El protocolo IP

El protocolo IP.

Descripción general.

Aunque mucha gente está conectada a Internet poca se para a pensar como es posible que su ordenador sea capaz de entenderse con otro que ni siquiera sabe como es y que se encuentra al otro lado del mundo. Esto es posible gracias a que todos los ordenadores conectados a esta red hablan un lenguaje común, una especie de "lingua franca", cuyo nombre TCP/IP es posible que también lo conozcan muchos, pero que pocos saben lo que es realmente.

TCP/IP, el conjunto de protocolos que es el fundamento de Internet, es la denominación que recibe una familia de protocolos diseñado para la interconexión de ordenadores, independiente de su arquitectura y del sistema operativo que ejecuten, de la tecnología usada a bajo nivel para conexión y que proporciona una conectividad universal a través de la red con reconocimiento de extremo a extremo.

Nació de la necesidad de conseguir dentro de ARPA, semilla de Internet, los objetivos citados anteriormente, y gracias a la expansión de Internet se ha convertido en un estándar de hecho debiendo parte de su gran popularidad al hecho de ir incorporado dentro de los sistemas operativos Unix, a su independencia del fabricante, a estar soportado en múltiples tecnologías y a que puede funcionar en máquinas de cualquier tamaño.

En 1969 DARPA (Defense Advanced Research Projects Agence de E.E.U.U.) fundó ARPANET, red experimental formada por una serie de ordenadores que se conectaban mediante la utilización de un sistema de conmutación de paquetes experimental, funcionando en principio los sistemas en una relación cliente-servidor, pero se decidió más tarde implementar un protocolo de igual a igual, que recibió el nombre de protocolo de control de red (NCP Network Control Protocolo).

A medida que ARPANET fue creciendo surgió la necesidad de simplificar el proceso de interconectar muchos tipos diferentes de ordenadores por lo que se planteó el objetivo de desarrollar un método de interconexión que cumpliese dos premisas fundamentales: pudiese conectar muchos tipos diferentes de ordenadores y pudiese funcionar sobre muchos medios de transmisión diferentes.

En 1973, Bob Kahn de DARPA y Vinton Cerf de la Universidad de Stanford empezaron el desarrollo del grupo de protocolos "Protocolos de control de transmisión" (TCP Transmisión Control Protocol), completado en su mayor parte hacia 1978 cuando recibió el nombre actual TCP/IP (Transmisión Control Protocol / Internet Protocol) debido a la necesidad de dividir el protocolo TCP en dos: uno orientado a la conexión (TCP) y otro no (IP).

A finales de los años 70 hubo un intento fallido para integrar TCP/IP dentro de los protocolos OSI, pero su gran difusión actual ha dado lugar a que sea realmente el estándar de interconexión en lugar del OSI.

Clases de direcciones. Subredes y máscaras.

IP trabaja con casi todos los protocolos de redes de área local y extensa, usando un esquema de direccionamiento independiente del esquema de direcciones de la red.

Cada nodo en una red IP tiene una dirección numérica de 4 bytes (32 bit). Esta dirección se suele representar por cuatro números entre 0 y 255 separados por puntos, y tiene dos partes, la primera comenzando por la izquierda representa la red, y la segunda representa al nodo en ella. La longitud de cada una de las partes no es fija sino que depende de la dirección. Atendiendo a esta diferencia las direcciones IP se clasifican en cinco clases. También hay que hacer constar que dentro del número de nodo, hay dos especiales, el que tiene todos los bits a 0 que representa a la red y el que tiene todos los bits a 1 que representa a todos los nodos.

Veamos un ejemplo de dirección IP en los dos formatos, numérico y binario:

115.8.3.45 == 01110011 00001000 00000011 00101101

Aunque las direcciones en formas de secuencia de bit son perfectamente entendidas por las máquinas, y su traslación numérica nos las hacen fácilmente manejables para nosotros, son difíciles de recordar; además de que si la maquina se mueve de red, su dirección varía. Sería más fácil para nosotros referirnos a ella por un nombre, y que nuestro ordenador averiguase la dirección. Esto es lo que hace el servicio de traducción de nombre, servicio implementado en la Internet como un servicio de directorio distribuido y jerarquizado en el que están registrados los nombres y direcciones de máquinas de forma que a partir de uno podemos encontrar el otro. Esta traducción se realiza de forma automática por la mayoría de los programas clientes de servicios que usamos en INTERNET, aunque existen herramientas específicas para interrogar al servicio de DNS que veremos al hablar de forma más extensa del mismo.

Como hemos dicho existen 5 clases de direcciones que se caracterizan por el valor de los bits más significativos de la parte de la dirección que corresponde a la red:

- Direcciones de clase A. Se caracterizan por tener a 0 el primer bit del campo de red que tiene una longitud de 8 bit, correspondiendo los otros 24 a la dirección del nodo. Las direcciones correspondientes van de la 0.0.0.0 a la 127.255.255.255 por lo que hay 128 redes de clase A que ya están todas asignadas en INTERNET. La red 10 esta reservada para crear redes ocultas dentro de las organizaciones, redes que no son visibles para el resto de la INTERNET.

0->127.255	
Parte de red	Parte de nodo
0 XXXXXXX	XXXXXXXX XXXXXXXX XXXXXXXX

- Direcciones de clase B. Se caracterizan por tener los dos primeros bit del campo de red con la secuencia 10 y la longitud del mismo es de 16 bit al igual que la del campo de nodo. Las direcciones correspondientes van de la 128.0.0.0 a la 191.255.255.255. La gran mayoría de estas 16.384 redes están ya asignadas por lo que es difícil conseguir una. Dentro de este rango, desde la 172.16.0.0 a la 172.31.0.0 están reservadas para la construcción de redes ocultas.

128.0 -> 191.255	
Parte de red	Parte de nodo
10 XXXXXX XXXXXXXX	XXXXXXXX XXXXXXXX

- Direcciones de clase C. Se caracterizan por tener un campo de red de 24 bit de longitud que comienza por 110 correspondiendo 8 bit a la dirección del nodo. Las direcciones correspondientes van de la 192.0.0.0 a la 223.255.255.255 de las que de la 192.168.0.0 a la 192.168.255.0 están reservadas para redes ocultas. Por tanto de clase C existen 2.097.152 redes.

192.0 -> 223.255	
Parte de red	Parte de nodo
110 XXXXX XXXXXXXX XXXXXXXX	XXXXXXXX

- Direcciones de clase D. Se caracterizan porque su dirección comienza con la secuencia de bit 1110 y corresponden a las direcciones desde la 224.0.0.0 a la 239.255.255.255. Estas direcciones reciben el nombre de “multicast” y en ellas desaparecen el concepto de red. Cada una no designa a un nodo sino a un grupo de nodo. Un paquete dirigido a una dirección “multicast” en entregado a todas las máquinas que componen el grupo.
- Direcciones de clase E. Se caracterizan porque su dirección comienza con la secuencia 1111 y van de la 240.0.0.0 a la 255.255.255.255. Son direcciones especiales, reservada por la

IANA y solo está asignada la 255.255.255.255 que corresponde a todas las máquinas conectada a un soporte físico.

De lo indicado anteriormente se podría deducir que una dirección IP identifica a una máquina lo que no es verdad siempre. Si tenemos un encaminador que está conectado a dos redes, cada una de las puertas tiene una dirección IP distinta y perteneciente a cada una de las redes. Aquí la dirección IP identifica a la puerta del nodo en la red. Por tanto pueden existir nodos con muchas direcciones IP.

Dentro de las direcciones posibles existen algunas que son consideradas especiales y que significan algo distinto de la dirección de un nodo:

- Dirección con todos los bit a 0 que identifica al propio nodo y que sólo puede usarse en el arranque del sistema.
- Dirección con todos los bit de red a 0 que identifica al nodo en la propia red y que sólo puede usarse también en el arranque del sistema.
- Dirección con todos los bit a 1, conocida como dirección de difusión de red local, y que permite enviar un mensaje a todos los nodos de una red local, estén o no en la misma red IP.
- Dirección con todos los bit del campo de nodo a 1, llamada dirección de multidifusión limitada a la propia red IP, que permite enviar un solo paquete a todos los nodos de la red IP del nodo emisor.
- Dirección 127.x.x.x, llamada de bucle local, utilizada para pruebas y comunicación entre procesos en la máquina local. Cualquier paquete enviado a esta dirección será entregada al propio nodo por el módulo ip sin enviar nada a la red.

