Universitat Autònoma de Barcelona

Escola Tècnica Superior d'Enginyeria
(Secció d'Enginyeria superior en Informàtica)

El protocolo IPv6 y sus extensiones de seguridad IPSec

Memoria del proyecto de fin de carrera correspondiente a los estudios de ***Ingeniería Superior en Informática***, presentado por **Gabriel Verdejo Alvarez** y dirigido por **Joan Borrell Viader**.

Bellaterra, Febrero del 2000.

El firmante, Sr. Joan Borrell Viader, profesor del departamento de informática de la Universidad Autónoma de Barcelona certifica:

Que la presente memoria ha sido realizada bajo su dirección por Gabriel Verdejo Alvarez.

Bellaterra, Febrero del 2000.

Firmado: Joan Borrell Viader

"*Conviene que me contradiga de tanto en tanto, más que nada para que parezca que no estoy solo.*" Comentó el rey Sol a un cortesano

A todos aquellos que me han ayudado durante toda mi vida. A los demás, no.
A Meritxell, por todo. Ya tienes tu nombre en un libro.

Índice

CAPITULO 1

Introducción

En un principio (años 60), la conexión entre ordenadores sólo era posible entre modelos de una misma marca, llegándose incluso a dar el caso de que diferentes modelos de un mismo fabricante no podían comunicarse entre sí. Esta limitación venía dada por el *hardware* que componía la máquina y cómo esta era gobernada por el *software* que la hacía funcionar. Además, esta comunicación sólo era posible entre dos ordenadores (vía línea telefónica, comunicación por un puerto serie o paralelo, etc.).

De esta forma, la incompatibilidad entre diferentes fabricantes de ordenadores era casi total, lo que implicaba tres serias limitaciones a la expansión de la informática:

- Se debían comprar todas las máquinas y *periféricos*[1] (Devices) al mismo fabricante. Esto permitía un monopolio absoluto de las pocas marcas existentes.

[1] Los términos de esta memoria que aparecen en cursiva están definidos en el glosario de términos.

- El precio de estos primeros ordenadores era muy elevado, de tal forma que tan sólo eran asumibles para gobiernos, universidades y grandes empresas.

- Las posibilidades de conexión entre diferentes equipos estaban limitadas a ordenadores de la misma marca, llegándose incluso al caso de ser sólo compatibles con su mismo modelo y no con el resto de ordenadores de la misma casa.

A partir de los años 70, las redes de interconexión de ordenadores ya habían avanzado bastante, permitiendo la comunicación de diferentes tipos de máquinas. Para conseguir este objetivo, fue básico que organismos internacionales (IEEE, ISO...) regularan los mecanismos y formas en que se debían realizar las comunicaciones entre equipos informáticos. Se definió un modelo teórico denominado *modelo OSI*, que especificaba un conjunto de 7 capas que permitían aislar el ordenador de la red a la que se encontraba conectado, permitiendo la interconexión de diferentes ordenadores a una misma red.

Una vez conseguida la conexión de ordenadores diferentes en una misma red, se pasó a una segunda fase que consistió en la comunicación entre los diferentes tipos de redes existentes hasta formar la gran red de redes que actualmente denominamos *INTERNET*.

La red INTERNET utiliza una familia de *protocolos* denominada TCP/IP (entre los que podemos destacar TCP, UDP e IP). **IP** (INTERNET Protocol) es un protocolo de interconexión de redes heterogéneas que se encarga del transporte de los *datagramas* (paquetes de datos) a través de la red. **UDP** (User Datagram Protocol) es un protocolo de comunicación entre ordenadores de nivel superior al IP, sin conexión y que no proporciona fiabilidad a la comunicación. **TCP** (Transmission Control Protocol) es también un protocolo de nivel superior al IP, pero orientado a conexión y fiable.

Los protocolos TCP/IP han gobernado (y seguirán haciéndolo) el funcionamiento de la red de redes INTERNET. No obstante, estos han ido sufriendo algunas revisiones en sus definiciones originales para corregir imperfecciones o ajustarse a las necesidades actuales. Estas revisiones han sido en forma de nuevas versiones de los protocolos, de esta forma, INTERNET en el año 2000 viene gobernada por la versión 4 del protocolo IP (también denominado IPv4).

Desde hace unos años se está trabajando en una nueva revisión del protocolo IP. Esta versión del protocolo IP es la número 6, que por cuestiones de nomenclatura ha pasado a denominarse IPv6 (IP versión 6) o IPng (IP next generation).

Esta revisión de las especificaciones actuales del protocolo IP ha sido motivada principalmente al hecho de que el sistema de direccionamiento (actualmente se utilizan 32 bits), se ha quedado pequeño debido al gran auge de INTERNET (en los últimos 10 años se ha experimentado un crecimiento superior al 600%), y no se puede absorber la demanda de nuevas direcciones. Este aumento ha llevado al limite las posibilidades de los diferentes protocolos de *encaminamiento* (Border Gateway Protcolol BGP, External Gateway Protocol EGP...), ralentizando en exceso el movimiento de los datagramas que circulan por INTERNET.

Además, los últimos servicios que se ofrecen sobre esta red (comercio electrónico o *e-comerce*, redes corporativas o Virtual Private Networks, video-conferencia...) requieren de unos componentes tanto de velocidad como de seguridad y autenticidad que la versión 4 del protocolo IP no contempla. Todo esto ha provocado la creación de algunas extensiones de seguridad (como el *Secure Socket Layer, SSL)*, que no forman parte de la definición del protocolo IP.

Debido a las carencias anteriormente citadas (y algunas que en menor medida representan limitaciones a las posibilidades de expansión), se ha motivado esta nueva revisión del protocolo IP. Esta debe realizarse de una forma transparente al usuario y rápidamente, puesto que si se mantiene el ritmo de crecimiento actual de INTERNET, entre los años 2005 y 2007 se habrán agotado todas las direcciones.

Nuestro proyecto denominado "**El protocolo IPv6 y sus extensiones de seguridad IPSec**", se propone los siguientes objetivos:

1. Estudiar las especificaciones de esta revisión del protocolo IP. Sus nuevas características y aportaciones así como sus semejanzas y diferencias con la versión 4.

2. Realizar una breve compilación de los sistemas de seguridad que actualmente se utilizan en INTERNET (Secure Socket Layer SSL, Secure Electronic Transaction SET,...).

3. Estudiar las extensiones de seguridad del protocolo IPv6 (que se denominan **IPSec,** abreviatura de *IP Security*) y cuales son los nuevos servicios que proporcionan.

4. Implementación del protocolo IPv6 en la plataforma LINUX y realización de diferentes pruebas para comprobar sus capacidades prácticas así como sus ventajas e inconvenientes respecto a la versión 4.

5. Finalmente reflejar los resultados, conclusiones y experimentos derivados de los puntos anteriores.

La estructura de esta memoria está basada en una serie de capítulos en las que se recogen los diferentes aspectos mencionados anteriormente según un esquema de comparación entre las dos versiones del protocolo IP:

Capítulo 2: Estado actual del protocolo IP versión 4 y los protocolos superiores TCP y UDP. Explicación de la evolución histórica de INTERNET hasta el año 2000 y de su esquema de funcionamiento interno (sistema de capas). Descripción de los principales protocolos que la gobiernan actualmente (IP, TCP y UDP).

Capítulo 3: Extensiones de seguridad sobre los protocolos TCP e IP versión 4. Explicación de los conceptos de comunicación y seguridad en una red de ordenadores. Clasificación de las diferentes amenazas que pueden producirse en una comunicación por INTERNET y las distintas soluciones que se aplican actualmente (sistemas de clave privada y clave pública). Explicación de los conceptos, algoritmos y legislación vigente referente a certificados y firmas digitales en el comercio electrónico (e-comerce).

Capítulo 4: El protocolo IP versión 6. Justificación de esta revisión del protocolo IP (falta de flexibilidad para adecuarse a las nuevas necesidades de los usuarios, limitaciones en su diseño, ausencia de seguridad...) y explicación de las características más relevantes de la versión 6 del protocolo IP (novedades, diferencias y puntos en común con la versión 4).

Capítulo 5: Las extensiones seguridad IPSec para IP versión 6. Introducción al estado actual de la seguridad en INTERNET y justificación de la necesidad de servicios de seguridad en INTERNET. Explicación de las especificaciones IPSec así como de sus características y servicios previstos más relevantes.

Capítulo 6: Experimentos realizados. En este capítulo se detallan las diferentes pruebas prácticas realizados en cada uno de los puntos anteriores así como los resultados obtenidos.

Capítulo 7: Conclusiones. En este capítulo se comentan las conclusiones extraídas tanto en la elaboración de este proyecto como en cada uno de los capítulos anteriores. Además se detallan posibles ampliaciones basadas en este estudio para futuros proyectos así como posibles nuevos enfoques para el área de seguridad en IPv6.

Señalar que las especificaciones existentes, comentadas y probadas en el momento de finalizar esta memoria **son provisionales**, ya que tanto IPv6 como IPSec permanecen aún en fase de diseño y por tanto están en fase **experimental**, con lo que en las especificaciones definitivas probablemente puedan diferir de lo aquí comentado y/o probado.

CAPITULO 2

Estado actual del protocolo IP versión 4 y los protocolos superiores TCP y UDP

2.1 Inicios y evolución de INTERNET

INTERNET ha revolucionado el mundo de los ordenadores y las comunicaciones de una forma radical sin precedentes. La invención del telégrafo y posteriormente la del teléfono, radio, ordenadores y redes de paquetes sentaron las bases para esta integración de capacidades que hasta este momento parecían utópicas. Además, INTERNET representa uno de los mayores ejemplos de los beneficios obtenidos mediante la investigación y el desarrollo en el campo de la información.

La siguiente información ha sido extraída en su mayor parte de las referencias [WWW14], [WWW16], [WWW18], [WWW19], [WWW22] y [WWW25]. También han sido consultados algunos capítulos de [Cer92], [Abo93], [Ric98] y [Hui97].

Los primeros indicios documentados sobre la creación de una red global para la conexión de ordenadores, pueden encontrarse en una serie de memorándums escritos por J.C.R. Licklider (*MIT, Massachusetts Institute of Technology)* en Agosto de 1962.
En estos escritos describía su idea de una *red galáctica* (*Galactic Network*), dónde todos los ordenadores estarían conectados entre sí, permitiendo el acceso rápido a cualquier tipo de información. Licklider fue el primer director del programa de desarrollo de ordenadores en el *DARPA (Defense Advanced Research Projects Agency)*, que se inició en Octubre de 1962. Desde este puesto impulsó su red galáctica e influenció a sus sucesores en el DARPA (Ivan Sutherland, Bob Taylor y al investigador del *MIT* Lawrence G. Roberts) de la importancia del proyecto.

En 1965 Lawrence G. Roberts en colaboración con Thomas Merill conecta un ordenador TX-2 (Massachusetts) con un ordenador Q-32 (California) mediante la línea telefónica, creando la primera red de gran alcance (*WAN*, Wide Area Network). En este experimento se demuestra la viabilidad del concepto de *conmutación de paquetes* (packet switching [Kle61] y [Kle65]).

En 1967 se publican los planes para el desarrollo de la conexión de todos los centros pertenecientes al DARPA mediante una red de ordenadores denominada ARPANET [Rob67]. A raíz de esta publicación, se descubre que paralelamente otros dos grupos ingleses (RAND [Bar64] y NPL) también estaban investigando el concepto de intercambio de paquetes como alternativa a la conmutación de circuitos. Finalmente se adopta de los trabajos del NPL la palabra *paquete* (packet) para el diseño de ARPANET.

En Septiembre de 1969 ARPANET empieza a caminar con la creación del primer nodo (Network Measurement Center, UCLA) y un mes después se añade el segundo nodo (Network Information Center, Stanford Research Institute). Después del intercambio del primer mensaje realizado por Leonard Kleinrock, se añade un tercer nodo en Noviembre

(Culler-Fried Interactive Mathematics, University of California Santa Barbara) y un cuarto en diciembre (Graphics, Utah).

También se introducen los RFC (Request For Comments), que son una serie de memorándums que sirven para comunicar informalmente diferentes ideas entre los grupos de investigación. Jon Postel es el encargado de su coordinación.

En 1970 y tras un crecimiento moderado de ARPANET, el Network Working Group (NWG) liderado por S. Crocker publica el primer protocolo de conexión entre parejas de ordenadores (host-to-host) para la red ARPANET, que pasará a denominarse NCP (Network Control Protocol).

En 1972 y coincidiendo la exposición internacional "International Computer Communication Conference" (ICCC), Kahn presenta el concepto de arquitectura abierta (Open Architecture) sobre el que se fundamentará ARPANET. Esta versatilidad de no depender de ningún ordenador o red en concreto, permite la interconexión de nodos heterogéneos en una red homogénea. Ray Tomlinson introduce la primera versión de un programa de correo (e-mail) que permitía leer y escribir mensajes. Se adopta del Tomlimson modelo 33 el signo @ (*at sign* en inglés, o *arroba* en castellano) para las direcciones de e-mail. Unos meses después, Roberts re-escribe el programa de correo añadiendo los servicios de re-envio de mensajes, la lectura selectiva de mensajes y el manejo de ficheros (FTP, File Transfer Protocol). Esta sencilla posibilidad de enviar y recibir mensajes se convierte en la estrella de ARPANET, aumentando su prestigio y facilitando el contacto entre los diferentes grupos conectados y el acceso a todos los RFC.

Conforme nuevos nodos se añadían a la nueva red ARPANET, se empiezan a descubrir las limitaciones del NCP:

- NCP no proporciona fiabilidad a las comunicaciones, fiándose de la propia ARPANET. No se proporciona ningún mecanismo de control sobre el número de paquetes enviados. Tampoco existe ningún tipo de control de errores entre los ordenadores que se están intercambiando paquetes de datos (end-to-end).

- NCP no proporciona ningún mecanismo de direccionamiento (para ordenadores y/o redes). Esto era debido a que inicialmente se supuso un numero ínfimo de nodos, y conforme fue aumentando de tamaño, las previsiones mas optimistas fueron desbordadas.

De esta manera, Kahn y Vincent Cerf deciden desarrollar una nueva versión del protocolo NCP que continue la filosofía de arquitectura abierta. Este protocolo se denominará Transmission Control Protocol / Internet Protocol (TCP/IP). Este nuevo diseño se basará en los siguientes principios:

1. Cada una de las redes conectadas debe ser independiente del resto, y no deberán realizarse cambios en estas para su conexión a ARPANET.

2. Si un paquete no alcanza su destino, deberá ser retransmitido por el origen.

3. Se utilizaran unas cajas negras para la interconexión de redes (mas tarde se les denominó *gateways* o *routers*) que **no** mantendrán información referente a cada una de las conexiones que se estén produciendo en cada momento, permitiendo una cierta tolerancia a fallos. Estas cajas negras tendrán la función de conducir los paquetes hacia los nodos de destino, lo que implica un direccionamiento dentro de la red.

4. Se deben permitir simultáneamente diferentes comunicaciones entre los ordenadores (*pipelining*) facilitando la interactividad. Esto posibilita la existencia de varias conexiones simultáneas en un mismo ordenador y obliga a la adopción de un sistema para diferenciarlas.

5. Es necesario un sistema de direccionamiento global para todos los nodos que forman parte de la red.

Después de algunas versiones de evaluación, se llegó a la conclusión de que el protocolo debería subdividirse en dos. Un protocolo simple denominado **IP** encargado de enviar paquetes individuales por la red hacia un nodo de destino, y otro mas complicado denominado **TCP** que se encargará de proveer un control de flujo de los

paquetes enviados, asegurando que lleguen a su destino de una forma correcta y ordenada. Para aquellas aplicaciones que no requieran un control tan estricto, se diseña el protocolo User Datagram Protocol (**UDP**), que al igual que TCP, utiliza los servicios del protocolo IP, pero sin dar fiabilidad (ver figura 2-1 y 2-2).

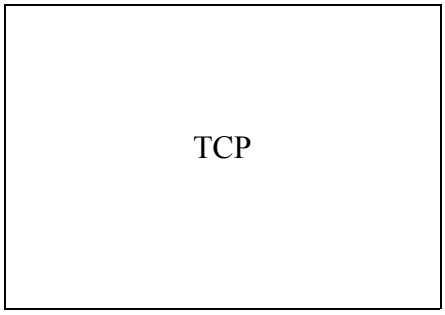


FIG. 2-1: Primer diseño del TCP/IP.

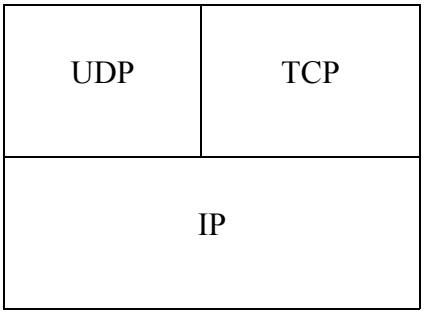


FIG. 2-2: Estructura jerárquica.

DARPA eligió tres grupos diferentes para que implementasen el nuevo protocolo. Estos grupos fueron las universidades de Standord y UCL, así como la empresa BBN. La definición del protocolo TCP (en la bibliografía consultada usualmente se habla simplemente de protocolo TCP, aunque implícitamente se está incluyendo el IP) como un protocolo abierto, permitió la interoperatividad de las diferentes implementaciones realizadas por los tres grupos de trabajo.

Con el auge de las redes locales (principalmente Ethernet, desarrollado por Bob Metcalfe en el XEROX PARC en 1973) y los ordenadores personales (Personal Computer de IBM, en la década de los 80) el número de nodos de ARPANET creció exponencialmente, lo que obligó a replantear el sistema de numeración de los nodos conectados en ARPANET. Fruto de este nuevo estudio, se desarrolló un sistema de direcciones de 32 bits y una subdivisión de estas en tres tipos (clases A, B y C).

En 1980 los protocolos TCP e IP fueron definitivamente adoptados por el departamento de defensa americano, lo que permitió su integración en ARPANET. El 1 de enero de 1983 se produjo la substitución del viejo protocolo NCP por los nuevos TCP/IP. A pesar del relativo gran tamaño de ARPANET, la substitución fue todo un éxito. Este gran

paso permitió a su vez la separación de ARPANET en dos grandes redes, **MILNET** (red militar) y **ARPANET** (red científica).

En 1985 Dennis Jennings perteneciente al NSF (National Science Foundation), adopta TCP/IP como el protocolo para la red NSFNET. De esta forma se integra al igual que otras redes en ARPANET.

A partir de 1988, los responsables de NSFNET, la espina dorsal (back bone) de ARPANET, empiezan a impulsar una política de limitación del uso de ARPANET al ámbito científico. Esto provoca una privatización controlada, ya que impulsó la aparición de las primeras redes privadas (PSI, UUNET, ANS CO+RE) para usos más lúdicos. Esto provoca la desaparición de ARPANET (1990) y el nacimiento de INTERNET.

En 1991 el CERN [WWW6] presenta un lenguaje de marcas (Tags) denominado HTML (Hyper Text Markup Languaje) que revoluciona INTERNET. Este lenguaje permite combinar de una forma fácil e interactiva textos e imágenes. Esta capacidad multimedia populariza el servicio Word Wide Web (WWW) que se basa el lenguaje HTML (ver figura 2.4). Philip Zimmerman presenta el Pretty Good Privacy (PGP), que añade extensiones de seguridad y privacidad al correo electrónico (e-mail).

En 1994 el NRCC (National Research Council Comitee) en colaboración con la NSF publica el informe "Towards a National Research Network". Este informe impulsó y sentó las bases para las futuras autopistas de la información. Finalmente, en 1995 culmina la política de privatización impulsada por NSF, lo que provoca la disolución de NSFNET en sub-redes locales.

La revolución impulsada por NSF hizo que en un plazo de 8 años y medio se pasara de 8 nodos conectados a 56Kbps a 21 nodos conectados a 45Mbps, favoreciendo el acercamiento de INTERNET a prácticamente todo el mundo y disparando su crecimiento (ver figuras 2-3 y 2-4).

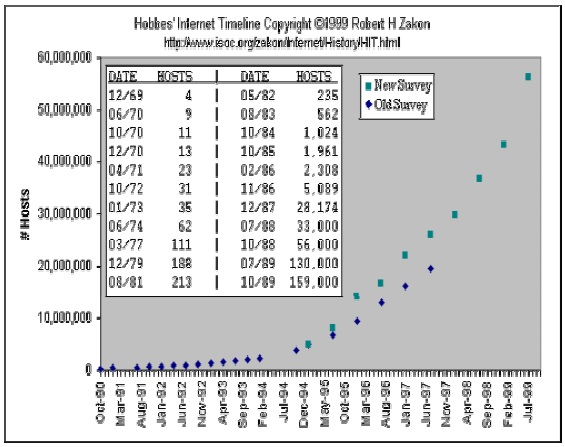


FIG. 2-3: Crecimiento de INTERNET.

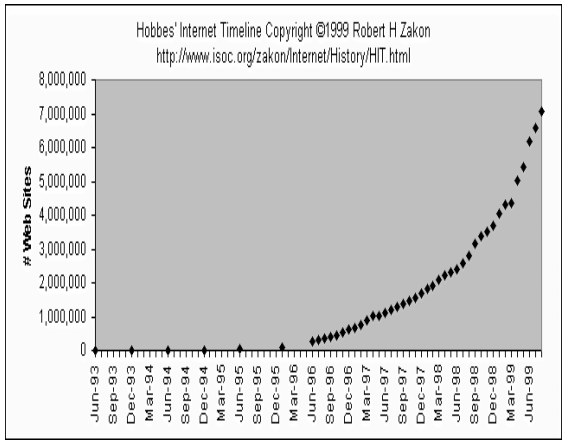


FIG. 2-4: Crecimiento de servicio WWW.

Pero ¿qué es INTERNET? Hay infinitas respuestas a esta pregunta, no obstante, nosotros señalamos la definición facilitada el 24 de Octubre de 1995 por el FNC (Federal Networking Council):

INTERNET hace referencia a un sistema global de información en que se está lógicamente conectado por un sistema global y único de direcciones basado en el Internet Protocol (IP) y que presenta las características siguientes:

1. *Es capaz de realizar comunicaciones usando el Transmission Control Protocol / Internet Protocol (TCP/IP) o cualquier extensión compatible con IP.*
2. *Y además proporciona, usa o hace accesible de forma pública o privada mediante servicios basados en un esquema de capas (layers) acceso a la información contenida en cualquiera de los nodos conectados.*

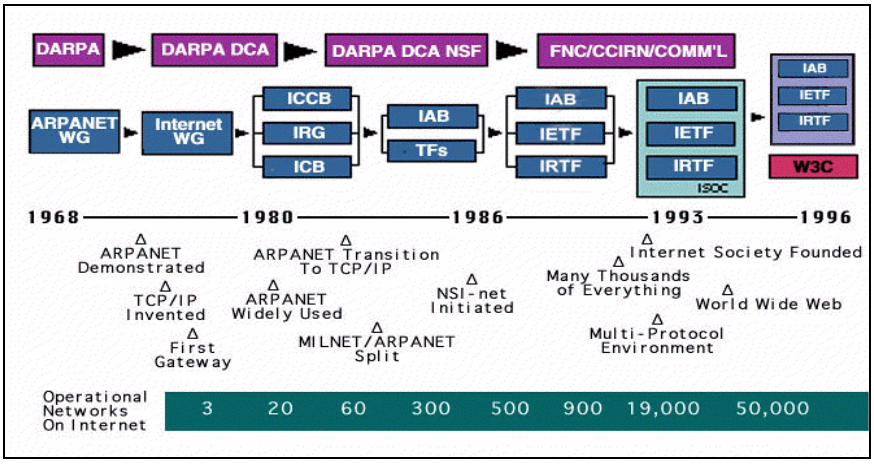


FIG. 2-5: Organismos oficiales que regulan INTERNET.

En la figura 2-6, podemos apreciar un resumen de la evolución histórica de INTERNET desde 1950 hasta el año 2000.

1950

1957 URSS lanza el satélite Sputnik. **1960**

1962 J.C.R. Licklider acuña el concepto de *Red Galáctica*

1966 Se presenta el primer plan para la creación de ARPANET.

1969 Creación de ARPANET con dos nodos iniciales. UCLA y SRI. Primer RFC **1970**

1970 Publicación del protocolo de ARPANET. Adopción del protocolo NCP.

1972 Mejora del e-mail. Elección del signo @. Publicación del protocolo TELNET [RFC318].

1973 El 75% del uso de ARPANET son e-mail's. Ethernet. Publicación del FTP [RFC454].

1974 Publicación de la primera versión del TCP/IP.

1978 División del protocolo TCP/IP en TCP e IP. **1980**

1980 El 27 de Octubre un virus produce una parada total de ARPANET.

1983 Substitución de NCP por TCP e IP. División en ARPANET y MILNET. Creación IAB.

1984 Introducción del DNS. Hay conectados más de 1.000 nodos.

1986 Creación IETF y IRTF. Primer acceso no gubernamental (Freenet).

1988 Creación IANA y CERT. Conexión a FIDONET. NSF impulsa la privatización.

1989 Se superan los 100.000 nodos conectados. **1990**

1990 Nace INTERNET substituyendo a ARPANET. Aparece el servicio de búsqueda *Archie*.

1991 El CERN presenta HTML, base del WWW. P. Zimmerman desarrolla el *PGP*.

1992 Creación de la ISOC. Ya hay más de 1.000.000 de nodos conectados a INTERNET.

1993 NSF crea INTERNIC para gestionar el registro de dominios.

1995 WWW se convierte en el servicio más utilizado. El Registro de un dominio cuesta 50$.

1998 Se presenta la privatización del DNS. Gran auge del comercio electrónico (*e-comerce*).

1999 Aparición del SETI@Home, búsqueda de vida extraterrestre utilizando INTERNET. **2000**

FIG. 2-6: Evolución histórica de INTERNET.

2.2 Funcionamiento de INTERNET: Los protocolos TCP/IP

Después de ver que es y cómo se organiza INTERNET, describiremos los protocolos que permiten su funcionamiento universal, independientemente de los ordenadores, sistemas operativos y/o redes que la conforman. A continuación citamos una posible definición de los protocolos TCP/IP extraída de [Ric98-1]:

> *"Las familias de protocolos TCP/IP permiten la comunicación entre diferentes tipos de ordenadores con independencia del fabricante, red a la que se encuentren conectados y sistema operativo utilizado."*

Las familias de protocolos que gobiernan INTERNET, se caracterizan por haber sido construidos siguiendo un esquema de *capas* (*layers*). Cada capa es la responsable de cada una de las diferentes facetas de la comunicación. De esta forma, se puede definir la familia de protocolos TCP/IP como una combinación de cuatro capas (ver figura 2-7). En este esquema, la capa superior accede únicamente a los servicios prestados por la capa situada justo en el nivel inferior a ella.

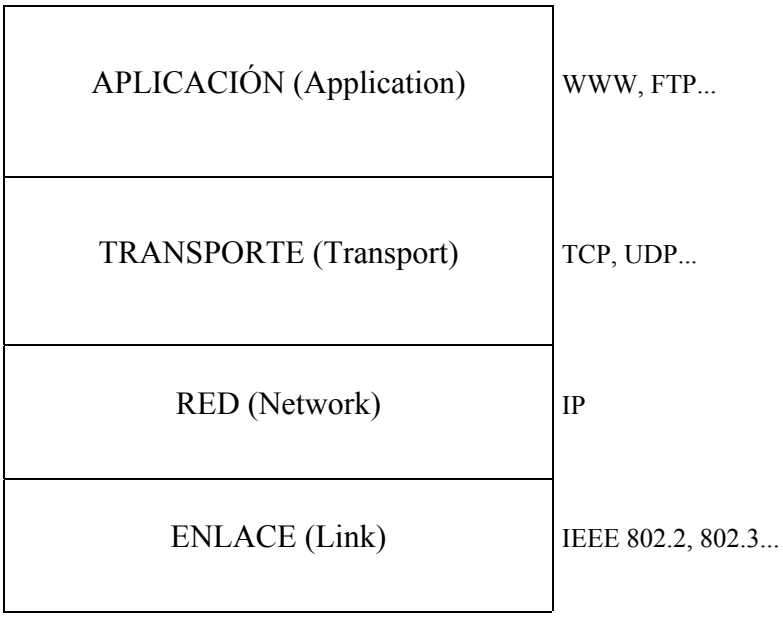


FIG. 2-7: Estructura de cuatro capas.

