Redes :

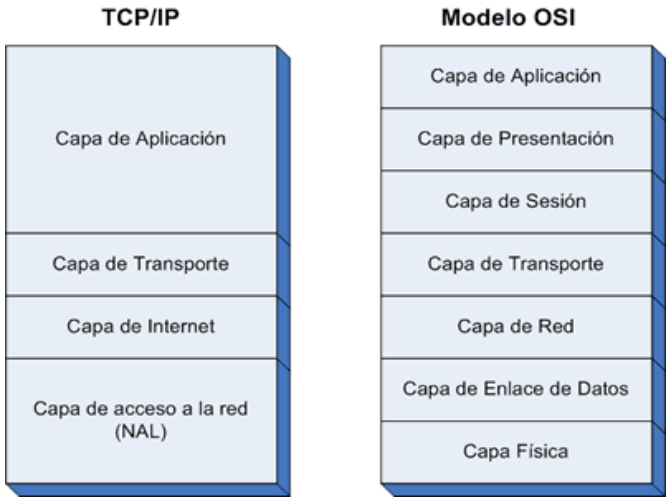
Antes de abordar este capitulo daremos una breve explicación del porque hemos decidido utilizar el modelo TCP/IP en lugar del OSI.

Cuando fue propuesto el modelo OSI, en el mercado había muchas arquitecturas de protocolos, algunas propietarias, otras abiertas, pero todas diferentes. El modelo OSI pretendía ser un marco para el desarrollo de estándares.

Por diferentes razones el modelo y las normas que del mismo se derivan, no han logrado tener la repercusión esperada, consideramos que los siguientes aspectos han sido determinantes:

• La complejidad del modelo.
• La complejidad de las normas desarrolladas a partir del modelo.
• El impulso del modelo Internet y la simplicidad de sus estándares.

Modelo TCP/IP vs OSI



Modelo OSI

Capa física define la manera en la que los datos se convierten físicamente en señales digitales en los medios de comunicación(pulsos eléctricos, modulación de luz, etc.).

Capa de enlace de datos define la interfaz con la tarjeta de interfaz de red y cómo se comparte el medio de transmisión.

Capa de red permite administrar las direcciones y el enrutamiento de datos, es decir, su ruta a través de la red.

Capa de transporte se encarga del transporte de datos, su división en paquetes y la administración de potenciales errores de transmisión.

Capa de sesión define el inicio y la finalización de las sesiones de

comunicación entre los equipos de la red.
Capa de presentación define el formato de los datos que maneja la capa de aplicación (su representación y, potencialmente, su compresión y cifrado) independientemente del sistema.

Capa de aplicación le brinda aplicaciones a la interfaz. Por lo tanto, es el nivel más cercano a los usuarios, administrado directamente por el software.

Modelo TCP/IP

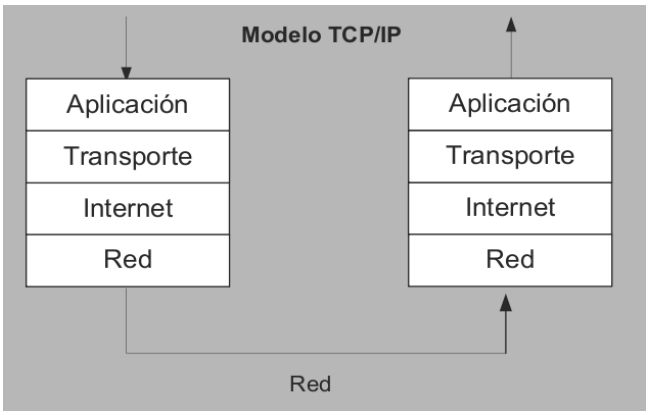
Durante la década de los 60, la agencia DARPA (Defense Advanced Research Projects Agency) de los Estados Unidos ante la posibilidad de un ataque que afectara su red de comunicaciones, financio la investigaciones que tenían por objetivo desarrollar una red distribuida.

Como resultado de sus investigaciones creo ARPANET, la cual era de carácter experimental, mas adelante en los 70 la agencia comenzó a investigar la interconectar de distintas redes, logrado en 1974 establecer las bases de desarrollo de los protocolos que actualmente utilizamos en las redes TCP/IP.

IP es un protocolo que brinda mecanismos de interconexión entre redes de área local y TCP es el encargado del control de flujo y el manejo de errores entre los extremos que realizan la comunicación.

La familia de protocolos TCP/IP se divide en las cuatro capas, cada protocolo se comunica con su par en la capa equivalente del sistema remoto. Cada protocolo solo debe ocuparse de la comunicación con su gemelo, sin preocuparse de las capas superior o inferior. Sin embargo, también debe haber acuerdo en como pasan los datos de capa en capa dentro de un mismo sistema, o sea de como una capa brinda servicio a otras.

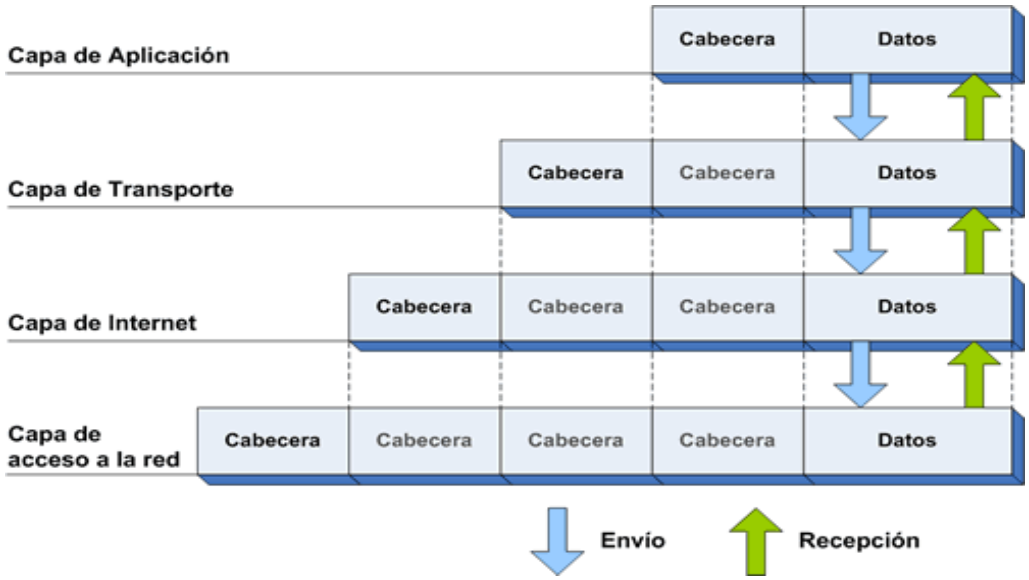
Las capas superiores delegan en las inferiores para la transmisión de los datos a través de la red. Los datos descienden por la pila, de capa en capa, hasta que son transmitidos a través de la red por los protocolos de la capa física (capa de red). En el sistema remoto, irán ascendiendo por la pila hasta la aplicación correspondiente.



Cada capa incorpora a los datos a ser enviados a capas inferiores, información de control la cual se coloca precediendo a los datos de la capa superior, de allí que también se la conozca con el nombre de cabecera.

Esta acción de incorporar datos de control se la conoce con el nombre de encapsulamiento.

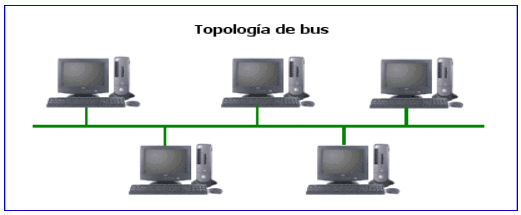
El proceso inverso tiene lugar al momento de querer enviar datos a capas superiores, en este caso se requiere quitar los datos de control, permitiendo así que la información ascienda por la pila.



Topológias de Red

Topología de bus

La topología de bus tiene todos sus nodos conectados directamente a un enlace y no tiene ninguna otra conexión entre nodos. Físicamente cada host está conectado a un cable común, por lo que se pueden comunicar directamente, aunque la ruptura del cable hace que los hosts queden desconectados.

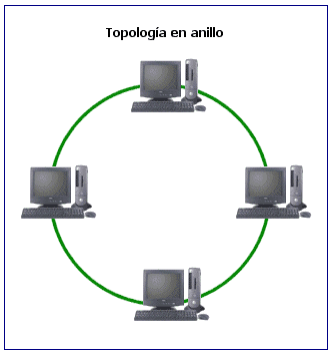


La topología de bus permite que todos los dispositivos de la red puedan ver todas las señales de todos los demás dispositivos, lo que puede ser ventajoso si desea que todos los dispositivos obtengan esta información. Sin embargo, puede representar una desventaja, ya que es común que se produzcan problemas de tráfico y colisiones, que se pueden paliar segmentando la red en varias partes.

Es la topología más común en pequeñas LAN, con hub o switch final en uno de los extremos.

