**Dirección IP** («IP Address»)

La dirección IP es un conjunto de números que identifica, de manera lógica y jerárquica, a una interfaz en la red (elemento de comunicación/conexión) de un dispositivo (computadora, laptop, teléfono inteligente) que utilice el protocolo (Internet Protocol) o, que corresponde al nivel de red del modelo TCP/IP.1​ La dirección IP no debe confundirse con la dirección MAC, que es un identificador de 48 bits expresado en código hexadecimal, para identificar de forma única la tarjeta de red y no depende del protocolo de conexión utilizado en la red.

La dirección IP puede cambiar a menudo debido a cambios en la red, o porque el dispositivo encargado dentro de la red de asignar las direcciones IP, decida asignar otra IP (por ejemplo, con el protocolo DHCP). A esta forma de asignación de dirección IP se le denomina también dirección IP dinámica (normalmente abreviado como IP dinámica). ​Los sitios de Internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen la necesidad de una dirección IP fija (comúnmente, IP fija o IP estática). Esta no cambia con el tiempo. Los servidores de correo, DNS, FTP públicos y servidores de páginas web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se permite su localización en la red.

Los dispositivos se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, para las personas es más fácil recordar un nombre de dominio que los números de la dirección IP. Los servidores de nombres de dominio DNS, "traducen" el nombre de dominio en una dirección IP. Si la dirección IP dinámica cambia, es suficiente actualizar la información en el servidor DNS. El resto de las personas seguirán accediendo al dispositivo por el nombre de dominio.

**Indice**

1 Direcciones IP

1.1 Direcciones privadas

1.2 Máscara de red

1.3 Creación de subredes

1.4 IP dinámica

1.4.1 Ventajas

1.4.2 Desventajas

1.4.3 Asignación de direcciones IP

1.5 IP fija

2 Direcciones IPv6

3 Google y Wikipedia

4 Obtención de la dirección IP

4.1 Microsoft Windows

4.2 GNU/Linux y subsistemas UNIX-like

5 Véase también

6 Referencias

7 Notas

8 Enlaces externos

**1. Direcciones IP**

Las direcciones IPV4 se expresan mediante un número binario de 32 bits permitiendo un espacio de direcciones de hasta 4.294.967.296 (232) direcciones posibles.

Las direcciones IP se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto está comprendido en el intervalo de 0 a 255 [el número binario de 8 bits más alto es 11111111 y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255].

En la expresión de direcciones IPv4 en decimal se separa cada octeto por un carácter único ".". Cada uno de estos octetos puede estar comprendido entre 0 y 255.

Ejemplo de representación de dirección IPv4: 10.128.1.253, 192.168.255.254

En las primeras etapas del desarrollo del Protocolo de Internet, los administradores de Internet interpretaban las direcciones IP en dos partes, los primeros 8 bits para designar la dirección de red y el resto para individualizar la computadora dentro de la red. Este método pronto probó ser inadecuado, cuando se comenzaron a agregar nuevas redes a las ya asignadas. En 1981 el direccionamiento internet fue revisado y se introdujo la arquitectura de clases. (classful network architecture). En esta arquitectura hay tres clases de direcciones IP que una organización puede recibir de parte de la Internet Corporation for Assigned Names and Numbers (ICANN): clase A, clase B y clase C.

En una red de clase A, se asigna el primer octeto para identificar la red, reservando los tres últimos octetos (24 bits) para que sean asignados a los hosts, 8​ de modo que la cantidad máxima de hosts es 2pow24 - 2 (se excluyen la dirección reservada para broadcast (últimos octetos a 1) y de red (últimos octetos a 0)), es decir, 16 777 214 hosts.

En una red de clase B, se asignan los dos primeros octetos para identificar la red, reservando los dos octetos finales (16 bits) para que sean asignados a los hosts,8​ de modo que la cantidad máxima de hosts por cada red es 2pow16 - 2, o 65 534 hosts.

En una red de clase C, se asignan los tres primeros octetos para identificar la red, reservando el octeto final (8 bits) para que sea asignado a los hosts,8​ de modo que la cantidad máxima de hosts por cada red es 2pow8 - 2, o 254 hosts.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Clase** | **Bits iniciales** | **Intervalo (\*)** | **N.º de redes** | **N.º de direcciones por red** | **N.º de hosts por red(‡)** | [**Máscara de red**](https://es.wikipedia.org/wiki/M%C3%A1scara_de_red) | [**Dirección de *broadcast***](https://es.wikipedia.org/wiki/Broadcast_(inform%C3%A1tica)) |
| A | 0 | 0.0.0.0 (\*\*) - 127.255.255.255 (†) | 126 | 16 777 216 | 16 777 214 | 255.0.0.0 | x.255.255.255 |
| B | 10 | 128.0.0.0 - 191.255.255.255 | 16 382 | 65 536 | 65 534 | 255.255.0.0 | x.x.255.255 |
| C | 110 | 192.0.0.0 - 223.255.255.255 | 2 097 150 | 256 | 254 | 255.255.255.0 | x.x.x.255 |
| D (Multicast) | 1110 | 224.0.0.0 - 239.255.255.255 |  |  |  |  |  |
| E (experimental) | 1111 | 240.0.0.0 - 255.255.255.254 |  |  |  |  |  |

(\*) La dirección que tiene los bits de host iguales a 0 sirve para definir la red en la que se ubica. Se denomina dirección de red. La dirección que tiene los bits correspondientes a host iguales a 1,1​ sirve para enviar paquetes a todos los hosts de la red en la que se ubica. Se denomina dirección de broadcast.

(\*\*) La dirección 0.0.0.0 es reservada por la IANA para identificación local.

(†) Las direcciones 127.x.x.x se reservan para designar la propia máquina. Se denomina dirección de bucle local o loopback.

(‡) La primera dirección se reserva para identificar la red (p.ej. 18.0.0.0), mientras que la última dirección se emplea como dirección de difusión o broadcast (p.ej. 18.255.255.255). Ese es el motivo por el que el número máximo de hosts en una red es siempre igual al número de direcciones disponibles en un rango específico menos dos.

El diseño de redes de clases (classful) sirvió durante la expansión de internet, sin embargo este diseño no era escalable y frente a una gran expansión de las redes en la década de los noventa, el sistema de espacio de direcciones de clases fue reemplazado por una arquitectura de redes sin clases Classless Inter-Domain Routing (CIDR)9​ en el año 1993. CIDR está basada en redes de longitud de máscara de subred variable (variable-length subnet masking VLSM), lo que permite asignar redes de longitud de prefijo arbitrario. Permitiendo por tanto una distribución de direcciones más fina y granulada, calculando las direcciones necesarias y "desperdiciando" las mínimas posibles.