- La ***capa de enlace*** (link layer), también denominada la capa de datos (data layer) o capa de acceso a red (network interface layer), incluye los mecanismos que permiten al sistema operativo enviar y recibir información a través de la red a la que está conectado (Ethernet, RDSI...).

- La ***capa de red*** (network layer), también denominada capa de INTERNET (INTERNET layer), es la encargada de mover los paquetes a través de las diferentes redes para llegar a su destino. En esta capa encontramos los protocolos de mas bajo nivel, destacando el IP (INTERNET Protocol).

- La ***capa de transporte*** (transport layer), es la encargada de proporcionar un flujo de datos entre dos ordenadores. Este flujo de datos puede ser fiable (Transmission Control Protocol, TCP) o no fiable (User Datagram Protocol, UDP).

- La ***capa de aplicación*** (application layer), es la encargada de manejar los detalles particulares relativos a las diferentes aplicaciones (WWW, TELNET, FTP...).

Siguiendo el modelo de capas descrito anteriormente, vemos que la comunicación entre dos ordenadores no se produce directamente. De esta forma, cada capa añade una información de control específica denominada *cabecera* (header) a los datos recibidos y los pasa a la capa inferior. Este proceso se repite en cada capa, hasta llegar a la capa de enlace, dónde se envían los datos por la red. Análogamente, cuando se recibe una información, la capa receptora elimina la cabecera y pasa el resultado a la capa superior.

Este sistema permite una independencia entre las diferentes capas y obliga a que la comunicación entre dos ordenadores se realice mediante una comunicación entre las capas de los dos ordenadores (ver figura 2-8).

La comunicación en INTERNET se produce mediante el intercambio de paquetes de información entre los distintos ordenadores. Estos paquetes de información (también denominados datagramas) viajan por los diferentes ordenadores que están conectados a INTERNET hasta que alcanzan su objetivo o son descartados por algún motivo.

De esta forma, en la comunicación de dos ordenadores por INTERNET podemos diferenciar dos tipos de funciones que pueden desempeñar los ordenadores por los cuales se transmiten los paquetes de información:

1. Ordenador ***emisor/receptor*** (end-system o end-host). Aquí se englobaría el ordenador origen o destinatario de la comunicación.

2. Ordenador ***intermedio*** (intermediate-system, router o gateway). Serían todos los ordenadores por los que van pasando los datagramas o paquetes de información hasta el ordenador destino de la comunicación o hasta el origen (en el caso de una respuesta).

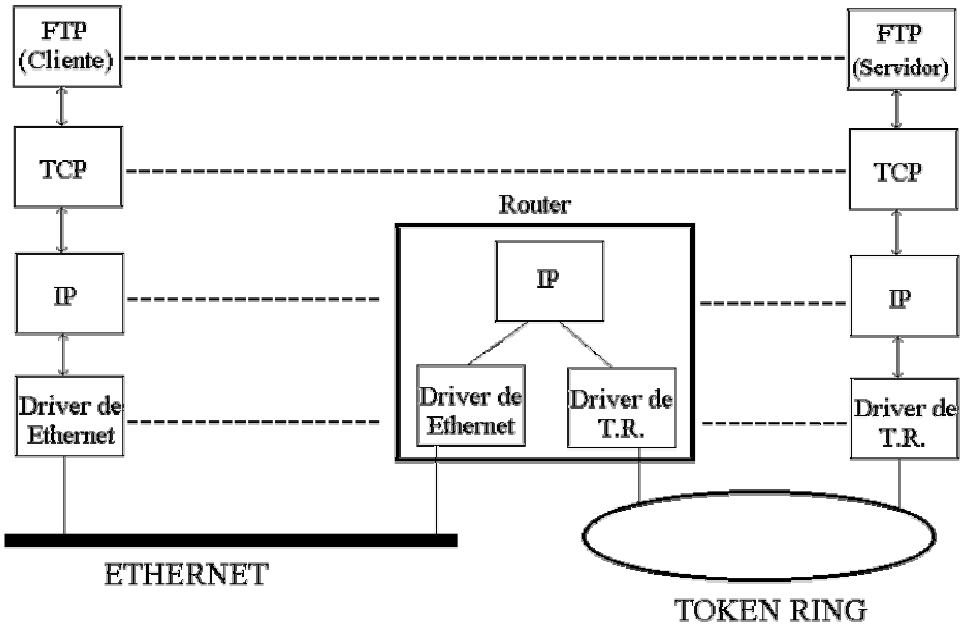


FIG. 2-8: Esquema de conexión de dos ordenadores en INTERNET.

Una vez vista cómo se efectúa una comunicación general entre dos ordenadores cualesquiera conectados a INTERNET, nos centraremos en cómo efectuar una comunicación entre dos ordenadores concretos.

De la misma manera que con el teléfono, para que se pueda establecer una comunicación entre dos ordenadores concretos conectados a una red dónde hay millones de ordenadores, es necesario un criterio que los diferencie. En la red telefónica

conmutada (RTC), el número de teléfono nos permite elegir un destino concreto entre los millones de teléfonos conectados. Análogamente, INTERNET tiene un sistema de numeración que permite diferenciar todos y cada uno de los ordenadores conectados (ver figuras 2-9 y 2-10).

En la versión 4 de los protocolos TCP/IP, estas direcciones han de cumplir dos requisitos básicos:

1. Deben ser únicas. No puede haber dos ordenadores con la misma dirección.

2. Las direcciones son números de 32 bits (4 bytes). Estas direcciones se representan mediante cuatro números decimales separados por un punto[2].

CLASE	**RANGO**		
A	0.0.0.0	Hasta	127.255.255.255
B	128.0.0.0	Hasta	191.255.255.255
C	192.0.0.0	Hasta	223.255.255.255
D	224.0.0.0	Hasta	239.255.255.255
E	240.0.0.0	Hasta	247.255.255.255

FIG. 2-9: Clases de direcciones IP en INTERNET

Clase A	0	Identificador de red (7 bits)							Número de ordenador (24 bits)																						
Clase B	1	0	Identificador de red (14 bits)														Número de ordenador (16 bits)														
Clase C	1	1	0	Identificador de red (21 bits)																	Número de ordenador (8 bits)										
Clase D	1	1	1	0	Identificador de red (28 bits)																										
Clase E	1	1	1	1	0	Reservado para futuro uso (27 bits)																									

FIG. 2-10: Subdivisión de los 32 bits para las clases A, B, C, D y E.

[2] Ej. 158.109.0.1

Este tipo de direccionamiento, nos permite una gran flexibilidad a la hora de definir redes que posteriormente conectaremos a INTERNET. Así, una clase A sería ideal para redes muy grandes, ya que permite 128 redes (2^7) de 16.777.216 (2^24) ordenadores cada una. Mientras que una clase B permite 16.384 (2^14) redes con 65.535 ordenadores, y una clase C permite 2.097.152 (2^21) redes de 256 ordenadores.

Las clases D (multicast) y E (reservada) se utilizan para diferentes posibilidades como la de tener ordenadores en redes diferentes y que se vieran como si estuvieran en la misma (ej. 2 ordenadores en la UAB, 1 en la UPC y 10 en la UB, y que todos recibieran la misma información. Como podría ser en una multi-conferencia). No obstante estas particularidades van mas allá de los propósitos de este trabajo, y se recomienda la lectura de [Ric98-1] para más información[3].

Una vez definido el direccionamiento de redes y ordenadores en INTERNET, mencionar la existencia de los DNS (Domain Name Server). Debido a que es más fácil de recordar un nombre (Centro de cálculo de la Universidad Autónoma de Barcelona, cc.uab.es) que una dirección numérica (158.109.0.4), se crearon los servidores de nombres (Domain Name Server, DNS), que son máquinas encargadas de transformar un nombre en su dirección correspondiente.

Al igual que las direcciones numéricas IP, los nombres tienen una jerarquía que permite la resolución de un nombre en su dirección numérica. La resolución del nombre siempre se realiza de derecha a izquierda. En nuestro caso (*cc.uab.es*), se buscaría primero *.es* (el último campo hace referencia al país: .es España, .us USA... o tipo de organización: .com Comercial, .gov Gubernamental, .mil Militar...). Después de conocer el país u organización, buscamos el segundo campo hacia la izquierda, *.uab* (este campo hace referencia al dominio dentro del país u organización, Universidad Autónoma de Barcelona en .es –España- en nuestro caso). Y finalmente *cc* (nombre del ordenador dentro del dominio .uab -Universidad Autónoma de Barcelona- del país/organización .es –España-).

[3] Destacar que como se usan 32 bits agrupados en 4 bytes (4 bytes x 8 bits/byte = 32 bits), los valores máximos son 255 (2^8 = 256 valores distintos. Del 0 al 255).

2.3 El protocolo IP versión 4

El protocolo IP (INTERNET Protocol) es la pieza fundamental en la que se sustenta el sistema TCP/IP y por tanto todo el funcionamiento de INTERNET. Su especificación está recogida en **[RFC791]**. La unidad de datos del protocolo IP es el *datagrama*, un esquema del cual puede verse en la figura 2-11.

El protocolo IP facilita un sistema **sin conexión** (connectionless) y **no fiable** (unreliable) de entrega de datagramas entre dos ordenadores cualesquiera conectados a INTERNET. IP da un servicio de entrega basado en el mejor intento (*best effort*).

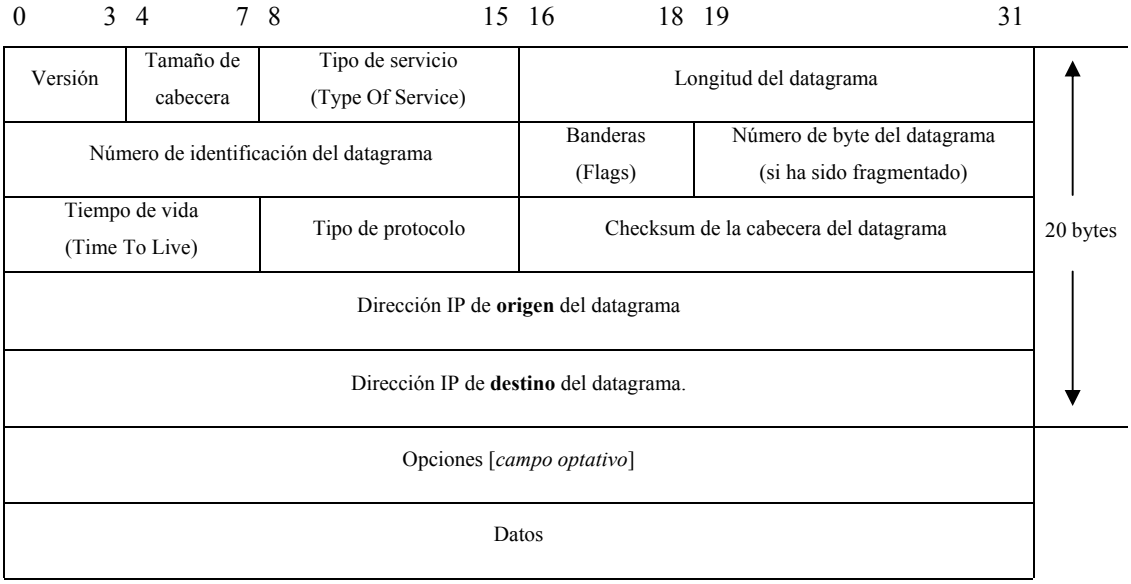


FIG. 2.11: Estructura de un datagrama IP v4.

Esto implica que cuando hay algún funcionamiento anómalo de INTERNET, como podría ser un *router* colapsado, se contempla un sistema muy simple de tratamiento de errores. Este mecanismo de control de errores viene regulado por el ICMP (INTERNET Control Message Protocol). En nuestro caso, el router colapsado descartaría el datagrama y enviaría un mensaje de error ICMP al ordenador de origen sin encargarse de la retransmisión del datagrama, lo que **no implica fiabilidad**. Además, no mantiene ningún tipo de información referente al estado de las conexiones. Cada datagrama es encaminado de forma independiente. Esto lo convierte en un **protocolo sin conexión**.

Debido a estas particulares características, puede pasar que se pierdan datagramas y/o que estos no lleguen en orden. De esta manera, cualquier fiabilidad que se necesite, deberá ser realizada por las capas superiores (TCP...).

En la figura 2-11 podemos ver cómo la estructura de un datagrama IP está estructurada en bloques de 32 bits (4 bytes). El datagrama IP se transmite enviando primero el bit 0, luego el bit 1, 2, 3... y así sucesivamente hasta finalizar el datagrama. Este orden se denomina **network byte order**. El orden es muy importante, puesto que los diferentes ordenadores tienen diferentes sistemas de almacenamiento de bits en memoria. El formato *little endian*, consite en almacenar los bits en orden inverso al network byte order (usando por ejemplo en los procesadores Intel), mientras que la otra posibilidad se denomina Big endian (usado por ejemplo en los procesadores Motorola).

La ***versión*** (4 bits), sirve para identificar a que versión específica (RFC) hace referencia el formato del datagrama. Esta información sólo es utilizada por los routers y capa IP de origen y final del datagrama. Esto permite la coexistencia de diferentes versiones del protocolo IP de una forma transparente al usuario. La versión actual es la 4 (conocida también cómo IPv4).

El ***tamaño de la cabecera*** (Header Length), son 4 bits (2^4 = 16 posiciones, 0..15) que indican el número de palabras de 32 bits que ocupa la cabecera. Estos 4 bits de tamaño máximo, nos limitan a un tamaño de cabecera máximo de 60 bytes (15 * 32 bits = 60 bytes). No obstante, el valor usual de este campo es 5 (5 * 32 bits = 20 bytes).

El ***campo del tipo de servicio*** (Type Of Service), se compone de 8 bits. Los primeros 3 bits tienen una función obsoleta y no se contemplan actualmente. Los 4 bits siguientes definen el tipo de servicio (ver figura 2-12). Y el último bit no se utiliza actualmente y debe tener valor 0. Solo 1 de los 4 bits del tipo de servicio puede estar activo a la vez.
El tipo de servicio determina la política a seguir en el envío del datagrama por INTERNET. Las opciones posibles son *minimizar el retraso* (minimize delay), *maximizar el rendimiento* (maximize throughput), *maximizar la fiabilidad del transporte* (maximize reliability) y *minimizar el coste económico* del transporte (minimize monetary cost).

Tipo de aplicación	Minimizar retraso	Maximizar rendimiento	Maximizar fiabilidad	Minimizar coste	Valor en hexadecimal
TELNET	1	0	0	0	0x10
FTP	0	1	0	0	0x08
SMTP	0	1	0	0	0x08
DNS (UDP)	1	0	0	0	0x10
DNS (TCP)	0	0	0	0	0x00
ICMP	0	0	0	0	0x00
BOOTP	0	0	0	0	0x00

FIG. 2-12: Valores típicos del tipo se servicio según la aplicación.

La ***longitud del datagrama*** (Total Length), es un número de 16 bits (2^16 = 65536, 0..65535) que indica la longitud total del datagrama. Este valor es muy importante, ya que nos permite saber que tamaño de memoria debemos reservar para la recepción del datagrama. Además, nos indica el número de bytes a leer, lo que nos permite un simple control de error. De esta forma, si el valor es incorrecto, el número de bytes leídos será como máximo de 65535, acotando el error. Además nos limita el número de bytes a enviar en un datagrama (Maximum Transfer Unit, MTU) a 65535 – 20 (tamaño típico de la cabecera) = 65515 bytes.

Si el tamaño del datagrama, es mayor que el tamaño máximo del paquete de red (Ej. Datagrama de 32000 bytes enviado sobre una ethernet, que tiene un tamaño máximo de paquete de 1500 bytes), se fragmenta en N trozos.

El ***número de identificación del datagrama*** (Identificartion Field), es un número de 16 bits que en caso de fragmentación de un datagrama nos indica su posición en el datagrama original. Esto nos permite recomponer el datagrama original en la máquina de destino. Este valor nos indica que un datagrama puede ser fragmentado en un máximo de 65535 fragmentos.

Las ***banderas*** (Flags) son 3 bits. El primero permiten señalar si el datagrama recibido es un fragmento de un datagrama mayor, bit M (More) activado. El segundo especifica si el datagrama no debe fragmentarse, bit DF (Don't fragment) activado. Y el tercero no se utiliza actualmente, asignándole el valor 0 [San99].

El ***número de byte en el datagrama*** (Fragmentation Offset), nos indica la posición en bytes que ocupan los datos en el datagrama original. Sólo tiene sentido si el datagrama forma parte de uno mayor que ha sido fragmentado. Este campo tiene un máximo de 13 bits (2^13 = 8192, como nos indica el desplazamiento en bytes 8192 * 8 bits = 65536). De esta forma, podemos reconstruir el datagrama original con los fragmentos.

El ***tiempo de vida*** (Time To Live), es un campo de 8 bits que indica el tiempo máximo que el datagrama será válido y podrá ser transmitido por la red. Esto permite un mecanismo de control para evitar datagramas que circulen eternamente por la red (por ejemplo en el caso de bucles). Este campo se inicializa en el ordenador de origen a un valor (máximo 2^8 = 256) y se va decrementando en una unidad cada vez que atraviesa un router. De esta forma, si se produce un bucle y/o no alcanza su destino en un máximo de 255 "saltos", es descartado. Entonces se envía un datagrama ICMP de error al ordenador de origen para avisar de su pérdida.

El ***tipo de protocolo*** (Protocol), es un valor que indica a que protocolo pertenece el datagrama (TCP, UDP, ICMP...). Es necesario debido a que todos los servicios de INTERNET utilizan IP como transporte, lo cual hace necesario un mecanismo de discriminación entre los diferentes protocolos.

El ***checksum de la cabecera del datagrama*** (Header Checksum), es una suma de comprobación que afecta sólo a la cabecera del datagrama IP. El resto de protocolos TCP, UDP, IGMP... tienen su propia cabecera y checksum. Su función es simplemente la de un mecanismo de control de errores. De esta forma, si se encuentra un error en el checksum de un datagrama IP, este es simplemente descartado y no se genera ningún mensaje de error. Esto implica que es deber de las capas superiores el control del flujo de los datagramas. Asegurándose que estos lleguen correctamente al destino, ya sea utilizando un protocolo fiable (TCP) o implementando internamente algún tipo de control.

Tanto la ***dirección IP de origen como la de destino*** (IP address), están formadas por dos números de 32 bits. Estas direcciones se corresponden a una distribución según la figura 2-10.

2.4 El protocolo UDP

El protocolo UDP (User Datagram Protocol) se podría definir como un ***protocolo simple y orientado a datagrama*** [Ric98-1]. Su definición se recoge en **[RFC7680]** publicado por Postel en 1980.

De esta forma, cada envío de datos se corresponde con un único envío de un datagrama independiente del resto de datagramas y de la misma comunicación (ver figura 2-13).

Esta característica lo diferencia de forma clara de otros protocolos orientados a flujo de datos (Stream Oriented) como el TCP. De esta forma, siguiendo los preceptos de encaminamiento de datagramas por INTERNET, la entrega al destino no está asegurada por el propio protocolo. Es mas, ni tan siquiera se asegura que los datagramas lleguen en el orden en el que fueron enviados.

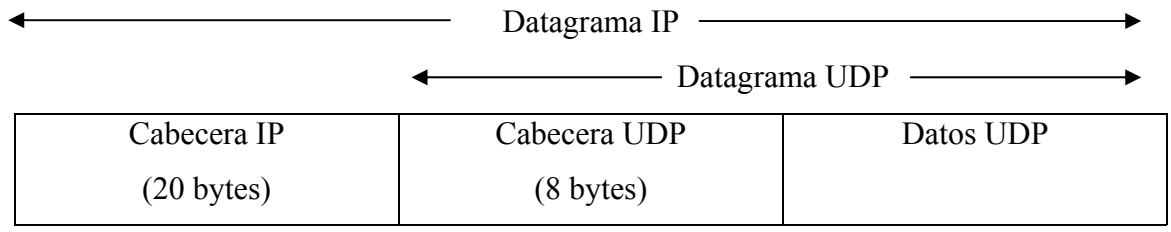


FIG. 2-13: Estructura de un datagrama UDP.

Como UDP no proporciona ningún tipo de fiabilidad, en caso de ser necesaria, debe ser proporcionada por la aplicación que hace uso del protocolo. O bien utilizar algún otro protocolo que si la proporcione, como TCP.

Debido a que UDP utiliza IP para su transporte por INTERNET, en caso de ser necesario (por diferentes tamaños de MTU, por ejemplo), este se fragmentará y ninguno de estos fragmentos (al igual que el datagrama original) proporcionara ningún tipo de seguridad o fiabilidad en la entrega.

Debido a la sencillez de este protocolo, el diseño de su cabecera (figura 2-14), es mucho más simple que el de IP.

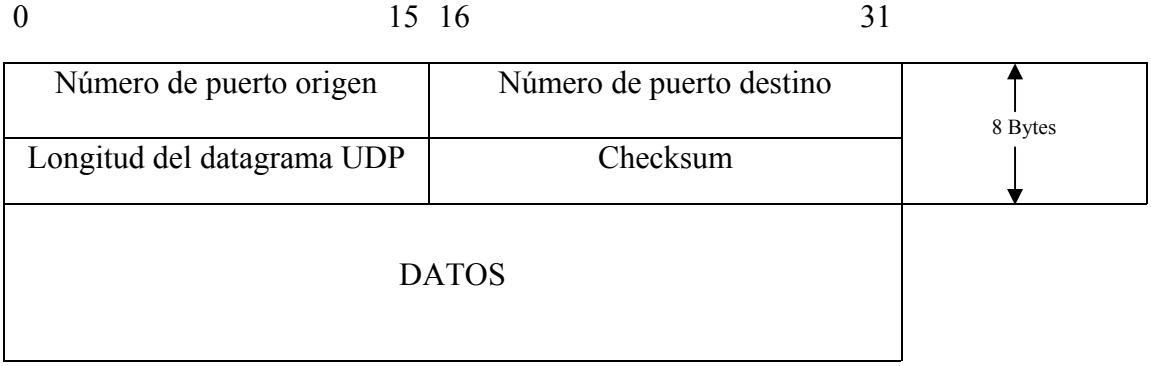


FIG. 2-14: Cabecera de un datagrama UDP.

El ***número de puerto*** (Port) se utiliza en la comunicación entre dos ordenadores para diferenciar las diferentes conexiones existentes. Si tenemos varias comunicaciones desde nuestro ordenador (por ejemplo un TELNET al *cc.uab.es* mirando el correo y un FTP a *blues.uab.es* bajando un fichero), al recibir un datagrama IP debemos saber a cual de las conexiones pertenece. Asignando un número de puerto a la comunicación podemos saber a que conexión pertenece. Al ser un número de 16 bits, podemos deducir que el número máximo de conexiones que un ordenador puede tener simultáneamente en uso es de 65535 (2^16).

La ***longitud del datagrama*** (UDP Length) hace referencia al tamaño del datagrama en bytes, y engloba la cabecera (8 bytes) mas los datos que transporta. El mínimo valor de longitud es 8 bytes (por lo tanto, el protocolo permite enviar un datagrama UDP con 0 bytes). Este campo es redundante, ya que utiliza IP para su transporte, y éste ya incorpora un campo para la longitud de los datos (ver figura 2-11) que sería la longitud del datagrama IP menos el tamaño de la cabecera.

El campo de ***checksum*** al igual que en IP, sirve como método de control de los datos, verificando que no han sido alterados. Este checksum cubre tanto la cabecera UDP como los datos enviados. Es necesario debido a que el checksum del protocolo IP tan sólo cubre la cabecera IP y no los datos que transporta. Si se detecta un error en el checksum, el datagrama es descartado sin ningún tipo de aviso (ver checksum en IP).

2.5 El protocolo TCP

El protocolo TCP (Transmission Control Protocol) se podría definir como un protocolo ***orientado a conexión***, ***fiable*** y ***orientado a un flujo de bytes*** [Ric98-1]. Su definición se recoge en **[RFC793]** publicado por Postel en 1981.

Aunque el protocolo TCP al igual que UDP utiliza los servicios de IP para su transporte por INTERNET (ver figura 2-15), es un protocolo **orientado a conexión**. Esto significa que las dos aplicaciones envueltas en la comunicación (usualmente un cliente y un servidor), deben establecer previamente una comunicación antes de poder intercambiar datos.

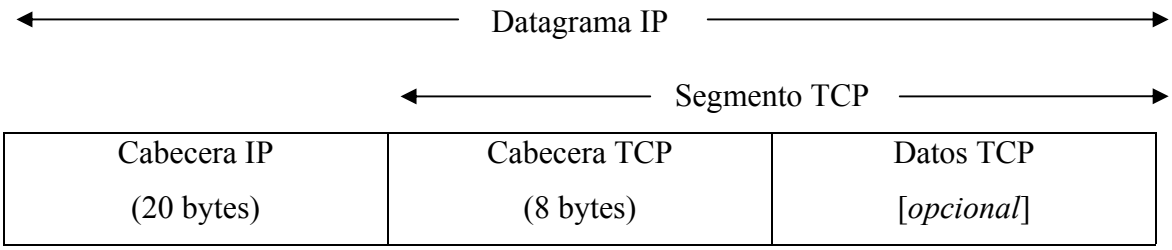


FIG. 2-15: Estructura de un segmento TCP.

TCP es también un protocolo **fiable**. La fiabilidad proporcionada por este protocolo viene dada principalmente por los siguientes aspectos:

1. Los datos a enviar son reagrupados por el protocolo en porciones (denominadas *segmentos* [Ric98-1]). El tamaño de estos segmentos lo asigna el propio protocolo. Esto lo diferencia claramente de UDP, dónde cada porción de datos generada corresponde a un datagrama.

2. Cuando en una conexión TCP se recibe un segmento completo, el receptor envía una respuesta de confirmación (Acknowledge) al emisor confirmando el número de bytes correctos recibidos. De esta forma, el emisor da por correctos los bytes enviados y puede seguir enviando nuevos bytes.

3. Cuando se envía un segmento se inicializa un timer. De esta forma, si en un determinado plazo de tiempo no se recibe una confirmación (Acknowledge) de los datos enviados, se retransmiten.

4. TCP incorpora un checksum para comprobar la validez de los datos recibidos. Si se recibe un segmento erróneo (fallo de checksum por ejemplo), no se envía una confirmación. De esta forma, el emisor retransmite los datos (bytes) otra vez.

5. Como IP no garantiza el orden de llegada de los datagramas, el protocolo TCP utiliza unos números de secuencia para asegurar la recepción en orden, evitando cambios de orden y/o duplicidades de los bytes recibidos.

6. TCP es un protocolo que implementa un control de flujo de datos. De esta forma, en el envío de datos se puede ajustar la cantidad de datos enviada en cada segmento, evitando colapsar al receptor. Este colapso sería posible si el emisor enviara datos sin esperar la confirmación de los bytes ya enviados.