Topología de anillo

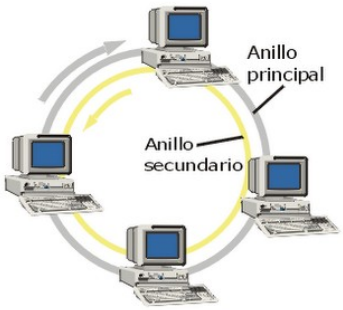
Una topología de anillo se compone de un solo anillo cerrado formado por nodos y enlaces, en el que cada nodo está conectado solamente con los dos nodos adyacentes.



Topología de anillo doble

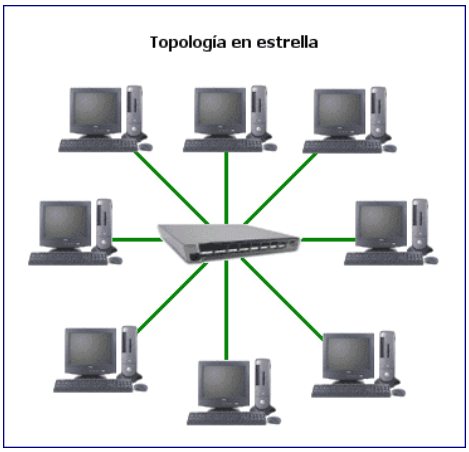
Una topología en anillo doble consta de dos anillos concéntricos, donde cada host de la red está conectado a ambos anillos, aunque los dos anillos no están conectados directamente entre sí. Es análoga a la topología de anillo, con la diferencia de que, para incrementar la confiabilidad y flexibilidad de la red, hay un segundo anillo redundante que conecta los mismos dispositivos.

La topología de anillo doble actúa como si fueran dos anillos independientes, de los cuales se usa solamente uno por vez.



Topología en estrella

La topología en estrella tiene un nodo central desde el que se irradian todos los enlaces hacia los demás nodos. Por el nodo central, generalmente ocupado por un hub, pasa toda la información que circula por la red.



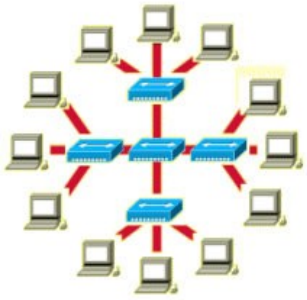
La ventaja principal es que permite que todos los nodos se comuniquen entre sí de manera conveniente. La desventaja principal es que si el nodo central falla, toda la red se desconecta.

Topología en estrella extendida:

La topología en estrella extendida es igual a la topología en estrella, con la diferencia de que cada nodo que se conecta con el nodo central también es el centro de otra estrella. Generalmente el nodo central está ocupado por un hub o un switch, y los nodos secundarios por hubs.

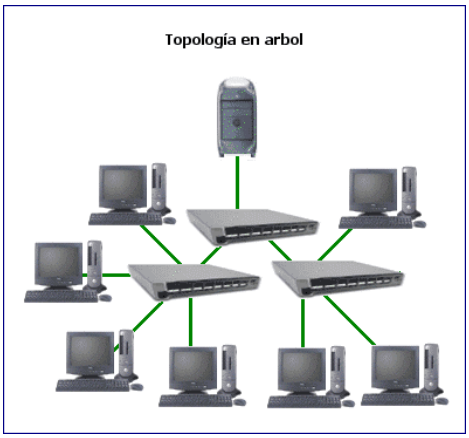
La ventaja de esto es que el cableado es más corto y limita la cantidad de dispositivos que se deben interconectar con cualquier nodo central.

La topología en estrella extendida es sumamente jerárquica, y busca que la información se mantenga local. Esta es la forma de conexión utilizada actualmente por el sistema telefónico.



Topología en árbol

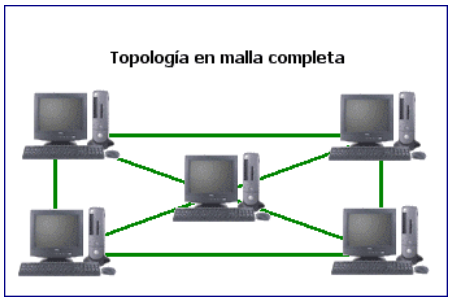
La topología en árbol es similar a la topología en estrella extendida, salvo en que no tiene un nodo central. En cambio, un nodo de enlace troncal, generalmente ocupado por un hub o switch, desde el que se ramifican los demás nodos.



El enlace troncal es un cable con varias capas de ramificaciones, y el flujo de información es jerárquico. Conectado en el otro extremo al enlace troncal generalmente se encuentra un host servidor.

Topología en malla completa

En una topología de malla completa, cada nodo se enlaza directamente con los demás nodos. Las ventajas son que, como cada todo se conecta físicamente a los demás, creando una conexión redundante, si algún enlace deja de funcionar la información puede circular a través de cualquier cantidad de enlaces hasta llegar a destino. Además, esta topología permite que la información circule por varias rutas a través de la red.

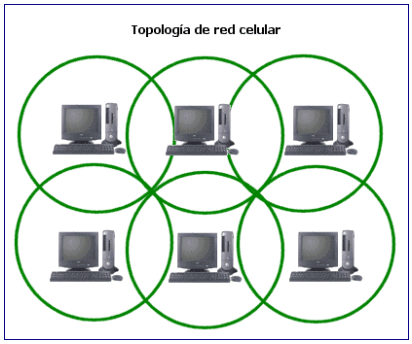


La desventaja física principal es que sólo funciona con una pequeña cantidad de nodos, ya que de lo

contrario la cantidad de medios necesarios para los enlaces, y la cantidad de conexiones con los enlaces se torna abrumadora.

Topología de red celular

La topología celular está compuesta por áreas circulares o hexagonales, cada una de las cuales tiene un nodo individual en el centro.



Capa de acceso a la red:

La capa de acceso a la red es la primera capa de la pila TCP/IP, normalmente formada por un red LAN o WAN, esta capa brinda los recursos que se deben implementar para transmitir datos a través de la red.

Desde el punto de vista físico, el hardware más utilizado para LAN es conocido como Ethernet (o FastEthernet o GigaEthernet). Sus ventajas son su bajo costo y la facilidad en su instalación.

Esta tecnología utiliza elementos intermedios de comunicación (hubs, switchs, routers) para configurar múltiples segmentos de red y dividir el tráfico para mejorar las prestaciones de transferencia de información. Normalmente, en las grandes instituciones estas LAN Ethernet están interconectadas a través de fibra óptica utilizando tecnología FDDI (fiber distributed data interface) que es mucho más cara y compleja de instalar, pero se pueden obtener velocidades de transmisión superiores a Ethernet y no tienen la limitación de la distancia (FDDI admite distancias de hasta 200 km).

Las interfaces Ethernet en GNU/Linux se llaman “ethX” (la X indica un número de orden comenzando por 0), los módems se llaman “pppX” (para PPP), para FDDI son “fddiX”. Estos nombres son utilizados por los comandos para configurarlas y asignarles el número de identificación que posteriormente permitirá comunicarse con otros dispositivos en la red.

¿Como ver las interfaces ethernet?

```
root@debian2:~# ifconfig -a

eth0    Link encap:Ethernet  HWaddr 48:5b:39:95:3d:23
          inet addr:192.168.1.20  Bcast:192.168.1.255
Mask:255.255.255.0
        inet6 addr: fe80::4a5b:39ff:fe95:3d23/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:78183 errors:0 dropped:0 overruns:0 frame:0
        TX packets:72318 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:84319533 (80.4 MiB)  TX bytes:8144722 (7.7 MiB)
        Interrupt:27 Base address:0xe000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:2537 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2537 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:252295 (246.3 KiB)  TX bytes:252295 (246.3 KiB)
```

Dirección de red Ethernet (Ethernet address o MAC address):

Es un número de 48 bits (por ejemplo 00:88:40:73:AB:FF en hexadecimal) que se encuentra en el dispositivo físico (hardware) del controlador (NIC) de red Ethernet y es grabado por el fabricante del mismo (este número debe ser único en el mundo, por lo que cada fabricante tiene un rango pre-asignado).

Se puede cambiar la MAC address en forma temporal :

```
# ifdown eth0
# ifconfig eth0 hw ether  48:5b:39:95:3d:15
# ifup eth0
```

O también instalando el paquete **macchanger**.

En este ejemplo nos muestra la mac address y la que pone automáticamente es la consecutiva de la que tiene definida.

```
# macchanger eth0


Current MAC: 00:09:a5:eb:23:f7 (Hansung Eletronic Industries
Development Co., Ltd)
Faked MAC: 00:09:a5:eb:23:f8 (Hansung Eletronic Industries
Development Co., Ltd)
```

Para volver al estado anterior realizamos lo siguiente :

```
# macchanger --endding eth0
```

Pone al azar una mac address.

```
# macchanger --another eth0
```

También se lo podemos poner a mano :

```
# macchanger --mac=00:09:a5:eb:23:f8 eth0
```

Para listar las mac de los vendedores :

```
# macchanger --list=Realtek
```

Trama Ethernet

Preámbulo. Este campo tiene una extensión de 7 bytes que siguen la secuencia <<10101010>>.

Inicio. Es un campo de 1 byte con la secuencia <<10101011>>, que indica que comienza la trama.