**1. 1. Direcciones privadas**

Existen ciertas direcciones en cada clase de dirección IP que no están asignadas y que se denominan direcciones privadas. Las direcciones privadas pueden ser utilizadas por los hosts que usan traducción de dirección de red (NAT) para conectarse a una red pública o por los hosts que no se conectan a Internet. Se reservan tres rangos no superpuestos de direcciones IPv4 para redes privadas.​ En una misma red no pueden existir dos direcciones iguales, pero sí se pueden repetir en dos redes privadas que no tengan conexión directa entre sí o que se conecten a través de un tercero que haga NAT. Las direcciones privadas son:

Esta sección es un extracto de IPv4 - Redes privadas

De los aproximadamente cuatro mil millones de direcciones definidas en IPv4, cerca de 18 millones de direcciones en tres rangos están reservadas para su uso en redes privadas. Las direcciones de paquetes en estos rangos no son enrutables en la Internet pública; son ignorados por todos los enrutadores públicos. Por lo tanto, los hosts privados no pueden comunicarse directamente con las redes públicas y requieren la traducción de direcciones de red en una puerta de enlace de enrutamiento para este propósito.

Rangos de red IPv4 reservados para redes privadas

Nombre Bloque CIDR Rango de direcciones Nº de direcciones Clase

bloque de 24-bit 10.0.0.0/8 10.0.0.0 – 10.255.255.255 16 777 216 Clase A.

bloque de 20-bit 172.16.0.0/12 172.16.0.0 – 172.31.255.255 1 048 576 Rango contiguo de 16 bloques de clase B.

bloque de 16-bit 192.168.0.0/16 192.168.0.0 – 192.168.255.255 65 536 Rango contiguo de 256 bloques de clase C.

Dado que dos redes privadas, por ejemplo, dos sucursales, no pueden interoperar directamente a través de la Internet pública, las dos redes deben conectarse a través de Internet a través de una red privada virtual (VPN) o un túnel IP, que encapsula los paquetes, incluidos sus encabezados que contienen el Direcciones privadas, en una capa de protocolo durante la transmisión a través de la red pública. Además, los paquetes encapsulados se pueden cifrar para que la transmisión a través de redes públicas asegure los datos.

**1.2 Máscara de red**

La máscara de red permite distinguir dentro de la dirección IP, los bits que identifican a la red y los bits que identifican al host. En una dirección IP versión 4, de los 32 bits que se tienen en total, se definen por defecto para una dirección clase A, que los primeros ocho (8) bits son para la red y los restantes 24 para host, en una dirección de clase B, los primeros 16 bits son la parte de red y la de host son los siguientes 16, y para una dirección de clase C, los primeros 24 bits son la parte de red y los ocho (8) restantes son la parte de host. Por ejemplo, de la dirección de clase A 10.2.1.2 sabemos que pertenece a la red 10.0.0.0 y el anfitrión o host al que se refiere es el 2.1.2 dentro de la misma.

La máscara se forma poniendo en 1 los bits que identifican la red y en 0 los bits que identifican al host.​ De esta forma una dirección de clase A tendrá una máscara por defecto de 255.0.0.0, una de clase B 255.255.0.0 y una de clase C 255.255.255.0 :los dispositivos de red realizan un AND entre la dirección IP y la máscara de red para obtener la dirección de red a la que pertenece el host identificado por la dirección IP dada. Por ejemplo:

Dirección IP: 196.5.4.44

Máscara de red (por defecto): 255.255.255.0

AND (en binario):

11000100.00000101.00000100.00101100 (196.5.4.44) Dirección ip

11111111.11111111.11111111.00000000 (255.255.255.0) Máscara de red

11000100.00000101.00000100.00000000 (196.5.4.0) Resultado del AND

Esta información la requiere conocer un router ya que necesita saber cuál es la red a la que pertenece la dirección IP del datagrama destino para poder consultar la tabla de encaminamiento y poder enviar el datagrama por la interfaz de salida. La máscara también puede ser representada de la siguiente forma 10.2.1.2/8 donde el /8 indica que los 8 bits más significativos de máscara que están destinados a redes o número de bits en 1, es decir /8 = 255.0.0.0. Análogamente (/16 = 255.255.0.0) y (/24 = 255.255.255.0).

Las máscaras de red por defecto se refieren a las que no contienen subredes, pero cuando estas se crean, las máscaras por defecto cambian, dependiendo de cuántos bits se tomen para crear las subredes.

**1. 3. Creación de subredes**

El espacio de direcciones de una red puede ser subdividido a su vez creando subredes autónomas separadas. Un ejemplo de uso es cuando necesitamos agrupar todos los empleados pertenecientes a un departamento de una empresa. En este caso crearíamos una subred que englobara las direcciones IP de estos. Para conseguirlo hay que reservar bits del campo host para identificar la subred estableciendo a uno los bits de red-subred en la máscara. Por ejemplo la dirección 173.17.1.1 con máscara 255.255.255.0 nos indica que los dos primeros octetos identifican la red (por ser una dirección de clase B), el tercer octeto identifica la subred (a 1 los bits en la máscara) y el cuarto identifica el host (a 0 los bits correspondientes dentro de la máscara). Hay dos direcciones de cada subred que quedan reservadas: aquella que identifica la subred (campo host a 0) y la dirección para realizar broadcast en la subred (todos los bits del campo host en 1).

Las redes se pueden dividir en redes más pequeñas para un mejor aprovechamiento de las direcciones IP que se tienen disponibles para los hosts, ya que estas a veces se desperdician cuando se crean subredes con una sola máscara de subred.

La división en subredes le permite al administrador de red contener los broadcast que se generan dentro de una LAN, lo que redunda en un mejor desempeño del ancho de banda.

Para comenzar la creación de subredes, se comienza pidiendo “prestados” bits a la parte de host de una dirección dada, dependiendo de la cantidad de subredes que se deseen crear, así como del número de hosts necesarios en cada subred.

**1. 4. IP dinámica**

Una dirección IP dinámica es una IP asignada al usuario, mediante un servidor DHCP (Dynamic Host Configuration Protocol). La IP que se obtiene tiene una duración máxima determinada. El servidor DHCP provee parámetros de configuración específicos para cada cliente que desee participar en la red IP. Entre estos parámetros se encuentra la dirección IP del cliente.

DHCP apareció como protocolo estándar en octubre de 1993. El estándar RFC 2131 especifica la última definición de DHCP (marzo de 1997). DHCP sustituye al protocolo BOOTP, que es más antiguo. Debido a la compatibilidad retroactiva de DHCP, muy pocas redes continúan usando BOOTP puro.

Las IP dinámicas son las que actualmente ofrecen la mayoría de operadores. El servidor del servicio DHCP puede ser configurado para que renueve las direcciones asignadas cada tiempo determinado.

**1. 4. 1. Ventajas**

Reduce los costos de operación a los proveedores de servicios de Internet (ISP).

Reduce la cantidad de IP asignadas (de forma fija) inactivas.

El usuario puede reiniciar el modem o router para que le sea asignada otra IP y así evitar las restricciones que muchas webs ponen a sus servicios gratuitos de descarga o visionado multimedia en línea.

**1. 4. 2. Desventajas**

Obliga a depender de servicios que redirigen un host a una IP.