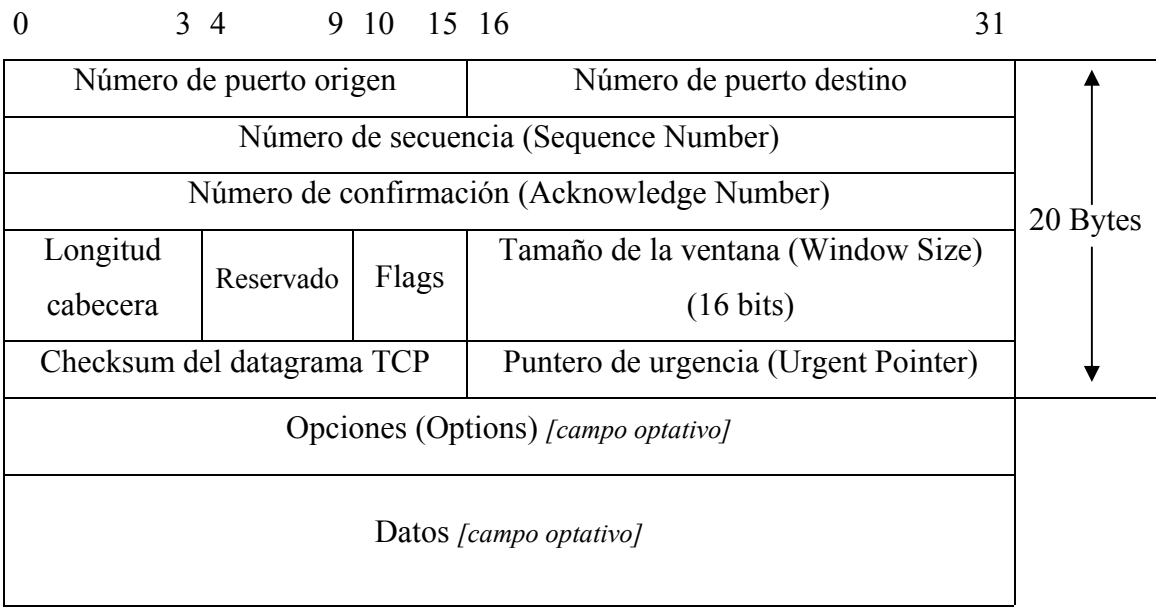


FIG. 2-16: Cabecera de un segmento TCP.

La cabecera del segmento TCP (figura 2-16), es bastante más compleja que la de UDP. Esto es debido a que la comunicación es más elaborada, ya que debe proporcionar fiabilidad. Esto implica una serie de información adicional que debe mantenerse para poder conocer el estado de la comunicación en cualquier momento.

El número de ***puerto origen*** y número de ***puerto destino***, sirven para diferenciar una comunicación en un ordenador de las demás. Cumple la misma función que en el datagrama UDP (ver punto 2.4). La tupla formada por la dirección IP y el número de puerto se denomina *socket*. Este término se utilizó luego en la especificación de la interficie de programación de Berkeley. Análogamente, la agrupación de las dos tuplas que definen una conexión entre dos ordenadores se denomina *socket pair*.

El ***número de secuencia*** (Sequence Number), identifica el byte concreto del flujo de datos que actualmente se envía del emisor al receptor. De esta forma, TCP numera los bytes de la comunicación de una forma consecutiva a partir del número de secuencia inicial. Cuando se establece una comunicación, emisor y receptor eligen un número de secuencia común. Esto nos permite mecanismos de control, como asegurar que los datos lleguen en el orden adecuado. Es un número de 32 bits, con lo que podemos enviar 2^32-1 bytes antes de que cicle.

El ***número de confirmación*** (Acknowledge Number), es el número de secuencia más uno. De este modo se especifica al emisor que los datos enviados hasta este número de secuencia menos uno son correctos. De aquí la importancia de la elección al principio de la comunicación de un número de secuencia común.

La ***longitud de la cabecera*** (header Length), especifica en palabras de 32 bits (4 bytes) el tamaño de la cabecera del segmento TCP incluyendo las posibles opciones. De esta forma el tamaño máximo es 15 * 4 = 60 bytes. No obstante, lo usual es tener un tamaño de 20 bytes (cabecera dónde no se incluyen opciones).

Las ***banderas*** (Flags), son las encargadas de especificar los diferentes estados de la comunicación. Así mismo, también validan los valores de los distintos campos de la cabecera de control. Pueden haber simultáneamente varios flags activados. En la figura 2-17 podemos ver los flags existentes y su significado.

FLAG	**Significado**
URG	Si es válido el puntero de urgencia.
ACK	El valor situado en el campo de confirmación (acknowledge) es válido.
PSH	El receptor debe pasar los datos a la aplicación o antes posible.
RST	RESET de la conexión
SYN	Inicio de comunicación. Búsqueda de un número de secuencia común.
FIN	El emisor finaliza el envío de datos.

FIG. 2-17: Flags del segmento TCP.

El ***tamaño de la ventana*** (Window Size), es el número de bytes desde el número especificado en el campo de confirmación, que el receptor está dispuesto a aceptar. El tamaño máximo es de (2^16) 65535 bytes. De esta forma, el protocolo TCP permite la regulación del flujo de datos entre el emisor y el receptor.

El ***checksum del segmento*** TCP, al igual que el del UDP o IP, tiene la función de controlar los posibles errores que se produzcan en la transmisión. Este checksum engloba la cabecera TCP y los datos. En caso de error, el datagrama/segmento queda descartado. El propio protocolo es el encargado de asegurar la retransmisión de los segmentos erróneos y/o perdidos.

El ***puntero de urgencia*** (Urgent Poiner), es válido sólo si el flag de **URG** se encuentra activado. Consiste en un valor positivo que se debe sumar al número de secuencia. Especificando una posición adelantada dónde podemos enviar datos urgentes.

Las ***opciones*** (Options), nos permiten especificar de forma opcional características extras a la comunicación. Un ejemplo de las opciones es el MSS (Maximum Segment Size), que especifica el tamaño máximo de datos que el emisor desea recibir [Ric98-1]. Esta opción se indica al inicio de la comunicación (flag SYN activado).

Los ***datos*** (Data) son opcionales. Esto significa que podemos enviar simplemente cabeceras TCP con diferentes opciones. Esta característica se utiliza por ejemplo al iniciar la comunicación o en el envío de confirmaciones. De esta manera, minimizamos el *overhead* ya que tan sólo enviamos/recibimos lo necesario para establecer o confirmar la comunicación.

2.5.1 Establecimiento de conexión TCP

TCP es un protocolo orientado a conexión. Esto implica que se ha de realizar un paso previo antes de poder intercambiar datos. Este paso es el de establecimiento de conexión.

Este paso es fundamental para poder garantizar la fiabilidad del protocolo. Es en este paso previo dónde se obtienen los números de secuencia que permitirán gestionar cualquier intercambio entre lo dos extremos de la comunicación. El método escogido para establecer la conexión se denomina **protocolo de 3 pasos** (*three way handshake,* ver figura 2-18).

En este esquema, podemos diferenciar un cliente activo que inicia la conexión (active open) y un servidor pasivo que tan sólo se limita a contestar (passive open) a las peticiones de conexión.

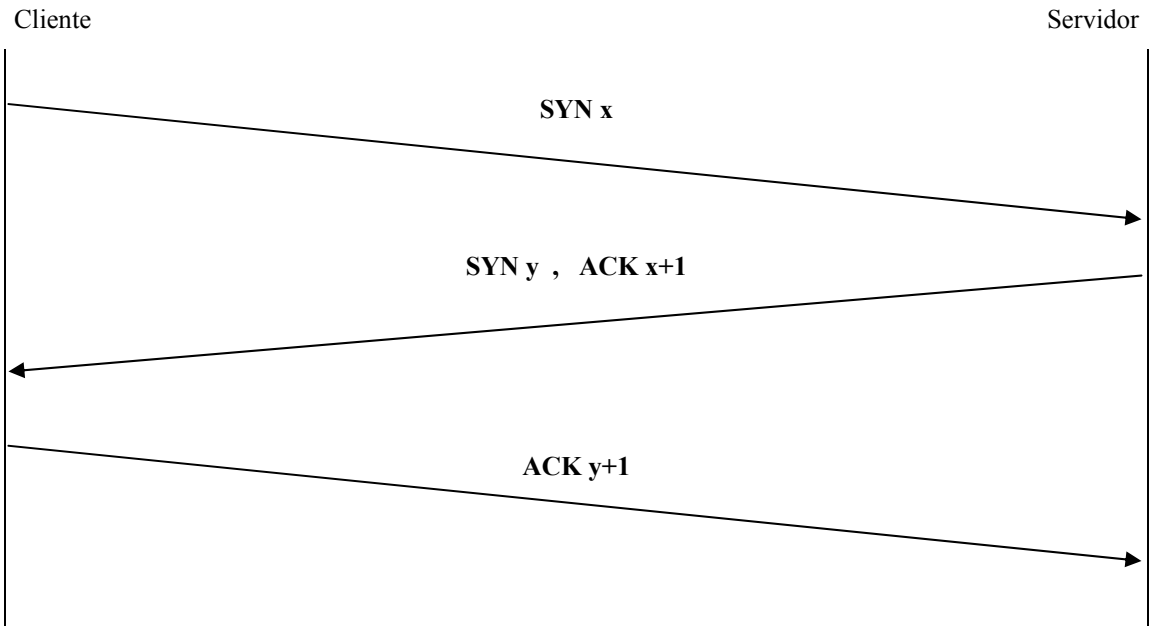


FIG. 2-18: Establecimiento de una conexión TCP utilizando el *three way handshake*.

1. El cliente (ordenador que desea iniciar la comunicación) selecciona un número aleatorio de secuencia (***x*** en la figura 2-18). A continuación activa el flag de SYN en el campo de opciones. Finalmente envía un segmento TCP al ordenador destino.

2. El servidor (ordenador con el que se quiere establecer una comunicación) recibe la petición y almacena el número de secuencia ***x***. Elige un número aleatorio (***y*** en la figura 2-18) que utilizará como número de secuencia. Activa los flags SYN y ACK. Finalmente envía un segmento con el número de secuencia elegido y con una confirmación del valor recibido mas uno (***ACK x+1*** en la figura 2-18).

3. El cliente almacena el número de secuencia (***y*** en la figura 2-18). Activa el flag de ACK. Y finalmente envía una confirmación del número recibido mas uno ***(ACK y+1*** en la figura 2-18).

En el caso de que este tercer paso no se realice, después de un cierto tiempo (entre 70 y 130 segundos, dependiendo del sistema operativo) la conexión se liberará. Esto es así porque cuando se inicia una conexión también se inicializa un timer.

Indicar que un ataque de *denegación de servicio* (Deny Of Service, DOS) muy famoso en INTERNET (denominado SYN FLOODING) consistía en no finalizar el establecimiento de las conexiones (no realizar el tercer paso). De esta forma, si se repetía continuamente se podía incluso llegar a bloquear las comunicaciones del ordenador al saturar los recursos de comunicaciones (para más información [San99]).

2.5.2 Finalización de una conexión TCP

Tal y como hemos visto en el punto anterior, se necesitan tres acciones para iniciar una conexión TCP. Para finalizarla se necesitan cuatro. La necesidad de estas cuatro acciones para finalizar la conexión, es debido a que la comunicación es *full duplex* (los datos pueden ser enviados y/o recibidos independientemente y en cualquier dirección de la comunicación). Esto obliga a que cada sentido de la comunicación deba ser finalizado independientemente (ver figura2-19).

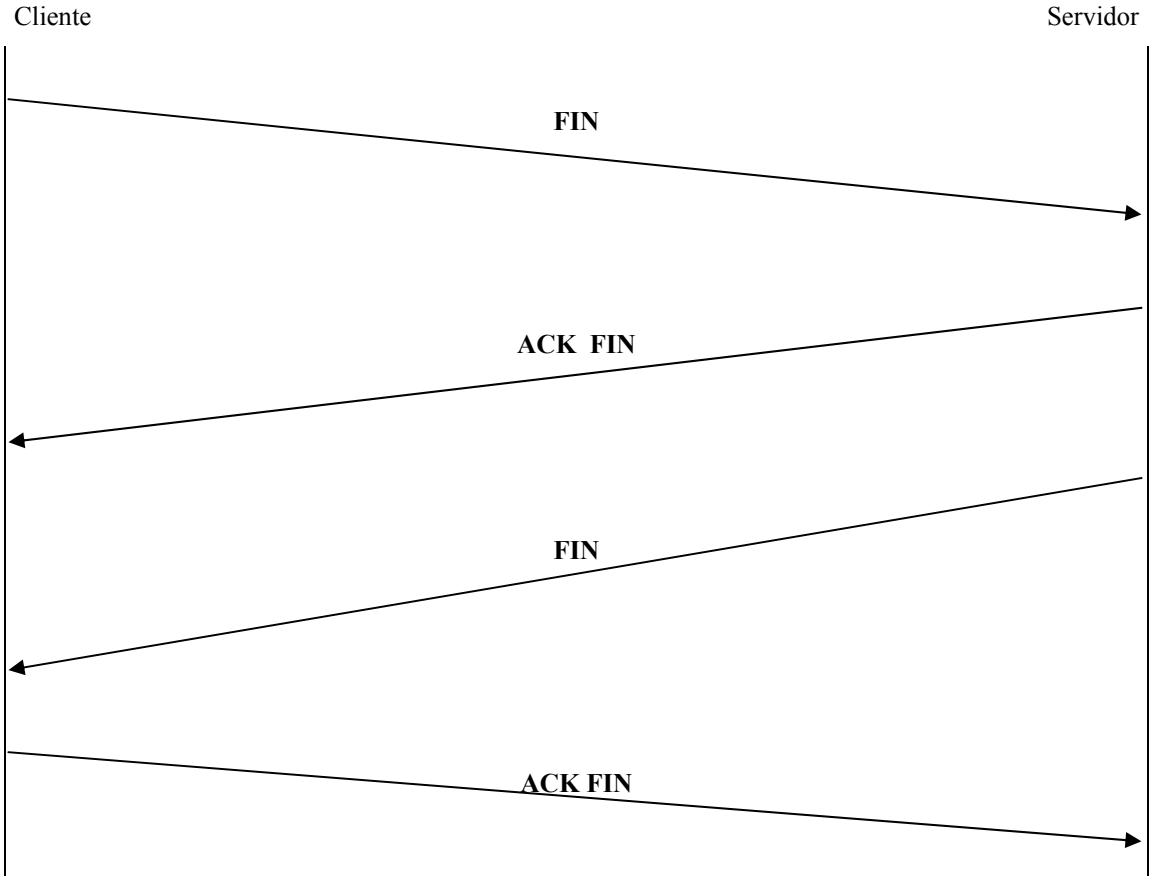


FIG. 2-19: Finalización de una conexión TCP.

Debido a la posibilidad de que cualquiera de los dos extremos implicados en la comunicación pueda enviar y/o recibir datos, tenemos la posibilidad de que cualquiera de los dos extremos finalice la comunicación (enviando una señal ***FIN***) hacia su sentido una vez finalizado el proceso de enviar/recibir datos. Esto nos obliga a realizar dos medias finalizaciones (*half-close*) de conexión, una hacia cada sentido.

TCP proporciona la capacidad de que cualquiera de los dos extremos de la conexión pueda finalizar su salida (*output*) de datos permitiéndole todavía recibir datos del otro extremo. Esta capacidad se denomina ***haf-close***. En el caso de finalizar sólo un sentido de la comunicación (de cliente a servidor por ejemplo). El cliente puede seguir recibiendo datos del servidor. Enviando únicamente reconocimiento de datos (acknowledge, ***ACK***) al servidor. De esta forma cuando el servidor envíe una señal de ***FIN*** la conexión TCP finalizará totalmente.

El envío se una señal *FIN* tan solo indica que no se producirá mas intercambio de datos en ese sentido. De esta forma es posible que en una conexión TCP se continúen enviando datos después de recibir una señal *FIN*. Esta posibilidad es muy poco aprovechada en las aplicaciones que utilizan TCP actualmente.

En un proceso de *half-close* podemos diferenciar de forma clara un extremo activo (*active close*) y un extremo pasivo (*passive close*). El extremo activo es el que envía la señal de FIN para finalizar ese sentido de la comunicación. Mientras que el extremo pasivo se limita a aceptar la petición y enviar un reconocimiento (*acknowledge*) de final.

De esta forma tenemos que para finalizar una conexión TCP debemos finalizarla en cada uno de los dos sentidos. Esto obliga a que alternativamente el cliente y el servidor adopten los papeles activo y pasivo en un proceso que consta de 4 fases:

1. (El cliente adopta el papel activo) El cliente decide finalizar la comunicación en su sentido. Enviando al servidor una señal de finalización (***FIN)*** con un número de secuencia.

2. (El servidor adopta el papel pasivo) El servidor recibe esta señal y responde con un reconocimiento (***acknowledge, ACK***) de señal. Enviando el número de secuencia recibido mas uno. Cabe señalar que la finalización (FIN) al igual que la señal de petición de conexión (***SYN***) consume un número de secuencia. Finalización de la primera *half-close*.

3. (El servidor adopta un papel activo) El servidor decide finalizar la conexión en su sentido y envía una señal de finalización (***FIN***) de conexión al cliente.

4. (El cliente adopta un papel pasivo) El cliente acepta la petición de finalizar la conexión respondiendo con un ***ACK*** y enviando el número de secuencia recibido mas uno. Finalización de la segunda *half-close*. Finalización de la conexión TCP.

2.6 Resumen

En este segundo capítulo se ha realizado una breve explicación de la historia de INTERNET (sus iniciales expectativas y su vertiginoso crecimiento en la década de los 90) y se han repasado los diferentes protocolos que la gobiernan actualmente:

IP (INTERNET Protocol), que es un protocolo **no fiable** y **no orientado a conexión** que se encarga del transporte de los datos por la red.

UDP (User Datagram Protocol), protocolo **simple** y **orientado a datagrama** que se encarga de enviar datagramas usando los servicios del protocolo IP.

TCP (Transmission Control Protocol) que es un protocolo **fiable**, **orientado a conexión** y **orientado a byte**. Utilizando los servicios del protocolo IP, este protocolo establece una conexión previa con el destino (mediante el *three way handshake*) que le permite regular el flujo de bytes enviados y su correcta recepción, retransmitiendo los bytes recibidos incorrectamente o perdidos. Finalmente realiza una fase de desconexión del receptor (mediante el sistema de *half close*) antes de dar por concluida la comunicación.

CAPITULO 3

Extensiones de seguridad sobre los protocolos TCP e IP versión 4

3.1 Necesidad de la seguridad en INTERNET

Definiremos *comunicación* como un intercambio de *información*. Así mismo definiremos una red (Network) como un mecanismo físico que permite la comunicación entre dos o más ordenadores separados entre sí mediante el uso de un protocolo común.

Desde el mismo momento en que nos conectamos a una red (ya sea una red local en una empresa o la propia INTERNET), no dejamos de enviar y recibir información. Esta información circula por la red de forma que el destinatario pueda recibirla y enviar una respuesta en caso necesario. Tenemos pues que cuando dos o más ordenadores están conectados a una misma red, existe la posibilidad de estos ordenadores puedan acceder

a esta información, e incluso a los otros ordenadores conectados. Esta característica deseable en la mayoría de ocasiones (cualquier biblioteca estaría encantada de que todas las personas pudieran acceder a la información que contienen) puede volverse en nuestra contra fácilmente (ningún banco estaría encantado de que cualquiera pudiera ver todos los extractos de cuentas corrientes).

Desde el mismo momento en que interesa que una información no sea pública (sea cual sea el motivo) necesitamos controlar el acceso a toda información para decidir si debe ser pública o debemos protegerla mediante sistemas de seguridad. La necesidad de seguridad siempre viene precedida de una amenaza (justificada o no). En nuestro contexto utilizaremos el término amenaza tal y como lo describe G. A. Marañón del CSIC:

"*Se entiende por amenaza una condición del entorno del sistema de información (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad (confidencialidad, integridad, disponibilidad o uso legítimo).*"

En nuestro contexto (una red dónde millones de ordenadores intercambian innumerable cantidad de información mediante el envío y recepción de datagrama según el conjunto de protocolos TCP/IP) podemos distinguir tres cualidades básicas deseables en toda comunicación:

1. **Secreta**. No solo respecto al contenido de la información enviada/recibida, sino también al origen y destino de la información.

2. **Auténtica**. Los datos no han sido modificados, provienen del origen legítimo (no hay suplantación de un usuario por otro y no pueden ser repudiados) y son ciertos en el momento de su recepción (evitar reutilización de la información).

3. **Accesible**. Tanto el emisor como el receptor han de poder intercambiar información cuando sea necesario.

Las amenazas a su vez pueden ser clasificadas principalmente en cuatro grupos distintos:

I. **Interrupción**. El acceso a un recurso/comunicación se ve interrumpido ya sea físicamente (destrucción de la red...) o lógicamente (se modifica la localización, los derechos de acceso...).

II. **Intercepción**. Alguien no autorizado consigue tener acceso al recurso/comunicación (pinchar la línea de red, sniffing...).

III. **Modificación**. Obtención no sólo de acceso no autorizado al recurso/comunicación, sino también de la capacidad de modificarlo (modificación de los datos enviados/recibidos entre dos ordenadores...).

IV. **Fabricación**. Además de conseguir acceso al recurso/comunicación, se tiene la posibilidad de insertar información falsa.

Vistos los diferentes tipos de amenazas que pueden existir en una comunicación, podemos clasificar los posibles ataques en pasivos y activos.

En los **ataques de tipo pasivo** el atacante no altera la comunicación, tan sólo tiene acceso a ella. De esta forma puede saber que información circula por el canal, a que horas, la frecuencia y entre que personas. Este tipo de ataque es muy difícil de detectar ya que no aparece ningún signo que nos pueda advertir de que estamos siendo atacados.

Por el contrario, en los **ataques activos** el atacante modifica el flujo de datos transmitidos o incluso crea uno falso, permitiendo incluso la suplantación de un usuario legítimo. Esto tipo de ataques es mucho más grave ya que además de conseguir interceptar la comunicación, puede modificar su contenido falseándola, lo que implica que en caso de usar algún sistema de seguridad este ha sido violado y se ha descubierto su clave de acceso o una de ellas (si es que hay) y el método utilizado para cifrar y descifrar la información. Estos ataques suelen acabar detectándose tras un cierto tiempo (por ejemplo un alumno que cambia las notas y un día un profesor que las consulta no lo tiene claro y mira el examen o una persona que cambia su saldo del banco, al final

cuando se produce una auditoria del banco los descuadres delatan el ataque) su problema principal es si se detectan demasiado tarde.

3.2 Seguridad en INTERNET bajo IP versión 4

Después de analizar brevemente las posibilidades de que una comunicación sea atacada, podemos fácilmente concluir que la seguridad debe ser un pilar básico en cualquier tipo de intercambio de información, y por tanto la comunicación a través de INTERNET no debería ser una excepción.

Desde los inicios de INTERNET y hasta finales de los años 90, casi toda la información que circulaba por INTERNET no utilizaba ningún tipo de cifrado, atravesando todas las máquinas que encontraba hasta en destinatario tal y cómo había sido enviada (*plain text*). Esto se debía principalmente a que en un principio el carácter divulgativo y científico de ARPANET lo hacía sólo accesible a ciertas universidades y laboratorios de investigación, motivo que propició que en las definiciones de los diferentes protocolos (TCP, UDP, IP...) de esos años, no se contemplase ninguna posibilidad de seguridad, ni tan sólo en los protocolos de nivel superior (como ejemplo baste decir que cuando realizamos un TELNET a una máquina remota, nuestro password circula tal cual, sin ningún tipo de cifrado o sistema de seguridad).

Posteriormente su expansión hacia el carácter lúdico y el público en general, hizo que el número de usuarios (así como sus fines) creciera exponencialmente. La proliferación de gestas realizadas por desaprensivos que aprovechaban las debilidades de unos protocolos diseñados sin ninguna seguridad, empezó a inundar los medios informativos, haciendo que la opinión pública (en algunos países más que otros) exigiera un fin de estas actuaciones. Esto obligó a los diseñadores de protocolos a añadir algunos parches y extensiones de emergencia a unos protocolos que no habían sido diseñados para ello.

De forma paralela el comercio electrónico (*e-comerce*) se empezaba a desarrollar, y diferentes soluciones fueron adoptadas por las empresas implicadas. Todas estas soluciones siempre se englobaron en las capas superiores de los protocolos (TCP).

3.2.1 Sistemas de criptografía

Definiremos una **clave** (*Key*) como la llave que permite cifrar o descifrar la información recibida de forma correcta. Los sistemas de criptografía actualmente se pueden clasificar en dos grandes grupos dependiendo de cómo distribuyan sus claves:

1. Sistema de **clave privada o simétricos** (ver figura 3-1). Este tipo de sistema se caracteriza por la existencia de una única clave que permite cifrar y descifrar los mensajes. Esto implica que tanto el emisor como el receptor comparten la misma clave. El principal inconveniente radica en cómo dar a conocer la clave privada únicamente al receptor. Este sistema fue el único utilizado hasta 1976.

2. Sistema de **clave pública o asimétrica** (ver figura 3-2). Este sistema desarrollado por Whitfield Diffie y Martin Hellman (ver [DH76] y [Rif95]) en 1976 se basa en que tanto el emisor como el receptor disponen de dos claves (una pública y otra privada). Estas dos claves están relacionadas entre sí matemáticamente, pero la relación entre ellas no es trivial, con lo que el conocimiento de la clave pública e incluso de texto en claro y texto cifrado no compromete la clave privada.

   Se dispone de una clave pública que se hará conocer a todos los posibles emisores (por e-mail o por autoridades de certificación) para que puedan cifrar los mensajes a enviar. La clave privada es guardada en secreto puesto que es la que permitirá descifrar el mensaje cifrado con la clave pública. El mayor inconveniente de este sistema es autenticar la procedencia de los datos, ya que como cualquiera tiene acceso a la clave pública, puede cifrar información y enviarla con otro nombre. Un uso muy común del sistema de clave pública es en la firma de documentos, dónde el emisor firma (cifra) un documento con su clave privada y cualquiera puede verificarlo:

$$D_{pub}(C_{priv}(mensaje)) = mensaje = D_{pub}(C_{priv}(mensaje))$$

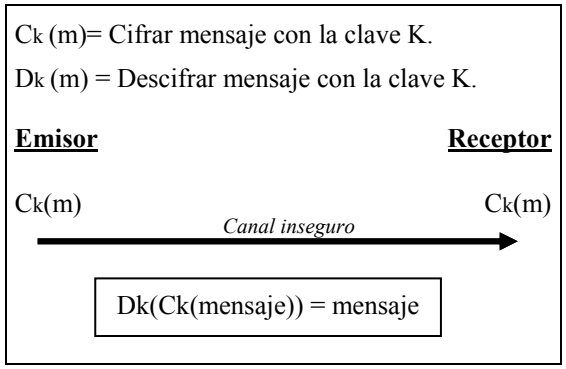


FIG. 3-1: Sistemas de clave privada o simétricos.

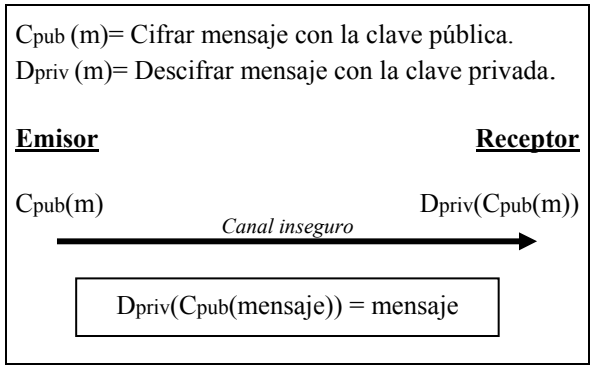


FIG. 3-2: Sistemas de clave pública o asimétricos.

3.2.2 Sistemas de clave privada o simétrica

En los sistemas de clave privada existen dos formas de cifrado, el cifrado en bloques y el cifrado en flujo.

