para la calidad del servicio, de la misma forma que el campo con el mismo nombre en el paquete IPv4. Además, los 2 bits de menor orden se usan para señalar las indicaciones explícitas de congestión, de nuevo en la misma forma que IPv4 [19].

Etiquetas de ujo: El campo *Etiqueta de ujo* proporciona el medio para que un origen y un destino marquen grupos de paquetes que tengan los mismos requerimientos y que la red deba tratar de la misma forma, para formar una seudoconexión. Por ejemplo, un ujo de paquetes de un proceso en cierto host de origen dirigido a cierto proceso en un host de destino puede tener requisitos muy estrictos de retardo y, por lo tanto, necesitar un ancho de banda reservado. El ujo se puede establecer por adelantado, dándole un identificador. Cuando aparece un paquete con una etiqueta de ujo diferente de cero, todos los routers pueden buscarla en sus tablas internas para ver el tipo de tratamiento especial que requiere. En efecto, los ujos son un intento de tener lo mejor de ambos mundos:

la exibilidad de una red de datagramas y las garantías de una red de circuitos virtuales. Para fines de la calidad del servicio, cada ujo está designado con base en la dirección de origen, la dirección de destino y el número de ujo. Este diseño significa que pueden estar activos hasta 220 ujos al mismo tiempo entre un par dado de direcciones IP. También significa que, incluso si dos ujos provenientes de hosts diferentes pero con la misma etiqueta de ujo pasan por el mismo router, éste será capaz de distinguirlos mediante las direcciones de origen y de destino. Lo ideal es que las etiquetas de flujo se escojan al azar, en vez de asignarlas de manera secuencial comenzando por el 1, de modo que los routers tengan que usar hashing [19].

Longitud de carga útil: El campo *Longitud de carga útil* indica cuántos bytes van después del encabezado de 40 bytes. El nombre se cambió de Longitud total en el IPv4 porque el significado cambió ligeramente: los 40 bytes del encabezado ya no se cuentan como parte de la longitud (como antes). Este cambio significa que ahora la carga útil puede ser de 65.535 bytes en vez de sólo 65.515 bytes [19].

Siguiente encabezado: El campo *Siguiente encabezado* revela el secreto. La razón por la que pudo simplificarse el encabezado es que puede haber encabezados adicionales (opcionales) de extensión. Este campo indica cuál de los seis encabezados de extensión (en la actualidad), siguen de éste, en caso de que los haya. Si este encabezado es el último encabezado de IP, el campo Siguiente encabezado indica el protocolo de transporte (por ejemplo, TCP, UDP) al que se entregará el paquete [19].

Límite de saltos: El campo *Límite de saltos* se usa para evitar que los paquetes vivan eternamente. En la práctica es igual al campo Tiempo de vida del IPv4; esto es, un campo que se disminuye en cada salto. En teoría, en el IPv4 era un tiempo en segundos, pero ningún router lo usaba de esa manera, por lo que se cambió el nombre para reejar la manera en que se usa realmente [19].

Dirección de origen: dirección IPv6 del host origen [19].

Dirección de destino: dirección IPv6 del host destino. [19].

3.7. Direcciones IPv6

El Protocolo IPv6 utiliza un modelo de direcciones de 128 bits, que contienen un campo de "alcance" que determina que aplicación es conveniente para esa dirección. El Protocolo IPv6 no soporta dirección broadcast, pero en compensación utiliza direcciones multicast para resolver ese problema. Adicionalmente, el Protocolo IPv6 implementa un nuevo tipo de dirección llamada anycast.