Dirección de destino. Es un campo de 2 o 6 bytes que contiene la dirección del destinatario. Aunque la norma permite las dos longitudes para este campo, la utilizada en la red de 10 Mbps es la de 6 bytes. Esta dirección puede ser local o global. Es local cuando la dirección sólo tiene sentido dentro de la propia red, y suele estar asignada por el administrador de red.

Una dirección global (dirección MAC o dirección Ethernet) es única para cada tarjeta de red, normalmente codifica la compañía constructora de la tarjeta y un número de serie. El bit de mayor orden de este campo, que ocupa el lugar 47, codifica si la dirección de destino es un único destinatario (bit puesto a 0) o si representa una dirección de grupo (bit puesto a 1). Una dirección de grupo es la dirección a la que varias estaciones tienen derecho de escucha (transmisión de uno a varios). Cuando todos los bits del campo dirección están a 1, se codifica una difusión o «broadcast », es decir, codifica una trama para todas las estaciones de la red. El sistema sabe si se trata de una dirección local o global analizando el valor del bit 46.

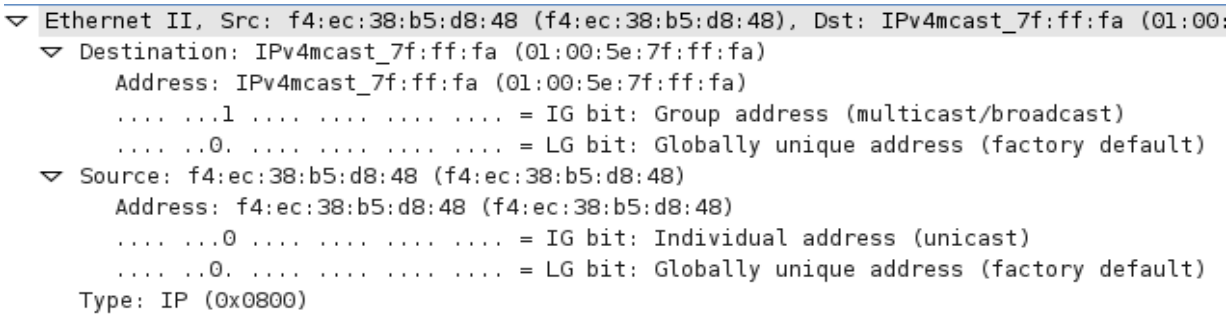
Dirección de origen. Es semejante al campo de dirección de destino, pero codifica la dirección MAC de la estación que originó la trama, es decir, de la tarjeta de red de la estación emisora.

Longitud. Este campo de dos bytes codifica cuántos bytes contiene el campo de datos. Su valor oscila en un rango entre 0 y 1 500.

Datos. Es un campo que puede codificar entre 0 y 1500 bytes en donde se incluye la información de usuario procedente de la capa de red.

Relleno. La norma IEEE 802.3 especifica que una trama no puede tener un tamaño inferior a 64 bytes, por tanto, cuando la longitud del campo de datos es muy pequeña se requiere rellenar este campo para completar una trama mínima de al menos 64 bytes. Es un campo que puede, por tanto, tener una longitud comprendida entre 0 y 46 bytes, de modo que la suma total de la trama sea al menos de 64 bytes.

CRC. Es el campo de 4 bytes en donde se codifica el control de errores de la trama.



El principio de transmisión

Todos los equipos de una red Ethernet están conectados a la misma línea de transmisión y la comunicación se lleva a cabo por medio de la utilización un protocolo denominado CSMA/CD (Carrier Sense Multiple Access with Collision Detect) que significa que es un protocolo de acceso múltiple que monitorea la portadora: detección de portadora y detección de colisiones).
Con este protocolo cualquier equipo está autorizado a transmitir a través de la línea en cualquier momento y sin ninguna prioridad entre ellos. Esta comunicación se realiza de manera simple:

- Cada equipo verifica que no haya ninguna comunicación en la línea antes de transmitir.
- Si dos equipos transmiten simultáneamente, entonces se produce una colisión (o sea, varias tramas de datos se ubican en la línea al mismo tiempo).
- Los dos equipos interrumpen su comunicación y esperan un período de tiempo aleatorio, luego una vez que el primero ha excedido el período de tiempo, puede volver a transmitir.

Este principio se basa en varias limitaciones:

- Los paquetes de datos deben tener un tamaño máximo.
- Debe existir un tiempo de espera entre dos transmisiones.

El tiempo de espera varía según la frecuencia de las colisiones:

- Luego de la primera colisión, un equipo espera una unidad de tiempo.
- Luego de la segunda colisión, un equipo espera dos unidades de tiempo.
- Luego de la tercera colisión, un equipo espera cuatro unidades de tiempo.
- ... Por supuesto, con una cantidad menor de tiempo aleatorio adicional.

Ethernet conmutada

La topología de Ethernet descripta hasta ahora ha sido la de Ethernet compartida (cualquier mensaje transmitido es escuchado por todos los equipos conectados y el ancho de banda disponible es compartido por todos los equipos).

Durante muchos años se ha dado un desarrollo importante: la Ethernet conmutada.
La topología física sigue siendo la de una estrella pero está organizada alrededor de un conmutador. El conmutador usa mecanismos de filtrado y conmutación muy similares a los utilizados por las puertas de enlace donde se han utilizado estas técnicas por mucho tiempo.
Inspecciona las direcciones de origen y destino de los mensajes, genera una tabla que le permite saber qué equipo se conecta a qué puerto del conmutador (en general este proceso se hace por auto aprendizaje, es decir, de manera automática pero el administrador del conmutador puede realizar ajustes adicionales).

Al conocer el puerto receptor, el conmutador sólo transmitirá el mensaje al puerto adecuado mientras que los otros puertos permanecerán libres para otras transmisiones que pueden ser realizadas simultáneamente.
Como resultado, cada intercambio puede llevarse a cabo a una velocidad nominal (mayor división de ancho de banda), sin colisión.

La capa de Internet

Esta capa permite el enrutamiento de datagramas (datos a los que se les agrega un encabezado que contiene información sobre su transporte) a equipos remotos junto con la administración de su división y ensamblaje cuando se reciben.

Comenzaremos definiendo que es una dirección IP y como se asignan para luego poder abordar los protocolos mas relevantes (IP, ARP y ICMP) de esta capa.

Dirección IP

Es utiliza universalmente para identificar los ordenadores sobre una red o Internet de manera univoca, está compuesta por cuatro números en el rango 0-255 separados por puntos (por ejemplo 192.168.0.1).
La traducción de nombres (“PC01”) en direcciones IP es realizada por un servidor DNS (domain name system) que transforma los nombres de nodo (legibles por humanos) en direcciones IP.

Ej: Obtener el host name y el servidor DNS de nuestro equipo.

```
root@debian2# more /etc/hostname
debian2

root@debian2# more /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.0.1
```

¿Cómo se asigna una dirección IP?

Una dirección IP tiene dos campos. El izquierdo representa la identificación de la red y el derecho la identificación del nodo.

Existen algunas restricciones, 0 (por ejemplo, 0.0.0.0) en el campo de red está reservado para el routing por defecto y 127 (por ejemplo, 127.0.0.1) está reservado para la autorreferencia (local loopback o local host), 0 en la parte de nodo se refiere a esta red (por ejemplo, 192.168.0.0) y 255 está reservado para paquetes de envío a todas las máquinas (broadcast) (por ejemplo,198.162.255.255).

En las diferentes asignaciones se puede tener diferentes tipos de redes o direcciones:

Clase A (red.host.host.host): 1.0.0.1 a 126.254.254.254 (126 redes, 16 millones de nodos) definen las grandes redes. El patrón binario es: 0 + 7 bits red + 24 bits de nodos.

Clase B (red.red.host.host): 128.1.0.1 a 191.255.254.254 (16K redes, 65K nodos) generalmente se utiliza el primer byte de nodo para identificar subredes dentro de una institución). El patrón binario es 10 + 14 bits de red + 16 bits de nodos.

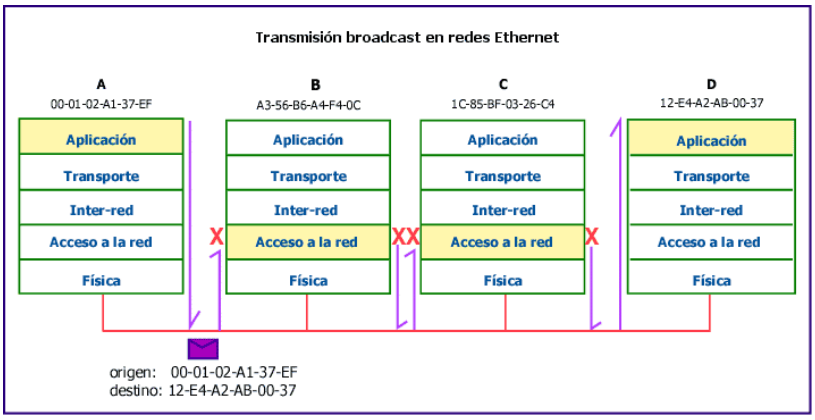
Clase C (red.red.red.host): 192.1.1.1 a 223.255.255.254 (2 millones de redes,254 de nodos). El patrón binario es 110 + 21 bits red + 8 bits de nodos.