Los sistemas basados en una única clave pueden **cifrar la información en bloques** (Block Ciphers) dónde la información se divide en bloques de una misma longitud y son precisamente estos bloques los que son cifrados. Estos bloques se cifran independientemente unos de otros, utilizando un modo denominado ECB (Electronic Code Block). De esta forma, hasta que no tengamos la suficiente cantidad de información (un bloque) esta no puede ser cifrada y por tanto enviada (una solución que se adopta a veces es rellenar el espacio que falta hasta completar el tamaño del bloque con ceros o espacios en blanco, por ejemplo cuando el último bloque a transmitir no está completo y no deseamos enviar más información).

No obstante, este tipo de cifrado hace que cifrar el mismo texto de la misma salida cifrada. Para evitar esto existe en modo CBC (Cypher Block Chaining) dónde el bloque ya cifrado se aprovecha para cifrar el siguiente. De esta forma la misma información no se cifrará igual dos veces.

- El sistema **DES** (Digital Encryption System) ha sido el adoptado por la mayoría de empresas y bancos, lleva más de 15 años como estándar de cifrado comercial en USA (para una explicación detallada ver [WWW13] y [Rif95]). Utiliza un tamaño de bloque y una clave de 64 bits (56 bits aleatorios + 8 bits de paridad) que le permite cifrar tanto en ECB como en CBC. Su mayor problema consiste en que 56 bits es un espacio de claves pequeño para la potencia de los ordenadores actuales.

- La extensión del algoritmo anterior se denomina **triple DES** y realiza tres veces el DES, aumentando la longitud de clave a 192 bits (64 x 3). Así, si utilizamos una llave formada por la triple concatenación de una misma clave DES, el resultado obtenido es el mismo que aplicando tres veces el DES con la misma clave (este sistema se utiliza para la compatibilidad de ambos sistemas).

- Existe una versión denominada **DESX** (DES eXtension) es una extensión de la empresa RSA que eleva la clave DES a 120 bits.

- **IDEA** (International Data Encryption Algorithm) es otro sistema de cifrado que tiene un tamaño de bloque de 64 bits y una longitud de clave de128 bits reales (no hay bits de paridad como en el DES).

- **RC2** es un algoritmo propietario de la empresa RSA que tiene un tamaño de bloque de 64 bits. Permite utilizar los modos ECB y CBC. Fue desarrollado como alternativa al DES y tiene una longitud de clave variable que va de 64 a 256 bits.

- **RC5** también pertenece de la empresa RSA. Se caracteriza por permitir bloques de 32, 64 o 128 bits. Su tamaño de clave varía de 0 a 2040 bits (255 bytes).

La segunda forma consiste en **cifrar la información en flujo** (Stream Ciphers), de esta forma se escoge una unidad fundamental de información (por ejemplo un byte) y esta se va cifrando según es producida. No hay necesidad entonces de esperar hasta completar un bloque o rellenar el espacio sobrante en este.

- **RC4** se caracteriza por utilizar la misma información de entrada que ha de cifrar para la generación de un número pseudo-aleatório que utilizará como clave, realizando un XOR entre la entrada y la clave. Esto significa que tanto el cifrado como el descifrado son operaciones idénticas. No se debe utilizar la misma clave más de una vez, ya que al utilizar un XOR como operación básica un atacante podría fácilmente descubrirla (XOR(XOR(X)) = X). La clave varía de 8 a 2048 bits.

- **RC4 con MAC** (Message Authentication Code) es una extensión del RC4 que busca asegurar integridad en los datos mediante el uso de una función (MAC) que a partir del mensaje genera una secuencia de bits de tal forma que si es modificado (deliberadamente o no), el receptor puede saberlo.

3.2.3 Sistemas de clave pública o asimétrica

Los sistemas basados en clave pública se caracterizan por la presencia de un par de claves (una pública y otra privada) eliminando el mayor problema de los sistemas de clave privada, dar a conocer únicamente al receptor autorizado la clave usada en el sistema de cifrado/descifrado. No obstante introduce un nuevo problema, la autenticación del origen de los datos. Puesto que todo el mundo conoce la clave pública, se puede enviar un mensaje falseando la procedencia. En los sistemas de clave privada esto no pasaba, ya que la clave la compartían únicamente el emisor y el receptor de la información, asegurando la confidencialidad y la procedencia de la información.

Las clave pública y privada están relacionadas matemáticamente, con lo que a partir de una es posible obtener la otra. No obstante esta relación no es directa, con lo que el hecho de conocer la clave pública (e incluso información y cómo esta es cifrada) no

compromete la seguridad de este sistema. Para poder dar a conocer las claves públicas de los usuarios sin ningún riesgo, debemos asegurarnos que estas no pueden ser ni modificadas ni alteradas en ninguna forma. Con esta función se crearon **las autoridades de certificación** (Certification Authorities, CA), que son organismos encargados de distribuir las claves públicas y velar por ellas.

- **RSA** (Rivest Shammir Adleman) se inventó en 1977 y es el algoritmo de clave pública más conocido y difundido en la actualidad. Se basa en las exponenciación de números utilizando aritmética modular y números primos (para una explicación detallada consultar [Mass88] y [Rif95]).

    1. Se eligen dos números primos P y Q y se calcula N = P * Q.
    2. Se elige la *clave pública* tal que 1 < clave pública < (P-1) * (Q-1)
    3. Se calcula la *clave privada* buscando el inverso de la clave pública en Z módulo (P-1) * (Q-1).
    4. Se hacen públicos los valores {clave pública, (P-1) * (Q-1)}.
    5. Para cifrar: X = información ^ clave pública módulo (P-1) * (Q-1).
    6. Descifrar: información = X ^ clave privada módulo (P-1) * (Q-1).

- **Diffie-Hellman** desarrollaron en 1976 un algoritmo para el intercambio de una clave entre dos usuarios. De esta forma se puede lograr que dos personas compartan una misma clave de forma segura. Este algoritmo no proporciona ni autenticación ni cifrado.

    1. **Generación de parámetros**. Una autoridad central elige un número primo B denominado base y un número G < B.

    2. **Fase 1**. Cada uno de los dos usuarios (a y b) genera un valor privado X tal que X_a, X_b < B. Se calculan los valores públicos Y_a, Y_b según la fórmula Y = G ^ X. Se intercambian estos valores públicos.

    3. **Fase 2**. Ambos usuarios calcula el número secreto común Z según la fórmula Z = Y módulo P.

- Actualmente se está trabajando en sistemas basados en **curvas elípticas** como el ECC (Elliptic Curve Cryptosystem) de la empresa Certicom y en la empresa RSA [WWW27].

3.2.4 Seguridad en las capas superiores

Tal y como se ha comentado en los puntos anteriores, la seguridad en la versión 4 de IP no fue contemplada en su diseño original, con lo que al querer introducir ampliaciones en las especificaciones IP (*¿IP versión 5?*) se encontraron muchos problemas, entre ellos la gran cantidad de software que debía modificarse para adoptar esta ampliación debido al gran tamaño que ya tenía INTERNET. Además se tardó mucho tiempo en finalizar las nuevas especificaciones, con lo que al desarrollarse el comercio electrónico (e-comerce) las empresas de venta por INTERNET puesto que no podían modificar ninguna definición de los protocolos (IP, TCP, UDP...) fueron desarrollando e imponiendo los suyos en los niveles que podían modificar, los correspondientes a la capa de aplicación (ver figura 3-3).

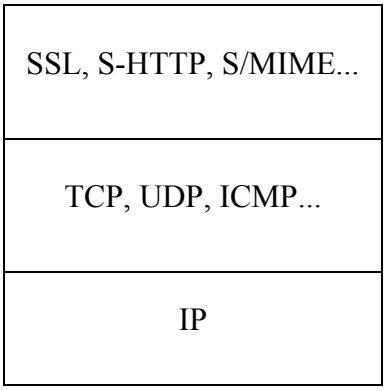


FIG. 3-3: Ubicación de los protocolos en capas superiores.

A continuación se detallan unos ejemplos de soluciones de seguridad adoptadas unánimemente en INTERNET.

- **SSL** (Secure Socket Layer) es un protocolo ampliamente utilizando que se basa en una arquitectura de tipo cliente/servidor y que permite una comunicación segura entre dos aplicaciones. Este protocolo permite la negociación de un algoritmo de cifrado y de las claves necesarias para asegurar un canal de seguridad (Channel Security) entre el cliente y el servidor. Este canal tiene tres propiedades principalmente:
    1. El canal garantiza la *privacidad.* Después del negociado de la clave privada todos los mensajes son cifrados.

    2. El canal garantiza la *autenticidad.* El servidor siempre se autentifica mientras que los clientes pueden hacerlo o no.

    3. El canal garantiza la *fiabilidad*. Los mensajes incluyen una integridad proporcionada por el uso del sistema MAC.

- **S-HTTP** (Secure Hypertext Transfer Protocol) es una extensión del protocolo HTTP utilizado en el servicio WWW que proporciona seguridad en el intercambio de documentos multimedia. Proporciona servicios de confidencialidad, autenticidad, integridad y no repudio (poder demostrar a una tercera persona que la información recibida proviene realmente del emisor). Asimismo permite múltiples algoritmos de cifrado (DES, DESX, IDEA y RC2) y de intercambio de claves (RSA, Kerberos, Out-band e In-band).

- **S/MIME** (Secure/Multipurpose INTERNET Mail Extensions) es una extensión del protocolo MIME [RFC1521] que añade las características de firma digital (ver punto 3.2.5) y cifrado. Los mensajes electrónicos constan de una cabecera [RFC822] dónde se especifican todas las opciones junto con el origen/destinatario del mensaje y el mensaje (body). En S/MIME el mensaje incluye un mensaje del tipo PKCS#7 que se calcula utilizando varios campos de la cabecera y del mensaje.

- **SET** (Secure Electronic Transaction) es un protocolo desarrollado por las empresas VISA [WWW30] y MASTERCARD [WWW21] para las transacciones electrónicas (e-comerce). Soporta los protocolos DES y RSA para el intercambio de claves y el cifrado de datos, además proporciona los siguientes servicios:

    1. Transmisiones confidenciales.
    2. Autenticación de los dos usuarios.
    3. Comprobación de la integridad en los pagos y las cantidades.
    4. Autenticación cruzada (del comerciante ante el usuario y del usuario al comerciante).

3.2.5 Firmas digitales y comercio electrónico

Para la definición de firma digital utilizaremos la definición proporcionada en la referencia [WWW20]:

"*Una **firma digital** es un bloque de caracteres que acompaña a un documento (o fichero), acreditando quién es su autor (**"autenticación"**) y que no ha existido ninguna manipulación posterior de los datos (**"integridad"**).*

*Para firmar un documento digital, su autor utiliza su propia clave **secreta**, a la que sólo él tiene acceso, lo que impide que pueda después negar su autoría (**"no revocación"**). De esta forma, el autor queda vinculado al documento que firma.*

*Cualquier persona puede **verificar** la validez de una firma si dispone de la clave **pública** del autor.*"

De esta forma, una firma digital es una secuencia de caracteres calculados a partir del documento original mediante unas funciones de resumen (*Digest*) o Hash (funciones que dada cualquier entrada producen una salida asociada a un rango determinado). Un simple ejemplo de una función Hash sería contar el número de letras del mensaje, si es

par asociamos un 0 y si es impar un 1 (ver figura 3-4). El principal inconveniente de este sistema es que pueden existir **colisiones** (dos mensajes diferentes producen la misma salida) por lo que conviene que las funciones tengan un rango de salida lo suficientemente grande (128 bits o más) para poder considerarlas libres de colisión.

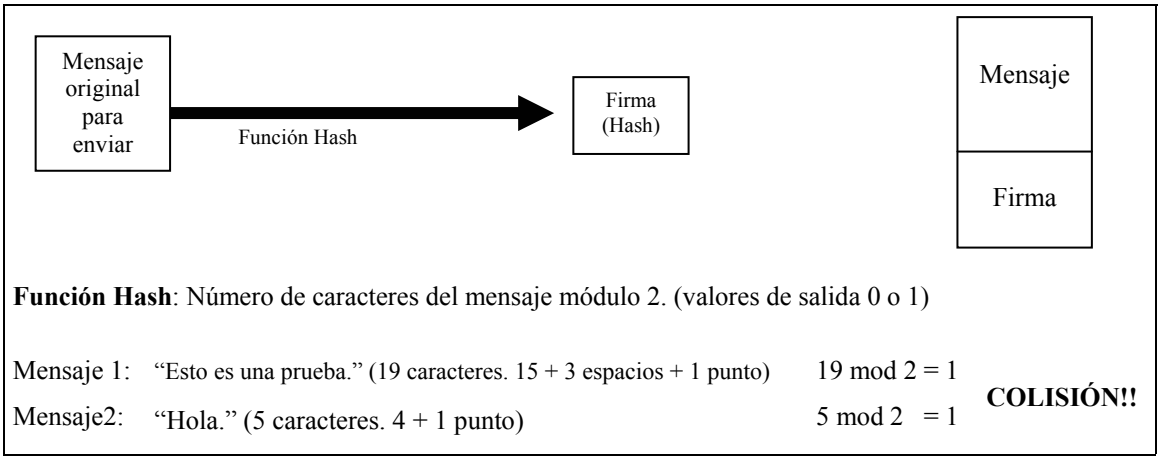


FIG. 3-4: Ejemplo de funciones Hash y colisión.

Los algoritmos más utilizados actualmente para el cálculo de la firma digital son los siguientes:

- **MD4** (Message Digest 4) es una función hash libre de colisión creada por Ron Rivest que produce una secuencia de 128 bits asociada al mensaje original. De esta forma, cualquier mensaje de cualquier longitud queda asociado de forma única a uno de los 2^128 valores posibles.

- **MD5** (Message Digest 5) es una mejora del algoritmo anterior. Utiliza bloques de entrada de 512 bits y retorna una salida de 128 bits. Para volúmenes grandes de datos, se recomienda primero comprimir los datos y luego aplicar la función Hash.

- **SHA** (Secure Hash Algorithm) es un algoritmo desarrollado por el NIST (National Standards and Technology Algorithm) que produce una salida de 160 bits.

Estas funciones producen una secuencia que acredita la autenticidad e integridad del documento ante terceras personas. La autenticación viene avalada por una autoridad de certificación (CA) en la cual se confía (ver figura 3-5). El principal inconveniente junto con la falta de una normativa internacional común, se encuentra cuando el emisor y el receptor no comparten la misma autoridad de certificación. La solución adoptada actualmente es crear una jerarquía de autoridades de certificación, de esta forma, aunque dependan de dos entidades distintas siempre se podrá subir al nivel superior hasta encontrar una entidad común.

Para la interacción entre las diferentes autoridades de certificación (CA) y su reconocimiento mutuo, se utilizan **infraestructuras de clave pública** (Public Key Infraestructure, **PKI**). Estas estructuras pueden negociar entre sí los certificados concedidos a sus usuarios y aceptar o denegar el de otras entidades. Este sistema de intercambio puede realizarse utilizando el estándar **X.509** (definido en [RFC2510] y [RFC2511]).

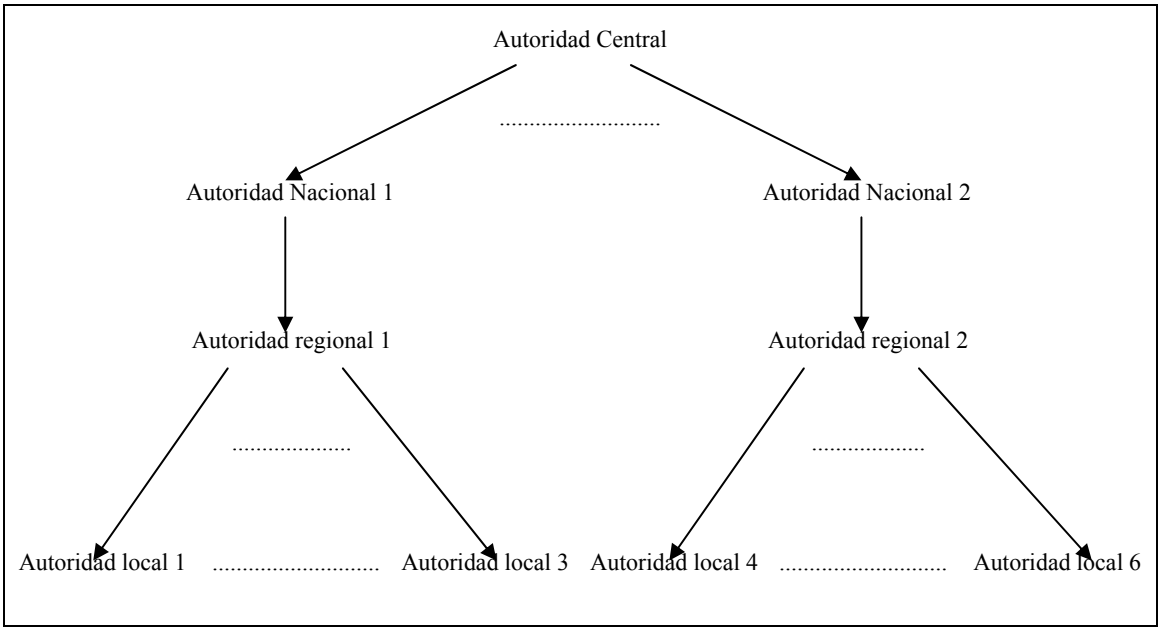


FIG. 3-5: Sistema jerárquico de autoridades de certificación (CA).

A continuación se citan brevemente los aspectos más relevantes del real decreto ley 14/1999 del 17 de septiembre de 1999 sobre la firma electrónica (BOE 18-09-1999) que sienta las bases legales que deben regular en nuestro país su funcionamiento.

Artículo 2. *Definiciones.*

A los efectos de este Real Decreto-ley, se establecen las siguientes definiciones:

a) «**Firma electrónica**»: Es el conjunto de datos, en forma electrónica, anejos a otros datos electrónicos o asociados funcionalmente con ellos, utilizados como medio para identificar formalmente al autor o a los autores del documento que la recoge.

b) «**Firma electrónica avanzada**»: Es la firma electrónica que permite la identificación del signatario y ha sido creada por medios que éste mantiene bajo su exclusivo control, de manera que está vinculada únicamente al mismo y a los datos a los que se refiere, lo que permite que sea detectable cualquier modificación ulterior de éstos.

c) «**Signatario**»: Es la persona física que cuenta con un dispositivo de creación de firma y que actúa en nombre propio o en el de una persona física o jurídica a la que representa.

d) «**Datos de creación de firma**»: Son los datos únicos, como códigos o claves criptográficas privadas, que el signatario utiliza para crear la firma electrónica.

e) «**Dispositivo de creación de firma**»: Es un programa o un aparato informático que sirve para aplicar los datos de creación de firma.

f) «**Dispositivo seguro de creación de firma**»: Es un dispositivo de creación de firma que cumple los requisitos establecidos en el artículo 19.

g) «**Datos de verificación de firma**»: Son los datos, como códigos o claves criptográficas públicas, que se utilizan para verificar la firma electrónica.

h) «**Dispositivo de verificación de firma**»: Es un programa o un aparato informático que sirve para aplicar los datos de verificación de firma.

i) «**Certificado**»: Es la certificación electrónica que vincula unos datos de verificación de firma a un signatario y confirma su identidad.

j) «**Certificado reconocido**»: Es el certificado que contiene la información descrita en el artículo 8 y es expedido por un prestador de servicios de certificación que cumple los requisitos enumerados en el artículo 12.

k) «**Prestador de servicios de certificación**»: Es la persona física o jurídica que expide certificados, pudiendo prestar, además, otros servicios en relación con la firma electrónica.

l) «**Producto de firma electrónica**»: Es un programa o un aparato informático o sus componentes específicos, destinados a ser utilizados para la prestación de servicios de firma electrónica por el prestador de servicios de certificación o para la creación o verificación de firma electrónica.

m) «**Acreditación voluntaria del prestador de servicios de certificación**»: Resolución que establece los derechos y obligaciones específicos para la prestación de servicios de certificación y que se dicta, a petición del prestador al que le beneficie, por el organismo público encargado de su supervisión.

Artículo 3. *Efectos jurídicos de la firma electrónica.*

1. La firma electrónica avanzada, siempre que esté basada en un certificado reconocido y que haya sido producida por un dispositivo seguro de creación de firma, tendrá, respecto de los datos consignados en forma electrónica, el mismo valor jurídico que la firma manuscrita en relación con los consignados en papel y será admisible como prueba en juicio, valorándose ésta según los criterios de apreciación establecidos en las normas procesales.

Se presumirá que la firma electrónica avanzada reúne las condiciones necesarias para producir los efectos indicados en este apartado, cuando el certificado reconocido en que se base haya sido expedido por un prestador de servicios de certificación acreditado y el dispositivo seguro de creación de firma con el que ésta se produzca se encuentre certificado, con arreglo a lo establecido en el artículo 21.

2. A la firma electrónica que no reúna todos los requisitos previstos en el apartado anterior, no se le negarán efectos jurídicos ni será excluida como prueba en juicio, por el mero hecho de presentarse en forma electrónica.

3.3 Pruebas realizadas

Los diferentes experimentos y pruebas realizadas referentes a este capítulo se detallan en el capítulo 6 (ver punto 6.1).

3.4 Resumen

En este capítulo se ha definido en qué consiste una comunicación, una comunicación segura y sus particularidades asociadas en su aplicación por INTERNET. Se han introducido brevemente los conceptos básicos relativos a la seguridad (privacidad, autenticidad y disponibilidad) y a la inseguridad (ataques activos y pasivos) en una comunicación realizada a través de INTERNET.

Se han comentado brevemente las alternativas criptográficas actuales (protocolos de clave privada o simétricos y protocolos de clave pública o asimétricos) así como los algoritmos más utilizados en su implementación práctica.

También se ha discutido la problemática de la seguridad asociada a la versión 4 del protocolo IP (principalmente que no fue diseñado para ser un protocolo seguro) y sus posibles soluciones (protocolos de nivel superior, autoridades de certificación y firmas digitales).

Finalmente se han repasado la legislación española actual (real decreto ley 14/1999) y los sistemas comerciales actuales o previstos que en un futuro que gobernarán el comercio electrónico (e-comerce) por INTERNET.

CAPITULO 4

El protocolo IP versión 6

4.1 Necesidad de revisar el protocolo IP versión 4

Como ya se ha visto en el capítulo 2, la versión 4 del protocolo IP es robusta y fiable. Además funciona adecuadamente y permite una independencia de los protocolos de las capas superiores (TCP, UDP...). Debemos entonces plantearmos seriamente las necesidades actuales y futuras para asegurarnos que una revisión del protocolo es necesaria y aportara ventajas. Esto es esencial debido al gran número de ordenadores conectados a INTERNET que utilizan TCP/IP. Lo que implica que cualquier modificación de cualquiera de estos protocolos afecte a una gran variedad de ordenadores y sistemas operativos, abarcando desde obsoletos VAX con VMS hasta modernos supercomputadores CRAY con sistemas operativos paralelos.

La versión 4 del protocolo IP utiliza un sistema de direcciones de 32 bits (2^32 = 4.294.967.296) subdivididas en cinco clases (ver punto 2.3 y figuras 2-10, 2-11). Con una simple revisión del crecimiento de INTERNET en los últimos 5 años, podemos observar que las direcciones a este ritmo se agotarán sobre los años 2005/2007 (ver figura 2-3).

Además las necesidades actuales han variado sensiblemente respecto las iniciales de 1978. En aquel momento tanto el número de ordenadores conectados como las expectativas de crecimiento eran mucho mas moderadas de lo que han sido realmente, y por tanto la suposición de que un tamaño de 32 bits sería suficiente parecía razonable.

De esta manera, podemos justificar la revisión de la versión 4 del protocolo IP desde dos puntos de vista principalmente:

1. **Técnicamente**: El sistema de direccionamiento es insuficiente para la demanda actual y futura prevista. Las tablas de encaminamiento (tablas de direcciones que almacenan los routers de forma interna, y que se utilizan para saber hacia dónde deben encaminar un datagrama) son excesivamente grandes debido a la gran cantidad de direcciones existentes actualmente y al sistema de encaminamiento utilizado, que obliga a los routers a mantener grandes cantidades de direcciones para conocer hacia dónde deben redireccionar los datagramas. Esto ralentiza excesivamente la circulación por INTERNET, ya que los routers deben consultar para cada datagrama estas tablas.

2. **Socialmente**: Las necesidades de los usuarios de INTERNET han aumentando espectacularmente, exigiendo nuevas capacidades (seguridad, privacidad, comercio electrónico, velocidad...) que la versión 4 no puede proporcionar.

4.2 El protocolo IP versión 6

Esta nueva revisión del protocolo IP se numerará con la versión 6. No se la denominará versión 5 para evitar posibles confusiones, ya que anteriormente a esta revisión se hicieron algunas pruebas añadiendo extensiones a la versión 4. Estas extensiones experimentales no acabaron de formalizarse en una nueva versión del protocolo, con lo que para evitar posibles conflictos de numeración y/o confusión, se optó por elegir el número de versión 6.

La información siguiente ha sido extraída en su mayor parte de las referencias [Hui98], [WWW14], [WWW16] y [WWW17].

La nueva cabecera del protocolo IP versión 6 (ver figura 4.1) no es mas que una evolución de la anterior versión. No se han introducido grandes cambios de contenido o estructura, sino que simplemente se ha mejorado y optimizado con los conocimientos y experiencias adquiridas durante los últimos 20 años. Se han suprimido algunos campos redundantes u obsoletos y se han ampliado algunas características para hacer frente a las nuevas necesidades de los usuarios (comunicaciones en tiempo real, seguridad...).

La nueva estructura de la cabecera del protocolo IP versión 6 se caracteriza principalmente por dos particularidades:

1. **Direcciones de 128 bits**. Se ha creado una nueva estructura de direccionamiento que aumenta su tamaño de 32 bits a 128 bits. Este aumento es consecuencia del gran aumento que ha sufrido INTERNET en los últimos años, agotando el número de direcciones existentes y colapsando las tablas de encaminamiento de los routers.

2. **Campos de longitud fija**. Con el objetivo de minimizar el tiempo necesario para procesar y encaminar los datagramas por INTERNET, se adopta un formato fijo. De esta forma se agiliza el tráfico de datagramas y se suprimen opciones poco utilizadas. No obstante se mantiene la posibilidad de especificar opciones, pero ya sin formar parte de la cabecera IP como ocurría anteriormente.

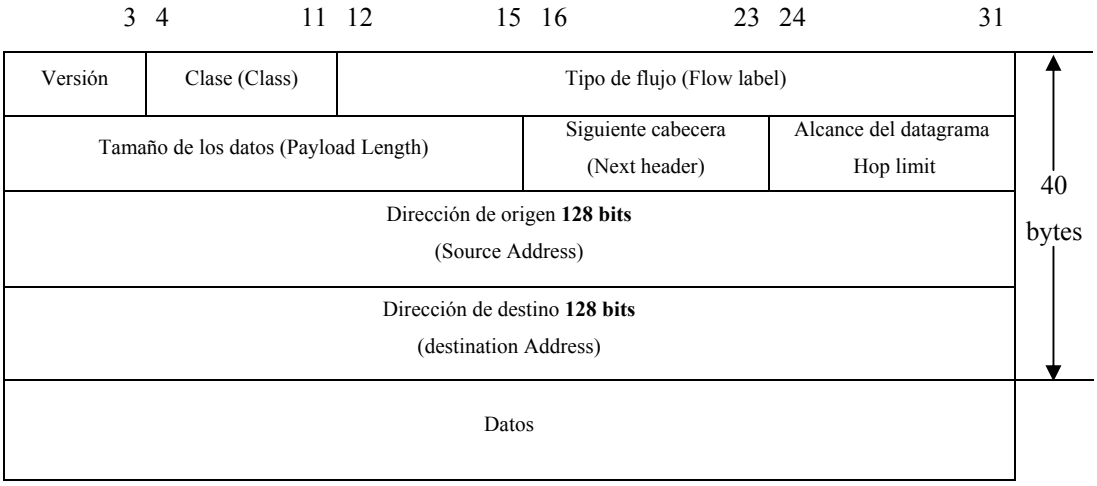


FIG. 4-1: Estructura de un datagrama IPv6.

