INDICE

DEDICATORIA

Esta tesis va dedicada con mucho cariño para mis padres, esposa e hijos quienes con su cariño y comprensión me han dado la fuerza suficiente para poder concluir esta meta importante en mi vida.

AGRADECIMIENTO

"Las oportunidades son incontables, sin duda una de ellos es el éxito". agradezco todos los catedráticos de la Universidad Politécnica de Madrid Especialmente a mi esposa e hijos ***Diana, Sebastián y Doménica.***

1. Protocolo IP (versión 4)

1.1 Introducción.

El protocolo IP (Internet Protocol) fue diseñado para interconexión de redes. IP se ocupa de la transmisión de bloques de datos, llamados Datagramas de origen a destino, donde orígenes y destinos son *hosts* identificados por direcciones de una longitud fija.

IP también se encarga de la fragmentación y reensamblado de Datagramas, si éste fuera necesario.

El protocolo IP implementa dos funciones básicas: Direccionamiento y fragmentación.

El módulo Internet usa las direcciones contenidas en la cabecera de los Datagramas para hacer llegar a estos a sus destinos. Así mismo, existen otros campos en la cabecera que permiten gestionar la fragmentación y posterior reensamblado de Datagramas, para poder transmitir a través de redes que trabajen con tamaños de paquete pequeños.

El módulo Internet reside en cada *host* integrado en la Internet, y en cada *gateway* interconectando redes. Estos módulos siguen reglas comunes para interpretar las direcciones y para realizar la fragmentación / reensamblado de Datagramas. Adicionalmente, estos módulos (especialmente en los *gateways)* están provistos de mecanismos para tomar decisiones sobre el enrutamiento de los datagramas.

1.2 Direccionamiento IPv4

1.2.1 Direccionamiento IPv4

Para identificar cada máquina en la Internet, se le asigna un número denominado dirección Internet o dirección IP. Este número es asignado de tal forma que se consigue una gran eficiencia al encaminar paquetes, ya que codifica la información de la red en la que está conectado, además de la identificación del *host* en concreto.

Cada dirección Internet tiene una longitud fija de 32 bits. Los bits de las direcciones IP de todos los host de una red determinada comparten un prefijo común. Conceptualmente, cada dirección IP es una pareja formada por identidad de red-identidad de host, donde la identidad de red identifica a la red, e identidad de host, a un host determinado dentro de esa red.

Para que exista una flexibilidad en la asignación de direcciones, existen tres formatos básicos de representación de direcciones. La elección de uno de estos formatos dependerá del tamaño de la red. Además de los tres formatos básicos, existe uno para *multicasting*, usado para envío de mensajes a un grupo de hosts, y otro reservado para futuro uso.

La estructura de los diferentes formatos es la que sigue :

Clase A

1	Identificador de red	identificador de host
	7 bits	24 bits

Clase B

1	0	Identificador de red	Identificador de host
		14 bits	16 bits

Clase C

1	1	0	Identificador de red	Identificador de host
			21 bits	8 bits

Clase D

1	1	1	0	Dirección multicast
				28 bits

Clase E

1	1	1	1	0	Espacio reservado para futuro uso

Vemos que los primeros bits identifican la clase de dirección IP, que va seguido de un prefijo de identificación de red, y seguido de un identificador de host. La clase D se usa para transmitir un mismo mensaje a un grupo de hosts determinado.

La clase A se usa para grandes redes que tengan más de 2^{16} (65536) hosts. La clase B se usa para redes de tamaño intermedio, entre 2^8 (256) y 2^{16} hosts. Finalmente, la clase C corresponde a redes con menos de 256 hosts. El cuarto tipo, el D, se dedica a tareas de *multicasting*.

Para asegurar que la parte de identificación de red de una dirección Internet es única, todas las direcciones son asignadas por una autoridad central, el Centro de Información de Red (NIC, Network Information Center).

Esta autoridad central tan sólo asigna el prefijo de red de la dirección y delega la responsabilidad de asignar las direcciones de host individuales a la organización solicitante. A las redes de área local con pocos ordenadores (menos de 255) se le asignan direcciones de la clase C, pues se espera que surjan un gran número de ellas. A redes muy grandes, como ARPANET, se les asigna la clase A, ya que se espera que no surjan demasiadas.

A la hora de trabajar con direcciones IP, usamos la notación decimal. La dirección expresada de esta forma vendrá dada por cuatro enteros positivos separados por puntos, donde cada entero se corresponde con el valor de un octeto de la dirección IP.

Según lo comentado, una dirección IP identifica a un host, pero esto no es estrictamente cierto. Por ejemplo, si un *gateway* está conectado a dos redes diferentes, no podemos asignarle una dirección IP única, ya que las dos redes tienen su propia dirección de red. En este caso, hay que asignar una dirección diferente según la conexión, con lo que la dirección IP no especificaría una máquina en particular, sino una conexión a una red.

Según esto, un gateway que conecte 'n' redes tendrá 'n' diferentes dirección IP, según la conexión establecida.

Otra consecuencia es que si un host se mueve de una red a otra, su dirección IP deberá cambiar según la red en la que se encuentre (como ejemplo, podemos imaginar un ordenador portátil).

Como debilidades del protocolo podemos indicar que si una red crece por encima de lo que su clase le permite direccionar (Una red de clase C que crezca por encima de los 255 host) deberá cambiar todas sus direcciones a la clase B, proceso muy costoso y en el que sería muy difícil encontrar errores.

El punto débil más importante de toda la estructura del direccionamiento Internet es que como el encaminamiento usa el prefijo de red de la dirección IP, depende de la dirección usada. Así, pueden existir situaciones en las que no sea posible acceder a un host determinado porque el camino escogido esté cortado temporalmente, pero que sí sería accesible usando otra dirección.

1.2.2 Direcciones especiales

Existen algunas combinaciones de 0's y 1's que no se asignan como dirección IP, sino que tienen asociado un significado especial.

Las distintas combinaciones son las indicadas a continuación :

Todo a 0's

Identifica al propio host.

Todo a 0's	Identificador de host

Identifica al host en su red.

Todo a 1's

Multidifusión limitada en la propia red.

Identificador de red	Todo a 1's

Multidifusión a todos los hosts de la red indicada

Bucle local.

Los dos primeros casos sólo pueden ser usados al arrancar el sistema (p.e. en máquinas sin unidad de almacenamiento fijo) y nunca se usan como una dirección de destino válida. En cualquier caso, sólo se usan de forma temporal mientras el host *aprende* su dirección IP.

El tercer caso es la denominada dirección de multidifusión de red local, o dirección de multidifusión limitada, que permite difundir un mensaje a toda la red local independientemente de su dirección IP asignada. Un host puede usar esta dirección como parte de un procedimiento de comienzo antes de conocer su dirección IP o la dirección IP de su red.

La dirección de multidifusión dirigida a una red nos permite enviar un mensaje a todas las estaciones situadas en una red determinada. Es una herramienta muy potente, ya que permite enviar un sólo paquete que será difundido en toda la red. Esta dirección se usa de forma restringida, ya que supone una gran carga de trabajo en redes grandes.

La dirección de bucle local está diseñada para pruebas y comunicación entre procesos en la máquina local. Si un programa envía un mensaje a esta dirección, el módulo internet le devolverá los datos sin enviar nada a la red. De hecho, nunca debe haber en la red un paquete de este tipo, ya no es una dirección de red válida.

1.2.3 Subredes

En el direccionamiento IP a cada red física se le asigna una única dirección de red, los hosts de esa red llevan su dirección de red incluida en su dirección individual.

Este esquema de direccionamiento tiene un fallo: el crecimiento exponencial de Internet. Cuando el protocolo IP fue diseñado, nadie imaginó que pudiera hacer cientos de miles de pequeñas redes de ordenadores personales.

Al existir tantas redes, aparte del problema administrativo de asignar direcciones a todas ellas, existe el problema de que las tablas de encaminamiento de los *gateways* son excesivamente largas, y la ocupación de ancho de banda de la red usada en transmitir esas tablas es alta.

Para solucionar esto, se debe disminuir el número de direcciones de red asignadas sin alterar el esquema de direccionamiento original. Para conseguir esto, hay que hacer que un mismo prefijo de red IP pueda ser compartido por múltiples redes físicas.

Para conseguir este objetivo deberán de modificarse los procedimientos de encaminamiento y todas las máquinas que se conectan a esas redes deben entender las convenciones usadas.

Para asignar una única dirección IP a varias redes físicas, se usa la *máscara de subred*, conceptualmente, el añadir subredes sólo varía la interpretación de las direcciones IP ligeramente. En lugar de dividir la dirección IP de 32 bits en un prefijo de red y un sufijo de host, lo que se hace es mantener el prefijo de red original, pero en lugar de una parte de host, lo que se tiene es una parte local, que puede ser asignada libremente en la red local.

El resultado es una forma de direccionamiento jerárquico que conlleva un encaminamiento jerárquico.

La ventaja de usar direccionamiento jerárquico es que permite el crecimiento con facilidad, ya un *gateway* no necesita conocer con tanto detalle los destinos remotos como los cercanos. Una desventaja es la dificultad de establecer el sistema, y mucho más de cambiarlo una vez establecido.

Para permitir la máxima flexibilidad al dividir las direcciones de subredes, se permite que cada red física pueda escoger independientemente su interpretación propia de subred. Una vez escogida, todas las máquinas en esta red deberán respetar su partición.

El estándar IP para subredes especifica que para cada red física en una localización que use subredes hay que escoger una *máscara de subred* de 32 bits. De esta forma, la parte local asociada con el identificador de host se puede dividir en dos, una asociada con el identificador de subred, y otra con el host en particular. En la máscara adquieren valor 1 los bits situados en las posiciones para la indicación de la clase de dirección y para el prefijo de red. Y dentro de la parte local, adquieren valor 1 los bits destinados a identificar la subred.

De esta forma, por ejemplo, una máscara 255.255.255.0 sobre una dirección de clase B determina que se pueden codificar 256 subredes, y sobre cada una de ellas 256 host.

1.3 El Datagrama Internet

1.3.1 El Datagrama Internet

Un Datagrama es la unidad básica de transferencia entre la Internet, y se descompone en cabecera y datos.

La estructura de un Datagrama Internet es la siguiente:

<table>
<tr><td>Versión</td><td>Long.cab</td><td>Tipo de servicio</td><td colspan="2">Longitud total</td></tr>
<tr><td colspan="3">Identificación</td><td>Flags</td><td>Offset fragmento</td></tr>
<tr><td>Tiempo de vida</td><td colspan="2">Protocolo</td><td colspan="2">FCS cabecera</td></tr>
<tr><td colspan="5">Dirección IP fuente</td></tr>
<tr><td colspan="5">Dirección IP destino</td></tr>
<tr><td colspan="4">Opciones</td><td>Relleno</td></tr>
<tr><td colspan="5">DATOS</td></tr>
</table>

A continuación describiremos cada uno de los campos :

1.3.2 Versión.

Este campo ocupa 4 bits, e indica el tipo de formato de datagrama. Para el formato descrito, su valor es 4 (IP versión 4).

1.3.3 Longitud de la cabecera.

Este campo ocupa 4 bits, y especifica la longitud de la cabecera medida en palabras de 32 bits, el mínimo valor posible para una cabecera correcta es 5 (5 32, 160 bits), ya que el campo de opciones puede estar presente o no.

1.3.4 Tipo de servicio.

Este campo ocupa 8 bits, e indica como deberá ser tratado el datagrama. Este campo se divide a su vez en cinco subcampos, de la forma siguiente:

Prioridad	D	T	R	Sin uso
3b	1b	1b	1b	2b

Los 3 bits de prioridad, con valores comprendidos entre cero (prioridad normal) y siete (control de red), permiten al remitente indicar la importancia del datagrama. Aunque la mayor parte del software y de los gateways no usa este campo, es un concepto importante porque permite que en un momento determinado los comandos de control tengan prioridad sobre los datos. Por ejemplo, sin este campo sería imposible implementar algoritmos de control de congestión que no se vieran afectados por la congestión que están intentando controlar.

Los bits D,T y R especifican el tipo de transporte que el datagrama solicita. Si están activos, sus significados son :

D activado El Datagrama solicita bajo retardo

T activado El Datagrama solicita alta capacidad

R activado El Datagrama solicita alta fiabilidad.

Es posible que en uno o varios nodos del camino no exista alguna de las facilidades solicitadas, así, estos bits son más una ayuda a los algoritmos de encaminamiento que una petición de servicio.

Así, si existen varios caminos disponibles a un destino determinado, el algoritmo de encaminamiento usará los bits del campo tipo de servicio para determinar, en función del hardware de red subyacente, el puerto de salida por donde enviar el Datagrama.