Representación de las direcciones IPv6: Las direcciones IPv6 consisten en 8 grupos de valores hexadecimales de 16 bits separados por dos puntos (:). Las direcciones IPv6 tienen un formato como este:

aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa

Cada "aaaa" es un valor hexadecimal de 16 bits, y cada "a" es un valor hexadecimal de 4 bits. Ejemplo de una dirección IPv6:

8000:0000:0000:0000:0123:4567:89AB:CDEF

Ya que muchas direcciones tendrán muchos ceros en ellas, se han autorizado tres optimizaciones. Primero, se pueden omitir los ceros a la izquierda dentro de un grupo, por lo que es posible escribir 0123 como 123. Segundo, se pueden reemplazar uno o más grupos de 16 bits cero por un par de signos de dos puntos. Por lo tanto, la dirección anterior se vuelve ahora: [20]

8000::123:4567:89AB:CDEF

Por último, las direcciones IPv4 se pueden escribir como un par de signos de dos puntos y un número decimal anterior separado por puntos, por ejemplo:

::192.31.20.46

Tipos de direcciones IPv6:

Unidifusión (unicast): un identificador para una interfaz individual. Un paquete enviado a una dirección de este tipo se entrega a la interfaz identificada por esa dirección [21].

Monodifusión (anycast): un identificador para un conjunto de interfaces (normalmente pertenecientes a diferentes nodos). Un paquete enviado a una dirección monodifusión se entrega a una de las interfaces identificadas por esa dirección (la más cercana, de acuerdo a la medida de distancia de los protocolos de ruteo) [21].

Multidifusión (multicast): un identificador para un conjunto de interfaces (normalmente pertenecientes a diferentes nodos). Un paquete enviado a una dirección multidifusión se entrega a todas las interfaces identificadas por esa dirección [21].

3.8. Encabezados de extensión [22]

Encabezado de opciones de salto a salto: El encabezado de opciones salto a salto transporta información opcional que, si está presente, debe ser examinada por cada dispositivo de ruteo a lo largo de la ruta. Este encabezado contiene

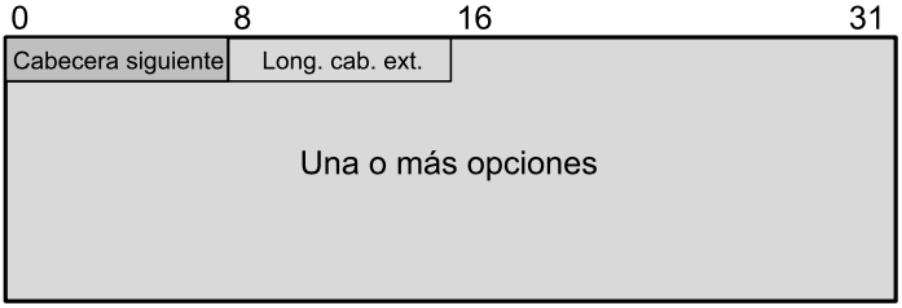

Figura 6. Encabezado de opciones de salto a salto - IPv6

los siguientes campos:

Cabecera siguiente (8 bits): identifica el tipo de encabezado que sigue inmediatamente a ésta.

Longitud del encabezado de extensión (8 bits): longitud del encabezado en unidades de 64 bits, sin incluir los primeros 64 bits.

Opciones: campo de longitud variable que consta de una o más definiciones de opción. Cada definición se expresa mediante tres subcampos: tipo de opción (8 bits), que identifica la opción; longitud (8 bits), que especifica la longitud en octetos del campo de datos de la opción; y datos de opción, que es una especificación de la opción de longitud variable.

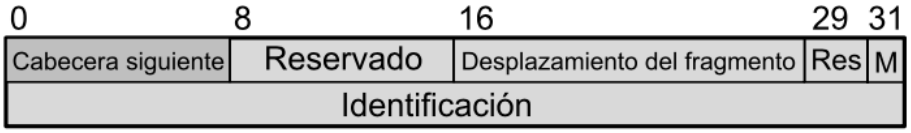

Figura 7. Encabezado de fragmentación - IPv6

Encabezado de fragmentación: En el Protocolo IPv6, la fragmentación sólo puede ser realizada por el nodo origen, no por los dispositivos de routers a lo largo del camino del paquete. Para obtener todas las ventajas del entorno de interconexión, un nodo debe ejecutar un algoritmo de obtención de la ruta, lo que permite conocer la unidad máxima de transferencia (MTU, Maximum Transfer Unit) permitida por cada red en la ruta. Con este conocimiento, el nodo origen

fragmentará el paquete, según se requiera, para cada dirección de destino dada. Si no se ejecuta este algoritmo, el origen debe limitar todos los paquetes a 1.280 octetos, que debe ser la mínima MTU que permitan las redes.