El método de direccionamiento de Internet asigna a cada red física una red IP de alguna de las clases anteriores. Este tipo de asignación tiene dos problemas: el primero ocasionado por el crecimiento espectacular de Internet que da lugar a que no haya suficientes número de redes para asignar. Por otro lado, si a una red de sólo cinco equipos le asignamos una red de clase C completa estamos desperdiciando 250 direcciones.

Para resolver estos dos problemas se utilizan las subredes. Las subredes nacen de modificar conceptualmente el formato de la dirección ip que pasa de ser <dirección de red><dirección del nodo> a ser <dirección de red><dirección de subred><dirección de nodo> donde el campo subred se obtiene tomando una parte de la dirección del nodo. Esta división, en la que se altera sólo la parte local de la dirección permite establecer un direccionamiento jerárquico, permitiendo la gran flexibilidad de este método que cada

red física pueda escoger su propia subred, que deberá ser la misma para todos los nodos conectados. Veamos un ejemplo de cómo se realiza esta división:

Consideremos un encaminador con cuatro puertas a cada una de las cuales está conectada la subred 3.1.0.0, 3.2.0.0, 3.3.0.0, 3.4.0.0. Si consideramos la red de clase A 3.0.0.0 la dirección 3.1.0.1 tendría como parte de red [00000011] y de nodo [000000010000000000000001]. En cambio si consideramos la misma dirección como parte de la subred 3.1.0.0 la parte de red es ahora [00000011 00000001] y la de nodo es [00000000 00000001]. De manera análoga funciona para las otras subredes.

En el ejemplo anterior hemos visto que la división en subredes se hace tomando el primer byte de la parte de dirección de nodo como dirección de subred, pero esto no es necesario hacerlo siempre así: podemos dividir en dos trozos cualesquiera el campo de dirección de nodo. Hemos de tener en cuenta que de cada una de las subredes que se obtienen perdemos dos direcciones, las que tiene nodos los bit a cero que identifica a la subred, y la que tiene todos los bit a 1.

Las posibilidades de partición explicada anteriormente hacen necesario un método para que los algoritmos de encaminamiento puedan distinguir las dos partes de la dirección. Este mecanismo, llamado máscara, es un campo de 32 bit que se añada a la dirección IP que siempre está formado por un grupo de bit a 1 que identifica la parte de red de la dirección IP y otro grupo de bit a 0 que representa la parte de nodo de la dirección IP. En el ejemplo anterior las máscaras para la red 3.0.0.0 considerada como una red de clase A es 11111111 00000000 00000000 00000000 mientras que si la consideramos dividida en subredes la máscara es 11111111 11111111 00000000 00000000.

Esta representación binaria de las máscaras es incómoda por lo que se suele emplear una notación decimal semejante a la de las direcciones IP. Así las dos máscaras anteriores serían ahora 255.0.0.0 y 255.255.0.0 que se corresponden con las máscaras genéricas para redes de clase A y B respectivamente. La máscara correspondiente a redes de clase C es 255.255.255.0.

La tabla de equivalencia entre notación binaria y decimal de máscaras es la siguiente:

00000000	0
10000000	128
11000000	192
11100000	224
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

Una herramienta muy cómoda para calcular subredes y datos relativos a ellas, como dirección de red, dirección de difusión, número de nodos, direcciónes de los nodos, etc. es el programa para Windows IP subnet Calculator que se incluye en este cdrom.

Modelo de niveles. La pila IP. Equivalencia con la pila OSI.

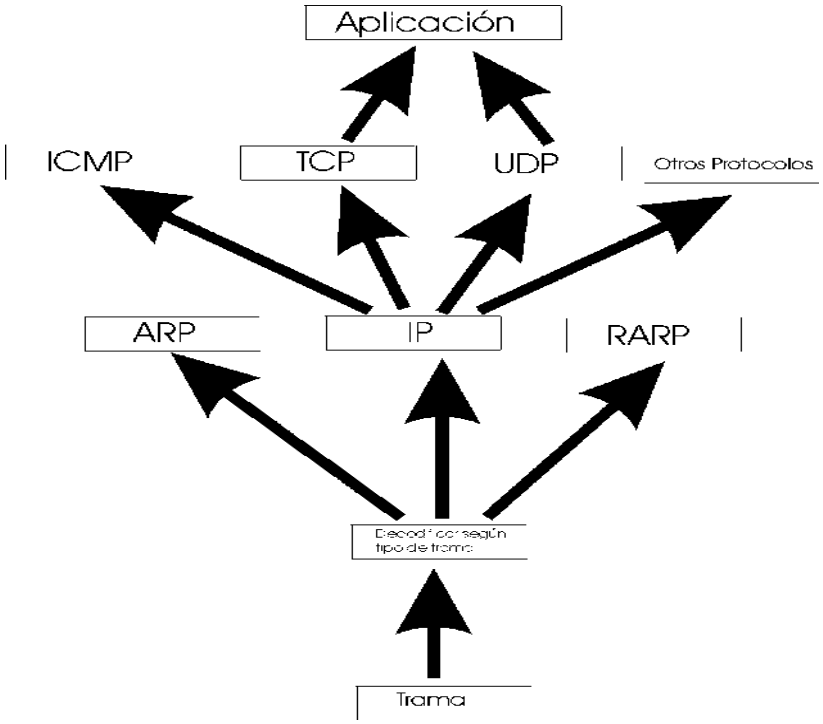
El conjunto de protocolos TCP/IP funciona con una estructura de niveles, apoyándose unos en otros, de la misma forma en que lo hacen los distintos niveles del modelo OSI. Sin embargo en el caso de TCP/IP los niveles que existen son sólo tres, y en cada uno de ellos puede existir más de un protocolo.

La siguiente tabla nos da una idea de la organización de la pila IP y su comparación con la pila OSI.

Aplicación	SNMP, FTP, SMTP, TELNET, DNS, HTTP, NTP
Presentación	
Sesión	
Transporte	TCP,UDP,ICMP,BOOTP
Red	IP,ARP,RARP
Enlace	
Físico	

Tenemos que tener en cuenta que debido a las condiciones puestas en su desarrollo con respecto al modo de transmisión que no tiene nivel físico ni de enlace de datos apoyándose en los estándares existentes para las diferentes tecnologías.

Una visión más clara de la jerarquía de funciones de los protocolos TCP/IP se ven en el siguiente diagrama:



Vamos a estudiar ahora los distintos protocolos existentes por capas si bien, en el caso de los protocolos que se pueden consideran pertenecientes a la capa de aplicación, sólo hablaremos de aquellos que no se tratan en otra parte de este curso.

Nivel de red.

Protocolo IP.

Dentro del nivel de red el protocolo básico es el IP (Internet Protocol), que es un protocolo de comunicación sin conexión, que proporciona un servicio de datagramas. IP se ocupa de la transmisión de los datagramas en función de las direccion de destino que va incorporada en la cabecera del mismo.

Dos son las funciones básicas que implementa el protocolo IP: el direccionamiento y la fragmentación.

Mediante el direccionamiento, el protocolo IP sabe encontrar un camino para el datagrama a fin de que llegue a su destino. Para ello está implementado no sólo en los nodos finales, sino también en los encaminadores que adicionalmente están provisto de mecanismos para tomar decisiones sobre el enrutamiento de los datagramas. Para conseguir este encaminamiento el IP debe encapsular los datos entregados por el protocolo de nivel superior poniéndole una cabecera propia en la que los datos más importantes son las direcciones de origen y destino; el protocolo encapsulado, necesario para saber a quien entregar el paquete en la máquina de destino, ya que cada protocolo de nivel superior genera e interpreta su propio formato de cabecera y los datagramas no son intercambiables, siendo también función de estos protocolos el reordenamiento y la verificación de perdidas de datagramas que pueden ocurrir ya que Ip maneja la congestión con el descarte de datagramas; y el código de verificación del contenido de la cabecera (Checksum) y sólo de la cabecera porque es más eficaz que el protocolo de nivel superior genere y verifique el código corrector de los datos, ya que de esta forma el tráfico de los datagramas en la red es mucho más ágil porque el código de verificación de Ip hay que recalcularlo en el paso por cada encaminador.