Clase D y E (red.red.red.host): 224.1.1.1 a 255.255.255.254 reservado para multicast (desde un nodo a un conjunto de nodos que forman parte de un grupo)y propósitos experimentales.

Algunos rangos de direcciones han sido reservados para que no correspondan a redes públicas, sino a redes privadas (máquinas que se conectan entre ellas sin tener conexión con el exterior) y los mensajes no serán encaminados a través de Internet, lo cual es comúnmente conocido como intranet).
Éstas son para la clase A 10.0.0.0 hasta 10.255.255.255, clase B 172.16.0.0 hasta 172.31.0.0 y clase C 192.168.0.0 hasta 192.168.255.0.

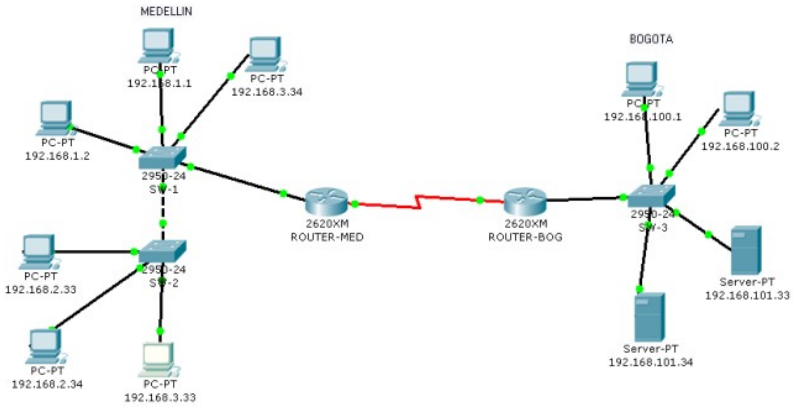
Clase de dirección	Bits de mayor peso	Intervalo de dirección del primer octeto	Número de bits en la dirección de red	Número de redes	Número de hosts por red
Clase A	0	0-127	8	126	16,777,216
Clase B	10	128-191	16	16,384	65,536
Clase C	110	192-223	24	2,097,152	254
Clase D	1110	224-239	28	No es aplicable	No es aplicable

La dirección de broadcast es especial, ya que cada nodo en una red escucha todos los mensajes (además de su propia dirección). Esta dirección permite que datagramas (generalmente información de routing y mensajes de aviso) puedan ser enviados a una red y todos los nodos del mismo segmento de red los puedan leer. Por ejemplo, cuando ARP busca encontrar la dirección Ethernet correspondiente a una IP, éste utiliza un mensaje de broadcast, el cual es enviado a todas las máquinas de la red simultáneamente. Cada nodo en la relee este mensaje y compara la IP que se busca con la propia y le retorna un mensaje al nodo que hizo la pregunta si hay coincidencia.

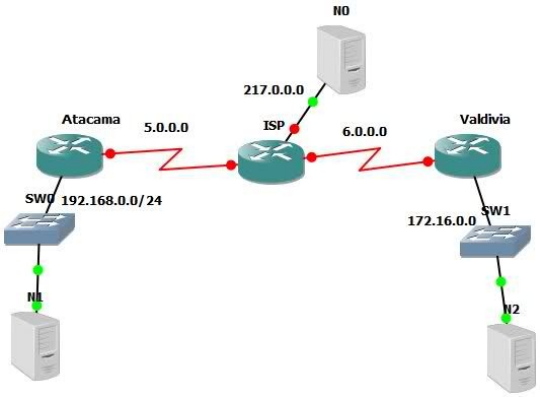


Subredes y routing

Subredes significa subdividir la parte del nodo en pequeñas redes dentro de la misma red para, por ejemplo, mejorar el tráfico. Una subred toma la responsabilidad de enviar el tráfico a ciertos rangos de direcciones IP extendiendo el mismo concepto de redes A, B, C, pero sólo aplicando esta redirección en la parte nodo de la IP. El número de bits que son interpretados como identificador de la subred es dado por una máscara de red(netmask) que es un número de 32 bits (igual que la IP). Para obtener el identificador de la subred, se deberá hacer una operación lógica Y (AND) entre la máscara y la IP, lo cual dará la IP de la subred.



El segundo concepto, routing, representa el modo en que los mensajes son enviados a través de las subredes.



Por ejemplo, sea una institución que tiene una red clase B con número 172.17.0.0, y su netmask es, por lo tanto,255.255.0.0. Internamente, esta red está formada por pequeñas redes (una planta del edificio por ejemplo). Así, el rango de direcciones es reasignado en 20 subnets (plantas para nosotros) 172.17.1.0 hasta 172.17.20.0. El punto que conecta todas estas plantas (backbone) tiene su propia dirección, por ejemplo 172.17.1.0.Estas subredes comparten el mismo IP de red, mientras que el tercero es utilizado para identificar cada una de las subredes dentro de ella (por eso se utilizará una máscara de red 255.255.255.0).

Por ejemplo, sean tres departamentos con subredes Ethernet:

1. Compras (subred 172.17.2.0),
2. Clientes (subred 172.17.4.0),
3. Recursos humanos, RR.HH., (subred 172.17.6.0)

4. Backbone con FFDI (subred 172.17.1.0).

Para encaminar los mensajes entre los ordenadores de las tres redes, se necesitarán tres gateways que tendrán cada uno dos interfaces de red para cambiar entre Ethernet y FFDI. Éstas serán:

1. ComprasGW IPs:172.17.2.1 y 172.17.1.1,
2. ClientesGW IPs:172.17.4.1 y 172.17.1.2
3. RRHHGW IPs:172.17.6.1 y 172.17.1.3, es decir, una IP hacia el lado de la subnet y otra hacia el backbone.

Cuando se envían mensajes entre máquinas de compras, no es necesario salir al gateway, ya que el protocolo TCP/IP encontrará la máquina directamente. El problema está cuando la máquina Compras0 quiere enviar un mensaje a RRHH3. El mensaje debe circular por los dos gateways respectivos. Cuando Compras0 “ve” que RRHH3 está en otra red, envía el paquete a través del gateway ComprasGW,que a su vez se lo enviará a RRHHGW y que a su vez se lo enviará a RRHH3. La ventaja de las subredes es clara, ya que el tráfico entre todas las máquinas de compras, por ejemplo, no afectará a las máquinas de clientes o RR.HH. (si bien significa un planteamiento más complejo y caro a la hora de diseñar, y construir la red).

----------GRAFICO

IP utiliza una tabla para hacer el routing de los paquetes entre las diferentes redes y en la cual existe un routing por defecto asociado a la red 0.0.0.0. Todas las direcciones que coinciden con ésta, ya que ninguno de los 32 bits son necesarios, son enviadas por el gateway por defecto (default gateway) hacia la red indicada.

Sobre comprasGW, por ejemplo, la tabla podría ser:

----------TABLA

El ‘-’ significa que la máquina está directamente conectada y no necesita routing. El procedimiento para identificar si se realiza el routing o no, se lleva a cabo a través de una operación muy simple con dos AND lógicos (subred AND mask y origen AND mask) y una comparación entre los dos resultados. Si son iguales no hay routing, sino que se debe enviar la máquina definida como gateway en cada máquina para que ésta realice el routing del mensaje.

Por ejemplo, un mensaje de la 172.17.2.4 hacia la 172.17.2.6 significará:

172.17.2.4 AND 255.255.255.0 = 172.17.2.0
172.17.2.6 AND 255.255.255.0 = 172.17.2.0

Como los resultados son iguales, no habrá routing. En cambio, si hacemos lo mismo con 172.17.2.4 hacia 172.17.6.6 podemos ver que habrá un routing a través del 172.17.2.1 con un cambio de interfaz (eth0 a ffdi0) a la 172.17.1.1 y de ésta hacia la 172.17.1.2 con otro cambio de interfaz (fddi0 a eth0) y luego

hacia la 172.17.6.6. El routing, por defecto, se utilizará cuando ninguna regla satisfaga la coincidencia. En caso de que dos reglas coincidan, se utilizará aquella que lo haga de modo más preciso, es decir, la que menos ceros tenga.

Para construir las tablas de routing, se puede utilizar el comando route durante el arranque de la máquina, pero si es necesario utilizar reglas más complejas(o routing automático), se puede utilizar el routing information protocol (RIP) o entre sistemas autónomos el external gateway protocol (EGP) o también el border gateway protocol (BGP). Estos protocolos se implementan en el comando gated.

Si se construye una red que nunca tendrá conexión a Internet, se pueden escoger las direcciones que se prefieran, pero es recomendable mantener un orden adecuado en función del tamaño de red que se desee tener y para evitar problemas de administración dentro de dicha red.

A continuación, se verá cómo se define la red y el nodo para una red privada (hay que ser cuidadoso,ya que si se tiene la máquina conectada a la red, se podría perjudicar a otro usuario que tuviera asignada esta dirección): dirección de nodo 192.168.110.23, máscara de red 255.255.255.0, parte de red 192.168.110., parte de nodo .23, dirección de red 192.168.110.0,dirección de broadcast192.168.110.255.