**1. 4. 3. Asignación de direcciones IP**

Dependiendo de la implementación concreta, el servidor DHCP tiene tres métodos para asignar las direcciones IP:

* manualmente, cuando el servidor tiene a su disposición una tabla que empareja direcciones MAC con direcciones IP, creada manualmente por el administrador de la red. Solo clientes con una dirección MAC válida recibirán una dirección IP del servidor.
* automáticamente, donde el servidor DHCP asigna por un tiempo preestablecido ya por el administrador una dirección IP libre, tomada de un intervalo prefijado también por el administrador, a cualquier cliente que solicite una.
* dinámicamente, el único método que permite la reutilización de direcciones IP. El administrador de la red asigna un intervalo de direcciones IP para el DHCP y cada ordenador cliente de la LAN tiene su software de comunicación TCP/IP configurado para solicitar una dirección IP del servidor DHCP cuando su tarjeta de interfaz de red se inicie. El proceso es transparente para el usuario y tiene un periodo de validez limitado.

**1. 5. IP fija**

Una dirección IP fija o IP estática es una dirección IP asignada por el usuario de manera manual, o por el servidor de la red,​ con base en la dirección MAC del cliente. Muchas personas asocian IP fija con IP pública e IP dinámica con IP privada.

Una IP puede ser privada ya sea dinámica o fija como puede ser IP pública dinámica o fija.

Una IP pública se utiliza generalmente para montar servidores en internet y necesariamente se desea que la IP no cambie. Por eso la IP pública se la configura, habitualmente, de manera fija y no dinámica.

En el caso de la IP privada es, generalmente, dinámica y está asignada por un servidor DHCP, pero en algunos casos se configura IP privada fija para poder controlar el acceso a internet o a la red local, otorgando ciertos privilegios dependiendo del número de IP que tenemos. Si esta cambiara (si se asignase de manera fuera dinámica) sería más complicado controlar estos privilegios (pero no imposible).

**2. Direcciones IPv6**

La función de la dirección IPv6 es exactamente la misma que la de su predecesor IPv4, pero dentro del protocolo IPv6. Está compuesta por 128 bits y se expresa en una notación hexadecimal de 32 dígitos. IPv6 permite actualmente que cada persona en la Tierra tenga asignados varios millones de IP, ya que puede implementarse con 2128 (3.4×1038 hosts direccionables). La ventaja con respecto a la dirección IPv4 es obvia en cuanto a su capacidad de direccionamiento.

Su representación suele ser hexadecimal y para la separación de cada par de octetos se emplea el símbolo ":". Un bloque abarca desde 0000 hasta FFFF. Algunas reglas de notación acerca de la representación de direcciones IPv6 son:

Los ceros iniciales se pueden obviar.

Ejemplo: 2001:0123:0004:00ab:0cde:3403:0001:0063 -> 2001:123:4:ab:cde:3403:1:63

Los bloques contiguos de ceros se pueden comprimir empleando "::". Esta operación solo se puede hacer una vez.

Ejemplo: 2001:0:0:0:0:0:0:4 -> 2001::4.

Ejemplo no válido: 2001:0:2001::2:0:0:1 o 2001:0:0:0:2::1.

**3. Google y Wikipedia**

Cuando se realiza una búsqueda en Google, se registra la dirección IP del dispositivo con el que se está buscando (PC, laptop, tableta, teléfono inteligente, etc) de esta manera Google sabe dónde enviar la respuesta.

Cuando se realiza un cambio en Wikipedia, se registra la dirección IP en el historial del artículo.

**4. Obtención de la dirección IP**

Dependiendo del sistema operativo en el que usted se encuentre, puede saber fácilmente cuál es su dirección IP:

**4. 1. Microsoft Windows**

En Windows, se puede averiguar la IP del equipo en cada interfaz de red con el siguiente comando:

C:\Windows\system32\ipconfig

El cual informará de la dirección IP**, de la máscara de** red usada y de la dirección de la puerta de enlace de cada interfaz de red conectada.

**4. 2. GNU/Linux y subsistemas UNIX-like**

En GNU/Linux y demás subsistemas UNIX y UNIX-like, se tienen dos comandos:

**IFCONFIG**

Nota: "ifconfig" puede no estar disponible en nuevas versiones de algunos subsistemas UNIX, está siendo sustituido por "ip"

user@host:~$ ifconfig [interfaz]

**IP**

user@host:~$ ip address **show dev [in**terfaz]

O también su ver**sión abrevia**da:

user@host:~$ ip a show dev [interfaz]

**Máscara de red**

La máscara de red es una combinación de bits que sirve para delimitar el ámbito de una red de ordenadores. Su función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

Índice

1 Metodología

2 Ejemplos

3 Tabla de máscaras de red

4 Clases de máscaras en subredes

5 Véase también

6 Referencias

7 Enlaces externos

**1. Metodología**

Mediante la máscara de red, un sistema (ordenador, puerta de enlace, enrutador, etc.) podrá saber si debe enviar un paquete dentro o fuera de la subred en la que está conectado. Por ejemplo, si el enrutador tiene la dirección IP 192.168.1.1 y máscara de red 255.255.255.0, entiende que todo lo que se envía a una dirección IP con formato 192.168.1.X, se envía hacia la red local, mientras que direcciones con distinto formato de dirección IP serán enviadas hacia afuera (Internet, otra red local mayor, entre otros).

**2. Ejemplos**

Suponiendo que tenemos un rango de direcciones IP desde 10.0.0.0 hasta 10.255.255.255, y si todas ellas forman parte de la misma red, su máscara de red sería 255.0.0.0, que también se puede expresar como 10.0.0.0/8.

Una máscara de red representada en binario son 4 octetos de bits:

11111111.11111111.11111111.11111111

La representación utilizada se define colocando en 1 todos los bits de red (máscara natural) y en el caso de subredes, se coloca en 1 los bits de red y 0 los bits de host usados por las subredes. Así, en esta forma de representación (10.0.0.0/8) el 8 sería la cantidad de bits puestos a 1 que contiene la máscara en binario, comenzando desde la izquierda. Para el ejemplo dado (/8), sería:

11111111.00000000.00000000.00000000

y en su representación en decimal sería 255.0.0.0.

Como se ve en el ejemplo anterior, la fila binaria de la máscara de subred determina que todas las direcciones IP de esa subred incluido el Gateway deben ser igu*ales ha*sta la línea y distintas después de la línea. La dirección IP completa se calcula realizando un AND lógico solo con aquellos bits que indique la máscara de subred (MS). El número total de direcciones IP que tiene esa subred es inversamente proporcional al número de bits encendidos en la máscara de red. Esa subred suele llamarse LAN.

La puerta de enlace puede ser cualquier dirección IP dentro de ese rango (subred) pero algunos adoptan la norma de que cumplan el que (IP & MS)+1 = GW (gateway, puerta de e*nlace).* Algunos controladores de protocolo TCP/IP rechazan todos los paquetes que no cumplen esta norma. La puerta de enlace la utilizan los protocolos de TCP/IP para enviar aquellos paquetes cuyo destino se encuentra fuera del rango de la subred definida por la máscara de red (si el paquete va destinado a algún ordenador cuya dirección IP se encuentre fuera del rango establecido por la máscara de red, utilizarán la puerta de enlace, que generalmente es un enrutador que se encarga de enviarlos a otras redes. De esta manera se optimiza el trabajo que realiza el PC.

A veces llamamos o confundimos router (enrutador) con puerta de enlace: la puerta de enlace es en definitiva la dirección IP del enrutador. Dirección que ha de estar dentro de la subred. La dirección IP del enrutador se programa en el mismo enrutador. La mayoría de los enrutador vienen con una dirección de fábrica, modificable a través de un puerto serie o por red mediante http, telnet u otros protocolos. Esta dirección modificable es la puerta de enlace de la red.