1.3.5 Longitud del Datagrama

Este campo ocupa 16 bits, e indica la longitud total del Datagrama, incluyendo la cabecera y los datos, la longitud se indica en octetos. Con esto, se permite especificar una longitud de hasta 65536 octetos, sin embargo, los datagramas largos resultan intratables a muchos hosts y redes. El mínimo tamaño que debería aceptar un host es de 576 octetos. Se recomienda que los hosts sólo envíen datagramas de más de 576 octetos y tienen la seguridad de que el destinatario podrá aceptarlos.

El tamaño de 576 octetos se elige para permitir un tamaño razonable del bloque de datos para ser transmitido junto con la cabecera. Así, este tamaño permite un tamaño para el bloque de datos de 512 octetos, junto con 64 octetos para la cabecera. El tamaño máximo de una cabecera es de 64 octetos, y una cabecera normal ronda los 20 octetos, proporcionando un margen de actuación.

Para que el datagrama se transmita de un nodo a otro de la red, deberá ser transportado en un paquete de la red física subyacente. La idea de transportar un datagrama en una trama de red se denomina *encapsulamiento*.

Para la red física subyacente, el datagrama IP es como cualquier mensaje intercambiado entre dos ordenadores, sin que reconozca ni el formato de datagrama ni la dirección de destino IP.

En el caso ideal, todo el datagrama IP cabría en una sola trama de red, haciendo que la transmisión fuese eficiente. Pero como el datagrama puede atravesar en su camino diferentes tipos de redes físicas, no existe una longitud máxima de datagrama que se ajuste a todas ellas. A la longitud máxima de transferencia de datos por trama de una red física se le conoce como unidad de transferencia máxima (MTU, Maximum Transmision Unit).

Cuando un datagrama se envía por una red con un MTU menor que su longitud, entonces el datagrama se divide en partes denominadas

fragmentos. Al proceso se le conoce como fragmentación, y será comentado posteriormente.

1.3.6 Identificación

Este campo ocupa 16 bits, y contiene un número entero que identifica al datagrama. Este número suele asignarse con un contador secuencial en la máquina origen que va asignándolos según nuevos datagramas. Este campo es indispensable en el proceso de reensamblado de fragmentos, cuando un datagrama fue fragmentado.

1.3.7 Flags

Este campo ocupa 3 bits, e incluye varios flags de control :

Bit 0: Reservado, debe ser 0

Bit 1: (DF) 0 = el datagrama puede fragmentarse,

1 = el datagrama NO puede fragmentarse

Bit 2: (MF) 0 = es el último fragmento

1 = existen más fragmentos

El primer bit significativo (bit 1) del campo flags es el de *no fragmentación*, se llama así porque si está activo implica que el datagrama no puede fragmentarse. Este bit resulta útil en casos de pruebas de redes y en algunas aplicaciones especiales donde se necesita que el datagrama llegue sin fragmentar. En el caso de que el gateway sea incapaz de enviarlo sin fragmentarlo, envía un mensaje de error a la máquina origen.

El bit de menor peso del campo flags (bit 2), es el bit de *más fragmentos*. Este bit es útil para la máquina destino, que así puede determinar si ha recibido todos los fragmentos correspondientes a un datagrama. Cuando el bit está a cero, indica que es el último fragmento del datagrama. Así, con este bit y con el campo de offset de fragmento, la máquina puede comprobar si ya ha recibido todos los fragmentos y puede reensamblar el datagrama original. La máquina destino no puede guiarse sólo por el bit

de *más fragmentos*, porque es posible que se reciba el último fragmento antes de recibir algún fragmento intermedio, ya que IP no provee un método para que los datagramas lleguen ordenados.

1.3.8 Offset del fragmento

Este campo ocupa 13 bits, y especifica el desplazamiento desde el comienzo del campo de datos del datagrama original hasta el comienzo del campo de datos del fragmento, expresado en múltiplos de 8 octetos.
Ejemplo:

Cabecera fragmento 1	DATOS 1

Offset : 0 octetos Contenido del campo offset de fragmento: 0

Cabecera fragmento 2	DATOS 2

Offset : 600 octetos Contenido del campo offset de fragmento: 75

Cabecera fragmento 3	DATOS 3

Offset : 1200 octetos Contenido del campo offset de fragmento: 150

1.3.9 Tiempo de vida

Este campo ocupa 8 bits, e indica cuanto tiempo, en segundos, está el datagrama autorizado a permanecer en el sistema internet. La idea es simple: cuando una máquina pone un datagrama en la internet, le asigna un tiempo máximo de existencia del mismo. Los gateways y hosts que van procesando el datagrama deben ir decrementando el campo tiempo de vida, y descartarlo de la internet cuando el tiempo haya expirado.
Es difícil para los gateways estimar el tiempo exacto transcurrido desde que el datagrama salió de la máquina anterior, ya que no conocen el retardo inducido por las redes. Para solventar este problema, se siguen dos normas:

1.- Cada gateway por el que pasa el datagrama decrementará en 1 el valor del campo.

2.- Para tener en cuenta los casos de gateways con gran retardo de tránsito, al llegar el paquete a un gateway, éste almacenará la hora local de llegada, y en el momento de enviarlo decrementará el valor del campo según el número de segundos que haya estado en el sistema esperando ser enviado.

Cuando el campo alcanza el valor cero, el datagrama es descartado y se envía un mensaje de error al origen. La idea del tiempo de vida es interesante porque evita que los datagramas estén eternamente circulando por la red en el caso de que las tablas de encaminamiento estén corruptas y los gateways envíen los datagramas en círculo.

1.3.10 Protocolo

Este campo ocupa 8 bits, e indica cuál fue el protocolo de alto nivel que ha creado los datos que están en el campo *datos*. La asignación de estos valores se hace por una autoridad centralizada (IANA, Institute Assigned Numbers Authority), para que exista acuerdo a través de toda Internet.

1.3.11 FCS cabecera

Este campo ocupa 16 bits, y asegura la integridad de la cabecera. La máquina origen ejecuta una serie de operaciones matemáticas sobre el conjunto de la cabecera y pone el resultado en este campo. El receptor hará la misma operación y comparará el resultado para asegurarse de que los datos de la cabecera son correctos. Sólo es verificada la cabecera, para no sobrecargar de trabajo a los gateways. Al entregarse estos datos sin comprobar, serán los protocolos de alto nivel los que realices su propio chequeo.

1.3.12 Dirección IP origen

Este campo ocupa 32 bits, e indica la dirección IP de la máquina origen .

1.3.13 Dirección IP destino

Este campo ocupa 32 bits, e indica la dirección IP de la máquina destino.

1.3.14 Opciones

Este campo tiene una longitud variable, y puede estar o no presente en la cabecera del datagrama. Esta opcionalidad se refiere a datagramas en particular, no a la implementación específica, cualquier módulo internet debe implementar esta funcionalidad, tanto en hosts como en gateways.
Cada opción tendrá un campo *código de opción*, de 1 octeto de longitud, que puede ser suficiente según la opción, si no es así, este campo vendrá seguido de un campo *longitud*, también de un octeto, y de un campo conteniendo los datos específicos de la opción de longitud variable.

La estructura de un campo *código de opción* es la siguiente:

Copia	Clase de opción	Número de opción
1 bit	2 bits	5 bits

- Copia.

El primer bit del campo es el de *copia*. Cuando este bit está a uno, indica que la opción deberá ser copiada a los diferentes fragmentos en caso de que el datagrama sea fragmentado. Si está a cero, entonces la opción deberá ser copiada sólo en el primer fragmento y no en el resto.

- Clase de opción.

Indica la clase de opción indicada, las diferentes clases son:

00 Datagrama o control de red

01 Reservado para uso futuro

10 Medida y control de errores

11 Reservado para uso futuro

- Número de opción.

Indica la opción específica.

En la siguiente tabla se muestran las diferentes opciones.

Clase	Número	Longitud	Descripción
0	0	1 octeto	Fin de la lista de opciones.
0	1	1 octeto	Sin operación.
0	2	11 octetos	Seguridad y restricciones de acceso.
0	3	variable	Encaminamiento de datagramas por rutas específicas.
0	7	variable	Grabación de ruta.
0	9	variable	Encaminamiento dirigido.
2	4	variable	Grabación de tiempo.

- Fin de la lista de opciones.

Indica el fin de la lista de opciones, que no suele coincidir con el tamaño de la cabecera. Se usa al final de todas las opciones, no tras cada una. Debe ser copiado en caso de fragmentación.

- Sin operación.

Esta opción puede usarse entre dos opciones, por ejemplo, para alinear el comienzo de la siguiente opción a 32 bits. Debe ser copiado en caso de fragmentación.

- Seguridad y restricciones de acceso.

Esta opción tiene una longitud de 11 octetos, y se divide en varios campos, indicando nivel de seguridad (desde *desclasificado*, hasta *alto secreto*), restricciones de acceso, etc.

- Opción de grabación de la ruta.

Esta opción permite que el origen cree una tabla de direcciones IP vacía, y obliga a cada gateway por el que pasa el datagrama a incluir su propia dirección IP en dicha lista. Su formato es el siguiente:

Código	Longitud	Puntero	primera dirección	segunda dirección	…	n-ésima dirección
1 octeto	1 octeto	1 octeto	32 bits	32 bits	…	32 bits

El campo *código* contiene el código de opción (7 para esta opción en concreto). El campo *longitud* especifica la longitud total ocupada por la opción, incluyendo los tres primeros octetos. El campo *puntero* indica la distancia (offset) al primer campo de dirección libre. A continuación, los campos de dirección contienen la dirección IP de los gateways que han ido encaminando el datagrama por la internet.

Cuando el datagrama llega al destino, la máquina leerá esta lista y la analizará, pero sólo sin antes se ha acordado entre origen y destino que sea así. El destino no analizará las direcciones sólo porque estas aparezcan con la opción activada.

Un detalle a tener en cuenta es que el nodo origen debe reservar suficiente espacio en la lista para las direcciones de todos los gateways que el datagrama encuentre en su camino. Si no es así, no se quedarán grabadas las direcciones de los últimos gateways que el datagrama atravesó.

- Opción de encaminamiento dirigido.

Esta opción permite al remitente indicar un camino determinado al datagrama a través de la internet. Es muy útil en caso de pruebas, pues permite dirigir el datagrama a un lugar por el que no pasaría normalmente. Para el usuario final no es muy interesante, tan sólo para la gente que conoce la topología de la red.

Existen dos tipos de encaminamiento dirigido : Estricto y aproximado.

En el encaminamiento estricto, el datagrama debe seguir exactamente el camino indicado en la lista. Si un gateway no puede conseguir esa ruta, se genera un error. En el caso de encaminamiento aproximado, el datagrama puede encontrar varios gateways entre dos direcciones IP de la lista sin que se genere ningún error.

En ambos casos los gateways van escribiendo su propia dirección IP en la lista, al igual que en la opción de grabación de ruta.

El formato de esta opción es similar al descrito en la opción *grabación de la ruta*, en este caso el código será el 137, y las direcciones IP corresponderán a las de los gateways que el datagrama debe atravesar.

- Grabación del tiempo.

La opción de grabación del momento funciona de forma similar a la de grabación de ruta, añadiendo además el momento en que el datagrama atravesó el gateway. Cada entrada en la tabla contiene dos partes de 32 bits, una será la dirección del gateway, y otra, el día y la hora expresada en milisegundos desde medianoche, según la hora universal (meridiano de Greenwich). Si no existe posibilidad de obtener esa hora, el gateway pone a 1 el bit más alto y escribe el día y la hora local. Por esto, la grabación de hora no debe considerarse como exacta, sino como una estimación.

La estructura de la opción es la siguiente :

Código	Longitud	Puntero	Desbordamiento	Flags
Primera dirección IP				
Primera marca de tiempo				
Segunda dirección IP				
Segunda marca de tiempo				

…
…
n-ésima dirección IP
n-ésima marca de tiempo

- Los campos código (en este caso 68), longitud y puntero son similares a los de las opciones comentadas anteriormente.

- Desbordamiento.

Este campo ocupa 4 bits e indica el número de gateways que no pudieron insertar sus marcas de tiempo porque no se reservó espacio suficiente.

- Flags.

Este campo ocupa 4 bits, e indica como deben los gateways de insertar exactamente el momento en que el datagrama fue tratado por ellos. Algunos de los valores son:

0 Grabar sólo el momento. Omitir la dirección IP

1 Poner la dirección IP antes del momento (este es el formato descrito anteriormente)

3 Las direcciones IP son indicadas por el remitente. Un gateway sólo graba el momento si la siguiente dirección IP coincide con la suya.

1.3.15 Relleno

La cabecera de un datagrama IP esta alineada a 32 bits. Este campo se usa para asegurar que sea así. El sobrante hasta conseguir un tamaño múltiplo de 32 (bits), se rellena con 0's.

1.4 Fragmentación

La fragmentación de un datagrama IP es necesaria cuando el tamaño de un datagrama resulta intratable para alguna de las redes que debe atravesar para llegar a su destino.