Hay un programa llamado ipcalc que nos sirve para sacar subredes.

```
# apt-get install ipcalc
```

ejemplo :

```
# ipcalc 192.168.0.1/24
# ipcalc 192.168.0.1 255.255.128.0 255.255.192.0
```

Configuración de la red en Debian

Configurar la interfaz de red implica dos pasos: asignar la dirección de red al dispositivo e inicializar los parámetros de la red al sistema. El comando utilizado para ello es el ifconfig (interface configure).

Un ejemplo será:

ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up

Lo cual indica configurar el dispositivo eth0 con dirección IP 192.168.110.23 y máscara de red 255.255.255.0. El up indica que la interfaz pasará al estado activo (para desactivarla debería ejecutarse ifconfig eth0 down).

El comando asume que si algunos valores no se indican, son tomados por defecto. En este caso, el kernel configurará esta máquina como Tipo-C y configurará la red con 192.168.110.23 y la dirección de broadcast con 192.168.110.255.

Por ejemplo:

ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up

Existen comandos como el ifup e ifdown, que permite configurar/desconfigurar la red en forma más simple utilizando el archivo /etc/network/interfaces para obtener todos los parámetros necesarios (consultar man interfaces para su sintaxis).

En Debian, con el fin de facilitar la configuración de la red, existe otra forma de configurar la red (considerada de alto nivel)que utiliza los comandos mencionados anteriormente ifup, ifdown y el archivo /etc/network/interfaces.

Si se decide utilizar estos comando no se debería configurar la red a bajo nivel,ya que estos comandos son suficientes para configurar/desconfigurar la red.Para modificar los parámetros de red de la interfaz eth0 se puede hacer (consultar man interfaces en la sección 5 del manual de Unix incluido con el sistema operativo para más información del formato):

```
   ifdown eth0                          para todos los servicios de red sobre eth0
   vi /etc/network/interfaces  edite y modifique los que necesite
   ifup eth0                            pone en marcha los servicios de red sobre
eth0
```

Supongamos que desea configurar sobre Debian una interfaz eth0 que tiene una dirección IP fija 192.168.0.123. y con 192.168.0.1 como puerta de enlace (gateway). Se debe editar /etc/network/interfaces de modo que incluya una sección como:

```
iface eth0 inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1
```

Si tiene instalado el paquete resolvconf puede añadir líneas para especificar la información relativa al DNS.

Por ejemplo:

```
iface eth0 inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1
        dns-search remix.org
        dns-nameservers 195.238.2.21 195.238.2.22
```

Después de activarse la interfaz, los argumentos de las opciones dns-search y dns-nameservers quedan disponibles para resolvconf para su inclusión en resolv.conf. El argumento remix.org de la opción dns-search corresponde al argumento de la opción search en resolv.conf (se verá más adelante) y los argumentos 195.238.2.21 y 195.238.2.22 de la opción dns-nameservers corresponde a los argumentos de las opciones nameserver en resolv.conf(consultar man resolv.conf). También se puede configurar la red a bajo nivel a través del comando ip (que es equivalente a ifconfig y route). Si

bien este comando es mucho más versátil y potente (permite establecer túneles, routing alternativos, etc) es más complejo y se recomienda utilizar los procedimientos anteriores para configuraciones básicas de la red.

Configuración de un red Wi-Fi (inalámbrica)

Para la configuración de interfaces Wi-Fi se utiliza básicamente el paquete wireless-tools (además de ifconfig o ip). Este paquete utiliza el comando iwconfig para configurar una interfaz inalámbrica, pero también se puede hacer a través del /etc/network/interfaces.

Por Ejemplo:

```
iface eth1 inet dhcp
   wpa-ssid "Nombre de la Wifi"
   wpa-psk 123456789e
   address 192.168.1.132
   netmask 255.255.255.0
   network 192.168.0.0
   broadcast 192.168.0.255
   gateway 192.168.1.1
```

Configuración del Name Resolver

El siguiente paso es configurar el name resolver que convierte nombres tales como pirulo.remix.com en 192.168.110.23. El archivo /etc/resolv.conf es el utilizado para tal fin. Su formato es muy simple (una línea de texto por sentencia). Existen tres palabras clave para tal fin: domain (dominio local), search (lista de dominios alternativos) y name server (la dirección IP del domain name server).

Ejemplo de /etc/resolv.conf

```
    domain remix.com
    search remix.com piru.com
    name server 192.168.110.1
    name server 192.168.110.65
```

Esta lista de servidores de nombre a menudo dependen del entorno de red,

que puede cambiar dependiendo de dónde esté o se conecte la máquina. Los programas de conexión a líneas telefónicas (pppd) o obtención de direcciones IP automáticamente (dhclient) son capaces de modificar resolv.conf para insertar o eliminar servidores; pero estas características no siempre funcionan adecuadamente y a veces pueden entrar en conflicto y generar configuraciones erróneas .

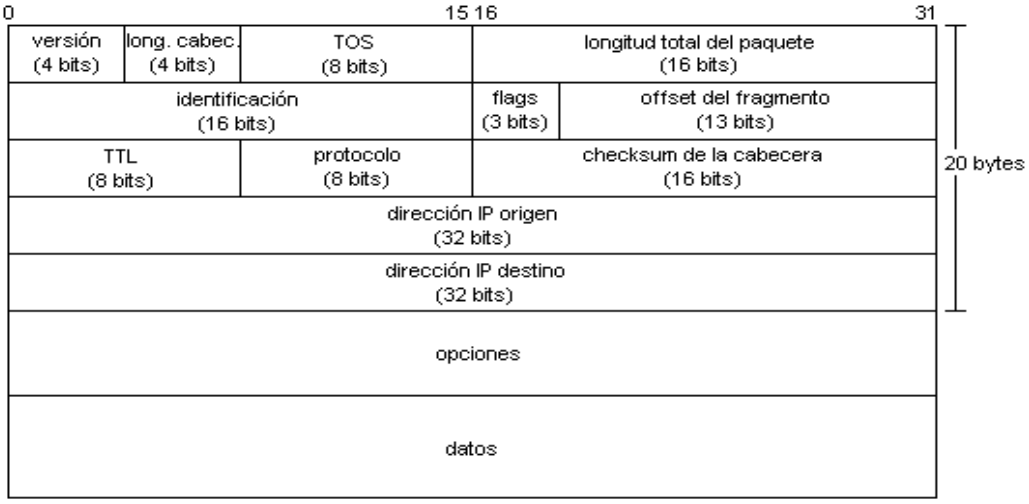
Protocolo IP

Se trata de un protocolo de red que permite la comunicación de ordenadores conectados a redes diferentes. Otros protocolos como TCP, UDP, ICMP, etc. usan datagramas IP para transportar su información. Tres características son especialmente significativas en este protocolo:

1. Es un protocolo no orientado a conexión (uso de datagramas). No se guarda información sobre los sucesivos datagramas que se envían o reciben. Esto conlleva que si dos paquetes cambian su orden relativo en el camino entre un host y otro, llegando antes el enviado más tarde debido a la variación en la ruta seguida, serán pasados desordenados a la capa superior.

2. Es un protocolo no fiable, no se garantiza que un paquete que se pase al protocolo para ser enviado llegue al otro extremo. Los paquetes pueden perderse o corromperse sin generar por ello comunicaciones de error ni retransmisiones. Si se detecta un error en un datagrama, el datagrama simplemente se descarta. La fiabilidad en la comunicación la debe aportar una capa superior (por ejemplo TCP, protocolo de transporte orientado a conexión).

3. Puede adaptarse a las diferentes características de las redes por las que ha de pasar un paquete, ya que permite la fragmentación de un datagrama en varios fragmentos. Los fragmentos deberán recomponerse de nuevo en un datagrama único al llegar al host destinatario. De nuevo aquí puede producirse un desorden en el orden de llegada de los fragmentos. Esta vez la capa IP deberá ser capaz de restablecer el orden adecuado, pues la fragmentación ha de ser totalmente transparente a la capa superior. Eso sí, como es lógico no se ejerce aquí tampoco ningún control sobre las perdidas o corrupciones de los fragmentos, que provocarán el descarte de un datagrama entero.



Campos que componen el paquete IP:

Versión: siempre vale cuatro (0100), para los paquetes de la versión actual (IPv4).

Longitud de la cabecera: da la longitud de la cabecera en palabras de 32 bits (4 bytes). Por tanto, el número de bytes de la cabecera tiene que ser múltiplo de 4. Asimismo, limita la longitud de la cabecera a 60 bytes (15 · 4), puesto que 15 es el máximo que se puede expresar con cuatro dígitos binarios. Si no hay campo de opciones, la cabecera tiene 20 bytes; por tanto, el campo en este caso valdría 5 (5 · 4 bytes = 20 bytes).

Tipo de servicio: este campo se encuentra dividido en varios subcampos. Permite pedir un trato especial para el paquete y rara mente se implementa.

Longitud total del paquete: da la longitud total del paquete (cabecera incluida) en bytes. Como este campo es de 16 bits, un paquete IP no puede tener más de 65.535 bytes (216 − 1).

Identificación del paquete: contiene un identificador que es diferente para cada paquete que genera la estación.

Indicadores (flags) y Posición de este fragmento: estos campos permiten gestionar la fragmentación de paquetes.