El encabezado de fragmentación contiene los siguientes campos:

Cabecera siguiente (8 bits): identifica el tipo de encabezado que sigue inmediatamente a ésta.

Reservado (8 bits): reservado para usos futuros.

Desplazamiento del fragmento (13 bits): indica dónde se sitúa en el paquete original la carga útil de este fragmento. Se mide en unidades de 64 bits. Esto implica que los fragmentos (excepto el último) deben contener un campo de datos con una longitud de múltiplo de 64 bits.

Reservado (2 bits): reservado para usos futuros.

Indicador M (1 bit): 1 % más fragmentos; 0 % último fragmento.

Identificación (32 bits): utilizado para identificar de forma única el paquete original. El identificador debe ser único para la dirección origen y dirección destino durante el tiempo que el paquete permanece en Internet. Todos los fragmentos con el mismo identificador, dirección origen y dirección destino son reensamblados para recuperar el paquete original.

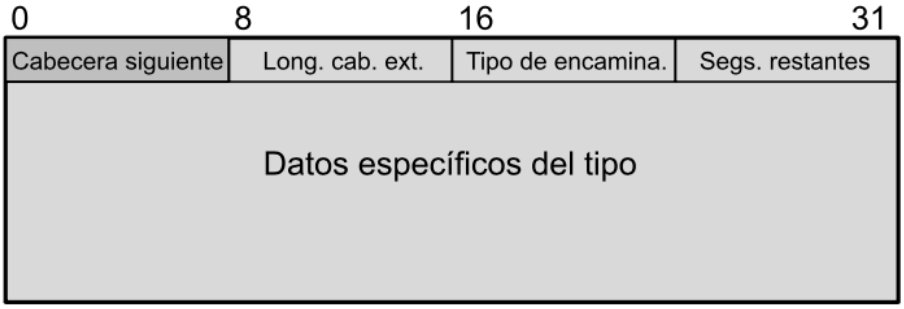

Figura 8. Encabezado de ruteo - IPv6

Encabezado de ruteo: El encabezado de ruteo contiene una lista de uno o más nodos intermedios por los que se pasa en el camino del paquete a su destino. Todas los encabezados de ruteo comienzan con un bloque de 32 bits consistente en 4 campos de 8 bits, seguido por datos de ruteo específicos al tipo de ruteo dado. Los cuatro campos de 8 bits son los siguientes:

Encabezado siguiente (8 bits): identifica el tipo de encabezado que sigue inmediatamente a ésta.

Longitud del encabezado de extensión (8 bits): longitud de este encabezado en unidades de 64 bits, sin incluir los primeros 64 bits.

Tipo de ruteo (8 bits): identifica una variante particular del encabezado de ruteo. Si un dispositivo de ruteo no reconoce el valor del tipo de ruteo, debe descartar el paquete.

Segmentos restantes (8 bits): número de segmentos en la ruta que quedan; esto es, el número de nodos intermedios explícitamente contenidos en la lista que se visitarán todavía antes de alcanzar el destino.

Encabezado de opciones para el destino: El encabezado de opciones para el destino lleva información opcional que, si está presente, se examina por el nodo destino del paquete. El formato de este encabezado es el mismo que el encabezado de opciones salto a salto.