El enrutador generalmente tiene dos direcciones IP, cada una en un rango distinto. Por ejemplo, una en el rango de una subred pequeña de 16 ordenadores y otra en otra subred más grande cuyo gateway o puerta de enlace da acceso a Internet. Solo se ven entre sí los equipos de cada subred o aquellos que tengan enrutadores y puertas de enlace bien definidas para enviar paquetes y recibir respuestas. De este modo se forman y definen las rutas de comunicación entre ordenadores de distintas subredes. Los enrutadores además realizan varias funciones, entre ellas la denominada NAT, que consiste en llevar la cuenta del origen de los paquetes para que cuando lleguen las respuestas sean enviadas al ordenador que procede.

Cuando un enrutador se comunica con un ISP o proveedor de servicios de Internet generalmente se les asigna una dirección pública o externa, la cual no es modificable sino asignada por la empresa suministradora (ISP) de ADSL/RDSI. En resumen, la máscara lo que determina es qué paquetes que circulan por la LAN se aceptan por algún ordenador de la LAN o qué paquetes han de salir fuera de la LAN (por el enrutador).

De esta manera, si se escribe en el navegador una dirección IP: 182.23.112.9, el equipo enviará la petición web, ftp, etc, directamente a la dirección especificada por la puerta de enlace (es decir, el router). Ningún equipo de la subred (LAN) atenderá estos paquetes por no estar dentro de su subred (LAN).

En el ejemplo anterior, la máscara da 6 bits (los que quedan a 0, es decir, 64 posibilidades, no de 1 a 64 sino 64 posibilidades) para programar las direcciones IP y la puerta de enlace de la LAN, es decir, el último byte para la dirección IP y la puerta de enlace, en nuestro ejemplo debería tomarse entre 11000000 y 11111111, es decir, entre 192 y 255. Lo normal es darle a la puerta de enlace (router) la dirección más baja, indicando que es el primer equipo que se instala en la LAN.

8bit x 4 octetos = 32 bit. (11111111.11111111.11111111.11111111 = 255.255.255.255)

8bit x 3 octetos = 24 bit. (11111111.11111111.11111111.00000000 = 255.255.255.0)

8bit x 2 octetos = 16 bit. (11111111.11111111.00000000.00000000 = 255.255.0.0)

8bit x 1 octetos = 8 bit. (11111111.00000000.00000000.00000000 = 255.0.0.0)

En el ejemplo 10.0.0.0/8, según lo explicado anteriormente, indicaría que la máscara de red es 255.0.0.0

Las máscaras de redes se utilizan como validación de direcciones realizando una operación AND lógica entre la dirección IP y la máscara para validar al equipo, lo cual permite realizar una verificación de la dirección de la Red y con un XOR y la máscara negada se obtiene la dirección del broadcasting.

**1. 3. Tabla de máscaras de red**

RED

Binario Decimal CIDR N.º hosts Clase

11111111.11111111.11111111.11111111 255.255.255.255 /32

11111111.11111111.11111111.11111110 255.255.255.254 /31

11111111.11111111.11111111.11111100 255.255.255.252 /30 2

11111111.11111111.11111111.11111000 255.255.255.248 /29 6

11111111.11111111.11111111.11110000 255.255.255.240 /28 14

11111111.11111111.11111111.11100000 255.255.255.224 /27 30

11111111.11111111.11111111.11000000 255.255.255.192 /26 62

11111111.11111111.11111111.10000000 255.255.255.128 /25 126

11111111.11111111.11111111.00000000 255.255.255.0 /24 254 C

11111111.11111111.11111110.00000000 255.255.254.0 /23 510

11111111.11111111.11111100.00000000 255.255.252.0 /22 1022

11111111.11111111.11111000.00000000 255.255.248.0 /21 2046

11111111.11111111.11110000.00000000 255.255.240.0 /20 4094

11111111.11111111.11100000.00000000 255.255.224.0 /19 8190

11111111.11111111.11000000.00000000 255.255.192.0 /18 16382

11111111.11111111.10000000.00000000 255.255.128.0 /17 32766

11111111.11111111.00000000.00000000 255.255.0.0 /16 65534 B

11111111.11111110.00000000.00000000 255.254.0.0 /15 131070

11111111.11111100.00000000.00000000 255.252.0.0 /14 262142

11111111.11111000.00000000.00000000 255.248.0.0 /13 524286

11111111.11110000.00000000.00000000 255.240.0.0 /12 1048574

11111111.11100000.00000000.00000000 255.224.0.0 /11 2097150

11111111.11000000.00000000.00000000 255.192.0.0 /10 4194302

11111111.10000000.00000000.00000000 255.128.0.0 /9 8388606

11111111.00000000.00000000.00000000 255.0.0.0 /8 16777214 A

11111110.00000000.00000000.00000000 254.0.0.0 /7 33554430

11111100.00000000.00000000.00000000 252.0.0.0 /6 67108862

11111000.00000000.00000000.00000000 248.0.0.0 /5 134217726

11110000.00000000.00000000.00000000 240.0.0.0 /4 268435454

11100000.00000000.00000000.00000000 224.0.0.0 /3 536870910

11000000.00000000.00000000.00000000 192.0.0.0 /2 1073741822

10000000.00000000.00000000.00000000 128.0.0.0 /1 2147483646

00000000.00000000.00000000.00000000 0. /0 4294967294

El número de hosts se determina como el número de IP's posibles menos dos, en cada subred hay una IP con todos los bits a ceros en la parte del host reservada para nombrar la subred y otra con todos los bits a uno reservada para la dirección de Broadcast.

Hay ciertos programas (p.e. Ethereal) que programan la tarjeta en un modo llamado 'promiscuo' en el que se le dice a la tarjeta de red que no filtre los paquetes según la norma explicada, aceptando todos los paquetes para poder hacer un análisis del tráfico que circula por la subred y poder ser escuchado por el PC.

Las máscaras 255.0.0.0 (clase A), 255.255.0.0 (clase B) y 255.255.255.0 (clase C) suelen ser suficientes para la mayoría de las redes privadas. Sin embargo, las redes más pequeñas que podemos formar con estas máscaras son de 254 hosts y para el caso de direcciones públicas, su contratación tiene un coste alto. Por esta razón suele ser habitual dividir las redes públicas de clase C en subredes más pequeñas. A continuación se muestran las posibles divisiones de una red de clase C. La división de una red en subredes se conoce como subne*tting.*

**1. 4. Clases de máscaras en subredes**

Clase Bits IP Subred IP Broadcast Máscara en decimal CIDR

A 0000 0.0.0.0 127.255.255.255 255.0.0.0 /8

B 1000 128.0.0.0 191.255.255.255 255.255.0.0 /16

C 1100 192.0.0.0 223.255.255.255 255.255.255.0 /24

D 1110 224.0.0.0 239.255.255.255 255.255.255.255 /32

E 1111 240.0.0.0 255.255.255.255 255.255.255.255 /32

**TCP - Protocolo de control de transmisión**

Protocolo de control de transmisión (en inglés Transmission Control Protocol o TCP) es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.