El campo *identificador* es usado junto con los de dirección origen, dirección destino y protocolo, para identificar fragmentos a reensamblar. El módulo Internet del origen del paquete debe asignar un identificador único para cada datagrama, que el destino usa para identificar a que datagramas originales pertenecen que fragmento.

El flag *más fragmentos*, está a 1 si el datagrama no es el último fragmento. El campo *offset de fragmento* identifica la localización del fragmento en el datagrama original, indicando el desplazamiento sobre su comienzo.

La estrategia de fragmentación está diseñada para que un datagrama sin fragmentar tenga toda la información relativa a fragmentación a 0's (*mas fragmentos* = 0*, offset fragmento* = 0*)*. Si un datagrama es fragmentado, todos sus fragmentos (menos el último) deben de estar alineados a 8 octetos (su longitud en bits debe ser múltiplo de 64).

Para fragmentar un datagrama Internet, un módulo Internet crean nuevos datagramas y copia los contenidos de la cabecera a todos ellos. El campo *datos* del datagrama original es dividido en n partes, las cuales deben estar alineadas a 8 octetos. La primera porción de datos se copia en el primer datagrama generado, y se cambia su campo *longitud*, haciéndolo coincidir con la longitud del primer datagrama. El flag *más fragmentos* es puesto a 1. La segunda porción de datos es copiada en el segundo datagrama, se cambia su campo *longitud* y *más fragmentos* de forma similar, y se especifica el desplazamiento en el campo *offset de fragmento*.

Este proceso se repite hasta el último fragmento generado, que tendrá el flag *más fragmentos* a 0, y que no deberá estar alineado a 8 octetos necesariamente.

Para reensamblar los fragmentos, el módulo internet en el destino, combina los fragmentos que tengan el mismo valor en los campos *identificador, dirección origen, dirección destino* y *protocolo*. La recombinación se hace copiando la parte de datos de cada fragmento en la posición relativa indicada en el campo *offset de*

fragmento. El primer fragmento deberá tener el campo *offset de fragmento* a cero, y el último fragmento el flag *más fragmentos* a cero.

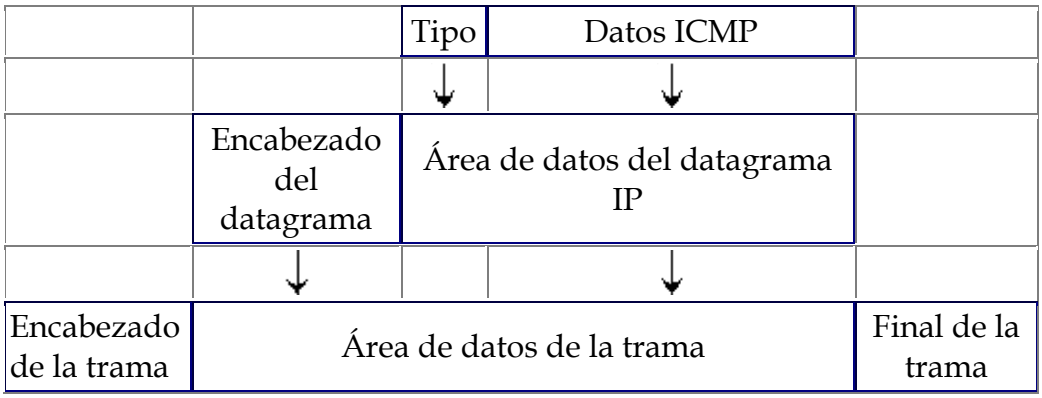
1.5 El protocolo ICMP

1.5.1 El protocolo ICMP

Debido a que el protocolo IP no es fiable, los datagramas pueden perderse o llegar defectuosos a su destino. El protocolo ICMP (*Internet Control Message Protocol*, protocolo de mensajes de control y error) se encarga de informar al origen si se ha producido algún error durante la entrega de su mensaje. Pero no sólo se encarga de notificar los errores, sino que también transporta distintos mensajes de control.

El protocolo ICMP está definido en la RFC 792 (en ingles y en español).

El protocolo ICMP únicamente informa de incidencias en la red pero no toma ninguna decisión. Esto será responsabilidad de las capas superiores. Los mensajes ICMP viajan en el campo de datos de un datagrama IP, como se puede apreciar en el siguiente esquema:



Debido a que el protocolo IP no es fiable puede darse el caso de que un mensaje ICMP se pierda o se dañe. Si esto llega a ocurrir no se creará un nuevo mensaje ICMP sino que el primero se descartará sin más.

Los mensajes ICMP comienzan con un campo de 8 bits que contiene el tipo de mensaje, según se muestra en la tabla siguiente. El resto de campos son distintos para cada tipo de mensaje ICMP.

El formato y significado de cada mensaje ICMP está documentado en la RFC 792.

Campo de tipo	Tipo de mensaje ICMP
0	Respuesta de eco (*Echo Reply*)
3	Destino inaccesible (*Destination Unreachable*)
4	Disminución del tráfico desde el origen (*Source Quench*)
5	Redireccionar (cambio de ruta) (*Redirect*)
8	Solicitud de eco (*Echo*)
11	Tiempo excedido para un datagrama (*Time Exceeded*)
12	Problema de Parámetros (*Parameter Problem*)
13	Solicitud de marca de tiempo (*Timestamp*)
14	Respuesta de marca de tiempo (*Timestamp Reply*)
15	Solicitud de información (obsoleto) (*Information Request*)
16	Respuesta de información (obsoleto) (*Information Reply*)
17	Solicitud de máscara (*Addressmask*)
18	Respuesta de máscara (*Addressmask Reply*)

1.5.2 *Solicitud y respuesta de eco*

Los mensajes de solicitud y respuesta de eco, tipos 8 y 0 respectivamente, se utilizan para comprobar si existe comunicación entre 2 hosts a nivel de la capa de red. Estos mensajes comprueban que las capas física (cableado), acceso al medio (tarjetas de red) y red (configuración IP) están correctas. Sin embargo, no dicen nada de las capas de transporte y de aplicación las cuales podrían estar mal configuradas; por ejemplo, la recepción de mensajes de correo electrónico puede fallar aunque exista comunicación IP con el servidor de correo.

La orden **PING** envía mensajes de solicitud de eco a un host remoto e informa de las respuestas. Veamos su funcionamiento, en caso de no producirse incidencias en el camino.

1. A envía un mensaje ICMP de tipo 8 (*Echo*) a B.
2. B recibe el mensaje y devuelve un mensaje ICMP de tipo 0 (*Echo Reply*) a A.
3. A recibe el mensaje ICMP de B y muestra el resultado en pantalla.

A>**ping** **172.20.9.7** **-n** **1**
Haciendo ping a 172.20.9.7 con 32 bytes de datos:
Respuesta desde 172.20.9.7: bytes=32 tiempo<10ms TDV=128

En la orden anterior hemos utilizado el parámetro "-n 1" para que el host A únicamente envíe 1 mensaje de solicitud de eco. Si no se especifica este parámetro se enviarían 4 mensajes (y se recibirían 4 respuestas).

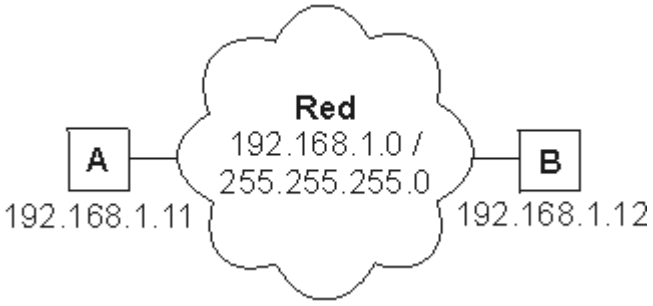
Si el host de destino no existiese o no estuviera correctamente configurado recibiríamos un mensaje ICMP de tipo 11 (*Time Exceeded*).

A>**ping 192.168.0.6 -n 1**
Haciendo ping a 192.168.0.6 con 32 bytes de datos:
Tiempo de espera agotado.

Si tratamos de acceder a un host de una red distinta a la nuestra y no existe un camino para llegar hasta él, es decir, los routers no están correctamente configurados o estamos intentando acceder a una red aislada o inexistente, recibiríamos un mensaje ICMP de tipo 3 (*Destination Unreachable*).

A>**ping 1.1.1.1 -n 1**
Haciendo ping a 1.1.1.1 con 32 bytes de datos:
Respuesta desde 192.168.0.1: Host de destino inaccesible.

Utilización de PING para diagnosticar errores en una red aislada



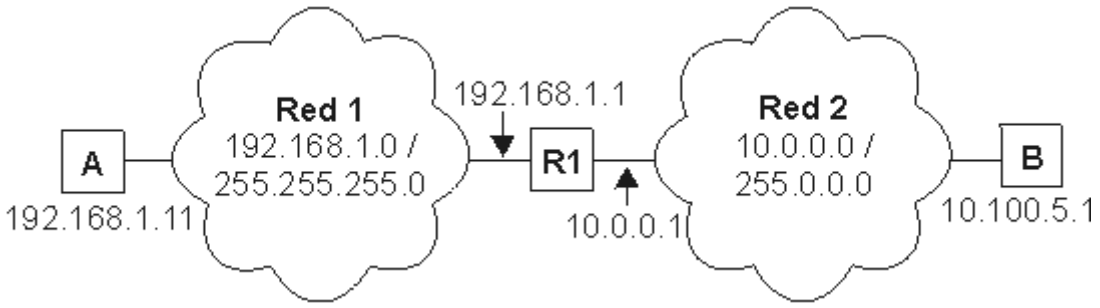
A>**ping 192.168.1.12**

- Respuesta. El cableado entre A y B, las tarjetas de red de A y B, y la configuración IP de A y B están correctos.
- Tiempo de espera agotado. Comprobar el host B y el cableado entre A y B.
- Host de destino inaccesible. Comprobar las direcciones IP y máscaras de subred de A y B porque no pertenecen a la misma red.
- Error. Probablemente estén mal instalados los protocolos TCP/IP del host A. Probar A>**ping 127.0.0.1** para asegurarse.

Nota: *El comando* ping 127.0.0.1 *informa de si están correctamente instalados los protocolos TCP/IP en nuestro host. No informa de si la tarjeta de red de nuestro host está correcta.*

Utilización de PING para diagnosticar errores en una red de redes

A continuación veremos un ejemplo para una red de redes formada por dos redes (1 solo router). La idea es la misma para un mayor número de redes y routers.



A>**ping 10.100.5.1**

- Respuesta. El cableado entre A y B, las tarjetas de red de A, R1 y B, y la configuración IP de A, R1 y B están correctos. El router R1 permite el tráfico de datagramas IP en los dos sentidos.
- Tiempo de espera agotado. Comprobar el host B y el cableado entre R1 y B. Para asegurarnos que el router R1 está funcionando correctamente haremos A>**ping 192.168.1.1**
- Host de destino inaccesible. Comprobar el router R1 y la configuración IP de A (probablemente la puerta de salida no sea 192.168.1.1). Recordemos que la puerta de salida (*gateway*) de una red es un host de su propia red que se utiliza para salir a otras redes.
- Error. Probablemente estén mal instalados los protocolos TCP/IP del host A. Probar A>**ping 127.0.0.1** para asegurarse.

En el caso producirse errores de comunicación en una red de redes con más de un router (Internet es el mejor ejemplo), se suele utilizar el comando PING para ir diagnosticando los distintos routers desde el destino hasta el origen y descubrir así si el fallo es responsabilidad de la red de destino, de una red intermedia o de nuestra red.

1.5.3 *Mensajes ICMP de tiempo excedido*

Los datagramas IP tienen un campo TTL tiempo de vida) que impide que un mensaje esté dando vueltas indefinidamente por la red de redes. El número contenido en este campo disminuye en una unidad cada vez que el datagrama atraviesa un router. Cuando el TTL de un datagrama llega a 0, éste se descarta y se envía un mensaje ICMP de tipo 11 (*Time Exceeded*) para informar al origen.

Los mensajes ICMP de tipo 11 se pueden utilizar para hacer una traza del camino que siguen los datagramas hasta llegar a su destino. ¿Cómo? Enviando una secuencia de datagramas con TTL=1, TTL=2, TTL=3,

TTL=4, etc... hasta alcanzar el host o superar el límite de saltos (30 si no se indica lo contrario). El primer datagrama caducará al atravesar el primer router y se devolverá un mensaje ICMP de tipo 11 informando al origen del router que descartó el datagrama. El segundo datagrama hará lo propio con el segundo router y así sucesivamente. Los mensajes ICMP recibidos permiten definir la traza.

La orden **TRACERT** (**traceroute** en entornos Unix) hace una traza a un determinado host. TRACERT funciona enviando mensajes ICMP de solicitud de eco con distintos TTL; traceroute, en cambio, envía mensajes UDP. Si la comunicación extremo a extremo no es posible, la traza nos indicará en qué punto se ha producido la incidencia. Existen algunas utilidades en Internet, como Visual Route , que conocen la localización geográfica de los principales routers de Internet. Esto permite dibujar en un mapamundi el recorrido que siguen los datagramas hasta llegar a un host.