3.9. Comparación de encabezados IPv4 - IPv6

Para fines instructivos, podemos comparar el encabezado de IPv4 con el de IPv6 para ver lo que se ha dejado fuera del IPv6. El campo IHL se fue porque el encabezado de IPv6 tiene una longitud fija. El campo Protocolo se retiró porque el campo Siguiente encabezado indica lo que sigue después del último encabezado IP (por ejemplo, un segmento UDP o TCP). Se eliminaron todos los campos relacionados con la fragmentación, puesto que el IPv6 tiene un enfoque distinto en cuanto a la fragmentación. Para empezar, todos los hosts que cumplen con el IPv6 deben determinar en forma dinámica el tamaño de paquete a usar. En resumen, cuando un host envía un paquete IPv6 demasiado grande, en vez de fragmentarlo, el router que no puede reenviarlo descarta el paquete y envía un mensaje de error de vuelta al host emisor. Este mensaje indica al host que divida todos los paquetes futuros para ese destino. Hacer que el host envíe paquetes del tamaño correcto en primer lugar es, en última instancia, mucho más eficiente que hacer que los routers los fragmenten sobre la marcha. Asimismo, el tamaño mínimo de paquete se incrementó de 576 a 1280 bytes para permitir 1024 bytes de datos y muchos encabezados.

Por último, el campo Suma de verificación desapareció porque al calcularlo se reduce en gran medida el desempeño. Con las redes confiables de hoy, además del hecho de que la capa de enlace de datos y las capas de transporte por lo general tienen sus propias sumas de verificación, la ventaja de tener otra suma de verificación no valía el costo de desempeño que generaba. Al quitar estas características, el resultado es un protocolo de capa de red compacto y sólido.

Por lo tanto, con este diseño se cumplió la meta del IPv6 (un protocolo rápido y exible con bastante espacio de direcciones).

4. IMPLEMENTACIÓN DEL PROTOCOLO IPv6 EN INTERNET

Debido a la gran predominancia del Protocolo IPv4 en Internet, la implementación del Protocolo IPv6 tomará mucho trabajo. En primer lugar, el Protocolo IPv4 y el Protocolo IPv6 no son compatibles entre sí pese a tantas similitudes, lo que implica utilizar nuevas técnicas para que ambos protocolos puedan coexistir en Internet. La técnica utilizada para abordar ese problema en la actualidad se llama Tunelización.

4.1. Tunelización

Es en extremo difícil manejar el caso general de hacer que dos redes distintas se interconecten. Sin embargo, hay un caso especial que se puede manejar, incluso para distintos protocolos de red. Este caso es cuando el host de origen y el de destino están en el mismo tipo de red, pero hay una red diferente en medio. Como ejemplo, piense en un banco internacional con una red IPv6 en París, una en Londres y una conectividad entre las oficinas a través de la Internet IPv4. Esta situación se muestra en la figura 9.

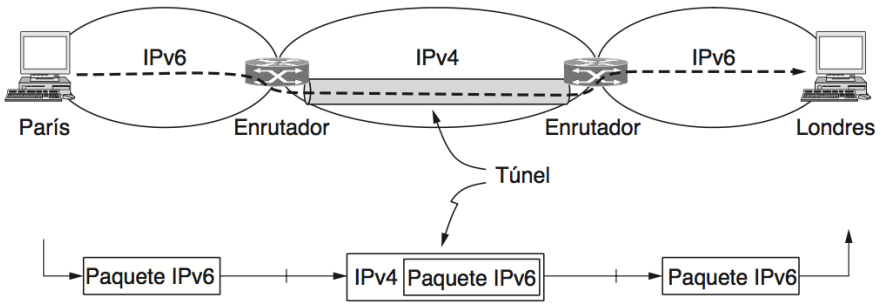

Figura 9. Tunelización - IPv6 - IPv4

La solución a este problema es una técnica llamada tunelización (tunneling). Para enviar un paquete IP a un host en la oficina de Londres, un host en la oficina de París construye el paquete que contiene una dirección IPv6 en Londres y la envía al router multiprotocolo que conecta la red IPv6 de París con la

Internet IPv4. Cuando este router recibe el paquete IPv6, lo encapsula con un encabezado IPv4 dirigido al lado IPv4 del router multiprotocolo que se conecta con la red IPv6 de Londres. Es decir, el router coloca un paquete (IPv6) dentro de un paquete (IPv4). Cuando llega este paquete envuelto, el router de Londres extrae el paquete IPv6 original y lo envía hacia el host de destino.