Muchos programas dentro de una red de datos compuesta por redes de computadoras, pueden usar TCP para crear “conexiones” entre sí a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

TCP da soporte a muchas de las aplicaciones más populares de Internet (navegadores, intercambio de ficheros, clientes FTP, etc.) y protocolos de aplicación HTTP, SMTP, SSH y FTP.

Índice

1 Objetivos de TCP

2 Información técnica

2.1 Funciones de TCP

2.2 Características del TCP

2.3 Formato de los segmentos TCP

3 Funcionamiento del protocolo en detalle

3.1 Establecimiento de la conexión (negociación en tres pasos)

3.2 Transferencia de datos

3.3 Números iniciales de secuencia

3.4 Checksums

3.5 Asentimientos y temporizadores

3.6 Ventanas deslizantes

3.7 Fin de la conexión

4 Puertos TCP

5 Desarrollo de TCP

6 Comparativa entre UDP y TCP

7 Bibliotecas de sockets TCP

**1. Objetivos de TCP**

Con el uso de protocolo TCP, las aplicaciones pueden comunicarse en forma segura (gracias al "acuse de recibo" -ACK- del protocolo TCP) independientemente de las capas inferiores. Esto significa que los routers (que funcionan en la capa de red) solo tienen que enviar los datos en forma de segmentos, sin preocuparse con el monitoreo de datos porque esta función la cumple la capa de transporte (o más específicamente el protocolo TCP).

**2. Información técnica**

TCP es usado en gran parte de las comunicaciones de datos. Por ejemplo, gran parte de las comunicaciones que tienen lugar en Internet emplean TCP.

**2. 1. Funciones de TCP**

En la pila de protocolos TCP/IP, TCP es la capa intermedia entre el protocolo de red (IP) y la aplicación. Muchas veces las aplicaciones necesitan que la comunicación a través de la red sea confiable. Para ello se implementa el protocolo TCP que asegura que los datos que emite el cliente sean recibidos por el servidor sin errores y en el mismo orden que fueron emitidos, a pesar de trabajar con los servicios de la capa IP, la cual no es confiable. Es un protocolo orientado a la conexión, ya que el cliente y el servidor deben anunciarse y aceptar la conexión antes de comenzar a transmitir los datos a ese usuario que debe recibirlos.

**2. 2. Características del TCP**

Permite colocar los segmentos nuevamente en orden cuando vienen del protocolo IP.

Permite el monitoreo del flujo de los datos y así evita la saturación de la red.

Permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.

Permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.

Por último, permite comenzar y finalizar la comunicación amablemente.

**2. 3. Formato de los segmentos TCP**

En el nivel de transporte, los paquetes de bits que constituyen las unidades de datos de protocolo TCP se llaman "segmentos". El formato de los segmentos TCP se muestra en el esquema segmento TCP.

**3. Funcionamiento del protocolo en detalle**

Las conexiones TCP se componen de tres etapas:

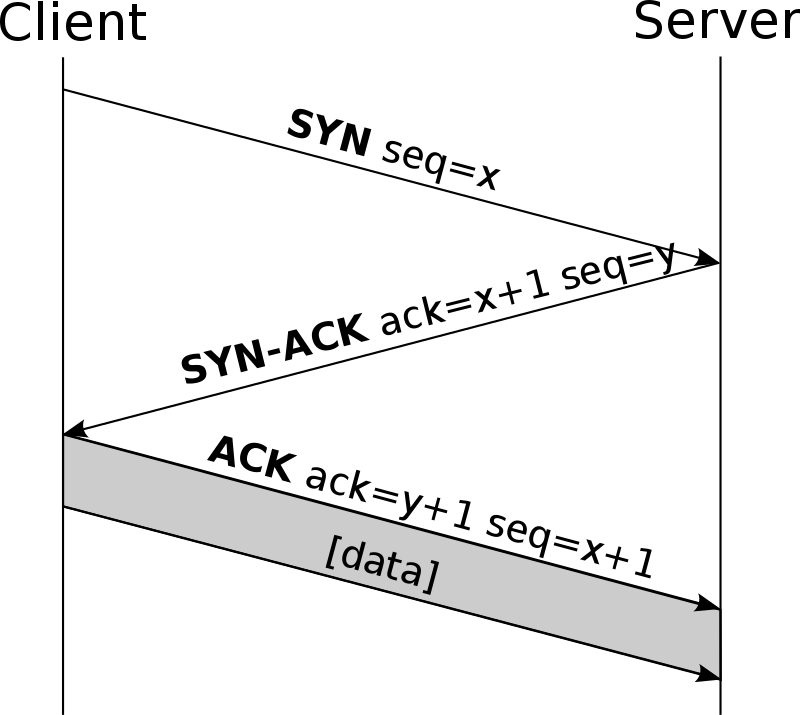
Establecimiento de conexión (3-way handshake).

Transferencia de datos.

Fin de la conexión.

Para establecer la conexión se usa el procedimiento llamado “negociación en tres pasos” (3-way handshake). Para la desconexión se usa una “negociación en cuatro pasos” (4-way handshake). Durante el establecimiento de la conexión, se configuran algunos parámetros tales como el número de secuencia con el fin de asegurar la entrega ordenada de los datos y la robustez de la comunicación.

**3. 1. Establecimiento de la conexión (negociación en tres pasos)**

Negociación en tres pasos o Three-way handshake

Aunque es posible que un par de entidades finales comiencen una conexión entre ellas simultáneamente, normalmente una de ellas abre un socket en un determinado puerto TCP y se queda a la escucha de nuevas conexiones. Es común referirse a esto como apertura pasiva, y determina el lado servidor de una conexión. El lado cliente de una conexión realiza una apertura activa de un puerto enviando un paquete SYN inicial al servidor como parte de la negociación en tres pasos. En el lado del servidor (este receptor también puede ser una PC o alguna estación terminal) se comprueba si el puerto está abierto, es decir, si existe algún proceso escuchando en ese puerto, pues se debe verificar que el dispositivo de destino tenga este servicio activo y esté aceptando peticiones en el número de puerto que el cliente intenta usar para la sesión. En caso de no estarlo, se envía al cliente un paquete de respuesta con el bit RST activado, lo que significa el rechazo del intento de conexión. En caso de que sí se encuentre abierto el puerto, el lado servidor respondería a la petición SYN válida con un paquete SYN/ACK. Finalmente, el cliente debería responderle al servidor con un ACK, completando así la negociación en tres pasos (SYN, SYN/ACK y ACK) y la fase de establecimiento de conexión. Es interesante notar que existe un número de secuencia generado por cada lado, ayudando de este modo a que no se puedan establecer conexiones falseadas (spoofing).

**3. 2 Transferencia de datos**

Durante la etapa de transferencia de datos, una serie de mecanismos claves determinan la fiabilidad y robustez del protocolo. Entre ellos están incluidos el uso del número de secuencia para ordenar los segmentos TCP recibidos y detectar paquetes duplicados, checksums para detectar errores, asentimientos y temporizadores para detectar pérdidas o retrasos y ventanas deslizantes para el control de flujo de datos.