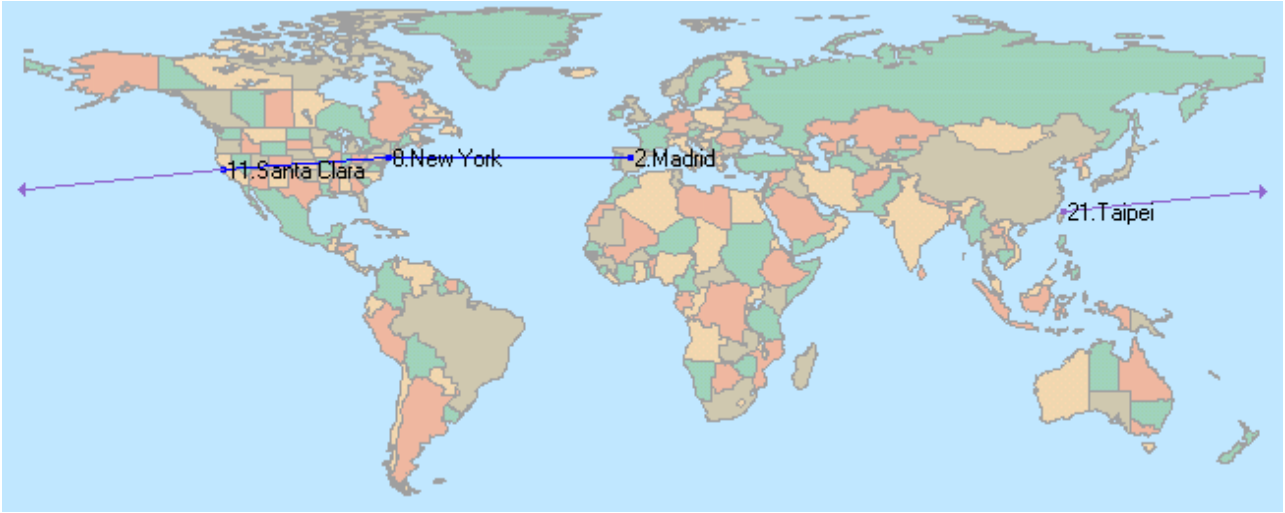
A>**tracert 130.206.1.2**

Traza a la dirección sun.rediris.es [130.206.1.2]
sobre un máximo de 30 saltos:

```
 1   1 ms   1 ms   1 ms  PROXY [192.168.0.1]
 2 122 ms 118 ms 128 ms  MADR-X27.red.retevision.es [62.81.1.102]
 3 143 ms 232 ms 147 ms  MADR-R2.red.retevision.es [62.81.1.92]
 4 130 ms 124 ms 246 ms  MADR-R16.red.retevision.es [62.81.3.8]
 5 590 ms 589 ms 431 ms  MADR-R12.red.retevision.es [62.81.4.101]
 6 612 ms 640 ms 124 ms  MADR-R10.red.retevision.es [62.81.8.130]
 7 259 ms 242 ms 309 ms  193.149.1.28
 8 627 ms 752 ms 643 ms  213.0.251.42
 9 137 ms 117 ms 118 ms  213.0.251.142
10 109 ms 105 ms 110 ms  A1-2-1.EB-Madrid00.red.rediris.es [130.206.224.81]
11 137 ms 119 ms 122 ms  A0-0-0-1.EB-Madrid3.red.rediris.es [130.206.224.86]
12 109 ms 135 ms 115 ms  sun.rediris.es [130.206.1.2]
```

Traza completa.

Ejemplo de Visual Route a una dirección IP de Taiwan (203.69.112.12):



1.6 El protocolo ARP

1.6.1 El protocolo ARP

Como hemos comentado en el punto anterior, dos máquinas que quieren comunicarse en la misma red física, sólo podrán hacerlo si conocen sus direcciones físicas.

Existen diversas formas de resolver el problema. Si en la red física pudiera escogerse la numeración de las estaciones (Red PRONET 10), entonces podemos hacer que su número sea una función simple de su dirección IP.

Pero en el caso de una red Ethernet, el problema no resulta tan sencillo de resolver. Cada interfaz Ethernet tiene asignada una dirección hardware de 48 bits, así pues, es imposible codificar la dirección hardware en una dirección IP, además, si se sustituye el interfaz Ethernet, cambia la dirección física de la estación.

Para resolver este problema, se diseño el *protocolo de resolución de direcciones* (ARP, Adress Resolution Protocol), válido para todas las redes que soportan multidistribución.

La idea es simple, si una máquina A necesita saber la dirección física de una máquina B, envía por multidifusión un paquete especial que pide a la máquina con la dirección IP indicada que responda con su dirección física. Una vez recibida la respuesta, A puede enviar paquetes a B directamente, pues conoce su dirección física.

Debido a que la multidistribución es un recurso costoso (consume recursos de red, ya que todos los receptores deben procesar el paquete enviado), suele evitarse su uso lo más posible. Una de las formas de hacer esto es manteniendo en cada máquina una tabla relacionando direcciones IP con direcciones físicas. Además, como en cada petición ARP se encuentra la dirección IP y la dirección física del remitente, todas las máquinas activas pueden actualizar su tabla con el nuevo dato.

Al enviar un mensaje ARP de una máquina a otra, este debe viajar en una trama física. Para que la máquina destino identifique la trama como ARP, debe llevar un valor en el campo de tipo de trama que lo identifique como tal. En Ethernet, este valor es 0806h (en hexadecimal).

El formato del mensaje ARP no es fijo, sino que depende que hardware de la red.

El formato de un mensaje ARP para Ethernet es el siguiente:

<table>
<tr><td colspan="2">Tipo de hardware</td><td>Tipo de protocolo</td></tr>
<tr><td>Long. dir. física</td><td>Long. dir. protocolo</td><td>Operación</td></tr>
<tr><td colspan="3">Dirección física remitente (octetos 0 a 3)</td></tr>
<tr><td colspan="2">Dirección física remitente (octetos 4 a 5)</td><td>dirección IP remitente (octetos 0 y 1)</td></tr>
<tr><td colspan="2">dirección IP remitente (octetos 2 y 3)</td><td>Dirección física destinatario (octetos 0 y 1)</td></tr>
<tr><td colspan="3">Dirección física destinatario (octetos 2 a 5)</td></tr>
<tr><td colspan="3">Dirección IP destinatario (completa, octetos 0 a 3)</td></tr>
</table>

El campo *tipo de hardware (16 bits)* especifica el tipo de interfaz hardware del que se busca la dirección (1 para Ethernet). El campo *tipo de protocolo (16 bits)* indica el tipo de protocolo del que el origen ha enviado la dirección (0800h para IP).

Los campos de longitud de direcciones física y de protocolo permiten usar ARP con diferentes hardware y protocolos.

El campo *operación* nos indica el tipo de operación en concreto, si es una petición ARP o una respuesta a una petición.

El resto de los campos indican las direcciones IP y físicas tanto del remitente como del destinatario.

1.6.2 El protocolo RARP

El protocolo RARP (Reverse Adress Resolution Protocol) es una variación de ARP, que permite a estaciones sin unidad de almacenamiento fija obtener su propia dirección IP.

Cuando una estación sin unidad de almacenamiento arranca, envía a la red un mensaje multidifusión con su dirección física (obtenida directamente del hardware). El servidor de direcciones buscará la dirección física del solicitante y le enviará un mensaje indicándole su dirección IP.

1.7 DNS - Domain Name System

1.7.1 Introducción

Los usuarios de las redes, prefieren utilizar nombres pronunciables, mas fáciles de recordar, en vez de la dirección IP de las maquinas conectadas a la red.

Inicialmente en Internet, el sistema de nombres escogido era una secuencia de caracteres arbitraria, administrada por el NIC (Network Information Center), que comprobaba la no existencia de otra maquina con ese mismo nombre. Como el numero de usuarios se incremento demasiado, el tener una única autoridad de asignación de nombres no era nada practico, debido al enorme trabajo administrativo que era mantenerla al día.

La solución hallada, que aun se utiliza, fue el descentralizar el mecanismo de asignación de nombres, delegando la autoridad, en parte de del espacio de nombres y distribuyendo la responsabilidad de asignar la relación entre nombres y direcciones.

La definición de asignación entre nombres y direcciones debe estar definida orientada a la traducción eficiente y que garantice el control autónomo de la asignación de nombres. Pongamos un ejemplo:
nombre_local.nombre_general
donde nombre_local seria el nombre administrado por una localización en concreto y nombre_general administrado por una autoridad general (nótese que ambos nombres están separados por puntos). Si aparece una nueva localización, la autoridad central incluirá su nombre en la lista de localizaciones validas y le daría capacidad para administrar todos los grupos de nombres que antecedan al nombre de esa nueva localización (separada por puntos). Los nombres se componen de combinaciones de los 26 caracteres anglosajones (A-Z y a-z), los dígitos (0-9) y el carácter "-". La longitud máxima de nombres de dominios o subdominios es de 63 caracteres y del nombre completo de 255 caracteres para el uso de cada uno de estos dominios.

Así llegamos al punto de tener una estructura jerárquica, subdividiendo el espacio de nombres hasta que este sea manejable, esto es :
disc.eps.ua.es
Donde "disc" sería el Departamento de Ingeniería de Sistemas y Comunicaciones, de la Escuela Politécnica Superior de Alicante "eps" de la Universidad de Alicante "ua", de España "es". Podríamos caer en la falacia de que los nombres asignados están relacionados (necesariamente) con la topología de la red o la estructura de las interconexiones físicas.
El mecanismo que implementa una jerarquía de nombres de maquinas en las redes se llama Sistema de Dominio de Nombres (Domain Name System, DNS ; a partir de ahora utilizaremos las siglas anglosajonas para referirnos a este sistema, por ser reconocidas internacionalmente, y mas familiares).

El DNS especifica la sintaxis de los nombres, y las reglas para delegar autoridad sobre los nombres; además de especificar la implementaron de un sistema distribuido que relaciona eficientemente nombres con direcciones.

1.7.2 Sintaxis de Nombres

La sintaxis de los nombres se compone de nombres de dominios separados por puntos. El nivel mas bajo se sitúa a la izquierda (esto facilita comprimir mensajes con múltiples nombres de dominios).

> En Internet la máxima autoridad para asignar las direcciones IP y los DNS es el IANA (Internet Assigned Numbers Authority), también es la encargada de delegar el segundo nivel de DNS a la organización IR (Internet Registry) o a registros regionales.

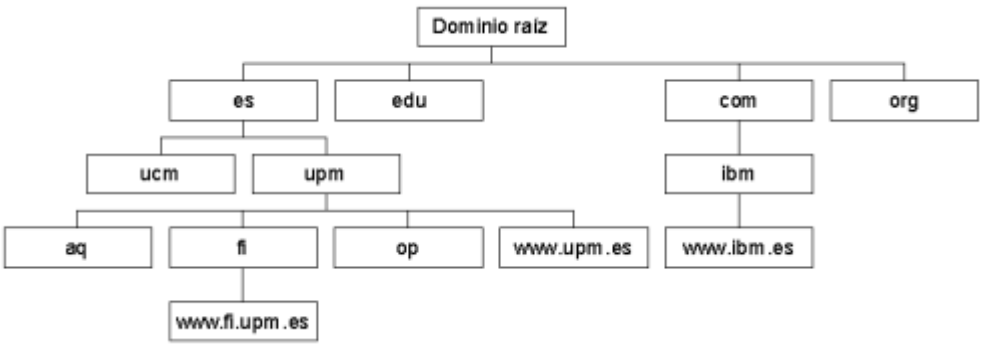
El sistema principal (root) no tiene nombre y es el que en un solo fichero (hosts.txt) tiene los nombres de los host y sus direcciones, este fichero es accesible vía ftp (rfc952 y rfc953). El "Top-level domain names" (TLDs, nivel superior del dominio de nombres) se divide en los siguientes DNS:

Nombre	Significado
Com	Organizaciones comerciales, se van a establecer subdominios
Edu	Instituciones educativas (registro de 2 a 4 años)
Gov	Instituciones gubernativas de EE.UU.
Mil	Grupos militares de EE.UU.
Net	Principales centros de soporte de red (NICs,NOCs ...)
Org	Otras organizaciones (diferentes a las anteriores)
Arpa	Dominio ARPANET temporal (obsoleto)
Int	Organizaciones internacionales
Código de país	Cada país (esquema geográfico)

Ejemplos de códigos de país, según la norma ISO-3166 :

Nombre	Significado
Es	España
Uk	Inglaterra (United Kindows)
De	Alemania (Deuschland)
Us	EE.UU.. (United States of América)
Fr	Francia
…	…

Conceptualmente, se permiten dos tipos diferentes de jerarquías: geográfica y organizacional. Cada organismo solicita con que tipo de esquema desea tener su nombre (en Internet el esquema geográfico es administrado por organismos generalmente públicos, en el caso particular de España, actualmente, este organismo solo permite solo permite que utilicen el nombre de dominio "es" las empresas S.A. y S.L.).