Podemos visualizar la ruta hacia la Internet IPv4 como un gran túnel que se extiende de un router multiprotocolo al otro. El paquete IPv6 simplemente viaja de un extremo del túnel al otro, bien acomodado en una caja bonita. No tiene que preocuparse por lidiar con IPv4 para nada. Tampoco tienen que hacerlo los hosts en París o en Londres. Sólo los routers multiprotocolo tienen que entender los paquetes IPv4 e IPv6. De hecho, el recorrido completo desde un router multiprotocolo al otro es como un salto a través de un solo enlace.

La tunelización se utiliza mucho para conectar hosts y redes aisladas mediante el uso de otras redes. La red que resulta se denomina red superpuesta (overlay), ya que realmente está superpuesta sobre la red base. Implementar un protocolo de red con una nueva característica es un motivo común. La desventaja de la tunelización es que no se puede llegar a ninguno de los hosts en la red que se tuneliza debido a que los paquetes no pueden escapar a mitad del túnel. Sin embargo, esta limitación de los túneles se convierte en una ventaja gracias a las redes VPN (Redes Privadas Virtuales, del inglés Virtual Private Network). Una VPN es simplemente una red superpuesta que se utiliza para proporcionar una medida de seguridad.

5. DISCUSIÓN

5.1. Controversia del Protocolo IPv6

Dados el proceso de diseño abierto y las fuertes opiniones de muchas de las personas participantes, no debería sorprender que muchas decisiones tomadas para el Protocolo IPv6 fueran tema de fuertes controversias. Ya se ha mencionado el argumento sobre la longitud de las direcciones. El resultado fue una solución intermedia: direcciones de 16 bytes de longitud fija.

Surgió otra pelea sobre la longitud del campo Límite de saltos. Una parte sentía que limitar la cantidad máxima de saltos a 255 (implícita al usar un campo de 8 bits) era un grave error. A fin de cuentas, hoy son comunes las rutas de 32 saltos, y en 10 años más podrán ser comunes rutas mucho más extensas. Esta gente argumentaba que usar una dirección con un tamaño enorme era ir demasiado lejos, pero que el uso de una pequeña cuenta de saltos era tener una visión miope.

La respuesta fue que podían existir buenas razones para aumentar todos los campos, lo que llevaría a un encabezado congestionado. Además, la función del

campo Límite de saltos es evitar que los paquetes vaguen por mucho tiempo, por lo cual 65.535 saltos son demasiados. Por último, a medida que crezca Internet se construirán cada vez más enlaces de larga distancia, mediante lo cual será posible ir de un país a otro en media docena de saltos, cuando mucho. Si se requieren más de 125 saltos para llegar del origen y el destino a sus puertas de enlace internacionales, algo está mal con las redes troncales nacionales. Los de 8 bits ganaron esta partida.

Otra "papa caliente" fue el tamaño máximo de paquete. La comunidad de las supercomputadoras quería paquetes de más de 64 KB. Cuando una supercomputadora comienza a transferir, el asunto realmente va en serio, y no quiere que se le interrumpa cada 64 KB. El argumento en contra de los paquetes grandes es que, si un paquete de 1 MB llega a una línea de 1.5 Mbps, ese paquete bloqueará la línea durante más de 5 segundos, produciendo un retardo muy notorio para los usuarios interactivos que comparten la línea. Se llegó a un punto medio: los paquetes normales se limitan a 64 KB, pero se puede usar el encabezado de extensión de salto por salto para permitir los jumbogramas (paquetes excesivamente grandes).