La segunda función, la fragmentación de los datagramas es necesaria ya que como los paquetes pueden circular por redes distinta, con distintos tamaños máximos de paquete (MTU, Maximun Transmission Unit), puede ser necesario su troceado en otros más pequeños.

La estructura general de una datagrama Ip es la siguiente donde cada una de las filas representa una palabra de 32 bit:

<table>
<tr><td>Vers</td><td>Header Length</td><td colspan="2">Type of Service</td><td>Total Length</td></tr>
<tr><td colspan="2">Identification</td><td>Flags</td><td colspan="2">Frangment offset</td></tr>
<tr><td>Time to live</td><td>Protocol</td><td colspan="3">Header checksum</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
</table>

Destinatión IP address
IP options
Datos

Veamos ahora la descripción de cada uno de los campos:

- **Vers** (Versión). Este campo tiene cuatro bit de longitud e indica la versión del protocolo, habitualmente tiene el valor de 4 que corresponde a IP versión 4.
- **Header Length** (Longitud de la cabecera). Este campo, también de cuatro bit de longitud indica el tamaño de la cabecera medido en palabras de 32 bit, pudiendo tener un valor mínimo de 5 que corresponde a una cabecera correcta sin el campo opciones (que puede no estar presente) con una longitud de 160 bit, siendo 64 octetos la longitud máxima que puede tener.
- **Type of Service** (Tipo de servicio). Campo de 8 bit de longitud que indica como debe ser tratado el datagrama. Está estructurado de la siguiente manera:

Prioridad	D	T	R	--

  El campo prioridad, de 3 bit, cuyo valor oscila entre 0 (prioridad normal) y 7 (control de red) permite al nodo remitente indicar la importancia del paquete. Aunque la verdad es que la mayoría de los programas y de los encaminadores no hacen aún caso del valor de este campo, conceptualmente es importante al permitir priorizar los paquetes según su contenido: por ejemplo las órdenes de control antes que los datos. Además permite también implementar algoritmos de control de congestión que no se vean afectados por la misma congestión.

  Los siguientes tres bit indican la calidad del transporte solicitado por el datagrama: el primero, D, indica un retardo bajo (low delay), el segundo, T, una alta tase de paquetes por segundo (Througput) y el tercero, R, solicita una alta fiabilidad, que no se descarten paquetes (Reliability). Estos bit constituyen una ayuda al encaminamiento cuando existen caminos alternativos.
- **Total Length** (Longitud total). Los 16 bit de este campo indican la longitud total del datagrama que no puede exceder de 65535 octetos (byte, 8 bit) ni se recomienda sea menor de 576 octetos.
- **Identification** (Identificación). Campo de 16 bit que contiene un número entero, que identifica al datagrama dentro de la secuencia en que fue emitido

por el nodo origen, en definitiva, el número de orden del datagrama en su emisión y por tanto indispensable para reensamblar los trozos cuando un datagrama fue fragmentado.

- **Flags** (Marcador, semáforo, bandera). Este campo de tres bit de los cuales el primero debe ser 0 ofrece información sobre la fragmentación del datagrama. Así el segundo bit del campo (DF) indica si el datagrama puede fragmentarse, valor 0, o no, valor 1. El tercer bit (MF) indica si el actual es el último trozo de un datagrama, valor 0, o si hay más, valor 1.
- **Fragment offset** (Situación del fragmento). Este campo tiene trece bit de longitud e indica el número de octetos a contar desde el comienzo del campo de datos del datagrama original hasta el punto donde que hay que colocar el campo de datos del fragmento en que se encuentra. El valor del campo viene dado en múltiplos de 8 octetos.
- **Time to live** (TTL, tiempo de vida). Los ocho bit del campo indican en segundos cuanto tiempo está autorizado el datagrama a circular por la red, y su utilidad es evitar que los datagramas estén eternamente circulando por la red si las tablas de rutas no están correctas y sigue un camino en círculos. El procedimiento que se sigue para esto es: cuando un nodo pone un datagrama en la red, le asigna un tiempo máximo de vida, y cada nodo que procese el datagrama debe decrementar ese valor hasta que sea 0, momento en el que deberá ser retirado de la red. Esta decrementación se realiza restando 1 al campo al pasar por un encaminador o bien almacenando la hora local de llegada al encaminador y comparándola con la del momento de envío restándole del campo la diferencia en segundos entre las horas anteriores.
- **Protocol** (Protocolo). Campo de ocho bit que indica que protocolo de alto nivel ha creado el contenido del campo datos. La asignación de los valores a los protocolos se hace por la IANA y algunos valores existentes son: 1 para ICMP, 6 para TCP, 17 para UDP.
- **Header Checksum** (Código verificador de la cabecera). Campo de 16 bit que representa un valor calculado por el nodo origen a partir del contenido de la cabecera y que será recalculado nuevamente por el nodo receptor para verificar que la cabecera se ha recibido correctamente.
- **Source IP Address** (Dirección de origen). Campo de 32 bit que indica la dirección Ip del nodo emisor del datagrama.
- **Destination IP Address** (Dirección de destino). Campo de 32 bit que indica la dirección Ip del nodo a quien va dirigido el datagrama.

- **IP options** (Opciones de IP). Campo de longitud variable y opcional dentro de la cabecera del datagrama. Tiene una estructura variable, ya que dentro de él pueden ir varias opciones, cuya estructura también es variable. Debe terminar con el campo "fin de opción" que es un octeto con todos sus bit a cero.

  Como hemos dicho la estructura de cada una de las opciones es variable pero todas tienen un primer octeto llamado código de opción cuya estructura es la siguiente:

  El primer bit del campo es el llamado bit de copia y cuando está a 1 indica que la opción debe ser copiada en cada uno de los trozos en que se puede dividir el datagrama. Si está a 0 deberá ser copiada sólo en el primer trozo.

  Los siguientes dos bit constituye la llamada clase de opción que puede ser 00 para indicar datagrama o control de red, o 10 para indicar medida y control de errores estando aún sin definir los valores 01 y 11.

  Los siguientes cinco bit forman el campo número de opción e identifica la opción requerida.

  Entre las posibles opciones que se pueden incluir están:

  - **Sin opción**. Clase 0, número 1. De longitud un octeto y suele usarse entre otras opciones para alineamientos. Debe ser copiada en caso de fragmentación.
  - **Seguridad y restricciones de acceso**. Clase 0, número 2. De once octetos de longitud, está dividida en varios campos, que incluyen nivel de seguridad, restricciones de acceso, etc..
  - **Opción de registro de la ruta**. Clase 0, número 7. De longitud variable, se utiliza para crear una tabla de los encaminadores por los que pasa el paquete. La estructura de la opción es:

    donde el campo de longitud indica la longitud total de la opción, y el campo puntero indica la situación del primer campo de dirección libre.

    En cada campo "dirección" se anota cada uno de las direcciones de los encaminadores por los que pase el datagrama, lista que será entregada al nodo destino por si procede analizarla. Si el espacio reservado a las direcciones no es suficiente no se anotarán las direcciones de los últimos encaminadores por donde pasó el datagrama.

- **Opción de encaminamiento prefijado**. Clase 0, número 9. De longitud variable permite prefijar un camino para el datagrama en la red. Si el encaminamiento es estricto, el datagrama debe seguir exactamente la ruta prefijada enviándose un mensaje de error al nodo origen si no es posible, mientras que en el encaminamiento aproximado se permite pasar por varios encaminadores entre los indicados en dos elementos consecutivos de la lista de direcciones entregadas.
- **Opción de registro de tiempos**. Clase 2, número 4. Funciona de manera similar a la de registro de ruta añadiendo a cada registro de dirección, la fecha y hora (universal) expresada en milisegundos a partir de la media noche. Si no es posible obtener la hora indicada anteriormente se usa la hora local, marcando con un 1 el bit más significativo del campo.
- **Fin de opciones**. Clase 0, número 0. Indicada anteriormente.

Por último se rellena el resto de la cabecera con bit a cero hasta que su longitud medida en bit llegue a un múltiplo de 32.