El protocolo IP versión 6 sigue siendo al igual que las versiones anteriores, un protocolo **no fiable** y **sin conexión**. Esto continua siendo así debido a que la experiencia ha enseñado que este sistema funciona y da flexibilidad a la comunicación. Además permite que sean los protocolos de las capas superiores (ver figuras 2-2, 2-7 y 2-8) los encargados de mantener un estado de conexión o fiabilidad según crean necesario, manteniendo la estructura en capas del modelo TCP/IP.

El único campo que se mantiene en la misma posición y con el mismo significado que en formatos anteriores es el de versión, debido a que durante el tiempo de implantación de la nueva versión convivirán simultáneamente la versión 4 y 6. De esta forma, los routers podrán saber rápidamente si el datagrama que reciben es de una versión u otra.

Se han suprimido seis campos (tamaño de cabecera, tipo de servicio, número de identificación del datagrama, banderas, numero de byte del datagrama fragmentado y el checksum) respecto la versión 4 del protocolo IP. Además se han redefinido los campos de longitud del datagrama, tiempo de vida y de tipo del protocolo.

La **versión** (4 bits) se sigue manteniendo como el primer campo del datagrama. Esto es así para mantener la compatibilidad con formatos anteriores y porque permite de una forma sencilla y rápida discriminar que versión de datagrama se recibe, facilitando a los routers el proceso de discriminar entre versión 4 y versión 6.

La **clase** (Class) es un número de 8 bits que hace referencia a la prioridad del datagrama. Este campo es una de las nuevas aportaciones para conseguir algunos tipos de aplicaciones (videoconferencia, telefonía...) puedan realizarse en *tiempo real*.

El **tipo de flujo** (Flow Label) se compone de 16 bits, que permiten especificar que una serie de datagramas deben recibir el mismo trato. Esto es aplicable por ejemplo a una serie de datagramas que van del mismo origen al mismo destino y con las mismas opciones. Junto con el campo de clase (Class) permiten aplicaciones en tiempo real.

El **tamaño de los datos**[4] (Payload Length) al igual que en la versión 4, es un número de 16 bits, lo que permite un tamaño máximo en principio de 2^16 = 65536 bytes (64K). No obstante, a diferencia de la versión 4, este número hace referencia sólo al tamaño de los datos que transporta, sin incluir la cabecera (Si en *IPv4* enviamos 100 bytes de datos utilizando TCP, tendríamos que el valor seria 100 bytes + 20 bytes de cabecera TCP + 20 bytes de cabecera IP versión 4 = **140**. En *IPv6* suponiendo los mismos valores nos darían un valor de **120**. No se contaría el tamaño de la cabecera IP).

La **siguiente cabecera**[5] (Next Header) es un valor de 8 bits que indica al router si tras el datagrama viene algún tipo de extensión u opción. Este campo substituye al campo de banderas (flags) de la versión 4. De esta manera, en lugar de complicar la cabecera IP con la interpretación de los diferentes bits de opciones, se sitúan fuera del datagrama básico (ver figura 4-2). En la versión 6 del protocolo IP se definen una serie de cabeceras de extensión (ver figura 4-3) que se colocan justo después de los datos en forma de cadena (*daisy chain*) y que permiten al usuario personalizar el tipo de datagrama. De tal forma que podemos tener varias extensiones de cabecera tan solo indicando en el campo de siguiente cabecera de cada una de ellas el tipo de la cabecera que vendrá a continuación.

[4] Ver explicación de los Jumbogramas (datagramas superiores a 64K) en el punto 4.6.
[5] Ver explicación de las cabeceras más importantes en el punto 4.3.

<table>
<tr><td>Cabecera IPv6
(siguiente = TCP)</td><td colspan="3">Cabecera TCP + Datos</td></tr>
<tr><td>Cabecera IPv6
(siguiente = routing)</td><td>Cabecera Routing
(siguiente = TCP)</td><td colspan="2">Cabecera TCP + Datos</td></tr>
<tr><td>Cabecera IPv6
(siguiente = routing)</td><td>Cabecera Routing
(siguiente = Fragment)</td><td>Cabecera Fragment
(siguiente = TCP)</td><td>Fragmento
Cabecera TCP + Datos</td></tr>
</table>

FIG. 4-2: Cadena de cabeceras en IP versión 6.

Valor decimal	**Abreviatura (keyword)**	**Descripción**
0	HBH	Opciones entre saltos
4	IP	IP en IP (encapsulación en IPv4)
5	ST	Stream
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol
51	AH	Authentication Header
52	ESP	Encrypted Security Payload
59	NULL	No Next Header
60	DO	Destination Options Header
194	JBGR	Jumbogram

FIG. 4-3: Muestra de algunos valores para los tipos de cabecera en IP versión 6.

El **alcance del datagrama** (Hop Limit) es un número de 8 bits que indica el número máximo de routers que puede atravesar un datagrama hasta llegar a su destino. Este campo es el equivalente al tiempo de vida (*TTL*) de la versión 4. Cuando un datagrama llega a un router y es encaminado hacia otro ordenador, este campo es decrementado en una unidad. Este campo es necesario para evitar que los datagramas circulen infinitamente por la red, eliminándose al llegar a 0 (su valor máximo es de 2^8 = 256). De este tamaño podemos deducir que para que exista comunicación entre dos ordenadores conectados a INTERNET deben de estar alejados como máximo por 255 routers. Es sorprendente que aunque se haya ampliado considerablemente (de 32 bits a 128 bits) el número de ordenadores que pueden conectarse a INTERNET, se mantenga la esperanza de que la distancia entre dos ordenadores no crecerá por encima de este valor. Los protocolos de nivel superior (TCP, UDP...) pueden a su vez implementar algún tipo de control sobre los paquetes duplicados o huérfanos, utilizando por ejemplo un sistema de control del tiempo usando relojes (*timestamps*) como se define en [RFC1323].

4.3 Cabeceras del protocolo IP versión 6

Tal y como se ha comentado en el punto 4.1, la cabecera IP versión 6 no contiene ningún tipo de opciones a diferencia de la versión 4. No obstante, en algunos casos se hace necesario poder especificar algunas características especiales a los routers intermedios para que traten el datagrama IP de una forma determinada. No todos los datagramas son datos que circulan de un usuario a otro por INTERNET, algunos son mensajes entre los diferentes routers (Ejemplo: comunicar que está congestionado o fuera de servicio para que no le envíen mas datagramas).

Un ejemplo típico podría ser la necesidad de especificar por que routers debe circular el datagrama. Si queremos una ruta fija entre dos ordenadores, ya sea porque no nos fiamos de los demás o simplemente porque queremos medir el rendimiento entre dos puntos, necesitamos especificar por dónde encaminarlo, evitando que sean los routers intermedios los que tomen la decisión. La manera de hacerlo es indicar en el campo siguiente cabecera (*Next Header*) de datagrama IP el número correspondiente a la cabecera que colocaremos tras el datagrama (ver figuras 4-2 y 4-3) de esta forma, el router sabe que antes de encaminar el datagrama, debe de tener en cuenta esa información extra.

La **cabecera de encaminamiento** (Routing Header) tiene la misma función que en la versión 4. Son cuatro bytes (valor máximo de cada opción 2^8 = 256) de cabecera a los que se añade una serie de direcciones de 128 bits que corresponden a los routers por los que debe pasar el datagrama hasta llegar a su destino (ver figura 4-4). El primer campo es el de *siguiente cabecera* (Next Header), ya que como se explicó en el punto 4.1, la versión 6 utiliza un sistema de cadena (daisy chain) dónde se pueden especificar múltiples cabeceras (ver figura 4-2). A continuación viene el *tamaño de la cabecera* (Header Extension Length) que es el tamaño total de la cabecera en palabras de 64 bits (incluyendo todas las direcciones especificadas). El *tipo de encaminamiento* (Routing Type) es la política que se debe seguir en el encaminamiento, actualmente sólo existe el tipo 0 (si el router aparece en la lista de direcciones especificadas, se quita de la lista, decrementa el campo de segmentos restantes y busca cual de la lista está mas cerca para enviar el datagrama. Si no aparece en la lista, se limita a encaminarlo ignorando esta

opción). El número de *segmentos restantes* (Segments Left) es un valor que indica el número de direcciones de encaminamiento que aún restan. De esta forma, al llegar a 0 significa que el datagrama ha alcanzado su destino.

0 7	8 15	16 23	24 31
Siguiente cabecera (Next Header)	Tamaño de la cabecera (Header Extension Length)	Tipo de encaminamiento (Routing Type)	Segmentos restantes (Segments Left)
Dirección 1 (128 bits)			
.....			
Dirección N (128 bits)			

FIG. 4-4: Cabecera de routing (tipo 0).

La **cabecera de fragmentación** (Fragmentation Header) en la versión 6 se diferencia respecto a la de la versión 4 en que no existe un bit de fragmentación, ya que no se fragmentan los datagramas. La experiencia ha demostrado que todo y la gran versatilidad de la fragmentación implementada en la versión anterior (si un router recibe un datagrama de tamaño superior al que puede enviar, lo fragmentaba en varios datagramas de menor tamaño. Estos datagramas se encaminaban independientemente, y por lo tanto, si uno sólo de ellos no llegaba a su destino o llegaba incorrecto, todo el datagrama original se desechaba y debía ser retransmitido.) era mas problemática que beneficiosa debido a la gran variedad de redes conectadas a INTERNET y al coste de retransmisión de todo el datagrama. De esta forma, si a un router le llega un datagrama de tamaño superior al que puede transmitir, lo descarta y envía al origen un datagrama de error ICMP. No obstante existe una cabecera de fragmentación (ver figura 4-5) para que en el origen (y no los routers intermedios como en la versión 4) pueda fragmentar un tamaño de datos superior al soportado por su red (Maximun Transfer Unit, *MTU*) en varios de tamaño inferior que son independientes entre si y pueden ser reenviados por separados en caso de necesidad. El primer campo de *siguiente cabecera* (Next Header) indica el siguiente tipo (si hay) de cabecera que nos encontraremos. El siguiente campo también es un byte que actualmente esta *reservado* y debe ser puesto a 0. El campo de *desplazamiento de fragmento* (Fragment Offset) indica los 13 bits más significativos del desplazamiento, asumiendo pues que la fragmentación es en múltiplos de 64. En la versión 4 se usaban también 13 bits, pero eran los menos significativos, obligando a

multiplicar por 8 para obtener el desplazamiento total del byte, cosa que ahora no es necesaria. Los 2 bits siguientes están reservados para futuros usos. Finalmente el último bit es el bit de *mas fragmentos* (More), que es puesto a1 en todos los fragmentos y a 0 en el último.

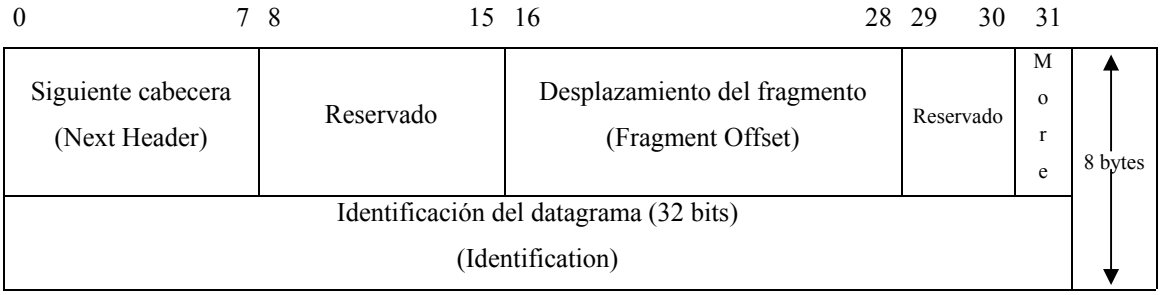


FIG. 4-5: Cabecera de fragmentación de datagramas.

La **cabecera de opciones de destino** (Destination Options Header) nos permite añadir opciones extra a los datagramas (ver figura 4-6) para que sean procesadas únicamente por el destinatario. Con este formato se permite que aquellos routers intermedios que no necesiten interpretarlas puedan evitarlas sin perder tiempo de proceso. El primer campo es como siempre el de *siguiente cabecera* (Next Header), que nos permite indicar la presencia de más cabeceras. A continuación tenemos el campo de *tamaño de la cabecera* (Extension Header Length) que en 8 bits especifica el tamaño de la cabecera en palabras de 64 bits sin incluir los primeros 64 bits. Esto permite tener un valor 0 en este campo, ya que si el tamaño cubriera toda la longitud, cada router debería examinar este campo para asegurarse que no es 0. Las *opciones* (options) son procesadas por el destinatario del datagrama, y su formato obliga a que sean múltiplos de 64 bits para poder ser especificadas en el campo de tamaño de la cabecera.

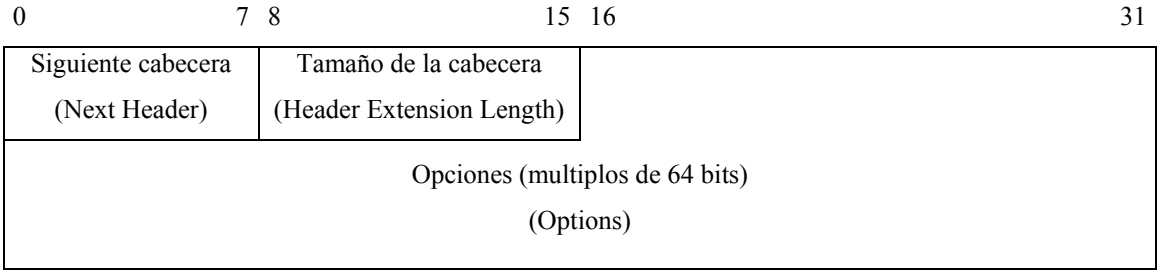


FIG. 4-6: Cabecera de opciones de destino.

La **cabecera de opciones entre saltos** (Hop-by-hop Options Header) permite especificar opciones que serán procesadas por todos los routers intermedios. Su formato es el mismo que el de la cabecera de opciones de destino (ver figura 4-6), aunque a diferencia de esta tan sólo es interpretada por el destinatario del datagrama. Cuando un datagrama llega una cabecera extra, especifica en el datagrama que tipo de cabecera le sigue con un código numérico (ver figura 4-3).

La **cabecera de autenticación** (Authentication Header) es una de las novedades más importantes en la versión 6 del protocolo IP (ver figura 4-7). Debe estar situada entre la cabecera IP y los datos del datagrama (ver figura 4-8). La presencia de una cabecera de autenticación no modifica de ninguna manera el comportamiento del resto de protocolos de nivel superior como TCP o UDP. Esta cabecera tan solo proporciona una seguridad implícita del origen del datagrama. De esta forma los protocolos superiores deben rechazar los paquetes que no estén adecuadamente autenticados. El primer campo indica la *siguiente cabecera* (Next Header) que nos encontraremos tras esta. A continuación nos encontramos el *tamaño de los datos* (Payload Length) especificado en palabras de 32 bits y un campo de 16 bits *reservado* que debe ser inicializado a 0. Después nos encontramos con el *índice de parámetros de seguridad*[6] (Security Parameters Index) y el *campo de número de secuencia*[7] (Sequence Number field) que ocupan 32 bits cada uno. Finalmente vienen los *datos de autenticación* (Authentication Data) que es un campo de longitud variable.

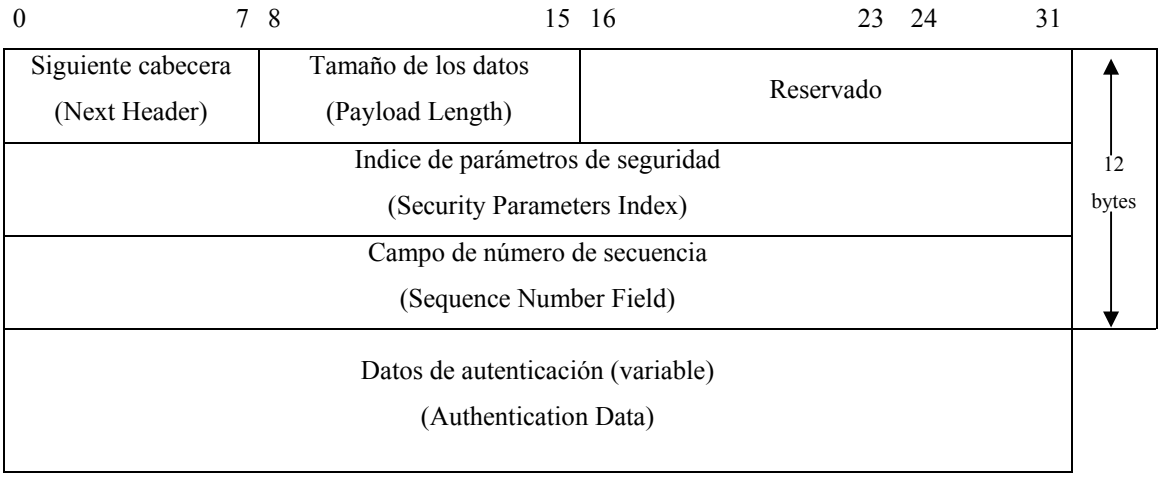


FIG. 4-7: Cabecera de autenticación de la versión 6.

[6] Se verá mas adelante en el punto 5.2.1.
[7] Se verá mas adelante en el punto 5.2.2.

Cabecera IPv6 (siguiente = **TCP**)	Cabecera de autenticación (Authentication Header)	Cabecera TCP + Datos	
Cabecera IPv6 (siguiente = **routing**)	Cabecera Routing	Cabecera de autenticación (Authentication Header)	Cabecera TCP + Datos
Cabecera IPv6 (siguiente = **routing**)	Cabecera de autenticación (Authentication Header)	Opciones al destino (End-to-end options)	Cabecera TCP + Datos

FIG. 4-8: Situación de la cabecera de autenticación.

Tal y como se ha visto, un datagrama puede incluir mas de una cabecera. Esto no debería suponer ningún problema para los routers intermedios encargados de encaminarlo hasta su destino. De esta forma, las cabeceras son procesadas por los routers a medida que estas llegan, sin ineficiencias. Este sistema de proceso ha sido comparado con las distintas capas de una cebolla (*onion-peeling*), dónde cada cabecera es una capa. De todas formas es importante señalas que hay algunas cabeceras con mayor importancia que otras, como la cabecera de autenticación (Authentication Header) que obliga a descartar todo el datagrama si es incorrecta o la cabecera de fragmentación (Fragmentation Header) que obliga al reensamblamiento de datagramas. Tenemos pues que el orden de las diferentes cabeceras es importante, y pese a no existir un formato rígido para establecer este orden, si que hay una recomendación en cuanto al orden adecuado de estas:

1. Cabecera IP versión 6 (IPv6 Header).
2. Cabecera de opciones entre saltos (Hop-by-hop Options Header).
3. Primera cabecera de opciones de destino (Destination Options Header).
4. Cabecera de encaminamiento (Routing Header).
5. Cabecera de fragmentación (Fragment Header).
6. Cabecera de autenticación (Authentication Header).
7. Segunda cabecera de opciones de destino (Destination Options Header).
8. Cabecera de protocolo de nivel superior (TCP, UDP...).

La presencia de cualquiera de estas cabeceras es opcional, con lo que por ejemplo no es necesario la especificación de una cabecera de opciones entre saltos (posición 2) si queremos insertar una cabecera de opciones de destino (posición 3).

Observamos que la cabecera de opciones de destino (Destination Option Header) se repite en dos posiciones distintas (3 y 7) esto es debido a que si necesitamos enviar datagramas encapsulados (*tunneling*) y queremos que estas opciones sean utilizadas por los routers intermedios debemos enviar estas opciones antes que las de encaminamiento. Por otro lado, si queremos pasar información que sólo sea interpretada por el destinatario del datagrama debemos colocar estas opciones justo antes de la cabecera del protocolo del nivel superior (posición 7).

4.4 ICMP y los mensajes de error

El protocolo de control de mensajes de INTERNET (INTERNET Control Message Protocol, **ICMP**) ya existía en la versión 4, y su principal objetivo es el de enviar mensajes entre los diferentes ordenadores (por ejemplo mensajes de error como destino desconocido o tiempo de respuestas excedido). Todo y ser un protocolo de nivel superior al IP, también ha sido adaptado a la versión 6 del protocolo IP. Se han suprimido muchos servicios redundantes o no utilizados, se ha impuesto un formato fijo para facilitar su tratamiento por los routers y se le han añadido características como la extensión de las direcciones a 128 bits (ver figura 4-9). Todo esto lleva a que la nueva revisión del ICMP para la versión 6 del IP (numerada como 2) sea incompatible con la versión anterior (identificada como 1) para IP versión 4. El primer campo de *tipo* (Type) indica la versión del protocolo ICMP, en el caso de ser compatible con la versión 4 es 1, y si es compatible con la versión 6 es 2. El *código* (Code) hace referencia a la naturaleza del mensaje que transporta (ver figura 4-10). El *checksum* es una suma de control de los datos que se envían, de forma que se pueda verificar que son correctos. Finalmente el mensaje (Body Message) es de longitud variable y contiene los datos.

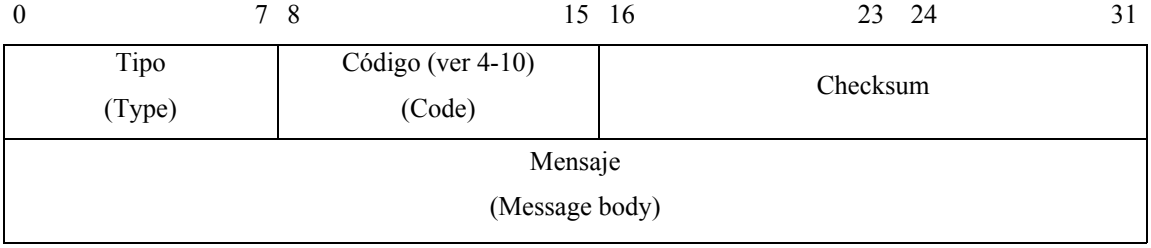


FIG. 4-9: Formato del ICMP versión 2 compatible con la versión 6 de IP.

Codigo	**Significado**
1	Destino inalcanzable (Destination Unrechable).
2	Datagrama demasiado grande (Packet too big).
3	Tiempo de respuesta agotado (Time Exceeded).
4	Parámetros incorrectos (Parameter Problem).
128	Solicitud de ECHO (ECHO Request).
129	Respuesta a ECHO (ECHO reply).
133	Solicitud de router (Router Solicitation).
135	Solicitud de vecino (Neighbour Solicitation).

FIG. 4-10: Tabla con los códigos más relevantes del ICMP versión 2.

Tal y como hemos visto en los puntos anteriores, en el caso de que un router descarte un datagrama, envía un mensaje ICMP al propietario del datagrama notificando la causa del error. Los cuatro primeros códigos (ver figura 4-10) indican los motivos por los cuales un router descarta un datagrama. Esto obliga a que los routers no envíen mensajes ICMP ante datagramas dirigidos a mas de un usuario a la vez (*multicast*[8]) para evitar avalanchas de respuestas. De la misma manera tampoco se responde a datagramas de tipo ICMP para evitar bucles infinitos de respuestas de error.

Destacar finalmente que el código de mensaje 2, *datagrama demasiado grande* (Packet too big) es el mecanismo utilizado para el *cálculo del tamaño máximo* de datos (Maximun Transfer Unit, **MTU**) que el router puede soportar. Esto permite saber al emisor cual es el tamaño de datagrama máximo que puede enviar al destino sin peligro de que sea descartado por algún router intermedio, optimizando de esta forma la comunicación entre dos ordenadores por INTERNET. Como este parámetro depende del camino que tome el datagrama (y por lo tanto de todos y cada uno de los routers intermedios que atraviese) hasta su destino, permite de una forma fácil y eficiente optimizar la comunicación dinámicamente.

[8] El concepto de multicast se explica en el punto 4.7.

4.5 Impacto en los protocolos superiores

El cambio que comporta la versión 6 del protocolo IP ha de tener forzosamente un impacto en los protocolos de nivel superior (TCP, UDP, ICPM...) puesto que estos hacen uso de los servicios de IP para el transporte a través de INTERNET. No obstante, el impacto en estos es mínimo, ya que debido a la estructura de encapsulamiento utilizada ya en la versión 4 (ver capítulo 2 y figura 4.2) los protocolos superiores permanecen separados de la cabecera IP. De esta forma se ha definido una nueva pseudo-cabecera para TCP y UDP (ver especificación para ICMP en el punto 4.4) en la que se amplía el campo de direcciones a 128 bits (ver figura 4-11) y se mantiene la filosofía de la versión 6 de IP de tener cabeceras de formatos fijos que faciliten su manipulación por los routers,. Primero nos encontramos con la *dirección de origen* (Source Address) que es un campo de 128 bits para proporcionar compatibilidad con la versión 6 de IP. A continuación está la *dirección de destino* (Destination Address) que también ha sido ampliada a 128 bits. El *tamaño de los datos* (Payload Length) incluye tanto las posibles cabeceras que encontremos tras esta pseudo-cabecera como los datos enviados. A continuación tenemos un campo de 24 bits no utilizado (Zero) que actualmente es 0 y finalmente el siguiente tipo de cabecera (Next Header) que sigue en la cadena. Esta pseudo-cabecera ya forma parte de la especificación de los protocolos TCP y UDP.

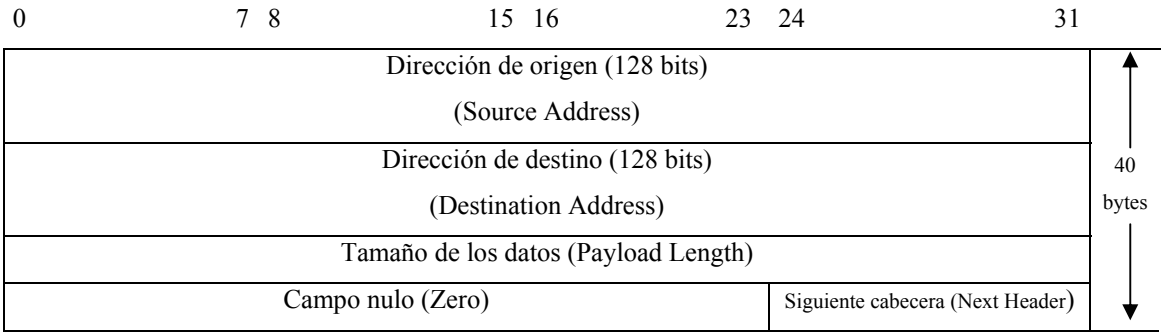


FIG. 4-11: Pseudo-cabecera para los protocolo TCP y UDP compatible con la versión 6 de IP.

Podemos observar que en esta pseudo-cabecera al igual que en la especificación de las demás cabeceras (incluyendo la del propio protocolo IP versión 6) ha desaparecido el campo de checksum. Esta decisión un tanto arriesgada de suprimir de las cabeceras los mecanismos de detección de errores viene dada por:

1. Los campos de checksum en la versión 4 tan sólo cubrían la posibilidad de posibles errores en las cabeceras, sin tener en cuenta posibles errores en los datos que transportan.

2. Cada vez que un datagrama pasa por un router, su tiempo de vida (TTL) es decrementado en una unidad, lo que implica un nuevo cálculo del checksum en cada router. Esta práctica choca con la filosofía de acelerar el encaminamiento en INTERNET. Además muchos routers para acelerar sus prestaciones ignoraban este campo, con lo que perdía todo su significado.