Un tercer tema candente fue la desaparición de la suma de verificación del IPv4. Para algunas personas esto representa algo parecido a quitarle los frenos a un automóvil. Hacerlo ciertamente aligera al automóvil y, por lo tanto, puede ir más rápido pero, de ocurrir un evento inesperado, tendremos problemas.

El argumento en contra de las sumas de verificación fue que cualquier aplicación a la que de verdad le importe la integridad de sus datos de todos modos tiene que tener una suma de verificación en la capa de transporte, por lo que tener otra en el IP (además de la suma de verificación de la capa de enlace de datos) es un exceso. Además, la experiencia mostraba que calcular la suma de verificación del IP representaba un gasto considerable en el IPv4. El bando en contra de la suma de verificación ganó esta partida, por lo que el IPv6 no tiene una suma de verificación.

Los hosts móviles también fueron tema de contienda. Si una computadora portátil vuela al otro lado del mundo, ¿puede continuar operando ahí con la misma dirección IPv6, o tiene que usar un esquema con agentes de base? Algunas personas querían incluir soporte explícito para hosts móviles en el IPv6. Este esfuerzo falló cuando no se pudo generar consenso para ninguna propuesta específica. Es probable que la batalla más importante fue sobre la seguridad. Todos estaban de acuerdo con que era esencial. La guerra fue sobre donde y cuando usarla. Primero donde. El argumento a favor de ponerla en la capa de red es que entonces se vuelve un servicio estándar que todas las aplicaciones pueden usar sin necesidad de planearlo por adelantado. El argumento en contra es que las aplicaciones realmente seguras por lo general no quieren nada menos que el cifrado de terminal a terminal, donde la aplicación de origen hace el cifrado y

la aplicación de destino lo deshace. Con cualquier otra cosa menos, el usuario está a merced de implementaciones de capa de red con fallas potenciales, sobre las que no tiene control. La respuesta a este argumento es que tales aplicaciones simplemente pueden abstenerse de usar las características de seguridad del IP y encargarse ellas mismas del asunto. La réplica a esto es que la gente que no confía en que la red lo haga bien no quiere pagar el precio de implementaciones de IP lentas y estorbosas que tengan esta capacidad, aun si está deshabilitada.

Otro aspecto sobre dónde poner la seguridad se relaciona con el hecho de que muchos países (pero no todos) tienen leyes de exportación muy estrictas en lo referente al cifrado. Algunos, en especial Francia e Irak, también restringen mucho su uso doméstico, de manera que la gente no pueda ocultar secretos al gobierno. Como resultado, cualquier implementación de IP que use un sistema de cifrado lo bastante robusto como para tener algún valor no podría exportarse de Estados Unidos (y de muchos otros países) a clientes mundiales. Tener que mantener dos conjuntos de software, uno para uso doméstico y otro para exportación, es algo a lo que la mayoría de los distribuidores de computadoras se oponen enérgicamente.

Un punto donde no hubo controversia es que nadie espera que la Internet IPv4 se apague un domingo por la mañana y reinicie como Internet IPv6 la mañana del lunes. En cambio, se convertirán islas de IPv6, que en un principio se comunicarán a través de túneles. A medida que crezcan las islas IPv6, se integrarán a islas más grandes. Tarde o temprano todas las islas se integrarán, y la Internet se habrá convertido por completo.

Por lo menos ése era el plan. La implementación ha demostrado ser el talón de Aquiles de IPv6. Se sigue usando muy poco, aun cuando todos los sistemas operativos principales tienen soporte completo para IPv6. La mayoría de las implementaciones son nuevos casos en los que un operador de red (por ejemplo, un operador de telefonía móvil) necesita una gran cantidad de direcciones IP. Se han de nido muchas estrategias para ayudar a facilitar la transición. Entre éstas hay formas de configurar de manera automática los túneles que transportan IPv6 sobre la Internet IPv4, y formas de que los hosts encuentren de manera automática las terminales de los túneles. Los hosts de pila dual tienen una implementación IPv4 y otra IPv6, de modo que puedan seleccionar qué protocolo usar dependiendo del destino del paquete. Estas estrategias optimizarán la implementación substancial que parece inevitable cuando se agoten las direcciones IPv4.