Veamos ejemplos de ambos tipos de jerarquías:

ozu.com ozu.es (dos empresas distintas, nacidas de la separación de Ozu)

La configuración de los host locales (se comenta en el rfc1033) pasa por unas especificaciones del administrador principal, este le provee de:

- La definición de su zona de actuación.
- El fichero maestro de datos.
- Le actualiza el fichero maestro.

Y el Sistema de Dominio proporciona los métodos estándar de:

- Formatos de recursos de datos.
- Métodos de búsqueda en las BD.
- Métodos NS para actualizar los datos locales sobre Servidores de Nombres.

En algunos países el segundo nivel de la jerarquía esta definida por categorías (AC, CO, GO, RE ...), en otras por políticas geográficas, por ejemplo en EE.UU. es de la
forma :

nombre-entidad.localidad.estado.us

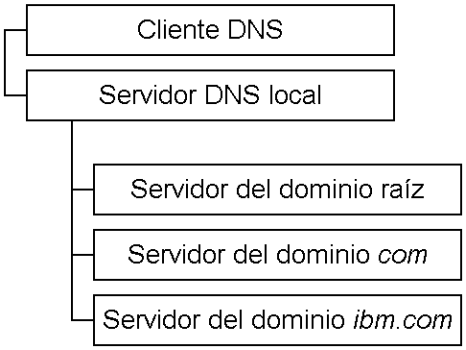
IBM.Armonk.NY.US

En EE.UU. existe un subdominio en el segundo nivel (a parte de los estados):

Nombre	Significado
k12	escuelas
Cc	colegios
Tec	escuelas técnicas
State	agencias estatales de gobierno
Cog	ayuntamientos
Lib	bibliotecas
mus	museos
....	ver RFC-1480 para otros

El IR se encarga de seleccionar y designar la administración diaria del DNS.

Destaquemos que usando solo la sintaxis de dominio de nombres no se puede distinguir los nombres de subdominios de maquinas individuales.



1.7.3 Normas para nombrar DNS

El registro de un nuevo nombre no implica derechos de marca (C, R, TM, LTD...), y la responsabilidad es de cada uno al elegir su nombre, asegurándose que no es una TM (marca registrada).

IANA no se encarga de decidir que es o no un país, estado... por ello la codificación de países la hace mediante la utilización del ISO 3166.

1.7.4 Registros, Consultas y derechos

Para registrarte en el segundo nivel, en COM, EDU, ORG, NET o GOV debe hacerlo IR en InterNIC y el segundo nivel de MIL debe registrarlo DDN como NIC.DDN.MIL, para INT debes dirigirte a PVM como ISI.EDU.

Para cualquier comentario sobre un nuevo TLD escribe a hostmaster@internic.net, si tienes algún problema con los registros regionales o los administradores de tu país contacta con:

Europa RIPE NCC ncc@ripe.net

Asia y el Pacifico APNIC hostmaster@apnic.net

Resto del mundo INTERNIC hostmaster@internic.net

Cuando se demuestra que un delegado de DNS no tiene competencia técnica suficiente, se le exige que en un determinado tiempo a unos estándares: que responda de un modo rápido a peticiones DNS, que disponga de Bases de Datos (BD) precisas, robustas y resistentes. Si en

este tiempo no arregla los problemas se le revoca la delegación de DNS y posiblemente se delegue esta a otro .

Para acceder BD del host rs.internic.net, y poder ver información sobre los TLDs, se debe utilizar:

whois -h rs.internic.net TLD-dom

Siendo TLD el nombre de cualquiera de ellos (us, edu, net ...).

1.7.5 Dominio de Nombres para Correo Electrónico

Los mensajes de correo son de la forma:

nombre_usuario@nombre_parte_de_dominio

pongamos un ejemplo:

ctpinos@yahoo.com

Donde ctpinos seria el nombre del usuario, este nombre es configurado por el administrador de la subred, la red local o incluso por un usuario de una maquina conectada directamente a Internet. @ nos indicaría que es una dirección de correo (el sistema de correo utiliza el DNS MX). Y yahoo.com seria el nombre del dominio (Yahoo).

1.7.6 Resolución de nombres en direcciones:

El esquema de dominio de nombres incluye un sistema eficiente, seguro, de propósito general y distribuido para relacionar nombres con direcciones. Este sistema esta compuesto por una serie de sistemas independientes, pero cooperativos denominados Servidores de Nombres; cada uno de ellos es un programa que funciona en un servidor y que soporta traducciones de direcciones IP a nombres y viceversa.

El programa cliente, denominado resolutor de nombres, necesitara usar uno o mas servidores al traducir un nombre.

Existen dos tipos de peticiones de resolución:

- Recurrida: Donde el servidor de nombres contara con otros servidores hasta

hallar la respuesta a la petición y la enviara al remitente.

- Iterativa: En la que el servidor, en el caso de que no pueda resolver la dirección por si mismo, mandara un mensaje al remitente diciéndole que no puede resolverla e indicándole la dirección del servidor de nombres al que debe dirigirse para hacerlo.

Para optimizar la búsqueda de nombres en servidores remotos, y para reducir el trafico en la red, los servidores utilizan la técnica caching, que consiste en:

1.- Cuando se recibe respuesta de un servidor remoto de una petición de resolución, se le añade un tiempo de vida (TTL) .

2.- Se mantiene durante un cierto tiempo en la memoria del servidor de nombres aquellas parejas nombre-direccion que hayan sido resueltos a petición de algún usuario de la red, junto con su TTL.

3.- Antes de enviar una petición a un servidor remoto, buscara en memoria si ya tiene esa dirección resuelta. Si existe en memoria, se la enviara al remitente informándole que pudiera no estar actualizada; enviara también la dirección servidor de nombres remoto, por si le interesa garantizar la veracidad de la información.

1.7.7 La Transmisión de Mensajes

La transmisión se produce con octetos. Cada octeto numeraremos los bits de izquierda a derecha, empezando por 0, siendo este el de mayor peso. En cuanto a los octetos, enviaremos en orden de significación. También podemos hacerlos en ASCII con paridad cero. Se utiliza TCP/IP sobre el puerto 53 (decimal). Retransmisión después de 2-5 segundos.

1.7.8 Formato del Mensaje de Dominio de Nombres

Este mensaje es usado por la aplicación, que debe comunicarse con una máquina y necesita resolver el nombre (que le ha introducido el usuario)

para hallar la dirección equivalente, la maquina lo mandara a un servidor de nombres local y este le contestará con otro mensaje (menor de 512 caracteres):

Identificación (16 bits)	Parámetro (16 bits)
Numero de pregunta	Numero de respuesta
Numero de autoridad	Numero de añadidos
Sección de pregunta Sección de respuesta Sección de Autoridad Sección de añadidos	

- Identificación: Usado por el remitente para comparar respuestas y preguntas.

- ***Parámetro:*** Especifica la operación pedida y el código de respuesta (ordenados los bits de izquierda a derecha):

Bit		significado
0	Operación:	0 Pregunta 1 Respuesta
1-4	Tipo de Pregunta:	0 Estándar 1 Inversa 2 Obsoleta (Terminación 1) 3 Obsoleta (Terminación 2)
5		1 Pregunta de autoridad
6		1 Mensaje Truncado
7		1 Se desea recursión
8		1 Recursión disponible
9-11		Reservado
12-15	Tipo de Respuesta:	0 Sin error

		1 error de formato en pregunta 2 Fallo de servidor 3 Nombre no existe

- **Numero de Preguntas**: Numero de entradas en la sección pregunta.
- **Numero de Respuestas**: Numero de entradas en la sección respuestas.
- **Numero de Autoridad**: Numero de entradas en la sección autoridad.
- **Numero de Añadidos**: Numero de entradas en la sección añadidos.
- **Sección de Pregunta**: Preguntas sobre las que se solicita respuesta.

El formato de cada pregunta es:

Dominio de Nombres de la Pregunta (32 bits)	
Tipo de pregunta (16 bits)	Clase de pregunta (16 bits)

- Dominio de nombres de la pregunta: Contiene el nombre solicitado. El primer octeto indica la longitud de cada etiqueta (en octetos). La ultima etiqueta es de longitud 0 para indicar el fin del nombre.

- Tipo de pregunta: ¿Es una maquina?, ¿es una dirección de correo?...

- Clase de pregunta: Nos permite usar este mensaje para otras direcciones que no sean Internet.

- Sección de respuesta: El servidor responderá a cada pregunta de la sección anterior, con una respuesta en esta sección. El formato de las secciones Respuesta, Autoridad y Añadidos es:

Petición Dominio de Nombre (32 bits)	
Tipo (16 bits)	Clase (16 bits)
TTL	Longitud de Petición
Datos de Petición …	

- **Petición Dominio de Nombre**: Nombre propio (del nodo que pide la resolución).

- **Tipo**: Petición disponible, según:

Tipo	**valor**	**significado y contenido**
A	1	Dirección de Host: Dirección IP 32 bits.
NS	2	Servidor de Nombres autorizado para el dominio.
MD	3	Obsoleto (destino de correo).
MF	4	Obsoleto (fuente de correo).
CNAME	5	Nombre canónico de un dominio.
SOA	6	Inicio de autoridad: Especifica que parte de la jerarquía de nombres esta implementada por un servidor de nombres.
MB	7	Experimental: MDN (Mailbox domain name).
MG	8	Experimental: Miembro de grupo de correo.
MR	9	Experimental: Renombre del MDN.
NULL	10	Experimental: Nulo.
WKS	11	Descripción de servicio conocido bueno.
PTR	12	Nombre del dominio como puntero.
HINFO	13	Nombre de la CPU y del S.O.
MINFO	14	Información de un buzón o lista de correo.
MX	15	16 bits prioritarios y nombre del host que actúa como central de correo para ese dominio.
TXT	16	Texto arbitrario: cadena ASCII sin interpretación.
AAAA	28	Dirección de Host: Dirección IPv6 128 bits.
AXFR	252	Petición de transferencia de una zona.
MAILB	253	Petición de campos de correo (MB,MG,MR)
MAILA	254	Obsoleto: Petición resolución de correo.

*	255	Petición de todos los registros.

Los mas utilizados son A y MX.

- ***Clase***:

tipo	**valor**	**significado y contenido.**
IN	1	Internet.
CS	2	Obsoleta: CSNET.
CH	3	CHAOS.
HS	4	Hesiod.
	255	Ninguna clase.

- ***TTL***: Tiempo que debe mantenerse en la memoria del servidor de nombres local, numero entero positivo del tipo de con signo de 32 bits.

- ***Longitud de datos recurso***: Numero de octetos en la sección datos recurso, 16 bits integer.

- ***Datos Recurso***: Aquí se hayan las respuesta a la pregunta solicitada.

1.8 Direccionamiento IPv4 En la actualidad. La estrategia CIDR

1.8.1 Preliminares

En los años 1992-1993 se planteo el problema del espectacular crecimiento de Internet (en mayúscula). Los problemas planteados eran:

- Se ve cercano el agotamiento de las direcciones Clase B. Las direcciones Clase C sólo permiten 255 *hosts*, mientras que las de Clase B permiten 65535. En la mayoría de las redes, la Clase C resulta demasiado pequeña, y la B demasiado grande.

- El crecimiento de las tablas de enrutamiento en los *routers* de Internet empieza a hacerlas intratables para el software y hardware existente.

- El espacio de 32 bits para direcciones comienza a resultar escaso.

Los dos primeros problemas son los que se intentan resolver mediante la estrategia CIDR, que explicaremos a continuación. El tercer problema (el

agotamiento del espacio de 32 bits para direcciones), resulta irresoluble en el marco de la versión 4 del protocolo IP.

Durante los años 1992-93 se produce una transición en Internet hacia CIDR, para solucionar los dos primeros problemas.

1.8.2 La estrategia CIDR (Classless Inter-Domain Routing)

La idea básica de la estrategia CIDR es la asignación de uno o más bloques de números de Clase C, y la introducción de una máscara que identifique al conjunto de direcciones. Además de frenar el agotamiento de las direcciones Clase B, conseguimos una estructura jerárquica que ayuda a frenar el crecimiento de las tablas de enrutamiento.

Un prefijo de red deja de ser fijo, dependiendo de la clase de dirección (A/B/C), y pasa a consistir de una tupla <Dirección IP-Máscara IP>.