El problema de la fragmentación.

Hemos dicho antes que cada tecnología de red soporta un tamaño máximo para las tramas que aproximadamente es de 1500 octetos para ethernet, 4500 para FDDI y 8000 para Frame Relay. Cuando un datagrama ip recorre el camino entre el nodo emisor y el receptor es habitual que pase por diferentes redes con MTU diferentes y normalmente menor que el tamaño del datagrama. Para que pueda pasar a través de ella es preciso fragmentar el datagrama en tantos trozos como sea necesario, siempre en trozos cuya longitud en octetos sea múltiplo de ocho (longitud en bit múltiplo de 64), rellenando con bit a cero el último fragmento si es preciso. Todos los trozos llevan una cabecera que es prácticamente copia de la del datagrama original, a excepción del bit MF del campo flag que será 1 en todos los trozos menos en el último, y del campo "offset fragment" que contiene la dirección del octeto, dado en múltiplos de 8, a partir del cual deben ponerse los datos que vienen en este trozo en el área de datos del datagrama original.

Hay que significar que este procedimiento de fragmentación no ocurre una sola vez en el momento de enviar el datagrama desde el nodo origen, sino que puede reiterarse cada vez que pase por un nuevo encaminador, así como que al ser tratado cada nuevo fragmento como un datagrama independiente y poder cada uno llegar a su destino por un camino diferente, los niveles de fragmentación pueden ser distintos, es decir, que al final al nodo receptor puede llegar el datagrama original en muchos trozos y de diversos tamaños. Para que este proceso funcione bien, los encaminadores deben

ser capaces de calcular los valores del campo "offset fragment" de los nuevos trozos teniendo en cuenta el valor del mismo campo en el trozo original.

La descripción exacta de los pasos de la fragmentación es:

1. Se comprueba el flag DF para ver si es posible fragmentarlo. Si no es así se descarta el datagrama.
2. Los datos son distribuidos en los trozos necesarios, siempre con longitud en octetos múltiplo de 8 y rellenado el último trozo si es preciso.
3. Se le añade una cabecera a cada trozo, copia de la del datagrama original salvo el flag MF que se pone a 1 salvo en el último fragmento, el campo "offset fragment" que se calcula para cada trozo, y según el tipo de opciones que aparezcan en el datagrama original, se copiaran o no a los trozos y por último se recalcula el código de verificación de cada cabecera.
4. Se transmiten cada uno de los trozos como datagramas ip independientes.

Tenemos ahora el problema subsiguiente para el nodo receptor: como recomponer correctamente el datagrama original a partir de los trozos. En primer lugar tenemos que tener en cuenta que no todos los trozos van a llegar juntos y en orden por lo que lo primero que es necesario fijar es el tiempo que vamos a estar esperando si nos falta algún trozo. Este tiempo máximo de espera, llamado tiempo de ensamble (Reassembly time) es determinado por el administrador de la red y empieza a contar en el momento en que el nodo receptor recibe el primer trozo y si transcurre sin haber recibido la totalidad de los trozos el datagrama completo se descarta, de la misma manera que se recibe algún trozo corrupto.

Una segunda cosa a tener en cuenta es que el nodo receptor no sabe la longitud real del datagrama hasta que no se recibe el trozo que lleva el flag MF a 0, calculando aquel valor a partir del valor del campo "offset fragment" y el valor del campo longitud del trozo último con la siguiente fórmula:

(offset x 8) + longitud = longitud total del datagrama en octetos

Para reensamblar la porción de datos del datagrama original, el nodo receptor cuando recibe el primer trozo prepara un "buffer" colocando este trozo en la posición indicada por el campo "offset" del mismo y así sucesivamente.

Una vez reconstruido el datagrama, el módulo Ip pasa los datos al protocolo de nivel superior correspondiente.

Protocolos "Auxiliares".

Cuando hemos estudiado anteriormente el protocolo IP hemos visto que en determinado ocasiones se realizaba el envío de mensajes al nodo emisor para indicar

una incidencia. Esta tarea, además de alguna otra, es realizada por el módulo de protocolo ICMP.

Por otra parte, cuando transmite una trama sobre una red, el módulo físico no puede dirigirla a la dirección ip de destino, sino a una dirección física de la red en que se encuentre. Nos hace falte por tanto una herramienta que relacione las direcciones físicas en la red de un nodo con su dirección ip. Esta herramienta está constituida por los módulos de protocolos ARP y RARP.

Protocolo ICMP.

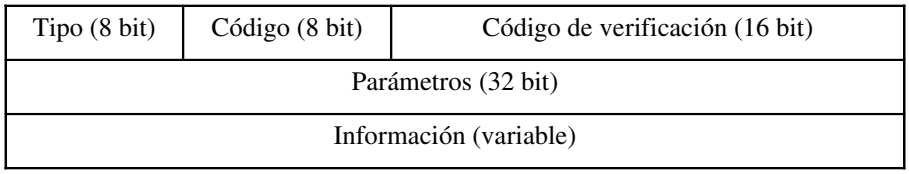
Como hemos dicho el protocolo IP no tiene ningún mecanismo para enviar mensajes de errores en las transmisiones de datos. Los recursos para esta tarea los pone el Protocolo para Mensajes de Control de Internet (ICMP Internet Control Message Protocol), que se utiliza para informar de errores en el proceso de los datagramas y proporciona los recursos necesarios para el tráfico de mensajes administrativos y de estado, siendo usado también como herramienta de diagnóstico.

El protocolo ICMP, cuyo número asignado es el 1, utiliza IP directamente y debe estar implementado como un subconjunto del propio protocolo IP en nodos finales y encaminadores y sus mensajes son generado y procesados por el propio TCP/IP y no por las aplicaciones de usuarios.

Antes de empezar a ver en detalle las tramas de ICMP debemos hacer notar que:

- ICMP es un usuario de IP. Los paquete ICMP van encapsulados en datagramas de IP y a su vez IP debe usar ICMP.
- ICMP informa de errores en los datagramas de IP pero no de errores en los paquetes ICMP.
- Si IP usa datagramas fragmentados, ICMP informa solo de errores en el primer trozo recibido.
- ICMP permite enviar información de control sobre la red.
- Los mensajes de error son enviados al nodo emisor del datagrama y no indican que encaminador causó el problema.

Veamos ahora cual es la estructura de un paquete ICMP:



Esta estructura está incluida dentro de la parte de datos de un datagrama IP cuyo campo protocolo de la cabecera lleva el valor 1. Los tres primeros campos de la estructura son obligatorios para todos los mensajes ICMP mientras que los otros dos son opcionales.

Los posibles valores del campo tipo son:

0	Respuesta de petición de eco
3	Destino inalcanzable
4	Control de flujo
5	Redirección
8	Petición de eco
11	Tiempo excedido
12	Parámetros ininteligibles
13	Petición de registro de tiempo
14	Respuesta de registro de tiempo
15	Petición de información
16	Respuesta informativa
17	Petición de máscara de dirección
18	Respuesta con máscara de dirección

Vamos a ver de una forma más amplia algunas de ellas.

Petición de eco y respuesta. Es una herramienta que se usa para determinar el estado de la red. Se puede enviar una petición de eco a cualquier dirección IP y la llegada de la respuesta implicará que el nodo que contesta está activo y accesible en la red. A nivel de usuario este servicio se conoce como PING, nombre de la orden que hay que dar para usarlo en máquinas unix, dos y otros, y de las opciones correspondientes de las utilidades Cyberkit y Netlab para máquinas Windows.

El formato de la utilidad Ping en una máquina Unix es el siguiente:

ping –qv –c count –i wait –p esquema –s tamaño direccion

donde:

- -q indica que no dé salidas intermedias y sólo el resumen final.
- -v indica que nos dé información accesoria.
- -c count indica el número de paquetes a enviar.
- -i wait indica el tiempo que debe transcurrir entre el envío de dos paquetes consecutivos.
- -p esquema indica que caracteres deben componer el paquete
- -s tamaño indica el tamaño de paquete
- dirección indica el nombre o dirección del nodo al que se envía la petición de eco.