**3. 3. Números iniciales de secuencia**

Durante el establecimiento de conexión TCP, los “números iniciales de secuencia” son intercambiados entre las dos entidades TCP. Estos números de secuencia son usados para identificar los datos dentro del flujo de bytes, y poder identificar (y contar) los bytes de los datos de la aplicación. Siempre hay un par de números de secuencia incluidos en todo segmento TCP, referidos al número de secuencia y al número de asentimiento. Un emisor TCP se refiere a su propio número de secuencia cuando habla de número de secuencia, mientras que con el número de asentimiento se refiere al número de secuencia del receptor. Para mantener la fiabilidad, un receptor asiente los segmentos TCP indicando que ha recibido una parte del flujo continuo de bytes. Una mejora de TCP, llamada asentimiento selectivo (Selective Acknowledgement, SACK) permite a un receptor TCP asentir los datos que se han recibido de tal forma que el remitente solo retransmita los segmentos de datos que faltan.

A través del uso de números de secuencia y asentimiento, TCP puede pasar los segmentos recibidos en el orden correcto dentro del flujo de bytes a la aplicación receptora. Los números de secuencia son de 32 bits (sin signo), que vuelve a cero tras el siguiente byte después del 232-1. Una de las claves para mantener la robustez y la seguridad de las conexiones TCP es la selección del número inicial de secuencia (Initial Sequence Number, ISN).

**3. 4. Checksums**

Un checksum de 16 bits, consistente en el complemento a uno de la suma en complemento a uno del contenido de la cabecera y datos del segmento TCP, agrupados en intervalos de 16 bits, es calculado por el emisor, e incluido en la transmisión del segmento. Se usa la suma en complemento a uno porque el acarreo final de ese método puede ser calculado en cualquier múltiplo de su tamaño (16-bit, 32-bit, 64-bit...) y el resultado, una vez plegado, será el mismo. El receptor TCP recalcula el checksum sobre las cabeceras y datos recibidos. El complemento es usado para que el receptor no tenga que poner a cero el campo del checksum de la cabecera antes de hacer los cálculos, salvando en algún lugar el valor del checksum recibido; en vez de eso, el receptor simplemente calcula la suma en complemento a uno con el checksum incluido, y el resultado debe ser igual a 0. Si es así, se asume que el segmento ha llegado intacto y sin errores.

Hay que fijarse en que el checksum de TCP también cubre los 96 bit de la cabecera que contiene la dirección origen, la dirección destino, el protocolo y el tamaño TCP. Esto proporciona protección contra paquetes mal dirigidos por errores en las direcciones.

El checksum de TCP es una comprobación bastante débil. En niveles de enlace con una alta probabilidad de error de bit quizá requiera una capacidad adicional de corrección/detección de errores de enlace. Si TCP fuese rediseñado hoy, muy probablemente tendría un código de redundancia cíclica (CRC) para control de errores en vez del actual checksum. La debilidad del checksum está parcialmente compensada por el extendido uso de un CRC en el nivel de enlace, bajo TCP e IP, como el usado en el PPP o en Ethernet. Sin embargo, esto no significa que el checksum de 16 bits es redundante: sorprendentemente, inspecciones sobre el tráfico de Internet han mostrado que son comunes los errores de software y hardware[cita requerida] que introducen errores en los paquetes protegidos con un CRC, y que el checksum de 16 bits de TCP detecta la mayoría de estos errores simples.

**3. 5. Asentimientos y temporizadores**

Los asentimientos (ACK o Acknowledgments) de los datos enviados o la falta de ellos, son usados por los emisores para interpretar las condiciones de la red entre el emisor y receptor TCP. Unido a los temporizadores, los emisores y receptores TCP pueden alterar el comportamiento del movimiento de datos. TCP usa una serie de mecanismos para conseguir un alto rendimiento y evitar la congestión de la red (la idea es enviar tan rápido como el receptor pueda recibir). Estos mecanismos incluyen el uso de ventana deslizante, que controla que el transmisor mande información dentro de los límites del búfer del receptor, y algoritmos de control de flujo, tales como el algoritmo de evitación de la congestión (congestion avoidance), el de comienzo lento (slow-start), el de retransmisión rápida, el de recuperación rápida (fast recovery), y otros.

**3. 6. Ventanas deslizantes**

TCP usa control de flujo para evitar que un emisor envíe datos de forma más rápida de la que el receptor puede recibirlos y procesarlos. El control de flujo es un mecanismo esencial en redes en las que se comunican computadoras con distintas velocidades de transferencia. Por ejemplo, si una PC envía datos a un dispositivo móvil que procesa los datos de forma lenta, el dispositivo móvil debe regular el flujo de datos.

TCP usa una ventana deslizante para el control de flujo. En cada segmento TCP, el receptor especifica en el campo receive window la cantidad de bytes que puede almacenar en el búfer para esa conexión. El emisor puede enviar datos hasta esa cantidad. Para poder enviar más datos debe esperar que el receptor le envíe un ACK con un nuevo valor de ventana.

El tamaño de la ventana de recepción TCP es la cantidad de datos recibidos (en bytes) que pueden ser metidos en el búfer de recepción durante la conexión. La entidad emisora puede enviar una cantidad determinada de datos pero antes debe esperar un asentimiento con la actualización del tamaño de ventana por parte del receptor.

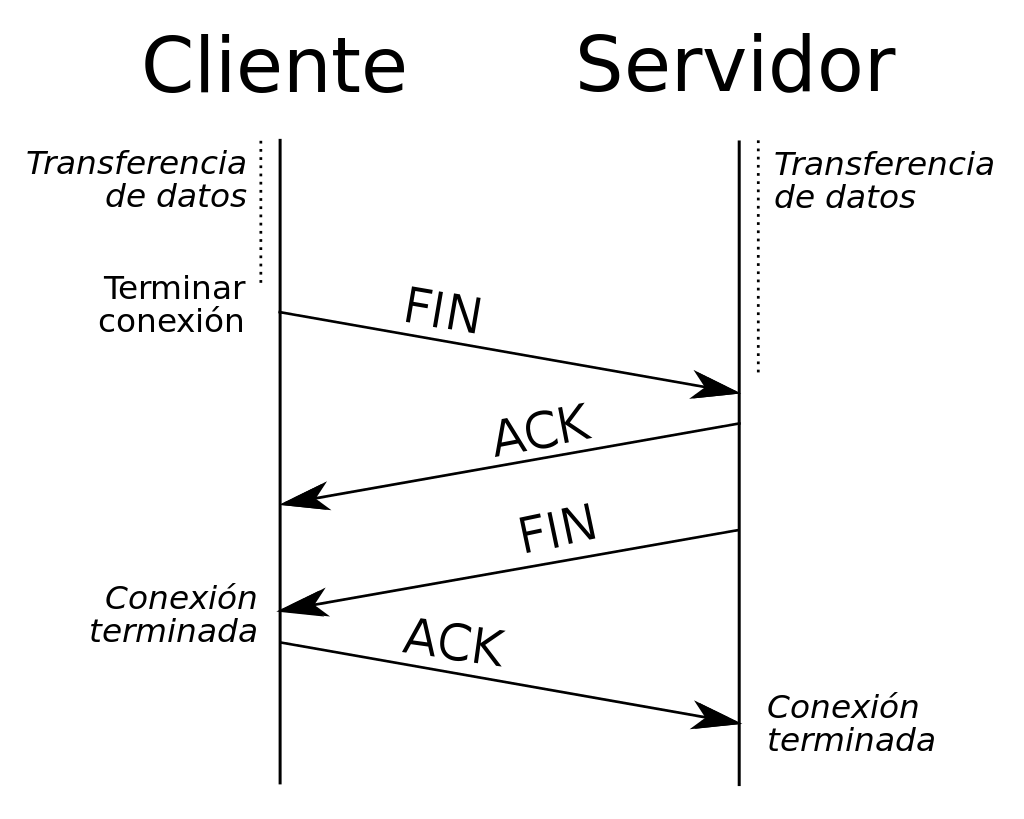
Un ejemplo sería el siguiente: un receptor comienza con un tamaño de ventana X y recibe Y bytes, entonces su tamaño de v**e**ntana será(X - Y) y el transmisor sólo podrá mandar paquetes con un tamaño máximo de datos de (X - Y) bytes. Los siguientes paquetes recibidos seguirán restando tamaño a la ventana de recepción. Esta situación seguirá así hasta que la aplicación receptora recoja los datos del búfer de recepción.