6. CONCLUSIÓN

El Protocolo IPv6 es la versión más nueva del Protocolo IPv4. El Protocolo IPv6 hereda las características funcionales del Protocolo IPv4, y elimina aquellas

características no funcionales. Es decir, todo lo que funcionaba perfectamente en IPv4 se ha mantenido, lo que no funcionaba se ha eliminado.

La principal causa que llevó a la creación del Protocolo IPv6 es la poca cantidad de direcciones IPv4 disponibles para la demanda actual.

El Protocolo IPv6 demostró ser más eficiente que el Protocolo IPv4, y también, más seguro. La eficiencia se logra mediante los encabezados variables del Protocolo IPv6. Esta eficiencia es bien apreciada en los routers del backbone.

En la actualidad, Internet es regida casi en su totalidad por el Protocolo IPv4, lo que lleva a un difícil despliegue total del Protocolo IPv6, ya que ambos protocolos, por mas parecidos que sean, no son compatibles entre sí en una misma red.

Para poder solucionar este problema de incompatibilidad entre protocolos, se utiliza una técnica llamada tunelización, que permite interconectar dos redes IPv6, atravesando un enlace IPv4 en el backbone de routers, y viceversa.

Como se pudo ver anteriormente, la creación del Protocolo IPv6 generó muchas controversias en distintas características del mismo, pero eventualmente dichas controversias fueron argumentadas con nuevas características del Protocolo IPv6 que pudiesen solucionar los problemas propuestos.

En una época donde la demanda de Internet crece exponencialmente, y donde ya no quedan direcciones IPv4 disponibles, la creación e implementación del Protocolo IPv6 demuestran ser una solución viable para sostener dicho crecimiento y demanda en el futuro próximo.

Referencias

1. Smith, Lucie; Lipner, Ian. February 2011. www.nro.net/news/ipv4-free-pool-depleted

2. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 394

3. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 390

4. Deering, S.; Hinden, R. 1998. *Internet Protocol, Version 6 (IPv6)*. RFC 2460

5. Bradner, S.; Mankin, A. 1995. *The Recommendation for the IP Next Generation Protocol*. RFC 1752

6. Postel, J. 1981. *Internet Control Message Protocol*. RFC 792

7. Postel, J. 1981. *Internet Protocol*. RFC 791. Pag. 5

8. Postel, J. 1981. *Internet Protocol*. RFC 791. Pag. 6

9. Postel, J. 1981. *Internet Protocol*. RFC 791. Pag. 7-9

10. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 379-381

11. Stallings, W. 2004. *Data and Computer Communications*. 7th Edition. Pag. 609-611

12. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 376-379

13. Postel, J. 1981. *Internet Protocol*. RFC 791. Pag. 11-16

14. Postel, J.; Reynolds, J. 1994. *Assigned Numbers*. RCF 1700

15. Bradner, S.; Mankin, A. 1995. *The Recommendation for the IP Next Generation Protocol*. RFC 1752. Pag. 21

16. Millán, R. 2001. *El Protocolo IPv6*. www.ramonmillan.com/tutoriales/ipv6_parte1.php

17. VanHaute, N. October 2017. *Protocolo IPv6*. es.ccm.net/contents/268-protocolo-ipv6

18. Stallings, W. 2004. *Data and Computer Communications*. 7th Edition. Pag. 619-620

19. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 392-393

20. Tanenbaum, A.; Wetherall, D. 2011. *Computer Networks*. 2nd Edition. Pag. 394

21. Stallings, W. 2004. *Data and Computer Communications*. 7th Edition. Pag. 624

22. Stallings, W. 2004. *Data and Computer Communications*. 7th Edition. Pag. 624-627