Al final ping nos da una estadística de uso que tiene la forma siguiente:

---nombre del nodo---

número de paquetes transmitidos, número de paquetes recibidos, porcentaje de paquetes perdidos.

Round-trip min/avg/max = tiempo mínimo de respuesta / tiempo medio de respuesta / tiempo máximo de respuesta ms.

Redirección. Servicio utilizado por los encaminadores para enviar información de rutas a otros encaminadores o nodos terminales. Es enviado por un encaminador al emisor de un datagrama que le ha llegado cuando existe una ruta mejor. El campo parámetros lleva la dirección del encaminador recomendado y el campo información la identificación del datagrama recibido, indicando el campo código que tipo de redirección se recomienda:

0 redireccionar datagramas para una red

1 redireccionar datagramas para un nodo

2 redireccionar datagramas para un tipo de servicio y red

3 redireccionar datagramas para un tipo de servicio y nodo

Destino inalcanzable. Mensaje enviado al emisor de una datagrama, bien por un encaminador para indicar que no puede entregar el datagrama, o bien por el nodo receptor para indicar que no es posible entregar el datagrama a un protocolo de nivel superior siendo descartado en ambos casos. En el campo de información se devuelve el encabezado y los primeros 64 bit del datagrama que causa el problema. Los códigos posibles son:

Codigo	**Descripción**
0	Red no accesible (Errores de rutas)
1	Nodo no alcanzable (Errores de entregas)
2	Protocolo no disponible
3	Puerto no disponible
4	Requerida fragmentación y está desactivada
5	Rutas fuentes no disponible
6	Red de destino desconocida
7	Nodo de destino desconocido
8	Nodo origen fuera de red
9	Prohibido el acceso a la red de destino
10	Prohibido el acceso al nodo de destino
11	Red no alcanzable para el tipo de servicio

12	Nodo no alcanzable para el tipo de servicio

Control de flujo. Se envía por parte de los encaminadores o nodos receptores al nodo emisor cuando los datagramas se reciben más rápidamente que pueden ser procesados y se descartan. El campo información lleva la cabecera y los primeros 64 bit del datagrama descartado.

Tiempo excedido. Se envía este mensaje cuando expiró el tiempo de vida de un datagrama, código 0, o cuando expira el tiempo de emsamble de fragmentos, código 1. El campo de información lleva la cabecera y los primeros 64 bit del datagrama descartado.

Parámetros ininteligibles. Enviado por el nodo receptor a un encaminador cuando no puede procesar la cabecera del datagrama. En el campo parámetro se envía un puntero al byte que ocasionó el problema en el proceso y en el campo información la cabecera del datagrama descartado y su primeros 64 bit.

Registro de tiempos. Se usa por los encaminadores par estimar los tiempos de tránsito de los datagramas a través de una red. La estructura del paquete es:

Tipo (8 bit)	Código (8 bit)	Código de verificación (16 bit)
Identificación (16 bit)		Número de Secuencia (16 bit)
Tiempo de envío (32 bit)		
Tiempo de receptión (32 bit)		
Tiempo de retransmision (32 bit)		

Donde los campos de tiempos están en milisegundos transcurridos a partir del tiempo GMT. Los valores de los campos identificación y número de secuencia son empleados por el nodo emisor para asociar peticiones con respuestas. El campo tiempo de envío es rellenado por el nodo emisor justo antes de enviar el datagrama, el campo tiempo de recepción es rellenado por el nodo receptor nada más recibir el paquete y el campo tiempo de reenvío rellenado por el nodo receptor en el momento de devolver el datagrama.

Petición de información y respuesta. Permite a un nodo indentificar la red a la que está unido. El nodo envía a la red un datagrama con los campos de dirección fuente y destino vacíos y un servidor autorizado le devuelve el datagrama con ambos campos rellenos.

Petición de máscara de red y respuesta. Usado por un nodo para obtener la máscara de subred empleada en la red a la que se está conectado, petición que puede ser enviada directamente a un encaminador o bien mediante un datagrama de difusión.

Protocolo ARP

Cuando un nodo está conectado a una red tiene una dirección física a la que deben dirigirse todas las tramas de red que deba recibir este nodo. Por tanto cuando un datagrama debe ser enviado a una dirección ip, debe ser encapsulado en tramas de red con la dirección física de destino. El módulo ARP es quien realiza esta traducción de direcciones y para ello tiene unas tablas donde se guardan asociadas las direcciones ip y físicas.

En esta tabla, para cada dirección IP se mantiene los cuatro campos siguientes:

- ifIndex que contiene la puerta física para el interfaz por donde se accede a esta dirección.
- Physical address que contiene la dirección física en la red.
- Ip address que contiene la dirección Ip correspondiente.
- Mapping type que puede tener uno de los siguientes valores:
  - 1 = otro
  - 2 = inválido
  - 3 = dinámico
  - 4 = estático

Cuando es preciso enviar un datagrama a una dirección ip, primero se busca en la tabla anterior. Si no está, el módulo ARP envía un mensaje de difusión a toda la red (cuya estructura depende del tipos de red), llamado petición ARP, que contiene la dirección ip a la que hay que enviar el datagrama. Si una de las máquinas reconoce la dirección IP en el paquete como suya devuelve una respuesta ARP con su dirección física al nodo emisor, que anotará esta dirección en la tabla de cache ARP y seguidamente enviará los datagramas.

Protocolo RARP

El protocolo RARP (Reverse Address Resolution Protocol) trabaja como su nombre indica de forma inversa al protocolo ARP y es utilizado por las estaciones que no conocen su propia dirección IP para solicitar información de servidor de información RARP.

La estructura del paquete, al igual que en protocolo ARP, depende del tipo de red empleado y su funcionamiento es similar: el nodo emite un paquete de difusión (broadcast) en la red que es interceptado por el servidor RARP y posteriormente devuelto con la información solicitada.

Este modo de actuación es adecuado cuando los nodos de la red no pueden mantener información fiable de configuración, bien por imposibilidad física (máquinas diskless) o por falta de fiabilidad del propietario del nodo.

Nivel de transporte.

Concepto de puerto

Los protocolos de Internet permiten que en un ordenador muchos procesos de usuario se comuniquen con el exterior simultáneamente. Necesitamos un método para identificar el proceso que debe recibir los datos que nos llegan por el canal de comunicación. Para ello, las interfaces de las aplicaciones de usuario con el protocolo de transporte se identifican a sí mismos con un número (entero de 16 bit) que es lo que se conoce como puerto. Por tanto podemos definir un puerto como un identificador que permite a los protocolos entre nodos identificar a los protocolos de alto nivel de los que se reciben y a los que se entregan los mensajes.

¿Cómo se identifica entonces cada proceso ante TCP/IP? Por la dirección ip del nodo y el número de puerto por el que se comunica con TCP/IP. Esta pareja de números constituyen asociados un identificador llamado socket. Por tanto, una conexión en TCP/IP está definida entre dos socket, pudiendo cada uno estos estar compartidos por más de una conexión. Otra característica a tener en cuenta de los puertos es que se usan asociados a un determinado protocolo de transporte (TCP, UDP), por lo que el mismo identificador se puede referir a distinto proceso de usuario según el transporte utilizado.

Cuando un cliente quiere acceder a un recurso de un servidor debe conocer no sólo la dirección ip del nodo, sino también el puerto asociado al recurso. Para hacer conocida esta información, determinado puertos con identificador menor de 255 (valores que están reservados para servicios mientras que el resto son de uso libre) han sido asociados a recursos predeterminados de los que los más habituales son:

20	FTP-DATA	Transferencias de datos en FTP
21	FTP	Control en FTP
23	TELNET	Servicio de terminal remoto
25	SMTP	Estafeta de correo SMTP
53	DOMAIN	Servicios del DNS
69	TFTP	Transferencias de ficheros
79	FINGER	Servicio de consulta de usuarios
80	HTTP	Servicios de WWW
110	POP3	Servicio de buzones de correo
139	NETBIOS-SS	Servicio de session de NETBIOS (WINDOWS)

Protocolo UDP

No está claro si UDP (User Datagram Protocol) debe ser visto como un protocolo de transporte no orientado a la conexión o simplemente como una interfaz de usuario al protocolo IP. No le añada a éste más fiabilidad ni control de flujo ni recuperación de errores y sólo sirve para mantener información sobre los socket usados en la conexión, por lo que podría parecer que es un protocolo orientado a la misma pero no incluye ninguna de las posibilidades de éstos y sólo usa el concepto de puerto para redirigir los datagramas a la aplicación adecuada, usándose por aquellos procesos de usuario que no necesitan los recursos más amplios de TCP, como TFTP, SNMP, etc.