3. En la actualidad, todas las redes de transporte utilizadas (Ethernet, FDDI, ATM...) ya incluyen como parte de su especificación un checksum de comprobación. Con lo que la integridad de los datos ya esta asegurada por el propio medio de transporte.

4. Finalmente, protocolos de nivel superior (TCP, UDP...) ya incluyen su propio checksum. Con lo que la inclusión de este en cada cabecera es redundante.

4.6 Datagramas que superan los 64K (Jumbogramas)

Una de las características más sorprendentes de la versión 6 del protocolo IP es la posibilidad de enviar cantidades de datos superiores a los 64K. Tal y como se describe en la cabecera IP de la versión 6 (ver punto 4.1 y figura 4-2), el tamaño de los datos se especifica en 16 bits (2^16 = 65536 = 64K), con lo que en principio este sería su máximo tamaño. El motivo por el cual el campo de la longitud del datagrama no se aumentó (como sería lógico para mantener la coherencia de cabeceras simplificadas y

rápidas de procesar por los routers) es porque la posibilidad de datagramas superiores a 64K es más una opción de futuro que una necesidad actual. Para poder enviar datagramas superiores a los 64K, tanto el origen como el destino y cada uno de los routers intermedios deben permitir en sus redes el envío de mas de 64K, cosa que actualmente no sucede (ver figura 4-12).

Esta opción fue una de las más discutidas tanto en la IETF [WWW16] como en los forums de debate en INTERNET. Pese a ignorar el principio de transmisiones de datos cortas (puesto que como los errores en las comunicaciones actuales no son tan poco frecuentes como sería deseable, tenemos que si enviamos N bytes y **tan sólo uno de estos bits** es erróneo, debemos retransmitir otra vez los N bytes) y su poco probable utilización a corto o medio plazo, fue aprobada con la denominación de jumbograma (ver figura 4-13).

La forma de especificar un jumbograma pasa por situar el tamaño de los datos (Payload Length) a 0 en el datagrama IP y utilizar el sistema de cabeceras de extensión definidas en la versión 6 (ver figura 4-3). La cabecera correspondiente al jumbograma deberá ser procesada por todos los routers intermedios. Se dispone de 32 bits (2^32 = 4.294.967.296 bytes = 4 Gigabytes) para la especificación del tamaño del datagrama.

Tipo de red	**Tamaño máximo de transacciones (MTU)**
ATM	8192 bytes (para TCP/IP)
Comunicaciones punto a punto (PPP)	296 bytes
X.25	576 bytes
IEEE 802.3/ 802.2	1492 bytes
Ethernet	1500 bytes
FDDI	4352 bytes
Token Ring	4464 bytes
Fast Token Ring	17914 bytes
Hyperchannel	65535 bytes

FIG. 4-12: Tamaño máximo de datos (MTU) de las redes más utilizadas actualmente.

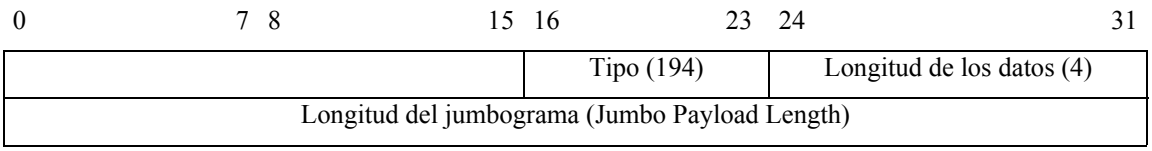


FIG. 4-13: Cabecera de un jumbograma.

4.7 Direccionamiento en IP versión 6

Una de las características más relevantes de la versión 6 del protocolo IP es el aumento de las direcciones de 32 a 128 bits. Una manera sencilla de entender este aumento sería coger el sistema de direccionamiento utilizado en la versión 4 (ver figuras 2-9 y 2-10) y aumentarlo simplemente añadiéndole más bits. Pero esto no seria cierto, puesto que uno de los motivos de este cambio es el de la ineficiente gestión de las direcciones, haciendo lento el encaminamiento por INTERNET. De esta forma, en la versión 6 se definen tres tipos de direcciones:

1. **Unicast.** Este grupo de direcciones se caracteriza por identificar un único punto final de destino (point-to-point). Un datagrama enviado a una dirección *unicast* será entregado a un solo punto de destino.

2. **Multicast.** Las direcciones *multicast* agrupan un conjunto de puntos finales de destino. Un datagrama enviado a una dirección *multicast* será entregado a un conjunto de destinos que forman parte de un mismo grupo.

3. **Anycast.** Este grupo de direcciones al igual que el *multicast* agrupa un conjunto de puntos finales de destino. La diferencia principal con el *multicast* está en sistema de entrega de datagramas. Un datagrama enviado a una dirección *anycast* es entregado solo a un punto de destino (el miembro más cercano del grupo al emisor del datagrama). Este tipo de agrupación no existía en la versión 4.

Las direcciones IP de la versión 6 están compuestas por 128 bits. Los diseñadores del protocolo optaron por representarlas en 8 agrupaciones de 16 bits (ver figura 4-14). De esta forma se puede utilizar la notación hexadecimal, que permite una representación más compacta que una ristras de 128 unos y ceros. Todo y esta simplificación, continua siendo bastante complicada de manipular y recordar (es posible recordar que cc.uab.es tiene la dirección 158.109.0.4, pero es imposible recordar que le corresponde la dirección IP versión 6 3FFE:3326:FFFF:FFFF:FFFF:FFFF:FFFF:1). Curiosamente se destacó esta cualidad para impulsar el uso de los nombres (cc.uab.es) por los usuarios.

Para compactar estas direcciones tan voluminosas, se aceptaron una serie de simplificaciones:

- Supresión de los ceros redundantes situados a la izquierda.

- Simplificación de los ceros consecutivos mediante el uso del prefijo '**::**'. Este prefijo tan sólo puede ser utilizado una vez en una misma dirección.

- Para las direcciones IP versión 6 obtenidas añadiendo 96 ceros a la dirección IP versión 4 (10.0.0.1 -> 0:0:0:0:0:0:A00:1) se permitirá el uso de la notación decimal (::10.0.0.1).

- La especificación de un prefijo de direccionamiento en la versión 6 se realizará mediante la forma dirección_ipv6/prefijo (Si tenemos el prefijo de 40 bits FEDC:BA98:76 en la dirección FEDC:BA98:7600::1 se especificará como FEDC:BA98:7600::1/40). Se debe tener mucho cuidado con las simplificaciones siempre que se indican prefijos, ya que puede pasar que con el prefijo de 64 bits FEDC:BA98:0: y la dirección FEDC:BA98:0

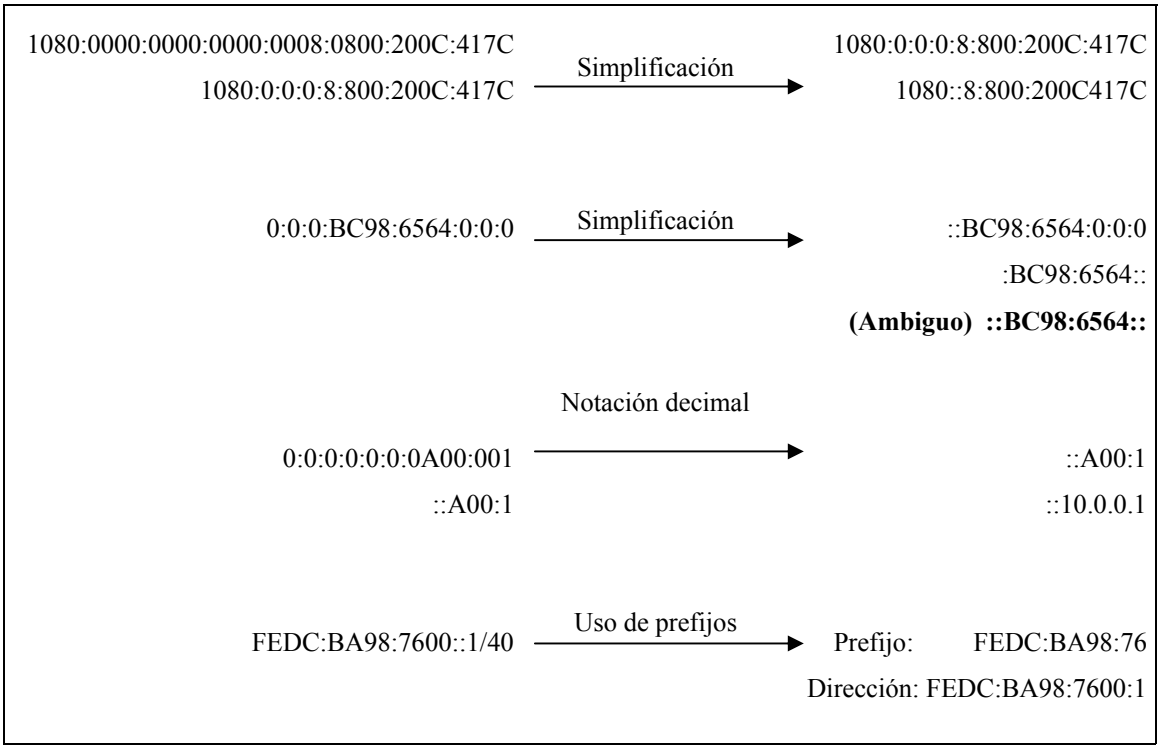


FIG. 4-14: Simplificaciones en el direccionamiento IP versión 6.

Después de la experiencia de observar cómo se van agotando las direcciones IP versión 4 sin poder ampliar o reestructurar el direccionamiento de una forma no traumática, los diseñadores de la versión 6 optaron por no consumir todo el espacio direccionable de los 128 bits, realizando una partición en subgrupos independientes (ver figura 4-15) para facilitar en un futuro la ampliación de tipos de direcciones o incluso un nuevo tipo de direccionamiento.

De esta forma, se han reservado algunos prefijos de direcciones para aquellos grupos específicos de direcciones (como direcciones compatibles NSAP o direcciones compatibles IPX) que se prevé que en un futuro pueden necesitar un rango de direcciones separado del resto de direcciones IP, incluso se ha reservado un rango de direcciones para un posible direccionamiento geográfico. Todo y esta partición del espacio de direcciones, aún queda más de un 70% del espacio total de direcciones sin asignar.

Grupo asignado	Prefijo	Fracción del espacio ocupado
Reservado	0000 0000	1/256
No asignado	0000 0001	1/256
Direcciones NSAP	0000 001	1/128
Direcciones IPX	0000 010	1/128
No asignado	0000 011	1/128
No asignado	0000 1	1/32
No asignado	0001	1/16
No asignado	001	1/8
Direcciones globales unicast	010	1/8
No asignado	011	1/8
Direcciones geográficas unicast	100	1/8
No asignado	101	1/8
No asignado	110	1/8
No asignado	1110	1/16
No asignado	1111 0	1/32
No asignado	1111 10	1/64
No asignado	1111 110	1/128
No asignado	1111 1110 0	1/512
Direcciones locales (Link Local)	1111 1110 10	1/1024
Direcciones locales (Site Local)	1111 1110 11	1/1024
Direcciones Multicast	1111 1111	1/256

FIG. 4-15: Distribución inicial del espacio de direcciones en la versión 6 de IP.

4.7.1 Direcciones unicast

El grupo de direcciones *unicast* representa aquellas direcciones que identifican un único punto final en una comunicación. Este grupo de direcciones presenta cinco subtipos de direcciones especiales:

1. La **dirección no especificada** (Unespecified Address) está compuesta por 16 bytes nulos (0:0:0:0:0:0:0:0) y sólo puede utilizarse como dirección inicial mientras se inicializa y se recibe una dirección fija. También puede utilizarse para funciones internas que requieran la especificación de una dirección IP.

2. La **dirección interna** (Loopback Address) se define como 15 bytes nulos y un byte con el último bit a 1 (0:0:0:0:0:0:0:1). Esta dirección es interna y de ninguna forma puede circular por la red o ser dirección de origen o destino de un datagrama. Su utilidad viene dada para los ordenadores que no dispongan de una conexión de red y deseen simular el comportamiento de conexión a una red mediante una dirección fantasma que nunca saldrá del propio ordenador.

3. **Direcciones tipo IP versión 4** (IPv4 Based Address) son aquellas direcciones que se obtienen añadiendo un prefijo de 96 ceros a una dirección IP versión 4 (10.0.0.1 pasaría a ser en la versión 6 ::10.0.0.1).

4. **Direcciones locales reservadas** (Site Local Address) son direcciones reservadas para intranets. Estas direcciones no son válidas por INTERNET y tan sólo sirven para que una organización (por ejemplo una empresa o una universidad como la UAB) pueda crear una organización de sus redes basada en un esquema TCP/IP sin la necesidad de estar conectados a INTERNET (en la versión 4 de IP, existen diferentes clases reservadas para este mismo fin, como por ejemplo 192.168.XXX.YYY)

5. Las **direcciones de inicialización locales reservadas** (Link Local Address) son direcciones que pueden utilizar los ordenadores conectados a una misma red local mientras se inicializa y no tiene asignada una dirección IP. Se diferencia de la dirección no especificada (0:0:0:0:0:0:0:0) en que a diferencia de esta, la dirección de inicialización local si puede circular por la red, permitiendo por ejemplo obtener el sistema operativo de un servidor en la misma red. Esta característica ya existe en la versión 4 del protocolo IP, que actúa conjuntamente con los protocolos ARP y RARP [Ric98-1]. Estas direcciones se construyen con el prefijo FE80::/10 y 64 bits que representan la dirección física (*MAC Address*) de la tarjeta de red.

4.7.2 Direcciones multicast

Las direcciones de tipo *multicast* (ver figura 4-16) fueron ya añadidas a la versión 4 del protocolo IP en 1988 con la definición de la clase D (ver capitulo 2 y figura 2-10). Aprovechando la experiencia obtenida desde entonces, y viendo su viabilidad se decidió añadirlas en la especificación de la versión 6. Este tipo de direcciones se caracteriza por ser comunes a un grupo de ordenadores (la misma dirección es compartida por todos los integrantes del grupo) de forma que un datagrama enviado a esta dirección será distribuido a todos los integrantes del grupo. Estas direcciones se forman mediante el prefijo FF**XY**:ZZZZ:ZZZZ:ZZZZ:ZZZZ:ZZZZ:ZZZZ:ZZZZ:ZZZZ/16.

El símbolo **X** agrupa un conjunto de 4 bits denominado *banderas* (Flags) dónde se especifican una serie de opciones, aunque actualmente los tres primeros están reservados y deben ser inicializados a 0 y el cuarto denominado transitorio (*transient)* e especifica si la dirección es local y una vez finalizada la comunicación debe liberarse (valor 0) o si la dirección es fija y debe conservarse (valor 1). El símbolo **Y** también agrupa 4 bits que definen el *alcance* (Scope) de la comunicación, evitando que los datagramas de una videoconferencia local salga a INTERNET o viceversa.

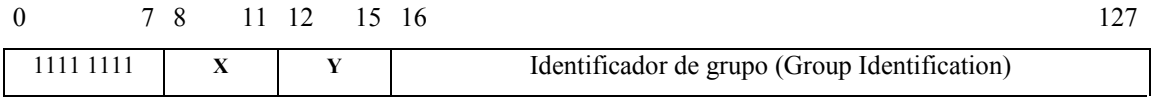


FIG. 4-16: Formato de una dirección de tipo multicast.

4.7.3 Direcciones anycast

Una de las nuevas características presentes en la versión 6 de IP es la inclusión de un nuevo tipo de direcciones denominado *anycast* (ver figura 4-17). Este tipo de direcciones aún en fase experimental se diferencia de las direcciones multicast en que el datagrama no es entregado a todos los miembros del grupo, sino que se entrega al integrante del grupo más cercano del origen del datagrama.

0	N-1 N	127
Prefijo de subred (Subnet prefix) N bits	Nulo (::0) 128 – N bits	

FIG. 4-17: Formato de una dirección de tipo anycast.

El formato de este tipo de direcciones es muy sencillo debido a que toda la carga se centra en el sistema de encaminamiento. De esta forma, para cada router debe guardar un solo registro que le indica cual es el miembro más cercano a él del grupo especificado, y al recibir un datagrama con una dirección de destino anycast comprobar la existencia de este registro especial en su tabla de encaminamiento o encaminar normalmente el datagrama.

4.8 Pruebas realizadas

Los diferentes experimentos y pruebas realizadas referentes a este capítulo se detallan en el capítulo 6 (ver punto 6.2).

4.9 Resumen

En este capítulo se han introducido los aspectos más importantes que han provocado la necesidad de una revisión del protocolo IPv4 y se han explicado con detalle los nuevos aspectos de la versión 6:

- Nuevo formato de los datagramas IP versión 6, que se caracterizan principalmente por tener los campos de longitud fija y ser una simplificación de la versión anterior.

- Nuevo sistema de extensión de cabeceras (cabecera de encaminamiento, fragmentación, de opciones de destino, de opciones entre saltos y de autenticación) que permite la especificación de opciones sin pérdidas de tiempo en su proceso.

- Sistema de direccionamiento de 128 bits. Se aumenta el tamaño de las direcciones de 32 a 128 bits y se clasifican en tres subgrupos (*unicast*, *multicast* y *anycast*).

- Aumento del tamaño de los datagramas (jumbogramas) para superar los 64K.

- Impacto en los protocolos de las capas superiores (TCP, UDP, ICMP...).

CAPITULO 5

Las extensiones de seguridad IPSec para IP versión 6

5.1 La seguridad en el protocolo IP

Tal y como se ha comentado en los capítulos 1 y 3, debido al carácter científico que en un principio tuvo INTERNET, la seguridad no fue contemplada históricamente en ninguna de las capas que forman la estructura TCP/IP. Con el auge de las tecnologías de la información y el aumento de personas y empresas conectadas a INTERNET, la necesidad de seguridad se fue convirtiendo en una necesidad. Además la proliferación de noticias sobre personas sin escrúpulos dedicadas a la piratería en INTERNET, creó un gran malestar social debido a la sensación de inseguridad por los ataques que sufrían tanto las empresas (bancos, universidades e incluso instituciones como la NASA) como los usuarios (utilización ilícita de números de tarjetas de crédito...).

La tardía reacción de las instituciones encargadas de la creación y modificación de los protocolos de INTERNET, propició la aparición de diferentes soluciones comerciales (SSL, SET...) para que los usuarios pudieran disfrutar de una seguridad que INTERNET no proporcionaba.

Aprovechando la necesidad de adaptar los diferentes protocolos al crecimiento de INTERNET, se optó por introducir una serie de especificaciones para garantizar la seguridad como parte implícita de las nuevas especificaciones de los protocolos. Estas especificaciones se conocen como IP Security o **IPSec**.

Una vez que se había consensuado la necesidad de introducir especificaciones de seguridad como parte intrínseca de los protocolos y no como simples extensiones voluntarias para los fabricantes de software (como paso con la versión 5?), se planteó un duro debate sobre que capa sería la idónea para proporcionar esta seguridad. Esta decisión era crítica, ya que en el mercado ya existían diferentes soluciones comerciales (SSL, SET...) que proporcionaban distintos grados de seguridad en la capa de usuario.

Finalmente para evitar duplicidades y asegurar un sistema seguro y auténtico en todas las capas, se optó por incluir las especificaciones en el nivel más bajo de la pila (Stack) de protocolos, en la especificación del protocolo IP versión 6.

5.2 Las especificaciones IPSec

La información siguiente ha sido extraída en su mayoría de las referencias [Hui98], [RFC2104], [RFC2401], [RFC2402], [RFC2403], [RFC2404], [RFC2405], [RFC2406], [RFC2407], [RFC2408], [RFC2410], [RFC2410], [RFC2411], [RFC2412], [RFC2451] así como de [WWW16].

Las especificaciones IPSec han sido definidas para trabajar en la capa inferior de la pila (Stack) de protocolos TCP/IP, funcionando por lo tanto en el nivel de datagrama y siendo independientes del resto de protocolos de capas superiores (TCP, UDP...).

La seguridad en IPSec se proporciona mediante dos aspectos de seguridad (Security Payload):

1. **Cabecera de autenticación** (Authentication Header, **AH**). Esta cabecera es la encargada de proporcionar autenticidad a los datos (datagramas) que se reciben en dos aspectos:

   - Los datagramas provienen del origen especificado. Se garantiza la autenticidad del origen de los datos (no pueden ser repudiados).
   - Los datagramas (y por tanto los datos que contienen) no han sido modificados.

2. **Cifrado de seguridad** (Encrypted Security Payload, **ESP**). De esta forma se garantiza que tan sólo el destinatario legítimo del datagrama (datos) pueda descifrar el contenido del datagrama.

La autenticidad y el cifrado de datos (o datagramas) requiere que tanto el emisor como el receptor compartan una clave, un algoritmo de cifrado/descifrado y una serie de parámetros (como el tiempo de validez de la clave) que diferencian una comunicación segura de otra. Estos parámetros conforman la **asociación de seguridad** (Security Association, SA) que permite unir la autenticidad y la seguridad en IPSec.

En un ordenador con múltiples conexiones (consultar el correo mientras se baja un fichero por FTP y se consulta el saldo bancario...) podemos tener varias asociaciones de seguridad (como mucho una por conexión). Para poder diferenciar entre ellas utilizaremos un **índice de parámetros de seguridad** (Security Parameter Index, SPI) que nos permitirá al recibir un datagrama saber a que asociación de seguridad hace referencia, y de esta forma poder autenticarlo y/o descifrarlo.

Al iniciar una comunicación que utilice los servicios IPSec con un único destino (direcciones unicast) este nos debe comunicar a que índice de parámetros de seguridad (SPI) debemos hacer referencia. Análogamente en una comunicación con varios destinos (direcciones multicast o anycast) todos los destinatarios deben compartir el mismo número de índice (SPI).

5.2.1 La cabecera de autenticación (AH)

La **cabecera de autenticación** (Authentication Header, AH) es una cabecera específica de la versión 6 de IP (ver figura 5-1) y se designa con el número 51 (ver capítulo 4, figura 4-3). Se suele situar justo antes de los datos, de forma que los proteja de posibles atacantes. No obstante ha sido diseñada de forma muy versátil, pudiendo incluirse antes que otras cabeceras (cabecera de opciones, cabecera de encaminamiento...) para asegurar así que las opciones que acompañan al datagrama son correctas.

De esta forma, la presencia de una cabecera de autenticación no modifica el funcionamiento de los protocolos de nivel superior (TCP, UDP...) ni el de los routers intermedios, que simplemente encaminan el datagrama hacia su destino.

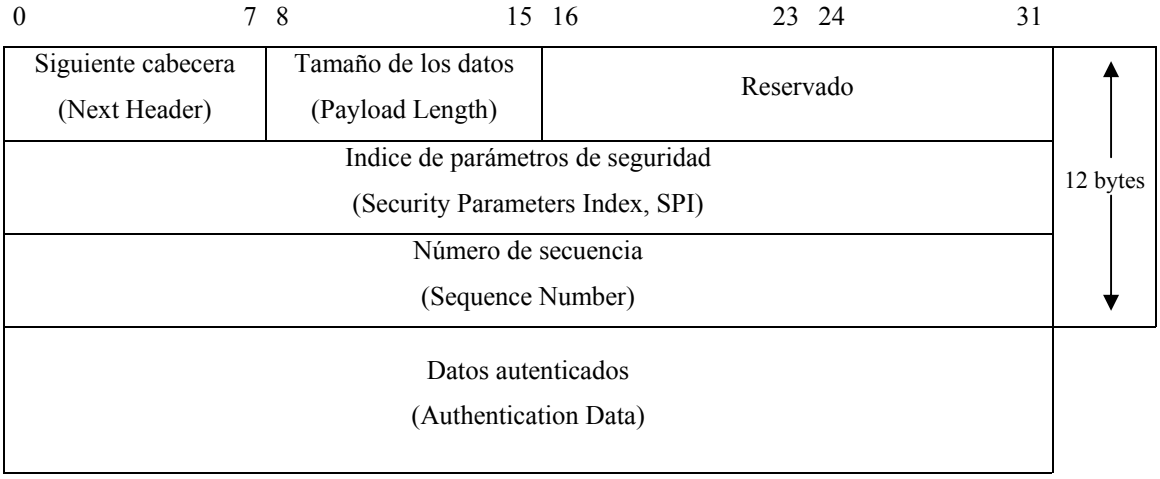


FIG. 5-1: Esquema de la cabecera de autenticación (AH).

El **tamaño de los datos** (Payload Length) especifica la longitud de los datos en palabras de 32 bits (4 bytes).

El **índice de parámetros de seguridad** (SPI) es un número de 32 bits, lo que nos permite tener hasta 2^32 conexiones de IPSec activas en un mismo ordenador.

El **número de secuencia** (Sequence Number) identifica en número del datagrama en la comunicación, estableciendo un orden y evitando problemas de entrega de datagramas fuera de orden o ataques externos mediante la reutilización (Replay Attacks) de datagramas (ver capítulo 3).

Los **datos autenticados** (Authentication Data) se obtienen realizando operaciones (depende del algoritmo de cifrar escogido) entre algunos campos de la cabecera IP, la clave secreta que comparten emisor y receptor y los datos enviados.

El principal problema al autenticar un datagrama es que algunos campos son modificados por los routers intermedios (como el alcance del datagrama, que se va decrementando en una unidad cada vez que pasa por un router para evitar bucles infinitos), esto hace imposible poder autenticar todo el datagrama, ya que durante su camino por INTERNET es modificado. El cálculo de los datos autenticados se realiza mediante un algoritmo de Hash (actualmente se sugiere el algoritmo MD5 que calcula un checksum de 128 bits, ver capítulo 3).

5.2.2 La cabecera de cifrado de seguridad (ESP)

La cabecera de autenticación (AH) no modifica los datos que transporta, circulando el texto en claro (Clear Text), simplemente les añade autenticidad (al origen y al contenido). De esta forma, los datos que circulan pueden ser interceptados y visualizados por un eventual atacante. Esto puede sernos útil por ejemplo cuando consultamos un documento oficial (BOE o las bases de unas oposiciones en la universidad...) ya que debe ser público y no tiene sentido cifrarlo, aunque si es básico que sea auténtico.

En el caso de necesitar confidencialidad (por ejemplo en consultas a un banco, no interesa que una tercera persona tenga acceso a nuestro saldo) se debe utilizar la **cabecera de cifrado de seguridad** (Encrypted Security Payload, **ESP**).

La **cabecera de cifrado de seguridad** (ver figuras 5-2 y 5-3) es siempre la última en el sistema de cabeceras en cadena (*Daisy Chain*). Esto es debido a que a partir de ella todo los datos vienen cifrados, con lo que los routers intermedios no podrían procesar las cabeceras posteriores.