Tiempo de vida o TTL (Time To Live): indica el número máximo de direccionadores que puede cruzar el paquete. Se utiliza para evitar que un paquete pueda quedar dando vueltas indefinidamente dentro de la red en caso de que haya algún problema al entregarlo. Cada direccionador disminuye el campo en uno; cuando uno de ellos detecta un paquete con TTL = 1, lo elimina y envía a quien lo ha emitido un mensaje de error por medio de un mensaje ICMP.

Protocolo: identifica el tipo de protocolo que transporta el paquete. Los valores más comunes de los diferentes tipos de protocolos son los siguientes:

TCP	6
UDP	17
ICMP	1

Checksum: realiza el control de errores en la cabecera. Elde datos no queda protegido por ningún checksum; es responsabilidad de los usuarios del IP (TCP, UDP, etc.) el control de los posibles errores en su contenido.

Dirección de origen IP: identifica la máquina que ha generado el paquete.

Dirección de destino IP: identifica la máquina a la que va destinado el paquete.

Opciones: existen diferentes servicios asignados a este campo,pero por lo general no se utilizan. Comentaremos algunos de los mismos cuando hablemos del ICMP. Sea como sea, la longitud total está limitada a 40 bytes (15 · 4 − 20).

IP Versión 6 (IP Next generation).

Desde hace tiempo ya se hacen evidentes algunas falencias que hoy tiene la actual versión del Protocolo IP (Versión 4). Algunas de ellas son la mala distribución que utiliza de sus cantidades de Host en cada una de sus redes (A, B y C); son muy pocas o ninguna las empresas que poseen los millones de Host que permite una dirección tipo A, hoy tampoco existen 2.100.000 empresas, por lo tanto, tanto en A como en C se desperdicia la asignación de direcciones, si bien hoy se asignan porciones de las mismas, este no fue el sentido con que fueron creadas.

Otra debilidad es la no posibilidad asignaciones geográficas , lo que representa una enorme carga de tablas en los router exteriores, casi ya imposible de controlar.

También se suman detalles de seguridad, criptogrfiado, Prioridades, longitud variable de cabecera, etc.

Características:

Las características principales de IPv6 son:

Direccionamiento: 128 bit.

Encaminamiento: Direccionamiento jerárquico.

Prestaciones: Cabecera simple de 40 Byte, alineada de a 64 bit.

Versatilidad: Formato flexible de opciones.

Multimedia: Id de flujos.

Multicast: Obligatorio.

Seguridad: Soporte de autenticación y cifrado.

Auto-configuración: Tres métodos PnP.

Movilidad: Surce routing, seguridad, detección de móviles, hand-off.

Fragmentación: Únicamente de extremo a extremo, es decir que sólo el origen puede fragmentar. Para implementar esto, hace uso de PMTU (Path MTU, RFC 1191), que es el mecanismo empleado para determinar la máxima unidad de datos que contendra un datagrama, una vez conocido este tamaño, armara todos los paquetes sin superar el mismo, por ende ningún router deberá fragmentarlo pues no será necesario.

Tamaño de datagrama: Mantiene el mismo concepto que la versión 4 y propone un nuevo modelo de datagrama, llamado Jumbograma, el cual se define a través de una cabecera en extensión, y permite transmitir datagramas de hasta 4 Gbyte. La idea de esta nueva aplicación es permitir la transmisión de grandes volúmenes de datos entre servidores, los cuales no necesitan incrementar con tanta redundancia de cabecera, siendo el mejor representante de esto el empleo de cluster de servidores.

Encabezado de IPv6:

Versión Clase de tráf. Rótulo de flujo
Longitud de carga útil Sig. Cabecera Límite Saltos
Dirección fuente
Dirección destino
Posibles cabeceras de extensión

Versión: (4), se mantiene el mismo tamaño para permitir distinguirlo del versión 4 y que puedan convivir durante algún lapso de tiempo.

Clase de tráfico (4): (Video, audio, datos, voz, etc).Cuanto más alto sea su valor más importante es, los valores que puede adoptar son:

0 tráfico sin caracterizar
1 tráfico “filler”
2 transferencia de datos no atendida, como E-mail
3 reservado
4 transferencia de bloques de datos atendida, como FTP
5 reservado
6 tráfico interactivo, como TELNET

7 tráfico de control de Internet, como protocolos de encaminamiento

Rótulo de flujo: (24), Todos los datagramas del mismo flujo (Ej: todos los datagramas de una misma FTP).

Longitud de carga útil: (16): Cantidad de octetos de datos.

Siguiente cabecera: (8), se permiten varias, todas ellas van después del campo Dirección destino y aquí se identifican cuales van.

Límite de saltos: (8), para evitar lazos infinitos.

Dirección origen y destino (128 c/u), aparece aquí aparte de Net y Host un nuevo identificador llamado Dirección de Agrupación, que identifica regiones topológicas.

Posibles cabeceras de extensión (Extension Headers): Irán colocadas antes del campo de datos, cada cabecera tendrá un primer campo (8 bit) que indica la próxima cabecera (Next Header) que indica si existe otra cabecera de extensión o si esta es la última:

-Cabecera salto por salto (valor 0): Lleva información para analizar en cada router.

-Cabecera extremo a extremo: Lleva información que solo se examinará en el destino.

-Cabecera de enrutamiento (valor 43): Ruta fija.

-Cabecera de fragmento (valor 44): Si existe fragmentación.

-Cabecera de verificación de autenticidad(valor 51): Permite verificar autenticidad de origen.

-Cabecera de confidencialidad: Los datos no deben ser leídos durante su paso por Internet.

Direccionamiento de IPv6:

Direcciones de 128 bit (16 octetos) (más de 1038 direcciones posibles).

A pesar de las restricciones de redes y reservadas aún quedan más de 1.500 direcciones por m2 de la superficie de la tierra.

Tres tipos de direcciones (unicast, anycast y multicast).

No existen clases, similar al concepto de CIDR.

Notación general

3FFE:2213:AE56:54AD:34EF:9888:33EA:AA21

Los ceros contiguos se pueden eliminar, es decir los siguientes pares de octetos se podrían representar como están indicados a su derecha:

:002E:	:2E:
:000A:	:A:
:6700:	:6700:

Protocolo ARP y RARP

En algunas redes (como por ejemplo IEEE 802 LAN que es el estándar para Ethernet), las direcciones IP son descubiertas automáticamente a través de dos protocolos miembros de Internet protocol suite: address resolution protocol (ARP) y reverse address resolution protocol (RARP).

ARP utiliza mensajes (broadcast messages) para determinar la dirección Ethernet (especificación MAC de la capa 3 del modelo OSI) correspondiente a una dirección de red particular (IP).

RARP utiliza mensajes de tipo broadcast (mensaje que llega a todos los nodos) para determinar la dirección de red asociada con una dirección hardware en particular. RARP es especialmente importante en máquinas sin disco, en las cuales la dirección de red generalmente no se conoce en el momento del inicio (boot).

Es fácil deducir que,en el funcionamiento normal de una estación, las peticiones ARP pueden ser muy habituales. Para evitarlo, cada estación dispone de una tabla con las parejas ya obtenidas, de forma que si una dirección IP ya se encuentra es dicha tabla, no hace falta enviar ninguna petición ARP. Dicha tabla se denomina caché ARP, y tiene el siguiente aspecto:

Tabla caché ARP

	Dirección IP	Dirección MAC	Interfaz
1	147.83.153.103	08:00:00:10:97:00	ether0
2	147.83.153.5	00:0c:aa:00:0f:e0	ether0

Cada fila corresponde a un mapeo IP-MAC y la interfaz para la cual se ha solicitado. Esta tabla se va llenando automáticamente a medida que se realizan nuevas peticiones, pero también se puede manipular con el comando arp. Con él se puede consultar, añadir o borrar entradas.

ICMP (Internet Control Messaging Protocol)

Este protocolo se encarga de la supervisión y control de la red. Un datagrama viaja entre router a través de la red hasta alcanzar su destino, si ocurre algún error o para controlar esta travesía es que se generan estos mensajes. ICMP no especifica las acciones a tomar, solamente sugiere la misma.

Las cabeceras de ICMP comienzan con tres campos:

TIPO: (8), especifica el mensaje ICMP.
CODIGO: (8), Brinda un poco más de información sobre el error.
CHECKSUM (16), CRC 16.

Los campos que continúan a estos tres, varían acorde al tipo de error, pero en la mayoría de ellos se encuentra incluido el encabezado del datagrama que generó el mensaje ICMP y también los 64 primeros octetos de este para dejar unívocamente establecido la identificación del error.

El estudio del funcionamiento del protocolo ICMP se puede entender básicamente desarrollando el significado del campo TIPO, el cual representa los distintos tipos de mensajes.

Tipos y códigos de los mensajes ICMP:

0 y 8: Eco de solicitud y de respuesta: No es ni más ni menos que el comando PING,que genera una solicitud y una respuesta (configurable), para determinar la continuidad del recorrido de un datagrama a lo largo de una red, su cantidad de saltos y el tiempo demorado.