Para una mayor eficiencia en redes de gran ancho de banda, debe ser usado un tamaño de ventana mayor. El campo TCP de tamaño de ventana controla el movimiento de datos y está limitado a 16 bits, es decir, a un tamaño de ventana de 65.535 bytes.

Como el campo de ventana no puede expandirse se usa un factor de escalado. La escala de ventana TCP (TCP window scale) es una opción usada para incrementar el máximo tamaño de ventana desde 65.535 bytes, a 1 Gigabyte.

La opción de escala de ventana TCP es usada solo durante la negociación en tres pasos que constituye el comienzo de la conexión. El valor de la escala representa el número de bits desplazados a la izquierda de los 16 bits que forman el campo del tamaño de ventana. El valor de la escala puede ir desde 0 (sin desplazamiento) hasta 14. Hay que recordar que un número binario desplazado un bit a la izquierda es como multiplicarlo en base decimal por 2.

**3. 7. Fin de la conexión**

Cierre de una conexión según el estándar. 

La fase de finalización de la conexión utiliza una negociación en cuatro pasos (four-way handshake), terminando la conexión desde cada lado independientemente. Sin embargo, es posible realizar la finalización de la conexión en 3 fases; enviando el segmento FIN y el ACK en uno solo.2​ Cuando uno de los dos extremos de la conexión desea parar su "mitad" de conexión transmite un segmento con el flag FIN en 1, que el otro interlocutor asentirá con un ACK. Por tanto, una desconexión típica requiere un par de segmentos FIN y ACK desde cada lado de la conexión.

Una conexión puede estar "medio abierta" en el caso de que uno de los lados la finalice pero el otro no. El lado que ha dado por finalizada la conexión no puede enviar más datos pero la otra parte si podrá.

**4. Puertos TCP**

TCP usa el concepto de número de puerto para identificar a las aplicaciones emisoras y receptoras. Cada lado de la conexión TCP tiene asociado un número de puerto (de 16 bits sin signo, con lo que existen 65536 puertos posibles) asignado por la aplicación emisora o receptora. Los puertos son clasificados en tres categorías:

1. bien conocidos,
2. registrados, y
3. dinámicos/privados.

Los puertos bien conocidos son asignados por la Internet Assigned Numbers Authority (IANA), van del 0 al 1023 y son usados normalmente por el sistema o por procesos con privilegios. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y se quedan a la escucha de conexiones. Algunos ejemplos son: FTP (21), SSH (22), Telnet (23), SMTP (25) y HTTP (80).

Los puertos registrados son normalmente empleados por las aplicaciones de usuario de forma temporal cuando conectan con los servidores, pero también pueden representar servicios que hayan sido registrados por un tercero (rango de puertos registrados: 1024 al 49151).

Los puertos dinámicos/privados también pueden ser usados por las aplicaciones de usuario, pero este caso es menos común. Los puertos dinámicos/privados no tienen significado fuera de la conexión TCP en la que fueron usados (rango de puertos dinámicos/privados: 49152 al 65535, recordemos que el rango total de 2 elevado a la potencia 16, cubre 65536 números, del 0 al 65535).

**5. Desarrollo de TCP**

TCP es un protocolo muy desarrollado y complejo. Sin embargo, mientras mejoras significativas han sido propuestas y llevadas a cabo a lo largo de los años, ha conservado las operaciones más básicas sin cambios desde el RFC 793, publicado en 1981. El documento RFC 1122 (Host Requirements for Internet Hosts), especifica el número de requisitos de una implementación del protocolo TCP.4​ El RFC 2581 (Control de Congestión TCP) es uno de los más importantes documentos relativos a TCP de los últimos años, describe nuevos algoritmos para evitar la congestión excesiva.5​

En 2001, el RFC 3168 fue escrito para describir la Notificación de Congestión Explícita (ECN), una forma de eludir la congestión con mecanismos de señalización. En los comienzos del siglo XXI, TCP es usado en el 95% de todos los paquetes que circulan por Internet.[cita requerida] Entre las aplicaciones más comunes que usan TCP están HTTP/HTTPS (World Wide Web), SMTP/POP3/IMAP (correo electrónico) y FTP (transferencia de ficheros). Su amplia extensión ha sido la prueba para los desarrolladores originales de que su creación estaba excepcionalmente bien hecha.

Recientemente, un nuevo algoritmo de control de congestión fue desarrollado y nombrado como FAST TCP (Fast Active queue management Scalable Transmission Control Protocol) por los científicos de California Institute of Technology (Caltech). Es similar a TCP Vegas en cuanto a que ambos detectan la congestión a partir de los retrasos en las colas que sufren los paquetes al ser enviados a su destino. Todavía hay un debate abierto sobre si este es un síntoma apropiado para el control de la congestión.[cita requerida]

**6. Comparativa entre UDP y TCP**

UDP: proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade la información necesaria para la comunicación extremo a extremo al paquete que envía al nivel inferior. Lo utilizan aplicaciones como NFS (Network File System) y RCP (comando para copiar ficheros entre computadores remotos), pero sobre todo se emplea en tareas de control y en la transmisión de audio y vídeo a través de una red. No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.[cita requerida]

TCP: es el protocolo que proporciona un transporte fiable de flujo de bits entre aplicaciones. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, pérdida de paquetes, orden en el que llegan los paquetes, duplicados de paquetes...) que gestiona el propio protocolo. Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes a enviar. Debido a que los paquetes para enviar tienen un tamaño máximo, cuanta más información añada el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete (el segmento TCP tiene una sobrecarga de 20 bytes en cada segmento, mientras que UDP solo añade 8 bytes). Por eso, cuando es más importante la velocidad que la fiabilidad, se utiliza UDP. En cambio, “TCP asegura la recepción en destino de la información para transmitir”.

**7. Bibliotecas de sockets TCP**

Se listan algunas de las bibliotecas de comunicaciones existentes, que utilizan los protocolos TCP y UDP para distintos sistemas operativos.