Veamos esto con algunos ejemplos:

Imaginemos un sistema que requiere direccionar menos de 1024 *hosts*, a este sistema se le asignarían 4 direcciones Clase C, Por ejemplo, de la 192.24.8.0 a la 192.24.11.0. Para referenciar al conjunto se usaría la dirección 192.24.8.0, con una máscara de dirección 255.255.252.0

Un sistema que requiriera menos de 512 direcciones de *host* tendría asignadas 2 direcciones Clase C, por ejemplo la 192.24.34.0 y la 192.24.35.0 Para referenciar el conjunto tendríamos la dirección 192.24.34.0 con la máscara 255.255.254.0

Como consecuencia de la implantación de la estrategia CIDR, se elimina el concepto de clases de dirección, y tenemos prefijos de red de longitud variable, longitud que viene especificada por la máscara de dirección.

CIDR ha tenido un gran impacto en los sistemas de enrutamiento en Internet. Los prefijos de red de longitud variable han permitido desarrollar múltiples niveles en el sistema de direccionamiento, introduciendo el concepto de *super redes*, esto es, una adecuada asignación y gestión de las máscaras de dirección permite referenciar

grupos de redes. El objetivo final es que la estructura de direcciones IP refleje en la mayor medida de lo posible la topología de Internet.

2. Protocolo IP (versión 6)

2.1 Introducción

El protocolo IP en su versión 6 (IPv6, a partir de ahora), surge como un sucesor de la versión 4, que pronto se quedará corta debido al crecimiento exponencial de Internet.

Los cambios de IPv6 respecto de IPv4 son, de forma resumida:

1.- Expansión de las capacidades de direccionamiento.

IPv6 incrementa el tamaño de las direcciones de 32 bits (IPv4) a 128 bits, para soportar más niveles en la jerarquía de direccionamiento, un número mayor de nodos direccionables, y un sistema de autoconfiguración de direcciones. Se añade un nuevo tipo de dirección, la llamada *anycast*, de forma que es posible enviar un paquete a cualquier nodo entre un grupo de ellos.

2.- Simplificación de la cabecera.

Algunos campos de la cabecera del IPv4 son eliminados o pasan a ser opcionales, tanto para reducir el coste de procesamiento como el tamaño de la cabecera.

3.- Mayor flexibilidad para extensiones y nuevas opciones.

En IPv6 no existe un campo *opciones*, como tal. (ver *el datagrama IP*). La gestión de opciones se realiza por un campo *siguiente cabecera* (*next header*). Eliminando así las limitaciones de tamaño en la cabecera, e introduciendo una gran flexibilidad en el desarrollo de nuevas opciones.

4.- Capacidades de control de flujo.

Se añaden capacidades que permiten marcar los paquetes que pertenezcan a un determinado tipo de tráfico, para el cual el remitente demanda una calidad mayor a la especificada por defecto o servicios en tiempo real.

5.- Capacidades de autenticación y privacidad de datos.

IPv6 provee extensiones para soportar autenticación, e integridad y confidencialidad de datos.

2.2 Formato de la cabecera IPv6

El formato de la cabecera IPv6 es el siguiente:

<table>
<tr><td>Versión</td><td>Prioridad</td><td colspan="3">Etiqueta de flujo</td></tr>
<tr><td colspan="3">Longitud carga</td><td>Siguiente cabecera</td><td>Límite de saltos</td></tr>
<tr><td colspan="5">Dirección origen
"
"
"</td></tr>
<tr><td colspan="5">Dirección destino
"
"</td></tr>
</table>

- Versión.

Este campo ocupa 4 bits, e indica la versión de IP. Para el formato descrito, la versíon es la 6, para IPv6 (también llamada IPng, Internet Protocol Next Generation).

- Prioridad.

Este campo ocupa 4 bits, e indica la prioridad que el remitente desea para los paquetes enviados, respecto a los demás paquetes enviados por él mismo. Los valores de prioridad se dividen en dos rangos, de 0 a 7, paquetes para los cuales el remitente espera una respuesta en caso de congestión (p.e. tráfico TCP). Y de 8 hasta 15, paquetes que no deben ser respondidos en caso de congestión, el valor más bajo (8), se usaría cuando el remitente está dispuesto a que sus paquetes sean descartados en caso de congestión (p.e. Video en alta calidad). Y el valor más alto (15), cuando el remitente está muy poco dispuesto a que algún paquete sea descartado (p.e. Audio de baja calidad).

- Etiqueta de flujo.

Este campo ocupa 24 bits, y es usado por el remitente para indicar que sus paquetes sean tratados de forma especial por los routers, como en servicios de alta calidad o en tiempo real. En este punto, se entiende el flujo como un conjunto de paquetes que requieren un tratamiento especial.

Todos los paquetes pertenecientes al mismo flujo deben tener valores similares en los campos dirección origen, dirección destino, prioridad, y etiqueta de flujo.

- Longitud de la carga.

Este campo ocupa 16 bits, e indica la longitud del resto del paquete que sigue a la cabecera, en octetos. Si su valor es cero, indica que el tamaño de la carga vendrá especificado como *Carga Jumbo*, en una opción *salto a salto* (ver apartado *Cabeceras extendidas*).

- Siguiente cabecera.

Este campo ocupa 4 bits, e identifica el tipo de cabecera que sigue a la cabecera IPv6. Es coherente con los valores del campo *protocolo* en IPv4.

- Límite de saltos.

Este campo ocupa un octeto. Es decrementado en una unidad por cada nodo que redirige el paquete hacia su destino. El paquete es descartado si el valor del campo llega a cero. Este campo sustituye al campo *tiempo de vida*, de IPv4.

- Dirección origen.

Este campo ocupa 128 bits, y corresponde a la dirección de origen. Se describirá con detalle en apartado *Direccionamiento IPv6*.

- Dirección destino.

Este campo ocupa 128 bits, y corresponde a la dirección de destino. Se describirá con detalle en apartado *Direccionamiento IPv6*.

2.3 Cabeceras extendidas

2.3.1 Cabeceras extendidas. Introducción

En IPv6, diferente información es codificada en cabeceras separadas, entre la cabecera IPv6 y la cabecera del nivel superior.

Existen una serie de cabeceras extendidas, cada una identificada con un valor en el campo *siguiente cabecera*. Un paquete IPv6 puede contener ninguna, una o más cabeceras extendidas.

Con la única excepción de las *opciones salto a salto*, que serán descritas posteriormente, las cabeceras extendidas no son examinadas ni procesadas hasta que el paquete llega a su destino.

En las *opciones salto a salto*, hay información que es necesario procesar a lo largo del camino del datagrama, incluyendo los nodos origen y destino, cuando esta cabecera está presente, debe situarse inmediatamente después de la cabecera IPv6.

Las cabeceras extendidas deben de ser procesadas en el orden en que aparezcan, el receptor no puede buscar una cabecera en concreto y procesarla antes que las anteriores.

Si en el procesamiento de las cabeceras, un nodo se encuentra con un valor en *siguiente cabecera* que le es desconocido, el paquete debe ser descartado, y un nivel superior (ICMP), se encargará de enviar un error al origen.

Cada cabecera extendida debe tener una longitud en octetos múltiplo de 8, para mantener la alineación a 8 octetos a cabeceras posteriores.

La implementación de IPv6 incluye soporte para las siguientes opciones extendidas :

- Opciones salto a salto
- Enrutamiento.
- Fragmentación.
- Opciones en destino.
- Autenticación.

- Opciones de seguridad para la carga.

2.3.2 Orden de las cabeceras

Cuando existe más de una cabecera extendida en el mismo paquete, es recomendable que éstas aparezcan en el siguiente orden:

- Cabecera IPv6.
- Opciones salto a salto.
- Opciones en destino. Para opciones que deban ser procesadas por el destino final y por los destinos marcados en la cabecera de enrutamiento.
- Enrutamiento.
- Fragmentación.
- Autenticación
- Opciones de seguridad para la carga.
- Opciones en destino. Para opciones que sólo deban ser procesadas por el destino final.
- Cabecera del nivel superior.

Cada cabecera debe aparecen tan solo una vez, con la excepción de las *opciones de destino*, que pueden aparecen dos veces, en el orden indicado anteriormente.

Si la cabecera del nivel superior es otra cabecera IPv6, ésta ira seguida de sus propias cabeceras, ordenadas de la forma indicada.

Los nodos que soporten IPv6, deben aceptar y procesar las cabeceras en cualquier orden en que aparezcan, y también si aparecen dos o más veces, a excepción de las opciones salto a salto, que deben aparecer inmediatamente después de la cabecera IPv6.

De todos modos, se recomienda que los nodos que envíen paquetes IPv6 sigan el orden recomendado.

2.3.3 Opciones TLV (Type-Leght-Value)

Dos de las cabeceras definidas (*salto a salto* y *opciones de destino)*, llevan un número variable de opciones, las cuales a su vez tiene longitud variable (TLV, type-lenght-value). Estas opciones tienen la siguiente estructura :

Tipo de opción	Longitud datos opción	Datos
8 bits	8 bits	Long. Variable

- Tipo de opción.

Este campo ocupa 1 octeto, y actúa como identificador de cada opción específica.

- Longitud datos.

Este campo ocupa 1 octeto, e indica la longitud del campo de datos, medida en octetos.

- Datos.

Este campo tiene una longitud variable, en él se contienen los datos específicos de cada opción.

La secuencia de opciones (ya que pueden aparecer varias), debe ser procesada en el orden en que aparezcan. El receptor no puede examinar la cabecera en busca de una opción y procesarla antes que las anteriores.

El campo *Tipo de opción* está codificado de tal forma que los dos bits de mayor peso especifican las acciones a tomar en caso que el nodo no reconozca la opción :

00 Descartar la opción y seguir procesando el paquete

01 Descartar el paquete entero

10 Descartar el paquete, y enviar un mensaje de error ICMP al origen.

11 Descartar el paquete, y enviar un mensaje de error ICMP al origen, si y sólo si el paquete no tiene una dirección destino *multicast.*

El tercer bit de mayor peso especifica si los datos específicos de la opción pueden cambiar durante el recorrido del paquete. Esto es

útil cuando existe una cabecera de autenticación. Cualquier mecanismo de autenticación deberá tomar como 0's los datos que puedan cambiar en ruta. Los valores son :

0 Datos de la opción NO pueden cambiar en ruta.

1 Datos de la opción pueden cambiar en ruta.

Las diferentes opciones pueden tener diferentes requerimientos de alineamiento, estos requerimientos se especifican en la forma xn+y. Por ejemplo, 2n significa que la opción debe encontrarse desplazada del comienzo de la cabecera en un número de octetos múltiplo de 2.

Para mantener el alineamiento, existen dos opciones de relleno. Si sólo se requiere un octeto, este se coloca todo a 0's sin más. Si se necesita más de un octeto, se usa la siguiente estructura:

Todo a 1's	Longitud relleno	Todo a 0's
8 bits	8 bits	(longitud relleno)-2 octetos

La longitud se especifica en octetos, sin tener en cuenta el preámbulo y el campo de longitud, por tanto, el número de octetos en el relleno propiamente dicho es longitud-2.

2.3.4 Cabecera de opciones Salto a Salto

La cabecera extendida de *Opciones Salto a Salto* se usa para contener información que deberá ser examinada por cada nodo que encamine el paquete hacia su destino. Este tipo de cabecera se identifica con valor 0 en el campo *siguiente cabecera*.

Su formato es el siguiente :

Siguiente cabecera	Longitud opciones	Opciones
8 bits	8 bits	Long. Variable

- Siguiente cabecera.

Este campo ocupa 1 octeto, e identifica el tipo de cabecera existente inmediatamente después de la de *Opciones Salto a Salto.*

- Longitud opciones.

Este campo ocupa 1 octeto, e indica la longitud de la cabecera, en octetos, sin incluir los ocho primeros.

- Opciones.

Este campo es de longitud variable, y contiene opciones del tipo descrito en el punto *Opciones.*

Además de las opciones con la estructura descrita, existe una opción especial, la *Carga Jumbo*. Con la siguiente estructura :

194 (identificador)	4 (long. opciones)	Longitud Carga Jumbo
8 bits	8 bits	32 bits

La opción *Carga Jumbo*, es utilizada para enviar paquetes con cargas superiores a los 65535 octetos. La longitud especificada por la *Carga Jumbo* es el tamaño total del paquete, excluyendo la cabecera IPv6, e incluyendo la cabecera de *Opciones Salto a Salto.*

La longitud determinada debe ser siempre superior a 65535, si se recibe un paquete con una *Carga Jumbo* que indique un tamaño de paquete igual o menor a 65535, ICMP se encargará de enviar un error.

Cada paquete cuya longitud esté especificada por un opción *Carga Jumbo*, debe tener a 0 el campo *longitud de la carga* en la cabecera IPv6, además, la opción *Carga Jumbo* no puede ser usada en un paquete conteniendo un fragmento. El incumplimiento de cualquiera de estas restricciones provocara un error ICMP.