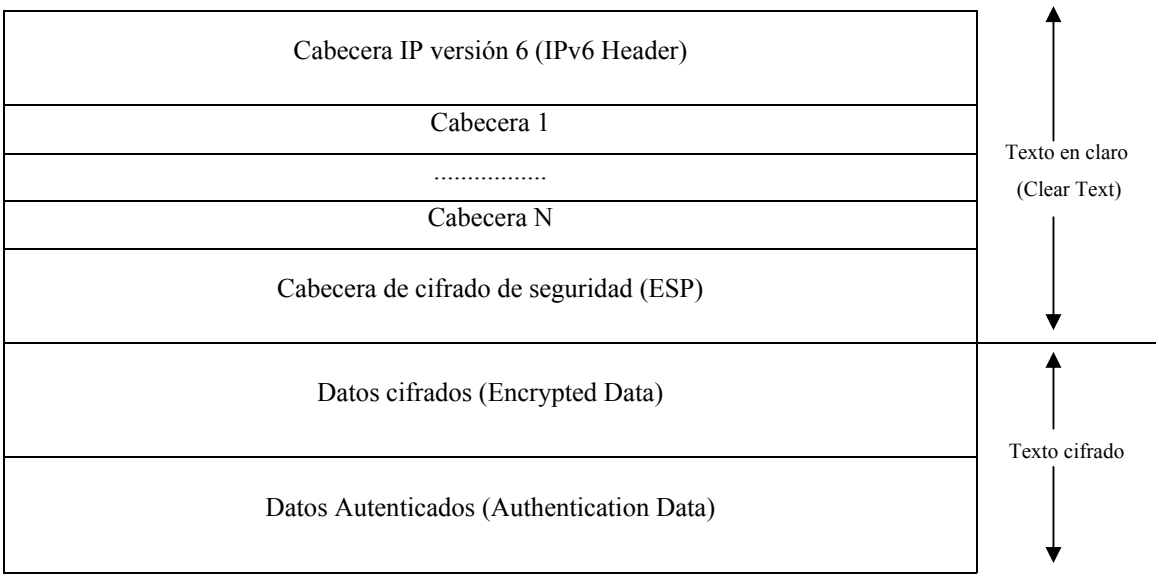


FIG. 5-2: Situación de la cabecera de cifrado de seguridad (ESP).

Al igual en con la cabecera de autenticación (AH), el algoritmo a utilizar se negocia con el receptor de la información antes de enviar un datagrama cifrado. Actualmente se propone e DES-CBC que es el algoritmo DES funcionado el modo de bloque CBC (ver capítulo 3).

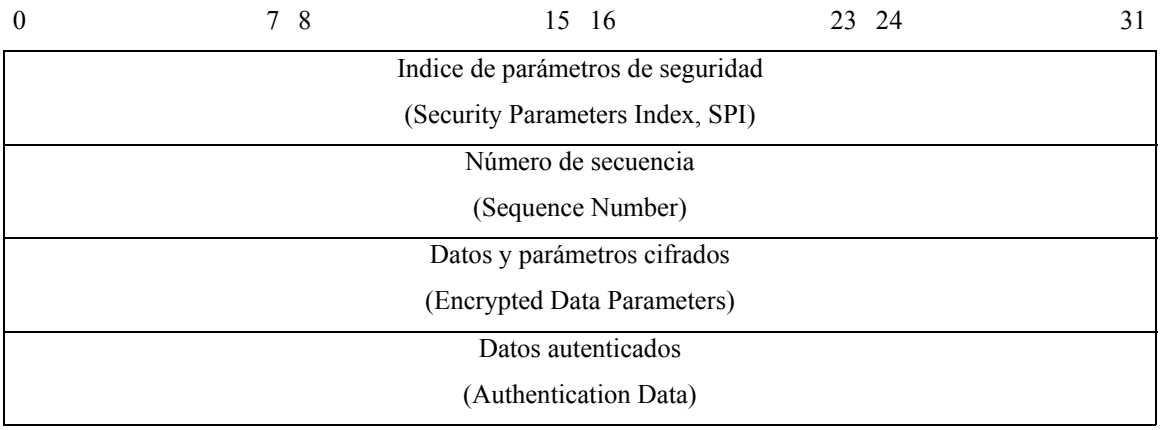


FIG. 5-3: Esquema de la cabecera de cifrado de seguridad (ESP).

A diferencia de la cabecera de autenticación (AH) no es necesario especificar el tamaño de los datos cifrados, ya que a partir de la cabecera de cifrado hasta el final del datagrama todo esta cifrado

El **índice de parámetros de seguridad** (SPI) y el **número de secuencia** (Sequence Number) tienen el mismo significado que en la cabecera de autenticación (AH).

Los **datos autenticados** (Authentication Data) aseguran que el texto cifrado no ha sido modificado utilizando un algoritmo de Hash (depende del algoritmo de cifrar escogido).

Debido a que tanto la cabecera de autenticación (AH) como la cabecera de cifrado de seguridad (ESP) pueden ser utilizadas independientemente, se recomienda que en el caso de ser necesario tanto la autenticidad como la privacidad se incluya la cabecera de cifrado tras la de autenticación. De esta forma autenticamos los datos cifrados.

5.2.3 El protocolo ISAKMP

El protocolo **ISAKMP** (INTERNET Security Association Key Management Protocol) parece ser el escogido para el intercambio de claves y parámetros de seguridad en IPSec. No obstante, debido a que aún se encuentra en fase experimental, no se puede asegurar que finalmente este sea el elegido, ya que varios algoritmos han sido propuestos durante los últimos años (SKIP, Phouturis, Oakley...).

ISAKMP es un protocolo que proporciona la infraestructura necesaria para la negociación de asociaciones de seguridad (SA) entre dos usuarios cualesquiera (ver figura 5-4). Definiremos una **transacción de configuración** (Configuration Transaction) como un doble intercambio dónde el emisor realiza un envío/petición (Set/Request) y el receptor contesta mediante un reconocimiento de petición/respuesta (Acknowledge/Reply).

De esta forma a un envío (Set) le corresponde un reconocimiento de envío (Acknowledge) y a una petición (Request) una respuesta (Reply).

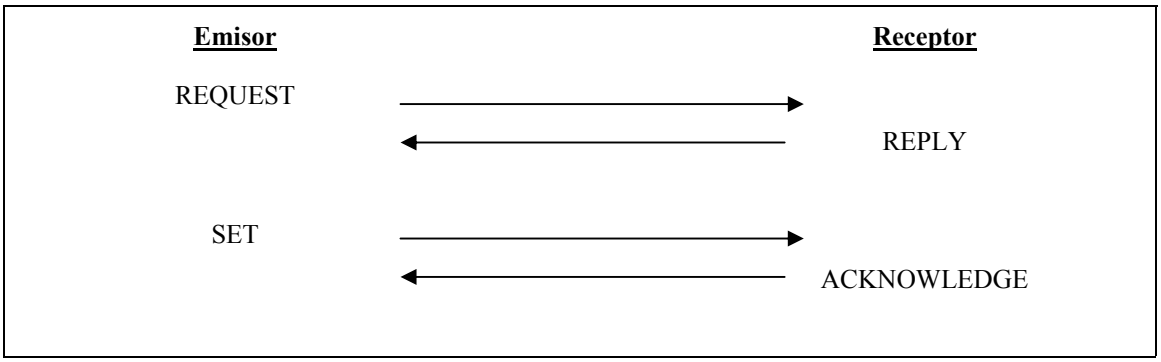


FIG. 5-4: Esquema de una transacción de configuración.

El inicio de la comunicación siempre es precedido de una transacción de configuración dónde se intercambian dos *cookies* (Request/Reply). Este esquema permite evitar ataques de denegación de servicio (DOS) ya que hasta que no recibamos la respuesta el esquema no continua. Posteriormente se producen los intercambios de información necesarios mediante envíos/reconocimientos de envío (Set/Acknowledge) dónde se negocian los diferentes parámetros de seguridad (SPI, clave común, tiempo de validez de la clave, algoritmo de cifrado a utilizar...) que gobernaran la comunicación.

El intercambio de mensajes mediante ISAKMP se realiza mediante el esquema de cabeceras de extensión (ver figura 5-5) ya utilizado en la definición del protocolo IP versión 6.

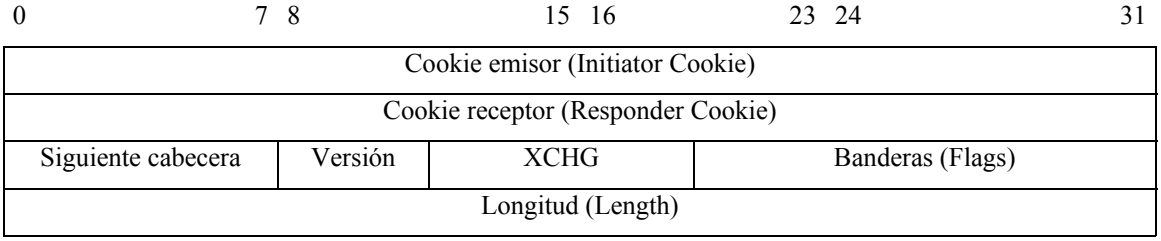


FIG 5-5: Formato de la cabecera del ISAKMP.

El intercambio de claves entre el emisor y el receptor se realiza utilizando el algoritmo de Diffie-Hellman (ver capítulo 3). En el caso de direcciones multicast (varios emisores/receptores en una misma comunicación) el algoritmo anterior resulta ineficiente, ya que está pensado para un emisor y un receptor. La solución adoptada es la de confiar en ordenadores servidores de claves.

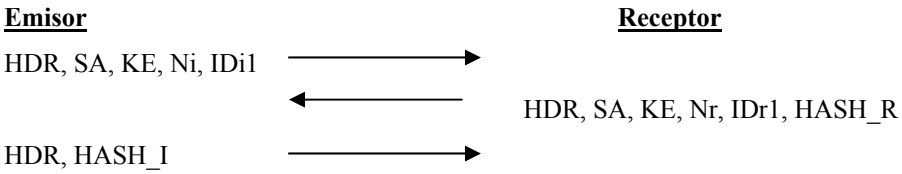
5.2.4 El protocolo IKE

El protocolo **IKE** (INTERNET Key Exchange) es un protocolo de dos fases para el establecimiento de un canal auténtico y seguro entre dos usuarios (Peers). Este protocolo utiliza la infraestructura de mensajes del protocolo ISAKMP para el intercambio de mensajes.

- **Fase1:** Se negocian las asociaciones de seguridad (SA). Se utiliza el protocolo Diffie-Hellman para el intercambio de una clave común y se establece el algoritmo de cifrado (3DES-CBC...), el algoritmo de Hash (MD5...) y del sistema de autenticación. En esta fase tanto el emisor como el receptor quedan autenticados mediante uno de los siguientes cuatro métodos:

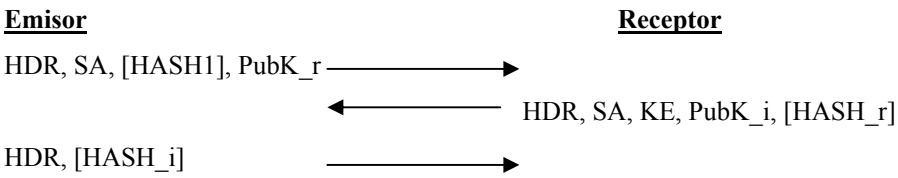
HDR: Cabecera ISAKMP.	HDR*: Cabecera cifrada
HASH: Función Hash.	KE: Valor público Diffie-Hellman.
SA: Asociación de seguridad.	Ni y Nr: Valor temporal (Nonce payload).
[Cert] y SIG: Certificado y firma digital.	ID: Identificador.
PubK: Clave pública.	[ ] : Opcional.

1. Autenticación con claves pre-compartidas (Pre-shared Keys).

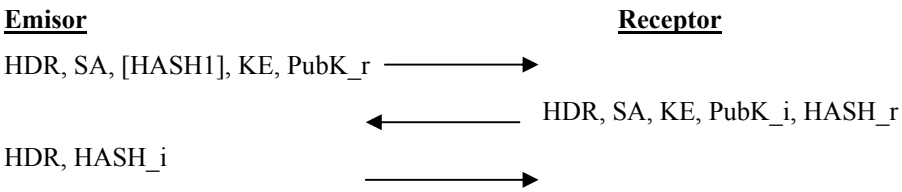


2. Autenticación mediante firmas digitales (Digital Signatures).

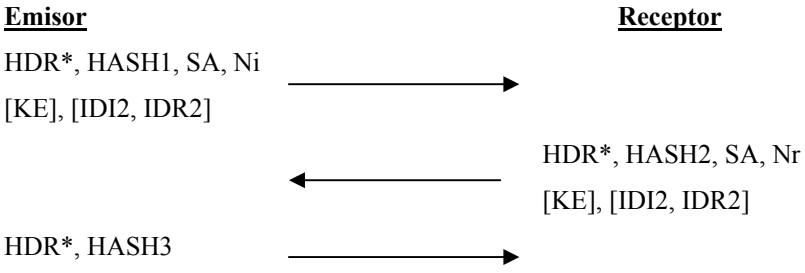
3. Autenticación mediante clave pública 1.



4. Autenticación mediante clave pública 2.



- **Fase 2:** Una vez establecidos los distintos parámetros iniciales (SA) y aprovechando la seguridad de la fase 1, se inicia un modo rápido (Quick Mode) dónde se vuelven a negociar asociaciones de seguridad (SA) con el objetivo de evitar ataques de reutilización (Replay) de los datagramas de la fase 1 por un atacante.



Esta combinación de algoritmos permite mantener una comunicación auténtica y privada entre dos usuarios (Peers), el problema principal radica en su complejidad, ya que pese a ser muy flexible (permite distintos métodos de autenticación y utilización de firmas digitales) es difícil su implementación práctica.

Además deja sin resolver el problema de comunicaciones seguras entre varios usuarios, ya que realizar este algoritmo entre todos ellos resulta en un alto coste de intercambio de datagramas. De esta forma para grupos (multicast o anycast) se debe utilizar un esquema dónde un servidor de claves (que debemos suponer seguro) sincroniza la clave común a todos los componentes del grupo.

5.3 Posibilidades y aplicaciones de IPSec

Las especificaciones IPSec tienen una gran versatilidad que les permite ser utilizadas en las distintas soluciones adoptadas actualmente en INTERNET (comunicación entre distintos cortafuegos (*Firewalls)*, configuración de ordenadores móviles...). El procedimiento de autenticación (fase1) permite que junto al protocolo de vecindad (Neighbor Discovery Protocol) se puedan asegurar intercambios seguros entre los distintos routers, evitando la intercepción de los datagramas.

Una de las soluciones más adoptadas actualmente para la implementación de la seguridad en INTERNET es el uso de Firewalls (ver figura 5-6). Este esquema de actuación consiste en no permitir un acceso directo de los ordenadores a INTENET, colocando una máquina intermedia (denominada cortafuegos o Firewall) que mediante un sencillo conjunto de reglas (dejar pasar los datagramas que vienen de la dirección A, no aceptar datagramas de las direcciones B y C, no aceptar datagramas que vayan al puerto X...) filtra todo el tráfico de INTERNET (entrante y saliente).

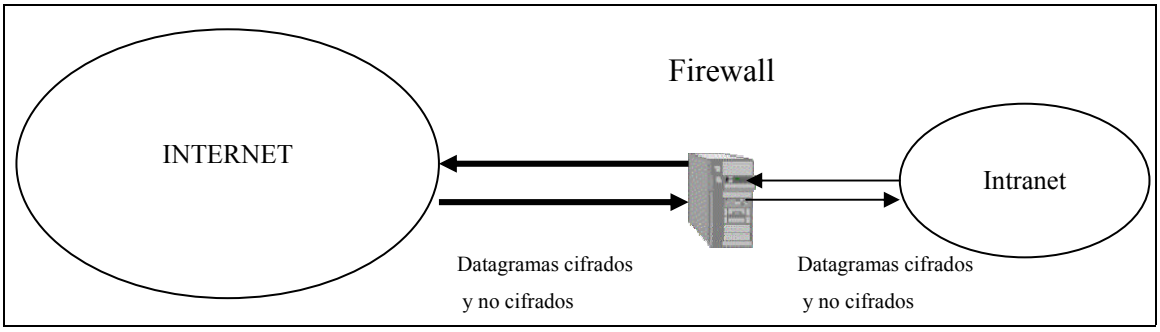


FIG. 5-6: Esquema de seguridad basado en un firewall.

La nueva configuración que se propone con la ayuda de las especificaciones IPSec consiste en realizar un túnel virtual seguro (Secure Tunnel) de forma que dos firewalls estén virtualmente conectados a través de INTERNET de una forma transparente para los usuarios (ver figura 5-7). De esta forma, el intercambio de información vendrá regulado por una comunicación entre los dos firewalls mediante datagramas IP versión 6 encapsulados en datagramas IP versión 6 autenticados (y cifrados si se requiere privacidad).

Las comunicaciones entre dos organizaciones (supongamos para nuestro ejemplo el MIT y la UAB) son realizadas de forma transparente y segura a través de los firewalls. Cuando un ordenador de la UAB desea conectarse a uno del MIT, envía el datagrama correspondiente al firewall. Este se encarga de encapsularlo en un datagrama auténtico (y cifrado si se desea privacidad) y enviarlo al otro extremo del túnel por INTERNET. Al recibir el firewall del MIT este datagrama, comprueba su autenticidad, lo descifra (si es necesario), lo desencapsula y lo envía al ordenador correspondiente.

De esta forma tan sencilla podemos proporcionar un canal seguro y auténtico entre dos puntos cualesquiera conectados a INTERNET. Esta configuración también es conocida como red privada virtual (Virtual Private Network, VPN).

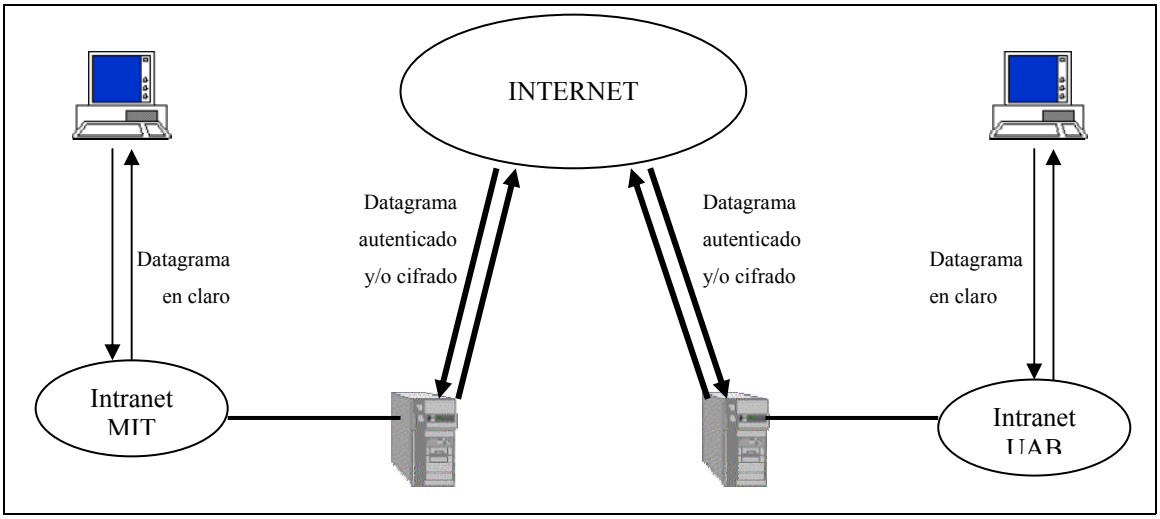


FIG 5-7: Esquema de seguridad proporcionado por IPSec basado en VPN.

5.4 Resumen

En este capítulo hemos analizado la necesidad de seguridad en las comunicaciones realizadas por INTERNET. Las distintas características necesarias (autenticidad y privacidad) para poder proporcionarla y los distintos niveles del esquema de comunicaciones de INTERNET (TCP/IP) dónde estos podían ser implementados.

Con la adopción de medidas de seguridad en la capa IP, se consigue proporcionar de manera independiente y totalmente transparente al usuario medidas de seguridad a todas las capas superiores (TCP, UDP...) sin ningún coste adicional.

IPSec es el conjunto de algoritmos que proporcionan autenticidad y/o confidencialidad a los datagramas IP que circulan por INTERNET. Se utiliza un protocolo de dos fases para el intercambio de claves denominado IKE, que aprovechando la infraestructura de intercambio de mensajes del protocolo ISAKMP proporciona un canal autentico y seguro entre dos usuarios cualesquiera conectados a INTERNET.

Finalmente se comentan algunos aspectos prácticos de conexiones seguras mediante la utilización de firewalls y cómo el esquema IPSec de redes privadas virtuales (VPN) la facilita y mejora.

CAPITULO 6

Experimentos realizados

6.1 Pruebas del capítulo 3

Las pruebas realizadas sobre este capítulo han consistido principalmente en el uso de autoridades de certificación (UAB) y la comprobación del sistema de intercambio de claves SKIP (Simple Key-management for INTERNET Protocols).

- **SKIP** (ver referencia [WWW28]) es un protocolo propiedad de la empresa SUN Microsystems que se basa en que cada dirección IP (la del origen y la del destino) tengan una clave pública (se requiere una autoridad de certificación dónde se proporcionan las claves públicas) y una privada, de esta forma y mediante el esquema de intercambio de Diffie-Hellman se consigue una clave común entre emisor y receptor.

Esta clave pública puede estar validada por diferentes métodos, como los certificados X.509 o el programa de seguridad de correo electrónico PGP. Para nuestras pruebas se utilizó en primera instancia la versión de SKIP para Solaris SUN 2.7 (no se pudo probar esta configuración, ya que después de conseguirla instalar fue imposible hacerla funcionar o desinstalarla, con lo que se tuvo que reconfigurar toda la máquina) y una adaptación para LINUX denominada EnSKIP (v.0.67).

A continuación se adjuntan algunos resultados obtenidos mediante el uso de diferentes algoritmos en ipsis2.uab.es (Pentium 90 con LINUX, Kernel 2.0.36 y 16MB de RAM):

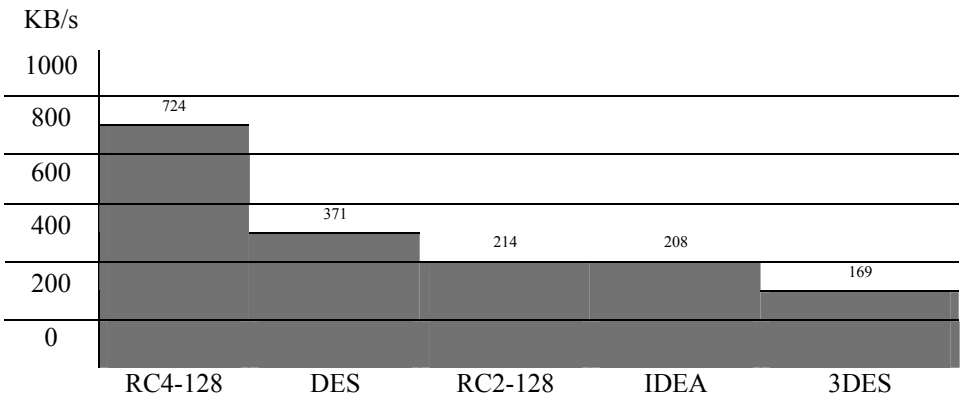


FIG. 3.6: Tabla comparativa de cifrado en un Pentium 90 y loopback.

Desgraciadamente, este protocolo así como los programas que lo implementan han sido abandonados por SUN, con lo que están obsoletos (por ejemplo EnSKIP sólo funciona con versiones obsoletas de Kernels 2.0.20 a 2.0.36).

- **Autoridad de certificación** de la UAB. En la UAB se están realizando pruebas para la implantación de una autoridad de certificación que pueda ser utilizada por profesores y alumnos. Para las pruebas se utilizó el laboratorio del departamento de CCD que tiene instalado un servidor de certificados de la empresa NETSCAPE [WWW23] y la versión 4.07 del NETSCAPE Navigator instalado en ipsis1.uab.es (Pentium 150 , con sistema operativo LINUX, Kernel 2.2.6 y 32MB de RAM). Las pruebas

consistieron en la demanda de un certificado al administrador (Sergi Robles) que fue utilizado para diferentes transacciones dentro de la UAB (recordar que en el momento de la realización de estas pruebas, los certificados tan sólo son válido en la UAB).

6.2 Pruebas del capítulo 4

Todo y que la versión 6 del protocolo IP es aún experimental y quedan varios años para que pueda ser utilizada normalmente en INTERNET, se fijó en los objetivos iniciales de este proyecto que no sería tan sólo un profundo estudio teórico del protocolo, sino que se llevarían a la práctica los conocimientos previos adquiridos. Los experimentos que se describen a continuación han sido posibles gracias al CCD de la UAB, Centro de Cálculo de la UAB, Centro de Supercomputación (CESCA), REDIRIS y 3COM (USA).

6.2.1 IP versión 6 en la UAB

Para el desarrollo de las diferentes pruebas se dispuso de dos ordenadores (*ipsis1.uab.es* e *ipsis2.uab.es*) conectados a la red de la UAB con sistema operativo LINUX. A partir de aquí se creó un acceso a INTERNET utilizando la infraestructura ya existente del centro de cálculo de la UAB y REDIRIS. Posteriormente se habló con los diferentes organismos (centro de cálculo de la UAB, CESCA y REDIRIS) para la obtención de una dirección IPv6.

Una vez gestionado los diferentes trámites administrativos se procedió a la creación de un túnel virtual que conectase la máquina ipsis2.uab.es al CESCA y a la red 6-BONE (red experimental que utiliza IP versión 6 encapsulado en IP versión 4 para su funcionamiento) según el esquema descrito en la figura 4-18.

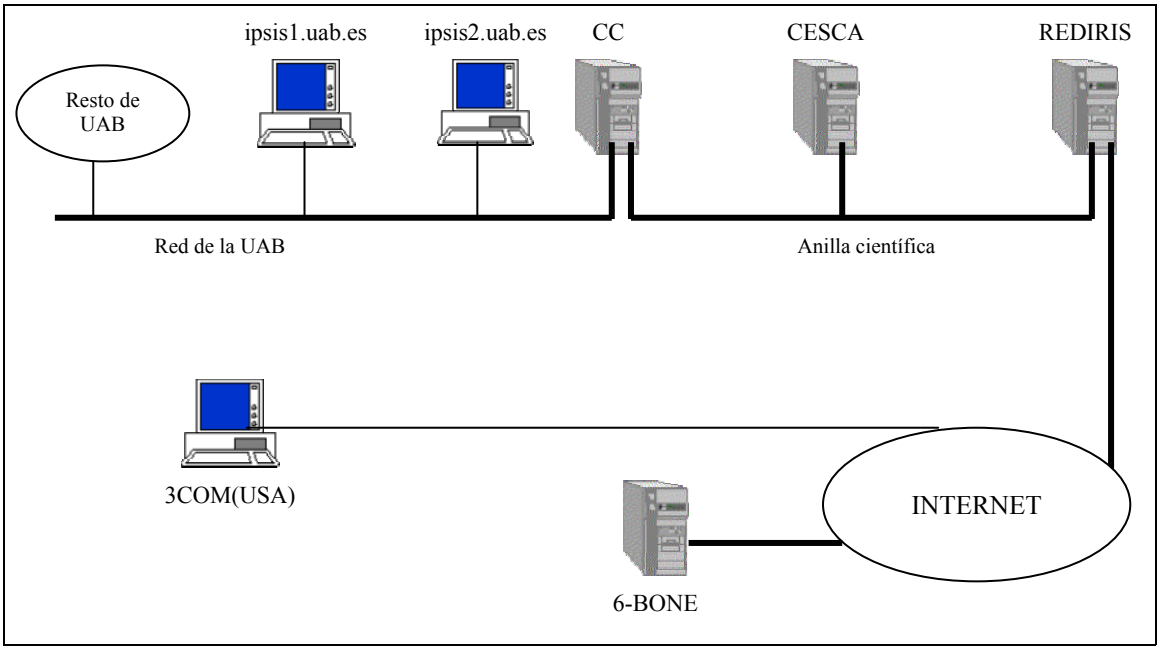


FIG. 4-18: Esquema de conexión a INTERNET y al 6-BONE.

Hardware utilizado en ese proyecto:

- Un ordenador Pentium 75 con 16 MB de RAM, 400 MB de disco duro y tarjeta de red (ipsis1.uab.es, 158.109.10.88).

- Un ordenador Pentium 150 con 32 MB de RAM, 1024 MB de disco duro y tarjeta de red (ipsis2.uab.es, 158.109.10.89).