La estructura del paquete UDP es la siguiente:

Source Port (16 bit)	Destination Port (16 bit)
Length (16 bit)	Checksum (16 bit)
Datos (variable)	

Veamos ahora la descripción de los campos:

- **Source port (Puerto de origen).** Identifica por que puerto ha enviado la aplicación emisora el paquete y tiene el valor 0 si no se usa.
- **Destination port (Puerto de destino).** Identifica el proceso (puerto) que recibirá la información en el nodo de destino.
- **Length (Longitud).** De valor mínimo 8, indica la longitud total del paquete.
- **Checksum (Código de vierificación).** Utilizado para verificar el contenido del paquete.

Protocolo TCP.

Cuando hablamos de IP y UDP vimos que estos protocolos no proporcionaban servicios orientados a la conexión, con la falta de recursos que ello implica. Para dar este servicio orientado a la conexión, requerido por muchos procesos de usuario, se usa el protocolo TCP (Transmission Control Protocol).

TCP es un protocolo orientado a la conexión que proporciona fiabilidad, control de flujo y recuperación de errores; protocolo punto a punto que suministra una conexión lógica entre pares de procesos, identificados cada uno de ellos por un socket, utilizando los números de puertos de éstos como comunicación con los procesos de nivel superior.

TCP tiene similitudes al nivel de transporte de OSI y muchas de sus propiedades han sido incluidas en la clase 4 de dicho transporte. Aunque habitualmente se usa con IP, TCP podría operar con otros protocolos.

¿Qué características tiene TCP? En primer lugar, al estar orientado a la conexión, es responsable ante los procesos de usuarios del flujo de información entre los dos puntos a través de la red, haciendo aparecer esta tarea ante ellos como si transmitieran los datos carácter a carácter (stream oriented), en el mismo orden en que

fueron enviados aunque, en su funcionamiento interior, deba agrupar los datos en segmentos ordenados para su transmisión sobre IP. Cuando el módulo receptor recibe un paquete, comprueba que no ha recibido datos dañados y envía un mensaje de confirmación positiva (ACK) al emisor y en caso contrario descarta el segmento e informa al módulo TCP emisor para su retransmisión. También debe descartar los posibles datos duplicados que pueda recibir y realizar la reordenación de los segmentos que puede recibir desordenados.

TCP debe realizar también el control de flujo. Para ello el módulo receptor va informando al módulo emisor de la cantidad de octetos que puede recibir sin problemas en cada lapso de tiempo, mediante un mecanismo llamado ventada deslizante (Window sliding) que veremos con detalle más adelante.

A fin de utilizar más eficientemente los recursos disponibles TCP ofrece la posibilidad de multiplexación entre distintos procesos de usuario, transmisión simultánea en los dos sentidos (full duplex) y la posibilidad de especificar niveles de seguridad y prioridad para las comunicaciones, así como un mecanismo llamado graceful close que asegura que la conexión no se cierra hasta no haber recibido confirmación de la recepción de todos los datos enviados.

Veamos ahora cual es la estructura del segmento TCP.

<table>
<tr><td colspan="5">Source Port (16 bit)</td><td colspan="4">Destination Port (16 bit)</td></tr>
<tr><td colspan="9">Sequence number (32 bit)</td></tr>
<tr><td colspan="9">Acknowledgment number (32 bit)</td></tr>
<tr><td>Data offset (4 bit)</td><td colspan="2">Reserved (6 bit)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Window (16 bit)</td></tr>
<tr><td colspan="5">Checksum (16 bit)</td><td colspan="4">Urgent Pointer (16 bit)</td></tr>
<tr><td colspan="7">Options (Variable)</td><td colspan="2">Padding</td></tr>
<tr><td colspan="9">Data (Variable)</td></tr>
</table>

El significado de los campos es:

- **Source Port (Puerto de origen).** Identifica al proceso (puerto) de origen.
- **Destination Port (Puerto de destino).** Identifica al proceso (puerto) de destino.
- **Sequence number (Número de orden).** Número de orden del byte que identifica la posición inicial de los datos del segmento con respecto al flujo de bytes original del emisor.
- **Acknowledgment number.** Indica el número de orden del byte que el receptor espera.

- **Data offset (Desplazamiento de los datos).** Indica la longitud de la cabecera de TCP medida en palabras de 32 bit.
- **Reserved.** Campo reservado para futuro uso que debe llevar todos sus bit a 0
- **Flags.** Los siguientes seis campos son indicadores para solicitar servicios o marcar la validez de otros campos de la cabecera. Su descripción es la siguiente:
  - URG: Si tiene valor 1 el campo Urgent Pointer es válido.
  - ACK: Si tiene el valor 1 el campo Acknowledgment number es válido.
  - PSH: Si tiene el valor 1 el segmento requiere un PUSH.
  - RST: Si tiene el valor 1 reinicializa la conexión.
  - SYN: Si tiene el valor 1 sincroniza los números de secuencia.
  - FIN: Si tiene el valor 1 el emisor llegó al final del flujo de caracteres.
- **Window (Ventana).** Indica el número de octetos que el receptor podría aceptar.
- **Checksum (Código de verificación).** Se usa para verificar la corrección de los datos contenidos en el segmento incluida la cabecera.
- **Urgent Pointer (Puntero a datos urgentes).** Este campo, válido sólo si el URG esta a 1, indica donde en el flujo de bytes están los datos considerados urgentes, que cada implementación tratará de manera diferente.
- **Options (Opciones).** Un campo para implementación de opciones que funciona de manera similar a como lo hace el campo opciones del datagrama IP. Cada opción tiene tres campos, un octeto que contiene el código de opción, un campo que indica la longitud de la opción y el tercer campo que incluye los valores propios de la opción. Las tres opciones disponibles actualmente son fin de lista de opciones, código 0, sin operación, código 1, y longitud máxima del segmento, código 2.
- **Padding.** Relleno de bit a cero para completar palabra de 32 bit.

Vamos a ver con un ejemplo como se asignan los puertos:

Supongamos que un nodo A quiere comunicarse con otro nodo B. Para ello le envía un segmento TCP donde el campo Destination port lleva el valor del puerto (proceso) con el que quiere comunicarse por ejemplo el 25. El campo Source port que indica que puerto (proceso) quiere establecer la conexión es puesto por el nodo origen, supongamos 700. Supongamos ahora que otro proceso del nodo A quiere establecer de nuevo una conexión con el nodo B y el mismo proceso 25. Se enviará un nuevo segmento desde el nodo A al nodo B con el mismo valor del campo Destination Port

pero con otro valor del campo Source Port, por ejemplo 701. Supongamos ahora que otro nodo C establece una conexión con el nodo B y el puerto 25 y que usa como puerto de origen también el 701.

Vemos que tres procesos comparten el mismo puerto de destino gracias a la capacidad de multiplexación del protocolo TCP, pero con todas estas conexiones activas el nodo C debe saber correctamente diferenciarlas a fin de no enviar datos cruzados, para ello tiene en cuenta lo siguiente. Las dos primeras conexiones se diferencian en el puerto origen del segmento mientras que la segunda y tercera se diferencian en la dirección ip origen del datagrama. Toda esta información, además de alguna otra, es necesaria guardarla para poder realizar correctamente todo el proceso de conexión, y esto se hace guardando para cada una de ellas el llamado TCB (Transmission control block) donde se almacena los socket local y remoto, los punteros a los buffer de envío y recepción, los punteros a la cola de retransmisiones, los valores de seguridad y prioridad para la conexión, el segmento actual y una serie de variables que retienen valores sobre los números de secuencia para el envío y recepción de datos.