2.3.5 Cabecera de enrutamiento

La *Cabecera de Enrutamiento*, es utilizada por el remitente para indicar uno o más nodos que el paquete debe visitar en su recorrido. Su formato es el siguiente :

Siguiente cabecera	Longitud cabecera	Tipo enrutamiento	Nodos restantes	Datos

8 bits	8 bits	8 bits	8 bits	Long.Variable

- Siguiente cabecera.

Este campo ocupa 1 octeto, e identifica el tipo de cabecera siguiente.

- Longitud opciones.

Este campo ocupa 1 octeto, e indica la longitud de la cabecera, en octetos, sin incluir los ocho primeros.

- Tipo de enrutamiento.

Este campo ocupa 1 octeto, e indica el tipo particular de cabecera de enrutamiento.

- Nodos restantes.

Este campo ocupa 1 octeto, e indica el número de nodos que restan por visitar, siempre sobre los nodos marcados explícitamente.

- Datos.

Este campo tiene un longitud variable, siempre múltiplo de 8 (en octetos), y su formato viene determinado por el tipo de enrutamiento específico.

Si un nodo encuentra un paquete con un tipo de enrutamiento desconocido, tomará alguna de estas dos medidas :

- Si el número de nodos restantes es cero, se ignora la cabecera y se pasa a la cabecera siguiente.

- Si el número de nodos restantes NO es cero, se descarta el paquete y se enviará un error ICMP

El Tipo 0 (*tipo de enrutamiento* = 0) tiene el siguiente formato:

Sig. Cabecera (8 bits)	Long. Cab. (8 bits)	Tipo (= 0) (8 bits)	Nodos rest. (8 bits)	Reservado (8 bits)	Req. vecinos (24 bits)
Primera Dirección					
"					
[…]					
[…]					

n-ésima Dirección
"

La longitud de la cabecera se especifica en unidades de 8 octetos, sin incluir los 8 primeros octetos, para el Tipo 0, es igual al doble de direcciones especificadas, y deber ser un número par menor o igual a 46, de igual forma, el máximo número de nodos que pueden especificarse es 23.

Existe un campo marcado como *reservado*, el remitente debe ponerlo a cero, y es ignorado por el receptor.

El campo *requerimiento vecinos* (*Strict / Loose Bit Map)* ocupa 24 bits, y es interpretado bit a bit, de izquierda a derecha, para bit, indica si la dirección especificada debe corresponder a un nodo vecino del anterior. 1 significa que el nodo debe ser vecino del anterior, 0 significa que no ha de serlo necesariamente.

Las direcciones a visitar se especifican una tras otra, se numeran de 1 a n y pueden aparecer como máximo 23.

En el Tipo 0 no pueden aparecer direcciones *multicast*. Si un paquete IPv6 tiene como destino una dirección *multicast,* no puede contener una cabecera de enrutamiento de Tipo 0.

2.3.6 Cabecera de fragmento

La *Cabecera de fragmento* es utilizada por el origen del paquete IPv6 para enviar paquetes cuyo tamaño excede el mínimo MTU (*Maximum Transmission Unit)* en el camino del paquete. Al contrario que en IPv4, la fragmentación sólo la lleva a cabo el origen. La cabecera sigue e siguiente formato :

Sig. Cabecera	Reservado 1	Ofsset fragmento	Reservado 2	flag M	Identificación
8 bits	8 bits	13 bits	2 bits	1 bit	32 bits

- Siguiente cabecera.

Este campo ocupa 1 octeto, e indica el tipo de la cabecera siguiente.

- Reservado 1.

Este campo ocupa 1 octeto. El origen lo pone a 0's y es ignorado en destino.

- Offset de fragmento.

Este campo ocupa 13 bits, e indica, en unidades de 8 octetos, el desplazamiento respecto de la parte fragmentable del paquete original.

- Reservado 2.

Este campo ocupa 2 bits. El origen lo pone a 0's y es ignorado en destino.

- Flag M.

Este campo ocupa un bit, es el flag de *más fragmentos*. Si 1, indica que quedan más fragmentos, si 0, indica que es el último fragmento.

- Identificación.

Este campo ocupa 32 bits, y sirve para identificar los fragmentos pertenecientes al datagrama original.

Ver apartado *Fragmentación en IPv6.*

2.3.7 Cabecera de opciones en destino

Esta cabecera se usa para contener información que sólo debe ser examinada por el nodo destino. La cabecera tiene el siguiente formato :

Siguiente Cabecera	Longitud cabecera	Opciones
8 bits	8 bits	Long. Variable

- Siguiente Cabecera.

Este campo ocupa 1 octeto, e indica el tipo de la cabecera siguiente.

- Longitud cabecera.

Este campo ocupa 1 octeto, e indica la longitud de la cabecera en unidades de 8 octetos, sin incluir los 8 primeros.

- Opciones.

Este campo tiene una longitud variable, siempre alineada a 8 octetos. Contiene una o más opciones de la estructura descrita en el apartado 2.3.3 *Opciones TLV*

Hay dos posibles formas de codificar opciones TLV, como una opción en la cabecera *Opciones en destino*, o como una cabecera extendida aparte.

Elegir una forma u otra dependerá de cual sea la acción deseada si el destino no entiende la información.

- Si se quiere que el destino, en caso de no reconocer la opción, descarte el paquete y envíe un error ICMP (sólo si la dirección destino no es *multicast)*. Entonces la opción deberá ser codificada en una cabecera extendida aparte.

- Si se requiere cualquier otra acción, entonces la opción se codificará dentro de una cabecera *Opciones en destino*, y la acción especificada vendrá dada por los dos bit de mayor peso del campo *Tipo de opción*, en la forma descrita en el apartado *Opciones TLV*.

2.3.8 No más cabeceras

El valor 59 en el campo *siguiente cabecera* de la cabecera IPv6 o extendida indica que no sigue ninguna cabecera.

2.4 Fragmentación en IPv6

Al contrario que en IPv4, el fragmentado de un paquete sólo lo puede llevar a cabo el origen, con lo que se obvia el flag *no fragmentable*, de IPv4 (ver *Fragmentación* y *Flags)*.

Para cada paquete que deba ser fragmentado, el origen le asigna un *identificador* (ver *Cabecera de fragmento)*, este identificador debe ser diferente del de cualquier otro paquete enviado recientemente con la mismas direcciones origen y destino.

El paquete original se diferencia en dos partes, fragmentable y no fragmentable.

La parte no fragmentable consiste en la cabecera IPv6 y las cabeceras extendidas que deban ser procesadas por los nodos intermedios en el camino del paquete.

La parte fragmentable consta del resto de cabeceras extendidas, de la cabecera del nivel superior y de la carga.

El paquete original se descompone en fragmentos cuya longitud debe estar alineada a 8 octetos (excepto el último). La parte no fragmentable del paquete original se copia a todos sus fragmentos, cambiando el campo *longitud de la carga* a la longitud de cada fragmento y el campo *siguiente cabecera* a 44 (valor que identifica a una cabecera de fragmento).

Cada fragmento está compuesto por:

- La parte no fragmentable del paquete original.
- La cabecera de fragmento. Ver apartado *Cabecera de fragmento*
- El fragmento propiamente dicho.

En destino, el paquete original es contruido según las siguiente normas:

- Los fragmentos del paquete original deben contener los mismos valores en los campos *Dirección origen, dirección destino* y *identificador de fragmento*.
- El campo *siguiente cabecera* de la cabecera IPv6 se obtiene del campo *siguiente cabecera* de la cabecera de fragmento del primer fragmento.
- El campo *tamaño de la carga* del datagrama original se calcula en base al tamaño de la parte no fragmentable, y al tamaño y *offset de fragmento* del último fragmento.

En el proceso de reensamblado, pueden producirse los siguientes errores:

- Si se ha recibido un número de fragmentos insuficientes para recomponer el paquete original pasados 60 segundos desde el primer fragmento recibido, se abandona el proceso y se descartan todos los fragmentos recibidos. Si se recibió el primer fragmento (*offset de fragmento* = 0), se envía un mensaje ICMP de error al origen.
- Si la longitud de un fragmento en octetos no es múltiplo de 8 y no es el último fragmento, se descarta el fragmento y se envía un mensaje ICMP de error al origen.
- Si la longitud y el *offset de fragmento* de un fragmento determinan que la longitud de la carga del paquete original es mayor de 65535 octetos, se descarta el fragmento y se envía un mensaje ICMP de error al origen.

2.5 Direccionamiento IPv6

2.5.1 Introducción. Tipos de direcciones

Las direcciones en IPv6 son identificadores de 128 bits para un interface o conjunto de interfaces, existen 3 tipos de direcciones IPv6 :

- *Unicast*. identificador para un solo interfaz. Un paquete IPv6 con una dirección destino *unicast* es encaminado a un único interfaz, especificado por la dirección.

- *Anycast*. identificador para un conjunto de interfaces. Un paquete IPv6 con una dirección destino *anycast* es encaminado a uno y sólo uno de los interfaces identificados por la dirección. El paquete será encaminado al interfaz más cercano, de acuerdo con las técnicas de medida de distancia de las estrategias de enrutamiento.

- Multicast. identificador para un conjunto de interfaces. Un paquete IPv6 con una dirección destino *multicast* es encaminado a todos y cada uno de los interfaces identificados por la dirección.

No existen direcciones *broadcast* en IPv6 (ver *Direcciones especiales)*, su función es realizada por las direcciones *multicast.*

Las direcciones IPv6 son asignadas a interfaces, no a nodos, cuando un nodo tiene más de un interfaz, el nodo puede direccionarse mediante la dirección de cualquiera de sus interfaces. Además, un interfaz puede tener asignada una o más direcciones, con dos excepciones :

- Un conjunto de interfaces puede tener asignada una sola dirección IPv6, esta agrupación elimina la posibilidad de que cada uno de los interfaces que comparten una dirección pueda tener asignada cualquier otra.

- *Los routers* pueden tener interfaces sin dirección asignada en enlaces PPP (ver IPv6 sobre PPP), los interfaces de enlaces PPP no necesitan dirección IP si no son origen o destino de datagramas IPv6.

2.5.2 Representación de direcciones IPv6

Existen 3 formas para representar direcciones IPv6 mediante cadenas de texto:

1.- La forma más indicada es mediante la estructura x:x:x:x:x:x:x:x, donde los valores x son los valores en hexadecimal de cada bloque de 16 bits de la dirección.

Ejemplos:

FEDC:BA09:6543:1234:FDCE:7564:BA98:7651

1080:0:0:0:8:800:200C:417A

2.- El segundo método permite agrupar largas series de 0's, para hacer más legibles las direcciones, el uso de "::" indica múltiples grupos de 16 bits a 0.

Ejemplos:

1080:0:0:0:8:800:200C:417A podría representarse como 1080::8:800:200C:417A

FF01:0:0:0:0:0:0:43 podría representarse como FF01::43

Sólo puede usarse "::" una vez en una dirección.

3.- El tercer método resulta el más indicado para representar direcciones IPv6 que contengan direcciones IPv4, los 2 últimos bloques de 16 bits se representan como 4 bloques de 8 bits mostrando sus valores en decimal, como en IPv4.

Ejemplos:

0:0:0:0:0:0:13.1.68.3 ó ::13.1.68.3

0:0:0:0:0:FFFF:129.144.52.38 ó ::FFFF:129.144.52.38

Los diferentes tipos de direcciones son especificados por los bits de mayor peso de la dirección, cada tipo tiene asignado un prefijo, de longitud variable para cada tipo

2.5.3 Direcciones Unicast

Una dirección *unicast* IPv6 tiene una estructura similar a una dirección IPv4 usando CIDR (ver *La estrategia CIDR).*

Existen múltiples formatos de dirección *unicast*, un nodo en Internet puede tener más o menos conocimiento de la estructura de las direcciones, dependiendo del papel que juegue en Internet. Como mínimo, un nodo considerará una dirección IPv6 como un identificador sin estructura interna.

Dirección IPv6
128 bits

Usando el valor de la máscara IP, pueden indicarse prefijos de red de longitud variable

Prefijo de red	Identificador de interfaz
n bits	(128-n) bits

Los nodos pueden tener un conocimiento más profundo de la jerarquía de direcciones, dependiendo del papel que desempeñen en la jerarquía de enrutamiento.

Ejemplos de direcciones *unicast :*

- Direcciones MAC (IEEE 802) para redes locales. (Ver *El protocolo ARP)*

Prefijo de grupo	Identificador de subred	Identificador de interfaz
n bits	(80-n) bits	48 bits

Siendo *Identificador de interfaz* la dirección MAC del interfaz. Para redes locales que no usen direcciones MAC, otros tipos de direcciones del nivel de enlace pueden ser usados.