3: Destino no alcanzable: Se genera cuando un datagrama no encuentra la dirección IP destino. También ocurre cuando el bit de no fragmentar de la cabecera IP esta puesto en 1, y la red destino no soporta bloques del tamaño de ese datagrama, por lo cual no podrá ser entregado a esa red, causando la no llegada a destino. Dentro de este tipo es interesante tener en cuenta el campo Código, pues brinda información adicional sobre las causas por las cuales no se llega a destino, en particular si lo que se desea es obtener información sobre ese extremo (Extremadamente usado para ataques a redes).

Los valores que toma son:

-0: Red inalcanzable.
-1: Host inalcanzable.
-2: Protocolo inalcanzable.
-3: Puerto inalcanzable.
-4: Fragmentación requerida y bit de no fragmentar puesto a 1 en el datagrama
origen.
-5: Falla en la ruta.
-6: Red desconocida.
-7: Host desconocido.
-8: Host origen aislado.
-9: Acceso a la red administrativamente prohibido.
-10: Acceso al Host administrativamente prohibido.
-11: Red inalcanzable por tipo de servicio.
-12: Host inalcanzable por tipo de servicio.

4: Fuente agotada: Sirve para regular el flujo de información. Implica un buffer lleno, causa por la cual sería conveniente que el Host transmisor dejara de hacerlo hasta que deje de recibir estos mensajes.

5: Se requiere redireccionamiento: Existe una ruta mejor.

11: Tiempo de vida excedido: El campo TTL llegó a 0.

12: Problemas con el parámetro: Error semántico o sintáctico en el encabezamiento IP.

13 y 14: Solicitud y respuesta de marcador de tiempo: Permite la sincronización de clock entre nodos, a través de la hora GMT (Greenwich Mean Time).

15 y 16: Solicitud y repuesta de información: Permite obtener información de un nodo. Este fue originariamente pensado para los protocolos BOOTP y R_ARP.

17 y 18: Solicitud y respuesta de máscara de dirección: Permite determinar las máscaras de las redes con que está conectada un nodo. Se emplea para el ruteo hacia esas redes.

La capa de transporte

Los protocolos de las capas anteriores permiten enviar información de un equipo a otro. La capa de transporte permite que las aplicaciones que se ejecutan en equipos remotos puedan comunicarse. El problema es identificar estas aplicaciones.

Su objetivo principal es establecer una comunicación extremo a extremo a través de una red. En otras palabras, actuar de interfaz entre los niveles orientados a la aplicación y los niveles orientados a la red de la jerarquía de protocolos (tanto OSI como TCP/IP).

El nivel de transporte oculta a los niveles altos del sistema el tipo de tecnología (red) al que está conectado el terminal.

En este apartado nos interesan los dos protocolos del nivel de transporte que se definen en la pila TCP/IP: UDP y TCP. UDP es no orientado a la conexión, mientras que TCP es orientado a la conexión.

En el nivel de transporte se definen dos direcciones que lo relacionan con los niveles superior e inferior:

• La dirección IP, que ya conocemos, es la dirección que identifica un subsistema dentro de una red.
• El puerto identifica la aplicación que requiere la comunicación.

Para identificar las diferentes aplicaciones, los protocolos TCP/IP marcan cada paquete (o unidad de información) con un identificador de 16 bits llamado puerto.

La verdadera utilidad de los puertos es que permiten multiplexar aplicaciones sobre protocolos del nivel de transporte. Ello significa que un mismo protocolo de transporte lleva información de diferentes aplicaciones y estas últimas son identificadas por el puerto.

Si alguna aplicación que corre en un terminal quiere establecer una comunicación con un servidor o con otro terminal, debe utilizar un protocolo de transporte: el TCP o el UDP. Como el destino puede encontrarse en una red remota, los protocolos de transporte necesitan el protocolo Internet para poder llegar al terminal o servidor remoto.

El UDP (user datagram protocol)

Es un protocolo no orientado a la conexión, de manera que no proporciona ningún tipo de control de errores ni de flujo, aunque
utiliza mecanismos de detección de errores. En caso de detectar un error, el UDP no entrega el datagrama a la aplicación, sino que lo descarta. Conviene recordar que, por debajo, el UDP está utilizando el IP, que también es un protocolo no orientado a la conexión. Lascaracterísticas más importantes del UDP son las siguientes:

• No garantiza la fiabilidad; es decir, no ofrece la seguridad de que cada datagrama UDP transmitido llegue a su destino; es un protocolo best-effort: el UDP hace todo lo posible para transferir los datagra mas de su aplicación, pero no garantiza su entrega.

• No preserva la secuencia de la información que le proporciona la aplicación. Como está en modo datagrama y utiliza un protocolo por debajo como el IP, que también está en modo datagrama, la aplicación puede recibir la información desordenada. La aplicación debe estar preparada para que haya datagramas que se pierdan, lleguen con retardo o se hayan desordenado.

El datagrama UDP consta de una cabecera y un cuerpo para encapsular los datos.

La cabecera consta de los elementos siguientes:

• Los campos Puerto de origen y Puerto de destino, que identifican las aplicaciones en los terminales de origen y de destino.Cada puerto tiene 16 bits.

• El campo Longitud indica la longitud, en bytes, del datagramaUDP incluyendo la cabecera UDP (es la diferencia de la longituddel datagrama IP menos la cabecera IP). Como la longitud máxima de un datagrama IP es de 65.535 bytes, con una cabecera estándar de 20 bytes, la longitud máxima de un datagrama UDP es de 65.515 bytes.

• El campo Checksum (16 bits) es opcional y protege tanto la cabecera como los datos UDP (es preciso recordar que el checksumdel datagrama IP sólo cubre la cabecera IP). Cuando el UDP recibe un datagrama y determina que hay errores, lo descarta y nolo entrega a ninguna aplicación.

Las aplicaciones que no requeiren de la funcionalidad del TCP, usan el UDP como protocolo de trasporte. Podemos poner dos ejemplos de estas aplicaciones:

• Aplicaciones en tiempo real. Estas aplicaciones requieren poco retardo (mejor

dicho, poca variabilidad en el retardo), y TCP puede introducir retardos considerables si tiene que esperar, por ejemplo que le llegue un paquete que se ha perdido.

• Aplicaciones interesadas en transmitir información en modo multicast o broadcast (a un grupo de usuarios o a todos los de una red). En este caso, no tiene sentido establecer una conexión como hace el TCP con cada una de las estaciones destino.

El TCP (transmission control protocol)

Como hemos podido observar, el UDP no garantiza la entrega de la información que le proporciona una aplicación. Tampoco reordena la información en caso de que llegue en un orden diferente de aquél en que se ha transmitido. Existen aplicaciones que no pueden tolerar dichas limitaciones. Para superarlas, el nivel de transporte proporciona un protocolo llamado TCP.

El TCP proporciona fiabilidad a la aplicación; es decir, garantiza la entrega de toda la información en el mismo orden en que ha sido transmitida por la aplicación de origen. Para conseguir esta fiabilidad, el TCP proporciona un servicio orientado a la conexión con un control de flujo y errores.

El TCP proporciona fiabilidad

Para proporcionar un servicio fiable a la aplicación, el TCP se basa en los principios siguientes:

1) Transmisión libre de error. El TCP debe entregar a la aplicación de destino exactamente la misma información que le entregó la aplicación de origen. De hecho, se trata de una entrega “casi libre” de errores.

2) Garantía de entrega de la información. El TCP garantiza que toda la información transmitida por la aplicación de origen se entregue a la aplicación de destino. Si no es posible, el TCP debe avisar a la aplicación.

3) Garantía de mantenimiento de la secuencia de transmisión.El TCP garantiza la entrega del flujo de información en el mismo orden en que le fue entregado por la aplicación de origen.

4) Eliminación de duplicados. El TCP garantiza que sólo entregará una copia de la información transmitida a la aplicación de destino. En caso de que reciba copias a causa del funcionamiento de la red o de los protocolos que se implementan por debajo del nivel de transporte, el TCP las eliminará.

La fiabilidad de la transmisión que proporciona TCP se consigue gracias a las siguientes estrategias:

• El TCP está orientado a la conexión: tiene una fase de establecimiento de la conexión, una de transmisión de datos y una de desconexión.

• El TCP utiliza el concepto buffered transfer: cuando se transfiere información, el TCP divide los flujos de datos (byte stream) que le pasa la aplicación en

segmentos del tamaño que le convenga. El TCP decide el tamaño de los segmentos tanto si la aplicación genera un byte de información, como si genera flujos de grandes dimensiones. En el primer caso, el TCP puede esperar a que la memoria intermedia se llene más antes de transferir la información, o bien la puede transferir de inmediato(mecanismo push). En caso de que los flujos sean muy grandes,el TCP puede dividir la información en tamaños más pequeños antes de transferirlos.

• El TCP utiliza una conexión full duplex: la transferencia de información es en ambos sentidos. La aplicación ve dos flujos independientes. En caso de que la aplicación cierre uno de los flujos, la conexión pasa a ser half duplex. Ello significa que uno de los extremos (el que no ha cerrado la conexión) puede continuar enviando información por el canal, mientras que el otro extremo (el que ha cerrado la conexión) se limita a reconocer la información. No obstante, no es normal encontrar este caso.Lo más habitual es que, si un extremo cierra la conexión, el otro también la cierre.