Se permiten en TCP dos formas de establecer la conexión: apertura pasiva y apertura activa. En la apertura pasiva un proceso de nivel superior (normalmente u servidor) ordena al TCP y al sistema operativo que espere la llegada de una conexión desde un sistema remoto. Cuando recibe esta petición el sistema operativo crea una identificación donde el socket remoto es 0, lo que le permite recibir llamadas desde cualquier usuario. En la segunda forma de conexión, el proceso de nivel superior designa un socket específico con el que se va a establecer la conexión.

Para mantener la información sobre las conexiones que están activas se usa la TPC connection table donde para cada conexión existe una fila con las siguientes cinco columnas:

- Connection state donde se mantiene es estado actual de la conexión.
- Local Address que contiene la dirección IP local para cada conexión.
- Local Port, puerto local usado por la conexión
- Remote Address dirección IP del nodo remoto con el que está establecida la conexión
- Remote port puerto usado en el nodo remoto para la conexión.

Para controlar la transmisión de información entre nodos e impedir que la llegada un excesivo número de octetos colapse la capacidad de proceso del nodo receptor, TCP usa un mecanismo llamado de ventana deslizante y cuyo funcionamiento vamos a ver con un ejemplo:

Consideremos un módulo TCP que está transmitiendo datos a otro B. En el módulo A llamamos ventada de envío a la máxima cantidad de octetos que puede enviar al módulo B sin recibir una confirmación de recepción. Dependiendo de su capacidad

de proceso de los datos recibidos, el módulo B puede ampliar o reducir este límite usando el campo Window del paquete TCP en que se confirma la recepción de los datos. Si consideramos el conjunto de datos a transmitir como una sucesión ordenada de octetos en módulo A tendría tres punteros a distintos octetos de esta sucesión:

- El primero apunta al último octeto transmitido y confirmado.
- El segundo apunta al último octeto transmitido pero no confirmado
- El tercero al límite de la ventana de transmisión, primer octeto que ya no se puede transmitir y que se calcula sumando al segundo puntero el tamaño de la ventana de transmisión.

Por otro lado, como TCP no tiene una confirmación negativa de los datos transmitido, necesita un mecanismo para saber cuando tiene que retransmitir los datos. Para ello se fija un tiempo de espera, pasado el cual se retransmite el segmento antes enviado.

El valor de este tiempo de espera no es fijo, sino que se calcula utilizando un algoritmo adaptativo, dependiente de la implementación, y en función de los retardos de los paquetes anteriores.

Protocolos especiales.

Dentro de este grupo vamos a considerar dos protocolos: el Bootp (Booststrap Protocol) y el IGMP (Internet Group Management Protocol).

Protocolo Bootstrap.

El protocolo Bootp (o protocolo de arranque), se puede considerar que actúa como una versión más amplia del protocolo RARP. Es usado por máquinas que no conocen su dirección IP al arrancar para obtener no sólo ésta de un servidor apropiado, sino también otros datos como ficheros de arranque, de configuración, etc.

Bootp usa datagramas IP para obtener su dirección y hace la carga de ficheros usando como protocolo UDP. La descripción del paquete Bootp es la siguiente:

<table>
<tr><td>Type</td><td>Header Type</td><td>H-Length</td><td>Hop count</td></tr>
<tr><td colspan="4">Transaction Id</td></tr>
<tr><td colspan="2">Seconds</td><td colspan="2">Padding</td></tr>
<tr><td colspan="4">Client IP Address</td></tr>
<tr><td colspan="4">Response IP Address</td></tr>
<tr><td colspan="4">Server IP Address</td></tr>
<tr><td colspan="4">Gateway IP Address</td></tr>
<tr><td colspan="4">Cliente Hardware Address (16 Octetos)</td></tr>
<tr><td colspan="4">Server Host Name (64 Octetos)</td></tr>
<tr><td colspan="4">Boot File Name (128 Octetos)</td></tr>
<tr><td colspan="4">Vendor-Specific Area (64 Octetos)</td></tr>
</table>

La descripción de los campos es la siguiente:

- **Type** (Tipo): Campo de un byte que identifica si el paquete es una solicitud o una respueta.
- **Header Type** (Cabecera): Campo de un byte de tamaño que identifica el tipo de dirección de hardware.
- **H-Length** (Longitud de la dirección hardware): Campo de un byte que mide la longitud de la dirección hardware en octetos.
- **Hop count** (Contador de saltos): Se utiliza cuando el protocolo BOOTP se utiliza a través de varios encaminadores, aumentando el contador en uno cada vez que pasa por un encaminador.
- **Transaction id** (Identificador de transacciones): Se utiliza para asignar respuestas a las solicitudes.
- **Seconds** (Segundos): Se utiliza para calcular el tiempo transcurrido desde el envío de la solicitud hasta la recepción de la respuesta. Longitud dos octetos.
- **Padding** (Relleno): Bit a ceros para completar a 4 octetos.
- **Client ip address** (Dirección IP del cliente): Puesta a 0 si el cliente no conoce su dirección. 32 bit.
- **Server IP address** (Dirección Ip del servidor): Puesta por el cliente si la conoce. Si es distinta de 0, sólo el servidor especificado responde a la petición.
- **Gateway IP address** (Dirección IP del encaminador): Puesta a 0 por el cliente y rellenada por el encaminador que obtiene la solicitud.
- **Server Host Name** (Nombre del servidor): Campo opcional.
- **Boot file name** (Nombre del fichero de arranque): El cliente lo pone a cero o indica un nombre genérico. El servidor lo sustituye por la ruta completa al fichero de arranque.
- **Vendor-specific area** (Área específica del fabricante): Dependiente de la implementación.

Protocolo IGMP.

El IGMP (Internet Group Management Protocol) es un protocolo, extensión del protocolo IP, que se usa por los miembros de una red multicast (de difusión uno a muchos, como por ejemplo la televisión por cable, distribución de vídeo conferencias, retransmisión de acontecimientos, etc) para mantener su pertenencia al grupo multicast, para la cual los encaminadores multicast envían mensajes de requerimiento de presencia cada cierto tiempo a los que deben contestar todos los nodos del grupo ya que si no son dados de baja en él.

También se usa este protocolo para propagar información de direccionamiento en redes multicast.

Nivel de aplicación.

La mayoría de los protocolos importantes que podemos considerar de la capa de aplicación se estudia en otros puntos del curso por lo que aquí nos limitaremos a una mera enumeración. Entre ellos están:

- **Ftp** protocolo de transferencia de ficheros.
- **SMTP** protocolo de comunicación de las estafetas de correo.
- **Telnet** protocolo de emulación de los servicios de terminal.
- **DNS** protocolo del servicio de traducción de nombres.
- **HTTP** protocolo de comunicación para servidores web.
- **NTP** protocolo de servicios de sincronización de tiempos.

La estructura cliente-servidor en Ip.

Sabemos que un servidor es cualquier programa que oferta un servicio a través de la red, mientras que un cliente es un programa que envía una petición a un servidor y espera una respuesta. También sabemos que el término servidor se puede extender a la máquina que oferta el servicio. Vamos a ver de una forma elemental como funciona la arquitectura cliente servidor en TCP/IP.

Un programa servidor en un nodo TCP/IP comienza su ejecución antes de recibir cualquier petición esperando éstas en un puerto predeterminado, reservado para el servicio, conocido por las aplicaciones clientes y normalmente está aceptando peticiones y enviando respuestas indefinidamente.

El cliente sin embargo reserva para la comunicación un puerto aleatorio y que no esté usado actualmente, y por tanto diferente en cada petición.

Los servidores pueden ser de dos tipos: secuenciales y concurrente. Los primeros atienden las peticiones de una en una siendo el sistema operativo el encargado de gestionar las colas de las mismas.

Los servidores concurrentes, que son aquellos capaces de atender múltiples peticiones de forma simultáneas, tienen una estructura más complicada: están formados por un proceso maestro que es el encargado de recibir las peticiones por el puerto predeterminado, y que cuando recibe una petición de un cliente crea un proceso esclavo que establece una nueva conexión con el cliente para atender la petición; una vez satisfecha ésta el proceso esclavo termina, mientras el proceso maestro se queda siempre en estado de espera.