Software utilizado en este proyecto:

- El sistema operativo seleccionado para las pruebas fue LINUX. La decisión viene dada porque es un sistema gratuito, tipo UNIX (lo que le confiere gran robustez), los códigos fuentes están disponibles (lo cual permite realizar modificaciones en cualquier parte del sistema) y está muy extendido en la comunidad científica y también se encuentran disponibles para LINUX la mayoría de aplicaciones realizadas para IP versión 6.

- Se han utilizado diferentes distribuciones de LINUX durante la duración del proyecto debido a que conforme avanzaba el tiempo estas iban proporcionando mas soporte e infraestructuras a la versión 6 del protocolo IP. Destacamos Slackware 3.0, Slackware 3.1, Debian 2.0, Red Hat 6.0 y Red Hat 6.1.

- Así mismo se han utilizado y modificado las diferentes versiones del *Kernel* de este sistema operativo que han ido apareciendo en el transcurso de la realización de este proyecto:

    1. Versiones **estables**: 2.0.36, 2.2.0, 2.2.4, 2.2.5, 2.2.6, 2.2.7, 2.2.10, 2.2.11, 2.2.12, 2.2.12-20 y 2.2.13.

    2. Versiones **experimentales**: 2.1.90, 2.1.91, 2.1.111, 2.1.112, 2.1.113, 2.1.115, 2.3.0, 2.3.2, 2.3.10, 2.3.14, 2.3.15 y 2.3.30.

- A continuación se detalla una lista con las aplicaciones utilizadas en las diferentes pruebas y que permitieron junto con las modificaciones realizadas en los Kernels tener dos ordenadores en la UAB utilizando IP versión 6.

    1. **Glibc** versión 2.1: Librerías del sistema básicas para la compilación de las aplicaciones basadas en la versión 6 del protocolo IP.

    2. **Libpcap** versión 0.46: Librerías del sistema básicas para la compilación de las aplicaciones basadas en la versión 6.

    3. **Inet6-apps** versión 0.36: Conjunto de aplicaciones básicas (FTP, TFTP...) y sus servidores (*Daemon*) adaptadas a la versión 6.

    4. **Net-tools** versiones 1.50, 1.52 y1.53: Aplicaciones y utilidades necesarias para la compatibilidad entre las versiones 4 y 6 de IP.

    5. **TCPdump** versión 3.4: Permite examinar los diferentes datagramas que circulan por la red (cabeceras y contenido).

6. **TELNET** versión 95.10.23: Aplicación de TELNET compatible con la versión 6.

7. **Traceroute** versión 1.4a5: Aplicación traceroute que permite examinar los diferentes caminos que toma un datagrama hasta llegar a su destino.

Finalmente comentar también la existencia de un servidor WWW (**apache** versión 1.3.6) y un navegador (**chimera** versión 2.0a14) compatibles con la versión 6 de IP.

6.2.2 Configuración de un ordenador para IP versión 6

La transformación de un ordenador compatible con la versión 4 de IP en un ordenador que utilice la versión 6, requiere una serie de cambios y modificaciones tanto en el kernel como en las aplicaciones que lo utilizan (Telnet, FTP...). Los cambios varían según el sistema operativo elegido (FreeBSD, Windows, LINUX...) con lo cual los pasos que se explican a continuación son para sistemas operativos compatibles con LINUX. La mayoría de los programas mencionados se obtuvieron de la bibliografía [FTP1] y [FTP8].

El primer paso es conseguir unas **librerías GLIBC versión 2.1** o superior y una versión de las librerías **libpcap** versión 0.46 o superior compatible con la versión 6 de IP. Algunas distribuciones recientes como Red Hat 6.1 ya las incorporan como parte del sistema operativo. Son esenciales puesto que las demás aplicaciones las utilizan y su compilación depende de estas.

El segundo paso constituye la **instalación de un kernel versión 2.2 o superior** (las versiones 2.1.115 y superiores también incluyen soporte IP versión 6, aunque suelen ser muy inestables). Este paso es básico ya que las versiones anteriores no incluyen soporte IP versión 6.

El tercer paso es la **reconfiguración del kernel** (C*ustomization*) para adecuar los servicios de red que gestiona a la versión 6 de IP. Además de las opciones específicas que necesitemos en nuestra máquina (Tarjeta de red, discos SCSI, sistemas de ficheros...) hemos de especificar las opciones de utilización de la versión 6. A continuación se especifican muy brevemente las opciones mínimas que es necesario activar:

- Compilación de soporte experimental (Prompt for experimental software).

- En el módulo de red (Network), activar el soporte *netlink* así como las opciones de IP versión 6. Todo y que estas opciones pueden ser compiladas como módulos separados del kernel (lo cual permite su carga y descarga en memoria siempre que lo deseemos) se recomienda no utilizar esta opción con versiones inferiores a la versión 2.2.12 debido a problemas en el código.

El cuarto paso después de la recompilación, instalación e inicialización del ordenador con el kernel compatible con la versión 6 de IP es la **compilación e instalación de las aplicaciones básicas** modificadas para utilizar la versión 6 de IP.

- Este punto es totalmente dependiente del sistema operativo escogido e incluso de la distribución seleccionada. Debido al carácter experimental de las aplicaciones y al poco soporte proporcionado por los creadores (cuando este soporte existe), las modificaciones del código han de ser realizadas "a mano" por el propio usuario, debiendo conocer perfectamente las funciones y protocolos utilizados. Las aplicaciones que se han modificado en este proyecto y/o aplicarles algún parche de [WWW27] y [FTP1] son:

  1. Apache (servidor WWW). No se consiguió hacerlo funcionar en Linux.
  2. Inet6-apps. Adaptación del código IPv6 a la distribución Red Hat.
  3. Libpcap. Modificación de los ficheros de cabecera (.h) y el Makefile.
  4. Traceroute. Adaptación a Linux. Programa escrito para FreeBSD.
  5. Tcpdump. Aplicación de parches.

El quinto y último paso consistió en la **instalación de los enlaces con el exterior** (CESCA, 6-Bone, 3-COM...) y la **asignación de las direcciones IP versión 6 asignadas a la UAB**.

- Debido al carácter concreto de estas pruebas, el CESCA asignó una única dirección IP versión 6 (3FFE:3326:FFFF:FFFF:FFFF:FFFF:FFFF:A/126) a la UAB. Esta dirección fue asignada a la máquina ipsis1.uab.es que actuó como router para las pruebas.

- Se realizaron dos túneles lógicos (encapsulación de datagramas de IP versión 6 en datagramas IP versión 4) que partían de ipsis2.uab.es:

  UAB (ipsis2.uab.es) ⟷ CESCA ⟷ 6-Bone

  UAB (ipsis2.uab.es) ⟷ 3COM (USA)

6.2.3 Pruebas realizadas

Los experimentos que se han realizado referentes a este apartado pueden clasificarse principalmente como **internos** (dentro de la red de la UAB) y **externos** (con el CESCA cortesía de Caterina Parals y 3COM USA cortesía de Somun Mathur.).

- **Internos:** Utilización de los programas TELNET, FTP y PING adaptados al protocolo IP versión 6, entre ipsis1.yab.es e ipsis2.uab.es.

  1. Creación de una red lógica IPv6 en la UAB utilizando encapsulación de datagramas IPv6 en datagramas IPv4. A esta red se conectaron los ordenadores ipsis1.uab.es (3FFE::2) e ipsis2.uab.es (3FFE::1).

  2. Configuración del programa de monitorización **tcpdump+ipv6** que permite la visualización en pantalla de los datagramas que circulan por la red así como su contenido.

3. Establecimiento de conexiones entre ipsis1 e ipsis2 mediante el protocolo UDP utilizando la aplicación **ping** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

4. Establecimiento de conexiones entre ipsis1 e ipsis2 mediante el protocolo TCP utilizando la aplicación **TELNET** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

5. Establecimiento de conexiones entre ipsis1 e ipsis2 mediante el protocolo TCP utilizando la aplicación **FTP** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

- **Externos:** Utilización de los programas TELNET, TRACEROUTE y PING adaptados al protocolo IP versión 6.

1. Creación del túnel lógico T_CESCA entre el ordenador ipsis2.uab.es (3FFE:FFFF:FFFF:FFFF:FFFF:FFFF:A) y quermany.cesca.es (3FFE:FFFF:FFFF:FFFF:FFFF:FFFF:9).

2. Creación del túnel lógico T_3COM entre el ordenador ipsis2.uab.es (3FFE:FFFF:FFFF:FFFF:FFFF:FFFF:A) y ipv6.usa.3com.com (3FFE:2000:4662::23).

3. Establecimiento de conexiones entre ipsis2 y quermany.cesca.es mediante el protocolo UDP utilizando la aplicación **ping** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

4. Establecimiento de conexiones entre ipsis2 y quermany.cesca.es mediante el protocolo UDP utilizando la aplicación **traceroute** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

5. Establecimiento de conexiones entre ipsis2 e ipv6.usa.3com.com mediante el protocolo UDP utilizando la aplicación **ping** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

6. Establecimiento de conexiones entre ipsis2 e ipv6.usa.3com.com mediante el protocolo UDP utilizando la aplicación **traceroute** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

7. Establecimiento de conexiones entre ipsis2 e ipv6.usa.3com.com mediante el protocolo TCP utilizando la aplicación **TELNET** para IPv6. Monitorización de los distintos datagramas enviados y recibidos entre ambos ordenadores mediante tcpdump.

CAPITULO 7

Conclusiones

Al iniciar este proyecto se establecían como objetivos principales el estudio de las especificaciones del protocolo IP versión 6 y las extensiones de seguridad IPSec que le acompañan. Así mismo también se pretendía la implementación práctica de estos protocolos en la plataforma LINUX.

Conforme fue avanzando el proyecto los objetivos tuvieron que ser ampliados, ya que debido a la gran complejidad y amplitud de aspectos cubiertos por estas especificaciones, resultaban imposibles de estudiar de forma aislada.

Primero se empezó por un estudio minucioso de la versión 4 del protocolo IP y los protocolos de nivel superior TCP y UDP. Esto era básico ya que la versión 6 del protocolo IP es una evolución de la versión 4, y por lo tanto está basada en potenciar sus mejores aspectos y minimizar sus inconvenientes. Además era necesario conocer la

evolución histórica de INTERNET para conocer cuales son sus necesidades actuales y preveer las futuras, evitando caer en los errores pasados.

Una vez profundizado en las especificaciones de la versión 4, se pasó al estudio de las especificaciones de la versión 6. En este punto empezaron a surgir las dificultades debidas principalmente al carácter provisional de los documentos de especificaciones de la versión 6. Estos fueron alterados varias veces en el transcurso de la realización de este proyecto, obligando en algunos casos a empezar de nuevo el estudio de diferentes puntos (cabeceras de extensión, ICMP, direccionamiento y encaminamiento).

Los cambios realizados tanto en los documentos de especificaciones como en las implementaciones software (Kernel de LINUX), obligaron a la utilización y recompilación de diferentes versiones de Kernels experimentales hasta encontrar una estabilidad en las comunicaciones realizadas bajo IP versión 6.

De esta primera parte podemos extraer que las especificaciones de la versión 6 de IP son muy beneficiosas para mejorar la velocidad de encaminamiento de los datagramas y mejorar el rendimiento global de INTERNET. La adopción tanto de una cabecera como de unos campos de longitud fija, permiten un procesamiento más rápido y eficiente de los routers, acelerando el proceso de encaminamiento por INTERNET. Además el esquema de cabeceras de extensión también ayuda a la flexibilidad del protocolo, permitiendo especificar opciones tanto al destino como a los routers intermedios.

Así mismo, el sistema de direccionamiento escogido (128 bits que se subdividen en los grupos unicast, multicast y unicast) parece proporcionar la infraestructura necesaria para las futuras necesidades de los usuarios de INTERNET (Videoconferencia, Game on-line...), aunque este aspecto no ha sido posible probarlo debido a que todas las pruebas se realizan bajo IP versión 4 (encapsulamiento de datagramas IPv6 en datagramas IPv4).

En cuanto a la seguridad proporcionada por la versión 6 y debido a su gran complejidad, pasó a ser estudiada en una segunda fase a posteriori de los apartados anteriores. Al igual que pasó con las especificaciones IP versión 4, se decidió aumentar el trabajo

incluyendo un breve estudio de las soluciones de seguridad que actualmente se utilizan sobre el esquema TCP en la versión 4 del protocolo IP.

La inclusión de un apartado dedicado a la seguridad en IPv4, se fundamenta principalmente en dar una perspectiva de la seguridad en INTERNET, conociendo los peligros potenciales, las necesidades reales, las soluciones y los algoritmos aplicados. Además gran parte de los algoritmos utilizados en IPSec son comunes a algunas soluciones ya adoptadas en TCP.

Las soluciones adoptadas en las capas superiores (SSL, SET...) tienen el gran inconveniente de que al no pertenecer a la definición del protocolo, son implementadas en el nivel de aplicación, dependiendo totalmente de la implementación de software que realice el proveedor del servicio. Además como no son universales, no todos los usuarios las contemplan e incluso pueden proporcionar la suya propia.

IPSec fue pensada para evitar todos los problemas anteriores, debido a que ya formaría parte de la versión 6 del protocolo IP, debiendo por lo tanto de ser soportada por todas las implementaciones de software. Además este esquema efectúa de forma mucho más óptima que la anterior, debido a que se implementa de forma totalmente transparente al usuario. Como está implementada en la capa inferior del conjunto de protocolos TCP/IP, proporciona seguridad a todas las capas superiores sin un coste extra de proceso.

Los problemas en este apartado llegaron al intentar implementar (total o parcialmente) los aspectos de seguridad en LINUX. Partiendo de unas especificaciones tan complejas como confusas (tanto los RFCs como los Drafts están llenos de contradicciones y explicaciones poco claras) ninguna de las implementaciones probadas llegó a funcionar.

Se buscaron algunas aproximaciones a estos esquemas (como la implementación KAME para el sistema operativo FreeBSD o FreeS/WAN adaptación libre de las especificaciones IPSec para LINUX), pero fue imposible llegar a verlos funcionar debido a la incompatibilidad entre ambas implementaciones.

El problema principal de las ambigüedades en las especificaciones IPSec ha llegado a tal punto, que en el momento de presentar esta memoria, en las listas de la IETF se están alzando voces contra IPSec (propiciadas por Bruce Schneider[9] y Steve Kent) pidiendo un abandono de estas y la creación de unos esquemas nuevos más simples.

Para finalizar estas conclusiones añadimos una reflexión propia sobre los aspectos más relevantes que a nuestro juicio debería contemplar el esquema de seguridad IPSec:

- La **autenticidad** permite tanto al origen como al destino asegurarse de que el datagrama proviene del origen legítimo. Resulta muy útil cuando se desea asegurar la fiabilidad de los datos y no es necesario su secreto (por ejemplo al leer las fechas de los exámenes en la universidad. Su autenticidad es vital, mientras que su confidencialidad no es necesaria).

- La **confidencialidad** permite mantener el secreto de los datos enviados entre un emisor y un receptor, evitando que una tercera persona pueda acceder a la información.

Pese a que el esquema anterior basado en dos aspectos diferenciados es muy flexible y permite autenticidad y/o confidencialidad, nuestra propuesta consistiría en unir autenticidad y confidencialidad en un único concepto, el de comunicación segura. De esta forma **siempre** que se estableciese una comunicación IPv6 entre dos ordenadores esta sería siempre auténtica y segura:

1. Crear unas especificaciones en que por defecto se autentifique el origen del datagrama evitaría muchos tipos de ataques que actualmente son viables en INTERNET (falseado de direcciones o Spoofing, robo de conexiones, denegación de servicio o DOS...). Como el propio protocolo descartaría los datagramas no auténticos, no serían necesarias medidas de seguridad extras para proteger los ordenadores y todos los ordenadores que usaran IPv6 estarían protegidos.

[9] Ver el articulo publicado en **www.counterpage.com/ipsec.pdf**

2. El cifrado de los datos del datagrama por defecto evitaría problemas como la circulación de passwords (TELNET, FTP...) en texto en claro por la red. Además como este cifrado se realiza por el propio protocolo IP de forma transparente al usuario, se proporciona automáticamente secreto a los protocolos de las capas superiores (TDP, UDP...) y se evitarían ataques de pinchado de red (Sniffing).

3. La especificación de un único método que aglutinara seguridad y confidencialidad (no de un único algoritmo, que podría ser escogido entre cualquiera que proporcionara autenticidad y secreto) simplificaría enormemente las especificaciones IPSec, facilitando su implantación, test y uso.

4. Este sistema es totalmente transparente a los routers intermedios, ya que estos tan sólo deberían encaminar los datagramas a su destino sin ningún tipo de penalización para la velocidad de INTERNET. Para el usuario final también sería transparente (evitándole la tediosa configuración de parámetros y opciones necesarias en una comunicación segura y auténtica) ya que sería el propio protocolo el encargado de verificar la validez del los datagramas (origen y los datos que contiene).

5. Actualmente se dispone de ordenadores suficientemente rápidos (mas de 1 Giga-Herzio) como para que la carga de cifrar/descifrar los diferentes datagramas no sea un cuello de botella.

Después de concluir el proyecto sobre IP versión 6 e IPSec hemos podido constatar que los objetivos iniciales eran muy ambiciosos para un único proyecto. No obstante se han conseguido la gran mayoría de objetivos fijados inicialmente en el plazo de un año.

La parte que menos se ha podido desarrollar es la referente a las especificaciones de seguridad (IPSec) debido a la gran cantidad de cambios y rectificaciones que vienen sufriendo sus especificaciones. Igualmente existe una gran falta de grupos de trabajo que implementen estas especificaciones para poder realizar pruebas de compatibilidad entre ellos (algo que sí ocurre con otros aspectos del IPv6).

Es en el campo de la seguridad dónde precisamente se podría profundizar más en un futuro proyecto, debido principalmente a la incertidumbre sobre el mantenimiento o no de las especificaciones IPSec actuales. Igualmente se podría realizar un proyecto que continuara el estudio de los demás aspectos de la versión 6 del protocolo IP, recogiendo las posibles variantes que se produzcan en los siguientes años.

"*Lo escrito permanece*"

Max Aub

Bellaterra, Febrero del 2000.

Gabriel Verdejo Alvarez.

GLOSARIO

Big endian: Sistema de almacenamiento de datos en la memoria del ordenador que consiste en situar los bits en orden ascendente (bit 0, bit 1, bit 2..., bit N).

Cabecera (Header): Información que suele situarse delante de los datos (por ejemplo en una transmisión) y que hace referencia a diferentes aspectos de estos (longitud...).

Capa (Layer): Cada una de los elementos que conforman una estructura jerarquica.

Comercio electrónico (E-comerce): Actividad que consiste en la compra o venta de artículos por INTERNET.

Daemon (Demonio): Proceso especial en los sistemas operativos tipo UNIX caracterizado por ser gobernado por el propio sistema operativo de forma autónoma.

Daisy Chain (Cadena de Margarita): Sistema de enlace de objetos dónde cada objeto contiene un apuntador al siguiente formado una lista.

DARPA: Defense Advanced Research Projects Agency.

Datagrama: Conjunto de estructurado de bytes que forma la unidad básica de comunicación del protocolo IP (en todas sus versiones).

Draft (Borrador): Documento de especificaciones que se expone públicamente para su discusión.

Encaminamiento (Routing): Procedimiento que consiste en conducir un *datagrama* hacia su destino a través de INTERNET.

Encapsulamiento: Sistema basado en colocar una estructura dentro de otra formando capas.

Firewall (Cortafuegos): Máquina encargada del filtrado del tráfico de INTENET (tanto de entrada como de salida) basado en reglas de comportamiento, que se sitúa entre lINTERNET y una Intranet.

Gateway: Ver router.

Half-close: Sistema de finalización de una comunicación establecida con el protocolo TCP.

ICMP (INTERNET Control Message Protocol): *Protocolo* encargado de la comunicación de mensajes entre nodos conectados a INTERNET.

IP (INTERNET PROTOCOL): Protocolo no fiable y sin conexión en el que se basa la comunicación por INTERNET. Su unidad es el *datagrama*.

IPng (IP Next Generation): Abreviatura escogida en IETF [WWW 16] con la que también se denomina la versión 6 del protocolo IP.

IPv6 (IP versión 6): Abreviatura escogida en IETF [WWW 16] con la que se denomina la versión 6 del protocolo IP.

Galactic Network: Ver red galáctica.

Kernel: Conjunto de servicios básicos que debe ofrecer un sistema operativo para poder funcionar.

LAN (Local Area Network): Red local. Es la encargada de conectar ordenadores en distancias inferiores a 1Km.

Little endian: Sistema de almacenamiento de números en la memoria del ordenador, que consiste en situar los bits en orden descendente (bit N, bit N-1, bit N-2...).

MAC Address: Dirección única que llevan las tarjetas de red grabadas en una ROM para identificarse y diferenciarse de las demás.

MIT (Massachussetts Institute of Technology): Universidad americana.

MTU (Maximun Transfer Unit): Siglas que denominan el tamaño máximo en unidades de transmisión que se permite en un canal de comunicación.

OSI (Modelo): Modelo teórico propuesto por IEEE que describe cómo deberían conectarse distintos modelos de ordenadores a diferentes tipos de red para poder comunicarse entre sí.

Overhead: Pérdida de rendimiento.

Paquetes: Ver datagrama.

Pipelining: Sistema consistente en la solapación de tareas (una tras otra) de forma que se mejore el rendimiento.

Periféricos: Cualquier tipo de dispositivo que pueda ser conectado a un ordenador.

Protocolos: Conjunto de reglas que establece cómo debe realizarse una comunicación.

Red: Dispositivo físico que conecta dos o más ordenadores.

RFC (Request For Comments): Documento de especificaciones que se expone públicamente para su discusión.

Router: Dispositivo físico u ordenador que conecta dos o más redes encargado de direccionar los distintos *datagramas* que le lleguen hacia su destino.

Socket: Tupla compuesta por una dirección IP y un número de port.

Socket pair: Pareja *sockets* que permiten definir una comunicación (origen y destino).

SSL (Secure Socket Layer): Protocolo que proporciona seguridad en INTERNET a partir del protocolo *TCP*.

TCP (Transmission Control Protocol): Protocolo de nivel superior que permite una conexión fiable y orientada a conexión mediante el protocolo *IP*.

Three Way Handshake: Protocolo de tres pasos en el que se basa el establecimiento de conexión en el protocolo TCP.

Tunneling: Ver encapsulamiento.

UDP (User Datagram Protocol): Protocolo no fiable y sin conexión basado en el protocolo *IP*.

WAN (Wide Area Network): Red de gran alcance. Este tipo de red suele utilizarse en la unión de redes locales (*LAN*).

BIBLIOGRAFIA

[Kle61] L. Kleinrock, "Information Flow in Large Communication Nets", RLE Quarterly Progress Report, 1961.

[Kle64] L. Kleinrock, "*Communication Nets: Stochastic Message Flow and Delay"*, Mcgraw-Hill, 1964.

[Rob67] L. Roberts, "Multiple Computer Networks and Intercomputer Communication", ACM Gatlinburg Conf., 1967

[Bar64] P. Baran, "On Distributed Communications Networks", *IEEE Trans. Comm. Systems,* 1964.

[DH77] W. Diffie, M. Hellman, "Privacy and authentication", IEEE, 1977.

[Mas88] J. L. Massey, "An introduction to contemporary cryptology", IEEE, 1988.

[Ste90] W. R. Stevens, "UNIX network programming", Prenctice-Hall, 1990.

[RH91] J. Rifà i Coma, LL. Huguet i Rotger, "Comunicación digital", Masson S.A, 1991.

[Cer92] V. Cerf, "How the INTERNET came to be", recopilado junto a otros artículos en [Abo93].

[Abo93] B. Aboba, "The Online user's encyclopedia", Addisson-Wessley, 1993.

[Rie95] A. Riera, "Xarxes d'ordinadors: apunts de teletractament I", publicación interna de la UAB, 1995.

[Rif95] J. Rifà i Coma, "Seguretat Computacional", Materials, Servei de publicacions UAB, 1995.

[Mar96] Fco. Manuel Márquez, "UNIX programación avanzada", Ra-ma, 1996.

[Rus97] D. A. Rusling, "The linux kernel", collección de drafts, 1997.

[Ric98-1] W. Richard Stevens, "TCP/IP Illustrated Volume 1: The protocols", Addison-Wessley, 1998.

[Ric98-2] W. Richard Stevens, "TCP/IP Illustrated Volume 2: The implementation", Addison-Wessley, 1998.

[Ric98-3] W. Richard Stevens, "TCP/IP Illustrated Volume 3: TCP transactions", Addison-Wessley, 1998.

[Hui98] Ch. Huitema, "IPv6:The New Protocol", Prentice-Hall, 1998.

[Sta98] W. Stallings, "Sistemas operativos" (2ª edición), Prenctice-Hall, 1998.

[Tor98] J. Torres Gonfaus, "Atacs de denegació de servei per manipulació de protocols TCP/IP", proyecto fin de carrera, UAB.

[San99] M. A. Sansó Oliver, "Desenvolupament d'assasyn, un monitor de comunicacions per protegir contra l'atac TCP SYN FLOODING", proyecto fin de carrera, UAB.

[WWW1] http://fujiwara-www.cs.titech.ac.jp/~nezz/CRYPT.html

[WWW2] http://home.mcom.com/info/security-doc.html

[WWW3] http://www.6bone.net

[WWW4] http://www.ascom.ch/web/systec/security/idea.htm

[WWW5] http://www.bsa.org/policy/encryption/cryptographers.html

[WWW6] http://www.cern.com

[WWW7] http://www.cert.es

[WWW8] http://www.crypto.com/key_study/report.shtml

[WWW9] http://www.cryptography.com

[WWW10] http://www.cs.hut.fi/crypto

[WWW11] http://www.cs.hut.fi/ssh/crypto/algorithms.html

[WWW12] http://www.cypher.net/pub/clipper/skipjack_interin_report.htm

[WWW13] http://www.geocities.com/SiliconValley/Heights/5265

[WWW14] http://www.iab.org

[WWW15] http://www.iec.csic.es/criptonomicon

[WWW16] http://www.ietf.org

[WWW17] http://www.ipv6.org

[WWW18] http://www.isoc.org/

[WWW19] http://www.isoc.org/zakon/Internet/History/HIT.html

[WWW20] http://www.kriptopolis.com

[WWW21] http://www.mastercard.com/set

[WWW22] http://www.mit.edu/people/mkgray/net/web-growth-summary.html

[WWW23] http://www.netscape.com/newsref/std/SSL.html

[WWW24] http://www.nist.gov/itl/div897/pubs/fip46-2.htm

[WWW25] http://www.nsa.gov:8080/museum

[WWW27] http://www.pjn.gov.ar/rquesada/

[WWW26] http://www.rediris.es/cert/keyserver.html

[WWW27] http://www.rsa.com

[WWW28] http://www.skip.org

[WWW29] http://www.uncitral.org

[WWW30] http://www.visa.com/cgi-bin/vee/sf/set/intro.html

[WWW31] http://www.wto.org

[WWW32] http://www.w3.org/hypertext/WWW/Security/Overview.html

[FTP1] ftp://ftp.bieriger.de/pub/linux/IPv6

[FTP2] ftp://ftp.cnb.uam.es/pub/misc/crypt

[FTP3] ftp://ftp.dit.upm.es/mirror/ftp.ripe.net/rfc/rfc2040.txt

[FTP4] ftp://ftp.funet.fi/pub/gnu/prep

[FTP5] ftp://ftp.kernel.org

[FTP6] ftp://ftp.rediris.es/pub/linux/kernel/sources

[FTP7] ftp://ftp.rediris.es/pub/rfc

[FTP8] ftp://ftp.redhat.com