Formato del segmento TCP

La unidad de información del protocolo TCP se llama segmento TCP y su formato el siguiente:

El segmento TCP consta de una cabecera y un cuerpo para encapsular datos. La cabecera consta de los campos siguientes:

a) El campo Puerto de origen identifica la aplicación en el terminal de origen.
b) El campo Puerto de destino identifica la aplicación en el terminal de destino.
c) El campo Número de secuencia identifica el primer byte del campo de datos. En el TCP no se numeran segmentos, sino bytes. Por tanto, el número de secuencia identifica el primer byte de los datos que envía el segmento: al principio de la conexión se asigna un número de secuencia inicial (ISN, del inglés initial sequence number), a partir del cual el TCP numera los bytes consecutivamente.

d) El campo Número ACK. El TCP reconoce datos por medio de la técnica de piggybacking. Al activar un bit de la cabecera (el bit
ACK), el TCP tiene en cuenta el número de secuencia ACK que indica al otro extremo TCP el próximo byte que está dispuesto a recibir. Dicho de otra manera, el número ACK menos uno indica el último byte reconocido.

e) El campo Longitud de la cabecera indica la longitud de la cabecera, que puede ser variable. La longitud típica es de 20 bytes; sin embargo, si el TCP utiliza el campo de opciones, puede llegar a una longitud máxima de 60 bytes. De este modo, el TCP sabe dónde empiezan los datos.

f) El campo Reservado, tal como su nombre indica, está reservado y se inicializa con ceros.

g) El campo Control está formado por seis indicadores independientes, cada uno de los cuales señala una función específica del protocolo cuando está

activo (a 1):

URG: indica que hay datos urgentes (y el campo Urgent pointer indica la cantidad de datos urgentes existentes en el segmento).

ACK: cuando este bit está activo, el campo Número ACK indica el byte siguiente que espera recibir la conexión TCP. Si este bit no está activo, el campo Número ACK no tiene ningún significado para el TCP.

PSH: invoca la función push en el protocolo. Esta función dice alreceptor que entregue a la aplicación todos los datos que tenga
disponibles en la memoria intermedia de recepción sin esperar a completarlos con datos adicionales. De este modo, los datos
no esperan en la memoria intermedia receptora hasta completar un segmento de dimensión máxima.

RST: realiza un reset de la conexión.

SYN: se utiliza para iniciar una conexión y también sirve para resincronizar los números de secuencia.

FIN: indica que el transmisor ha acabado la conexión.

h) El campo Ventana indica cuántos bytes componen la ventana de transmisión del protocolo de control de flujo por ventana deslizante. A diferencia de los protocolos del nivel de enlace, en que la ventana era constante y contaba tramas, en el TCP la ventana es variable y cuenta bytes. Con cada segmento transmitido, un extremo TCP advierte al otro extremo de la cantidad de datos que está dispuesto a recibir en cada momento. De este modo, el extremo que recibe un segmento actualiza el tamaño de su ventana de transmisión.

i) El campo Checksum se utiliza para detectar errores.

j) El campo Urgent pointer tiene sentido cuando el bit de control URG está activo. Indica que los datos que envía el origen son urgentes e identifica el último byte de dicho campo. La aplicación es la que indica que estos últimos son urgentes y lo sabe porque el TCP se lo indica en la recepción.

k) El campo Opciones TCP permite añadir campos a la cabecera para realizar las operaciones siguientes:

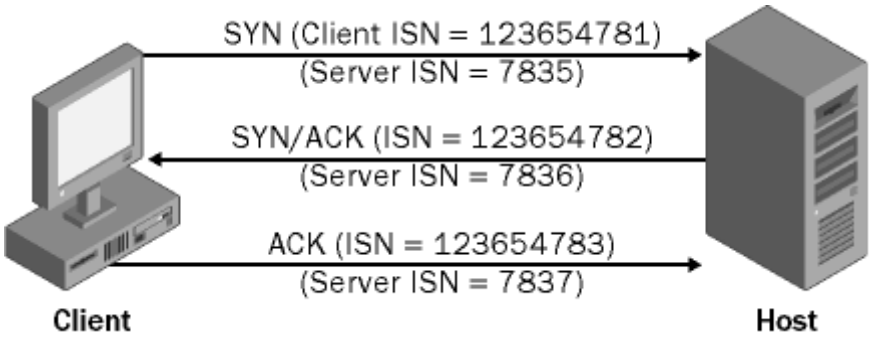
Marcar el tiempo (timestamp) en que se transmitió el segmento y de este modo poder monitorizar los retardos que experimentan los segmentos desde el origen hasta el destino.

Aumentar el tamaño de la ventana.

Indicar el tamaño máximo del segmento (MSS, del inglés maximum segment size) que el origen está preparado para recibir. Por tanto, el receptor no le puede transmitir segmentos por encima de este valor.

Establecimiento de la conexión

Para establecer una conexión, el TCP utiliza el protocolo three-way handshake. Este último necesita tres segmentos TCP para poder establecer la conexión.



Consideremos que el servidor está en un estado de escucha, llamado listen, y que el cliente quiere establecer una conexión con el servidor.

El TCP de la máquina cliente iniciará la petición de conexión TCP,que será contestada por el TCP de la máquina servidor.

Para que el cliente TCP pueda establecer una conexión TCP con el servidor, se siguen los pasos siguientes:

1) Petición de la conexión

El TCP cliente envía un segmento de petición de conexión al servidor.

Dicho segmento, que se conoce como segmento SYN porque tiene activado el bit SYN en el campo Control de la cabecera del segmento
TCP, especifica el número de secuencia inicial TCP del cliente (ISN).El número de secuencia inicial se elige al azar.

2) Confirmación de la conexión

El servidor responde a la petición de establecimiento de la conexión con un segmento SYN que indica el número de secuencia inicial que utilizará.Asimismo, este segmento contiene un reconocimiento (ACK) del segmento SYN del cliente que indica el ISN del cliente más 1 (el número de secuencia inicial del cliente más 1).

3) Reconocimiento de la conexión

El cliente reconoce el segmento SYN (K) del servidor con un reconocimiento que contiene el ISN servidor más 1. En la figura anterior, sería el segmento ACK (K + 1).

Terminación de la conexión

Cuando la transferencia de la información ha finalizado, el TCP dispone de un protocolo de terminación de la conexión para cerrarla.
En una conexión TCP full duplex, en la que los datos fluyen en ambos sentidos,

independientes el uno del otro, cualquier conexión debe cerrarse independientemente.

Los pasos que se siguen son los siguientes:

1) El cliente envía un segmento TCP del tipo FIN con el número de secuencia correspondiente (J). Ello significa que a partir de este momento no habrá más datos que fluyan en este sentido (cliente→ servidor).

2) El servidor envía una confirmación del cierre por medio de un ACK con el número de secuencia recibido más 1 (J + 1).

El TCP servidor indica a su aplicación que el cliente cierra la conexión.
La aplicación servidor indica a su TCP que la cierre a continuación.

3) El servidor envía un segmento TCP del tipo FIN al cliente con el número de secuencia correspondiente (K).

4) El TCP cliente responde automáticamente con un ACK (K + 1).

La capa de aplicación

La capa de aplicación se encuentra en la parte superior de las capas del protocolo TCP/IP. Contiene las aplicaciones de red que permiten la comunicación mediante las capas inferiores.

Por lo tanto, el software en esta capa se comunica mediante uno o dos protocolos de la capa inferior (la capa de transporte), es decir, TCP o UDP.

Existen diferentes tipos de aplicaciones para esta capa, pero la mayoría son servicios de red o aplicaciones brindadas al usuario para proporcionar la interfaz con el sistema operativo. Se pueden clasificar según los servicios que brindan:

servicios de administración de archivos e impresión (transferencia);
servicios de conexión a la red;
servicios de conexión remota;
diversas utilidades de Internet.

Conceptos básicos del HTTP

El servicio WWW permite “navegar’’ por un conjunto de elementos de información interconectados con referencias o enlaces entre sí que pueden contener datos con diferentes tipos de representaciones (texto, imágenes, audio, vídeo). El protocolo que proporciona el acceso a esta información es el HTTP.

El HTTP sigue el modelo general de peticiones y respuestas entre un medio del TCP. En este caso, el puerto por defecto para establecer las conexiones es el asignado oficialmente al servicio WWW, es decir, el 80.

En el HTTP1.0, el cliente establece una conexión con el servidor y le envía un mensaje HTTP con la petición; y, a continuación, el servidor envía al cliente otro mensaje HTTP con la respuesta y cierra la conexión. Si quiere efectuar más peticiones, el cliente debe establecer una nueva conexión para cada una.

En el HTTP/1.1, en cambio, es posible intercambiar diferentes peticiones y respuestas en una misma conexión que se denomina conexión persistente. Éste es el modo de funcionamiento por defecto en el HTTP/1.1.

FTP. (File Transfer Protocol).

Esta herramienta posibilita acceder a documentos y ficheros de un ordenador remoto, y traerlos a nuestro ordenador. Un programa, un texto, una foto,... cualquier cosa que esté en el ordenador con el que hemos conectado, mediante unos comandos, se instala en nuestro ordenador (es lo que los Internautas llaman "bajar" de la red).