- Para sistemas que requieran por su tamaño más niveles de jerarquía, la dirección puede dividirse en múltiples niveles, por ejemplo :

Identificador grupo	Identificador área	Identificador subred	Identificador interfaz
G bits	a bits	s bits	(128-g-a-s) bits

- Para direcciones basadas en proveedor, tenemos la siguiente estructura :

010	Id. registro	Id. proveedor	Id. suscriptor	Id. Intra-Suscriptor
3 bits	n bits	m bits	s bits	(125-n-m-s) bits

Esta estructura refleja la jerarquía, un registro asigna las direcciones de un grupo de proveedores de servicios (p.e. *backbones* o redes regionales), que asignan direcciones a sus suscriptores (p.e. *Sites* o campus universitarios), etc.

2.5.4 Direcciones especiales unicast

- Dirección 0:0:0:0:0:0:0:0, esta dirección no puede ser asignada a ningún nodo, de hecho, indica la ausencia de dirección. Puede usarse, por ejemplo, como dirección origen al inicializar nodos, antes de que éstos conozcan su propia dirección IP. En ningún caso podrá aparecer como dirección destino.

- Dirección 0:0:0:0:0:0:0:1, esta es la dirección del *bucle local*, puede ser usada por un nodo para enviarse un datagrama a él mismo. En ningún caso podrá aparecer como dirección origen. Un datagrama enviado a la dirección de *bucle local* no saldrá al medio, puede ser usada, por ejemplo, para comunicación entre los procesos de un nodo.

2.5.5 Direcciones unicast IPv6 conteniendo direcciones IPv4

Existen dos formas de codificar direcciones IPv4 en direcciones IPv6. La primera se usa en nodos que puedan gestionar ambos protocolos, tanto IPv6 como IPv4. Las direcciones se codificaran de la siguiente forma:

Todo a 0's	Todo a 0's	Dirección IPv4
80 bits	16 bits	32 bits

La segunda forma se usa para representar las direcciones de nodos que sólo soporten IPv4, antes de la conversión de IPv6 a IPv4, el datagrama llevará una dirección con la siguiente estructura:

Todo a 0's	Todo a 1's	Dirección IPv4
80 bits	16 bits	32 bits

2.5.6 Uso local de direcciones unicast IPv6

Existen dos tipo de direcciones IPv6 de uso local. Estos tipos son el *enlace local (Link-Local)* y el *grupo local* (*Site-Local).*

La estructura de dirección *enlace local* es la siguiente:

1111111010	Todo a 0's	Identificador interfaz
10 bits	n bits	(118-n) bits

Las direcciones de *enlace local* son usadas para direccionar un sólo enlace, para diferentes propósitos, como autoconfiguración de direcciones, descubrimiento de vecinos, o cuando no existe un *router.*

La estructura de dirección *Site-Local* es la siguiente:

1111111011	Todo a 0's	Identificador subred	Identificador interfaz
10 bits	n bits	m bits	(118-n-m) bits

Las direcciones *Site-Local* se usan en grupos de redes que no disponen de una conexión a Internet, no necesitando un prefijo de dirección para su direccionamiento en Internet. En el momento en que el grupo se conecte a Internet, el prefijo de *Site-Local* será sustituido por un prefijo que identifique al grupo en la estructura global de Internet.

2.5.7 Direcciones anycast

Una dirección IPv6 *unicast* es una dirección asignada a un grupo de interfaces, con la particularidad de que un paquete con una dirección *unicast* es llevada a sólo un interfaz, que será el mas cercano según las técnicas de medida de distancia en las estrategias de enrutamiento.

Las direcciones *anycast* usan los mismos formatos definidos para direcciones *unicast*, con la diferencia de que el campo *identificador de interfaz* estará todo a 0's.

Prefijo de subred	Todo a 0's
n bits	(128-n) bits

Una dirección *anycast* no podrá nunca aparecer como *dirección origen* en un paquete IPv6, ni podrá ser asignada a ningún *host.* Las direcciones *anycast* sólo podrán ser asignadas a un *router*.

2.5.8 Direcciones multicast

Una dirección IPv6 *multicast* identifica a un conjunto de nodos. Un nodo puede pertenecer a cualquier numero de conjuntos *multicast.*

Las direcciones *multicast* tienen la siguiente estructura :

11111111	Flags	Ambito	Identificador de grupo
8 bits	4 bits	4 bits	112 bits

- Flags.

Este campo ocupa 4 bits, los tres bits de mayor peso son reservados, y deben ser inicializados a 0. El cuarto bit indica si la dirección es fija o no.

4º bit a 0 La dirección es fija.

4º bit a 1 La dirección NO es fija (transición).

- Ambito.

Este campo ocupa 4 bits, e indica el ámbito del grupo *multicast*. Los valores son:

0 Reservado

1 Nodo local.

2 Enlace local.

3 Sin asignar

4 Sin asignar

5 *Site* local

6 Sin asignar

7 Sin asignar

8 Organización local

9 Sin asignar

A Sin asignar

B Sin asignar

C Sin asignar

D Sin asignar

E Global

F Reservado

- Identificador de grupo.

Este campo ocupa 112 bits, e identifica al grupo en el ámbito indicado, sea fijo o de transición. Las direcciones fijas tienen un significado independiente del ámbito que se indique.

2.6 DNS para IPv6

La resolución de direcciones DNS funciona de forma similar a los mecanismos vistos para IPv4, Además, se han añadido algunas características, que comentamos a continuación.

Las extensiones son:

- Un nuevo tipo de petición para soportar direcciones IPv6.

- Un nuevo dominio.

- Todos los procesos adicionales que se requerían para localizar direcciones

IPv4 son redefinidos para localización de direcciones tanto IPv4 como IPv6.

- La nueva petición:

Se llama AAAA y es el tipo 28 (en decimal). El formato de datos de AAAA es una dirección IPv6 de 128 bits, la resolución de este campo en la red es la de mayor peso el primer octeto.

- Nuevo dominio:

El IP6.INT es el nuevo dominio para IPv6, una dirección IPv6 se representa con un nombre en este dominio. La secuencia anterior a .IP6.INT se codifica de forma inversa. Por ejemplo la dirección IPv6:

4321:0:1:2:3:4:567:89ab

seria representada por:

b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT

CONCLUSIONES

Un conjunto de ordenadores conectados entre ellos que están situados por todo el mundo (más de 3.000.000 de ordenadores). La información contenida en cada uno de estos ordenadores es accesible desde cualquier otro ordenador conectado a esta red.

No existe ninguna compañía que se llame Internet. No existe ninguna organización que imponga reglas. Aquellas organizaciones que quieren conectar sus ordenadores a Internet no tienen mas que engancharse a otro ordenador que, a su vez, esté en Internet. Cada organización es responsable de sus propios ordenadores y de sus conexiones. Su mandato acaba donde terminan sus propios cables.

En Internet, las comunicaciones concretas se establecen entre dos puntos: uno es el ordenador personal desde el que usted accede y el otro es cualquiera de los servidores que hay en la Red y facilitan información.

El fundamento de Internet es el TCP/IP, un protocolo de transmisión que asigna a cada máquina que se conecta un número específico, llamado “Numero IP” (que actúa a modo de "número teléfono único") como por ejemplo 192.555.26.11.

El protocolo TCP/IP sirve para establecer una comunicación entre dos puntos remotos mediante el envío de información en paquetes. Al transmitir un mensaje o una página con imágenes, por ejemplo, el bloque completo de datos se divide en pequeños bloques que viajan de un punto a otro de la red, entre dos números IP determinados, siguiendo cualquiera de las posibles rutas. La información viaja por muchos ordenadores intermedios a modo de repetidores hasta alcanzar su destino, lugar en el que todos los paquetes se reúnen, reordenan y convierten en la información original. Millones de comunicaciones se establecen entre puntos distintos cada día, pasando por cientos de ordenadores intermedios.

La gran ventaja del TCP/IP es que es inteligente. Como cada intercambio de datos está marcado con números IP determinados, las comunicaciones no tienen por qué cruzarse. Y si los paquetes no encuentran una ruta directa, los

ordenadores intermedios prueban vías alternativas. Se realizan comprobaciones en cada bloque para que la información llegue intacta, y en caso de que se pierda alguno, el protocolo lo solicita de nuevo hasta que se obtiene la información completa.

TCP/IP es la base de todas las máquinas y software sobre el que funciona Internet: los programas de correo electrónico, transferencia de archivos y transmisión de páginas con texto e imágenes y enlaces de hipertexto. Cuando es necesario, un servicio automático llamado DNS convierte automáticamente esos crípticos números IP a palabras más inteligibles (como www.universidad.edu) para que sean fáciles de recordar.

Toda Internet funciona a través de TCP/IP, y razones históricas hacen que está muy ligado al sistema operativo Unix (y sus variantes). Por fortuna, los usuarios actuales no necesitan tener ningún conocimiento de los crípticos comandos Unix para poder navegar por la Red: todo lo que necesitan es un ratón.

La conclusión a la que se ha llegado tras realizar este trabajo ha sido la siguiente: El conjunto de protocolos TCP/IP ha sido de vital importancia para el desarrollo de las redes de comunicación, sobre todo para Internet. El ritmo de expansión de Internet también es una consecuencia de estos protocolos, sin los cuales, conectar redes de distintas naturalezas (diferente *Hardware*, sistema operativo, etc..), hubiera sido mucho mas difícil, por no decir imposible. Así pues, podemos decir que los protocolos TCP/IP fueron y son el motor necesario para que las redes en general, e Internet en particular, se mejoren y se pueda lograr una buena "autopista de la información".

BIBLIOGRAFÍA

UNIVERSIDAD POLITÉCNICA DE MADRID
www.master.etsit.upm.es

UNIVERSIDAD DEL AZUAY
www.uzauy.net.ec

Titulo: Domain Name System Structure an Delegation
RFC 1591
Organización: ISI
Fecha: Marzo 1994

Titulo: Domain Names, Concepts and Facilities
RFC 1034
Organización: ISI
Fecha: Noviembre 1987

Titulo: DNS Extensions to Support IPv6
RFC 1886
Organización: Bellcore, INRIA
Fecha: Diciembre 1994

Titulo: IPv4 Address Behaviour Today
RFC 2101
Organización: IAB.
Fecha: Febrero 1997

BIBLIOGRAFIA

Titulo: Curso de Sistemas de Comunicaciones Telemáticas. Módulo 5

Autor/es: Formación y Consultoría S.A.

Edita: INEM (Instituto Nacional de Empleo)

Fecha: 1994

Titulo: Sistemas para la Transmisión de Datos.

Autor/es: Fernando Torres Medina

Edita: Universidad de Alicante

Fecha: 1996

Titulo: Internet Protocol, DARPA Internet Program Protocol Specification.

RFC 0791

Organización: DARPA, Information Sciences Institute University of Southern California.

Fecha: Septiembre 1981

Titulo: Domain Name System Structure an Delegation

RFC 1591

Organización: ISI

Fecha: Marzo 1994

Titulo: Domain Names, Concepts and Facilities

RFC 1034

Organización: ISI

Fecha: Noviembre 1987

Titulo: IPv4 Address Behaviour Today

RFC 2101

Organización: IAB.

Fecha: Febrero 1997

Titulo: Supernetting: An Address Asignment and Aggregation Strategy.

RFC 1338

Organización: BARRNet, Cisco, Merit, OarNet.

Fecha: Junio 1992

Titulo: Classless Inter Domain Routing, CIDR: An Address Asignment an Aggregation

strategy.

RFC 1519

Organización: BARRNet, Cisco, Merit, OarNet.

Fecha: Septiembre 1993

Titulo: An Architecture for IP Address Allocation with CIDR

RFC 1518

Organización: Watson Research Center, IBM Corporation, Cisco Systems

Fecha: Septiembre 1993

Titulo: Internet Protocol, Version 6 (IPv6) Specification

RFC 1883

Organización: Xerox Parc, Ipsilon Networks

Fecha: Diciembre 1995

Titulo: IPv6 Addressing Architecture

RFC 1884

Organización: Xerox Parc, Ipsilon Networks

Fecha: Diciembre 1995

Titulo: An Architecture for IPv6 Unicast Address Allocation

RFC 1887

Organización: Cisco System

Fecha: Diciembre 1995

Titulo: DNS Extensions to Support IPv6

RFC 1886

Organización: Bellcore, INRIA

Fecha: Diciembre 1994

Titulo: The Point-to-Point Protocol (PPP)

RFC 1548

Organización: DayDreamer

Fecha: Julio 1994

Titulo: IPv6 over PPP

RFC 2023

Organización: Networks Inc.

Fecha: Octubre 1996

Titulo: PPP Internet Protocol Control Protocol Extensions for Name Server Addresses.

RFC 1877

Organización: Microsoft

Fecha: Diciembre 1995

Titulo: PPP Authentication Protocols

RFC 1334

Organización: DayDreamer

Fecha: Octubre 1992

Titulo: PPP Challenge Handshake Authentication Protocol

RFC 1994

Organización: DayDreamer

Fecha: Agosto 1996