SolarSockets es una biblioteca para C++ Multiplataforma y Multithread, gratuita para proyectos libres.6​ Fácil de usar y con varios ejemplos.7​

SDL NET proporciona funciones y tipos de dato multiplataforma para programar aplicaciones que trabajen con redes.8​

C++ Sockets Library es una biblioteca de clases en C++ bajo licencia GPL que 'mapea' el berkeley sockets C API, y funciona tanto en algunos sistemas Unix como en Win32.9​

GNU Common C++ es una biblioteca de propósito general que incluye funciones de red.10​

HackNet es una biblioteca de comunicaciones para crear juegos multijugador.11​

DirectPlay simplifica el acceso de las aplicaciones a los servicios de comunicación. Es una componente de DirectX.

UDP - Protocolo de datagramas de usuario

El protocolo de datagramas de usuario (en inglés: User Datagram Protocol o UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas (Encapsulado de capa 4 o de Transporte del Modelo OSI). Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción. Su uso principal es para protocolos como DHCP, BOOTP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión/desconexión son mayores, o no son rentables con respecto a la información transmitida, así como para la transmisión de audio y vídeo en tiempo real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tiene en estos casos.

Índice

1 Descripción técnica

2 Puertos

3 Uso en aplicaciones

4 Principales características

5 Comparativa entre UDP y TCP (Transmission Control Protocol)

6 Transmisión de vídeo y voz

7 Véase también

8 Enlaces externos

**1. Descripción técnica**

User Datagram Protocol (UDP) es un protocolo mínimo de nivel de transporte orientado a mensajes documentado en el RFC 768 de la IETF.

En la familia de protocolos de Internet UDP proporciona una sencilla interfaz entre la capa de red y la capa de aplicación. UDP no otorga garantías para la entrega de sus mensajes (por lo que realmente no se debería encontrar en la capa 4) y el origen UDP no retiene estados de los mensajes UDP que han sido enviados a la red. UDP solo añade multiplexado de aplicación y suma de verificación de la cabecera y la carga útil. Cualquier tipo de garantías para la transmisión de la información deben ser implementadas en capas superiores.

Bits 0 – 15 16 - 31

0 Puerto origen Puerto destino

32 Longitud del Mensaje Suma de verificación

64 Datos

La cabecera UDP consta de 4 campos de los cuales 2 son opcionales (con fondo rojo en la tabla). Los campos de los puertos origen y destino son campos de 16 bits que identifican el proceso de emisión y recepción. Ya que UDP carece de un servidor de estado y el origen UDP no solicita respuestas, el puerto origen es opcional. En caso de no ser utilizado, el puerto origen debe ser puesto a cero. A los campos del puerto destino le sigue un campo obligatorio que indica el tamaño en bytes del datagrama UDP incluidos los datos. El valor mínimo es de 8 bytes. El campo de la cabecera restante es una suma de comprobación de 16 bits que abarca una pseudo-cabecera IP (con las IP origen y destino, el protocolo y la longitud del paquete UDP), la cabecera UDP, los datos y 0's hasta completar un múltiplo de 16. El checksum también es opcional en IPv4, aunque generalmente se utiliza en la práctica (en IPv6 su uso es obligatorio). A continuación se muestra los campos para el cálculo del checksum en IPv4, marcada en rojo la pseudo-cabecera IP.

Bits 0 – 7 8 – 15 16 – 23 24 – 31

0 Dirección Origen

32 Dirección Destino

64 Ceros Protocolo Longitud UDP

96 Puerto Origen Puerto Destino

128 Longitud del Mensaje Suma de verificación

160 Datos

El protocolo UDP se utiliza por ejemplo cuando se necesita transmitir voz o vídeo y resulta más importante transmitir con velocidad que garantizar el hecho de que lleguen absolutamente todos los bytes.

**2. Puertos**

UDP utiliza puertos para permitir la comunicación entre aplicaciones. El campo de puerto tiene una longitud de 16 bits, por lo que el rango de valores válidos va de 0 a 65.535. El puerto 0 está reservado, pero es un valor permitido como puerto origen si el proceso emisor no espera recibir mensajes como respuesta.

Los puertos 1 a 1023 se llaman puertos "bien conocidos" y en sistemas operativos tipo Unix enlazar con uno de estos puertos requiere acceso como superusuario.

Los puertos 1024 a 49.151 son puertos registrados.

Los puertos 49.152 a 65.535 son puertos dinámicos y son utilizados como puertos temporales, sobre todo por los clientes al comunicarse con los servidores.

**3. Uso en aplicaciones**

La mayoría de las aplicaciones claves de Internet utilizan el protocolo UDP, incluyendo: el Sistema de Nombres de Dominio , donde las consultas deben ser rápidas y solo contaran de una sola solicitud, luego de un paquete único de respuesta, el Protocolo de Administración de Red, el Protocolo de Información de Enrutamiento (RIP) y el Protocolo de Configuración dinámica de host.

**4. Principales características**

Las características principales de este protocolo son:

1. Trabaja sin conexión, es decir que no emplea ninguna sincronización entre el origen y el destino.
2. Trabaja con paquetes o datagramas enteros, no con bytes individuales como TCP. Una aplicación que emplea el protocolo UDP intercambia información en forma de bloques de bytes, de forma que por cada bloque de bytes enviado de la capa de aplicación a la capa de transporte, se envía un paquete UDP.
3. No es fiable. No emplea control del flujo ni ordena los paquetes.
4. Su gran ventaja es que provoca poca carga adicional en la red ya que es sencillo y emplea cabeceras muy simples.

**5. Comparativa entre UDP y TCP (Transmission Control Protocol)**

UDP: proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade la información necesaria para la comunicación extremo a extremo al paquete que envía al nivel inferior. Lo utilizan aplicaciones como NFS (Network File System) y RCP (comando para copiar ficheros entre ordenadores remotos), pero sobre todo se emplea en tareas de control y en la transmisión de audio y vídeo a través de una red. No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.

TCP: es el protocolo que proporciona un transporte fiable de flujo de bits entre aplicaciones. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, pérdida de paquetes, orden en el que llegan los paquetes, duplicados de paquetes...) que gestiona el propio protocolo. Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes que enviar. Debido a que los paquetes para enviar tienen un tamaño máximo, cuanta más información añada el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete (el segmento TCP tiene una sobrecarga de 20 bytes en cada segmento, mientras que UDP solo añade 8 bytes). Por eso, cuando es más importante la velocidad que la fiabilidad, se utiliza UDP. En cambio, TCP asegura la recepción en destino de la información para transmitir.

**6. Transmisión de vídeo y voz**

UDP es generalmente el protocolo usado en la transmisión de vídeo y voz a través de una red. Esto es porque no hay tiempo para enviar de nuevo paquetes perdidos cuando se está escuchando a alguien o viendo un vídeo en tiempo real.

Ya que tanto TCP como UDP circulan por la misma red, en muchos casos ocurre que el aumento del tráfico UDP daña el correcto funcionamiento de las aplicaciones TCP. Por defecto, TCP pasa a un segundo lugar para dejar a los datos en tiempo real usar la mayor parte del ancho de banda. El problema es que ambos son importantes para la mayor parte de las aplicaciones, por lo que encontrar el equilibrio entre ambos es crucial.

Todo este tipo de protocolos son usados en telemática.