El encaminamiento

Recibe el nombre de encaminamiento el proceso de elegir un camino por el que enviar un datagrama ip a su destino. El encaminamiento es realizado por los propios nodos finales de la red ip y por unas pasarelas especializadas llamadas encaminadores (Routers). Estas pasarelas pueden ser cualquier nodo que incorpore la pila TCPI/IP completa. Los encaminadores se encuentran conectados normalmente a varias redes física y hacen de paso entre ellas. Para ello tienen lo que se llaman "tablas de rutas" que son listas de las redes a las que se puede acceder a través de cada una de sus conexiones.

El encaminamiento puede ser de dos tipos: directo o indirecto.

El encaminamiento directo es el que se produce cuando tanto el nodo emisor como el receptor se encuentran en la misma red, lo que averigua el emisor comparando la parte de red de su propia dirección ip con la dirección de destino del datagrama ip a enviar. El nodo emisor debe encapsular el datagrama en una trama física, averiguar la dirección de red correspondiente a la dirección ip de destino mediante el protocolo ARP y enviar la trama correspondiente directamente al destinatario. Este encaminamiento directo se produce siempre en el momento de entrega del datagrama.

El encaminamiento indirecto es más complicado y se produce siempre que el nodo destino no se encuentra en la misma red ip que el nodo emisor. En este caso, éste debe enviar el datagrama a uno de los encaminadores a los que tiene acceso en su red física para que este a su vez lo reenvíe a su destino o a otro encaminador si aquel no se encuentra en ninguna de las redes a las está conectado. Tenemos el problema de saber a que encaminador se debe enviar el datagrama lo que se resuelve consultado la tabla de rutas que tiene cada nodo. En estas tablas de rutas se guardan parejas compuestas por direcciones de redes y del encaminador al que hay que enviar el datagrama para que llegue hasta ella. Hay que hacer constar que también pueden haber rutas a nodos.

Para evitar que las tablas de rutas, que pueden ser estáticas, es decir, que se escriben en un fichero del nodo por el administrador, o dinámicas, es decir, que el nodo las va aprendiendo de los encaminadores sean demasiado grandes, se puede utilizar la técnica de la ruta por defecto, que consiste en poner una entrada diciendo a que encaminador se deben enviar todos los datagramas dirigidos a redes de las que no conocemos rutas de acceso.

¿Cómo funcionan los encaminadores? Cuando un encaminador recibe un paquete comprueba en primer lugar si viene dirigido a él, es decir, si la dirección ip de destino coincide con alguna de las direcciones de sus conexiones a red; si no es éste el caso, comprueba si la parte de red de la dirección ip de destino del paquete se corresponde con las de las redes a las que está conectado y en este caso la envía por encaminamiento directo en la forma mencionada antes. Si no ocurre así, consulta sus tablas de ruta para ver si la red de destino aparece en ellas, enviando en caso afirmativo el datagrama al encaminador correspondiente. En caso negativo, si tiene una ruta por

defecto envía el datagrama al encaminador indicado y si no, descarta el datagrama y envía de vuelta un paquete ICMP con el código de red inalcanzable.

Protocolos de encaminamiento (Routing protocols)

Hemos dicho anteriormente que los encaminadores tienen tablas de rutas para determinar el mejor camino que debe seguir el paquete para llegar a su destino. Aquí nos queda por contestar a dos posibles preguntas: ¿Cómo se averigua el mejor camino? ¿Cómo se comunica a los demás encaminadores?

Las tablas de rutas se pueden rellenar de forma estática, es decir, el administrador del nodo o encaminador escribe todas la direcciones de red y su correspondiente encaminador asociado, o bien la ruta por defecto para todas las redes. Está claro que este procedimiento no tiene en cuenta todas las posibles variaciones de la red y podría darse el caso de que el mejor camino a una red en el momento de escribir la tabla no lo fuese en otro instante, o incluso estuviese cortado impidiendo el envío de datagramas a dicha red aunque tuviésemos un camino alternativo.

Por eso parece mejor que los encaminadores aprendiesen cuales son los caminos a las distintas redes y se lo comunicasen entre ellos y a los distintos nodos de la red. Esto es la tarea que desempeñan los protocolos de encaminamiento (rounting protocols).

Los protocolos de encaminamiento pueden ser de dos tipos: de vector-distancia y de estados de los enlaces.

Los encaminadores que utilizan un protocolo de vector-distancia informan a sus homólogos conectados a las mismas redes que ellos dándoles cuenta de las redes a las que son capaces de acceder y su distancia a ellas, empezando por aquellas a las que están directamente conectados con lo que todas las rutas se van transmitiendo a todos los encaminadores. Cuando un encaminador recibe una información de este tipo la almacena en su propia tabla de rutas incrementando la distancia recibida con el valor de su distancia al encaminadore del que la ha recibido.

Los protocolos basados en los estados de los enlaces usan otros criterios en lugar de las distancias para el cálculo de rutas, criterio basados en factores del tipo de servicio (TOS), que son definidos por los administradores de red y que pueden incluir criterios como retrasos, anchos de banda y disponibilidad. Se escoge entonces las rutas en la Internet en base a la posibilidad de que los encaminadores y redes por las que se puede pasar puedan proporcionar los servicios requeridos.

Algunos protocolos de este tipo usan métodos dinámicos para actualizar las tablas de rutas reflejando el tráfico en los enlaces y la disponibilidad de los encaminadores. Otros protocolos usan métodos más estáticos y sólo actualizan sus tablas de rutas cuando se producen condiciones de fallos en los enlaces que dan lugar a un cambio de topología en la red.

Veamos ahora una pequeña descripción de alguno de los protocolos de encaminamiento más habituales:

- **GGP (Gateway-to-Gateway Protocol),** protocolo del tipo vector-distancia usado en los primeros años de la internet y que apenas se usa en la actualidad.
- **RIP (Routing Information Protocol),** es un protocolo de encaminamiento desarrollado bastante después de GGP y basado también en vector-distancia, del que existen dos versiones I y II soportando esta última manejo de subredes. Rip escoge una ruta para un destino entra varias posibles asignado a cada una de ellas un coste en función de los encaminadores por lo que deba pasar y esta información es enviada cada cierto tiempo (normalmente cada 30 segundos) a los demás encaminadores a los que tenga acceso directamente. Además del aumento de tráfico que ocasiona este paso de rutas, RIP tiene problemas para detectar los posibles lazos en la transmisión de rutas así como el hecho de no autentificar los intercambios de manera obligatoria, lo que puede provocar problemas de seguridad.
- **Hello Protocol,** protocolo similar al Rip salvo que el coste de las rutas es medido por el retardo en la entrega de los paquetes.
- **OSPF (Open Shortest Path First),** protocolo basado en el estado de los enlaces, e incluye entre sus capacidades direccionamiento de subredes y encaminamiento de tipo de servicio. Está considerado como un protocolo adaptativo y dinámico que proporciona unas tablas de rutas estables en un corto periodo de tiempo. Las redes es Ospf se encuentran agrupadas en áreas que pueden están ocultas para los otras y que se agrupan en sistemas autónomos. Distingue cuatro tipos de encaminadores: internal routers que se caracterizan porque todas las redes conectadas a estos están en el mismo área; borders routers que es todo encaminador que no es un internal router; backbone router que es un encaminador con interfaces al troncal del sistema autónomo y Boundary router, encaminadores que intercambian información con otros sistemas autónomos.

Para ver que ruta sigue un datagrama desde nuestro nodo hasta un determinado destino podemos usar la utilidad Traceroute en máquinas unix o las opciones equivalentes de las utilidades Cyberkit y Netlab en máquinas Windows.

La utilidad traceroute en máquinas Unix tiene el formato siguiente:

traceroute –FdInv –f ttl –i interface –m número –s dirección –w tiempo nombre

donde :

- Nombre es el nombre o dirección del nodo al que se traza la ruta.

- - f ttl pone el tiempo de vida inicial para el primer paquete que se envía.
- - F activa el bit de impedir fragmentación.
- -d activa el rastreo.
- -i interface indica el interface de red cuya dirección se usa como dirección fuente del paquete de salida.
- -I usa ICMP en lugar de UDP.
- -m número es el máximo número de encaminadores por los que se traza la ruta.
- -n no hace resolución inversa de nombres.
- -s dirección pone la dirección fuente del paquete de salida.
- -v da la máxima información.
- -w tiempo es el máximo tiempo de espera para la respuesta.