Suite de Protocolos TCP/IP

En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen:

- Protocolo de Control de Transmisión (TCP)
- Protocolo de Internet (IP)

que fueron dos de los primeros en definirse, y que son los más utilizados de la familia.

La pila TCP/IP

La suite de protocolos TCP/IP son el conjunto de protocolos utilizados para las comunicaciones en internet. Son muchos protocolos pero los referentes, TCP en la capa de transporte e IP en la capa de red, son los que le dan el nombre.

La capa Física posee características propias de cada una de las redes y enlaces al nivel mas bajo. La capa de Enlace esta firmemente vinculada a la capa Física. Ambas rigen la comunicación a nivel local.

¿Cómo logramos la comunicación en la red de redes? . Primero necesitamos identificar cada uno de los host que la integran. Para una identificación que mejor que un nombre… sin embargo, para las máquinas resulta mas sencillo utilizar números. Éstos números, tomados como direcciones, son las famosas direcciones IP (próximas transparencias). La información más importante de la capa de Red.

Luego necesitamos herramientas para poder hacer que la información viaje desde un punto a otro. El ruteo y los protocolos de ruteo hacen que esto último sea posible.

Aún falta solucionar algunas cosas más y una de ellas es poder ubicar los procesos dentro de los host en los extremos. Dichos procesos los asociamos a servicios y estos los identificamos mediante Puertos. Los PORTs o puertos constituyen la información mas importante que maneja la capa de Transporte.

Luego, el tipo de servicio que puede ser orientado a conexión (TCP) o no orientado a conexión (UDP).

Y la capa de Aplicación es donde se implementan todos los aplicaciones que son la cara más visible de internet.

Preguntas de Repaso

- ¿Qué es la pila TCP/IP?
- ¿Qué información importante maneja la capa de transporte?
- ¿Qué información importante maneja la capa de red?
- ¿Qué otra capa puede necesitar direcciones?

TCP/IP - Direcciones

IPV4

- Cuatro octetos de 8 bits cada uno (formato de notación de puntos divisorios).
- Por ejemplo: 10.0.123.34

IPV6

- Ocho grupos de cuatro dígitos hexadecimales separados por “:”
- Por ejemplo: 2001:0db8:85a3:0042:1000:8a2e:0370:7334

Algunas direcciones de red se reservan para propósitos especiales.

- 0.0.0.0 comodín(wildcard), ruteo por defecto
- 127.0.0.1 dirección loopback.

Las direcciones IP que vamos a estudiar son las IPV4. Vamos a entender su uso y funcionamiento. Son las mas usadas.

Hay una intención de cambio a IPV6, pero todavía está muy atrasado, sobre todo acá en nuestro país.

No veremos IPV6 ya que conociendo las IPV4 podemos luego entender con mayor facilidad las IPV6.

En los hosts nos vamos a encontrar con un par de datos importantes. La ruta por defecto está relacionada con el router default, gateway default, puerta de enlace, etc. Por ahora solo diremos que es la IP del router que nos permitirá conectarnos con internet.

La red 127.0.0.0 está reservada para el tráfico local IP del host. + Normalmente, la dirección 127.0.0.1 se asignará a una interfaz especial del host, la interfaz loopback, que actúa como un circuito cerrado.

- Cualquier paquete IP enviado a esta interfaz por TCP o UDP le será devuelto a cualquiera de ellos como si simplemente hubiese llegado desde alguna red.

- Esto permite desarrollar y probar software de red aunque no se esté usando una red real.

Máscara de Red

Otra información necesaria para el funcionamiento de las comunicaciones y que funciona en conjunto con las direcciones IP es la máscara. A cada IP asignada le corresponde su máscara que le permite al dispositivo poder determinar la red a la cual está conectado y por ende cuales direcciones no pertenecen a su red.

Los dispositivos de ruteo por lo general están conectados a varias redes y por cada una de ellas tendrá asignada una dirección IP y una máscara. Como son diferentes redes cada IP asignada pertenece a un grupo de direcciones de red diferente. Entonces el router cuando recibe un paquete con una dirección por algunas de las interfaces utiliza información adicional y las máscaras para enviarla por otra interface.

El protocolo IP está diseñado para que los paquetes viajen de origen a destino , pero no garantiza que lleguen ya que podrían perderse, dañarse o ser descartados por congestión por los routers.

Direcciones IPv4

Solo nos vamos a concentrar en las clases A, B y C. Las otras no son caso de estudio para el funcionamiento de internet.

- Clase A comprende redes desde 1 hasta 127 El número de red está contenido en el primer octeto. (8 bits red, 24 bits hosts) La red 127 se reserva para loopback y pruebas internas

- La clase B comprende las redes desde 128 hasta 191 el número de red está en los dos primeros octetos. (16 bits red, 16 bits hosts)

- La clase C va desde 192 hasta 223, con el número de red contenido en los tres primeros octetos. (24 bits red, 8 bits hosts)

Otras: + La clase D va desde 224 hasta 239 y están reservadas para multicast (a menudo usadas por aplicaciones de streaming de media) + La clase E va desde 240 hasta 255 reservadas para experimentación e investigación.

IPv4 - Especiales

Existen direcciones privadas para ser usadas en forma local y desconectada de internet. Veremos mas adelante que son importantes y que con un proxy o representante se pueden utilizar para conectarse a internet.

- La clase A comprende redes desde 10.0.0.0 hasta 10.255.255.255 El número de red está contenido en el primer octeto. (8 bits red, 24 bits hosts)

- La clase B comprende las redes desde 172.16.0.0 hasta 172.31.255.255; el número de red está en los dos primeros octetos. (16 bits red, 16 bits hosts)

- La clase C va desde 192.168.0.0 hasta 192.168.255.255, con el número de red contenido en los tres primeros octetos. (24 bits red, 8 bits hosts)

*Rango IPV6: fc00::/7

IPv4 - Formato

Para identificar las redes con los 3 primeros bits o con los rangos obtenemos la clase y con la máscara (Propia de cada clase) la red particular.

Las redes que empiezan con los primeros bits:

0 Clase A
- (quedan 7 bits para identificar la red y 24 para hosts)

10 Clase B
- (quedan 14 bits para identificar la red y 16 para hosts)

11 Clase C
- (quedan 21 bits para identificar la red y 8 para hosts)

Otras:
- ***1110*** Clase D Multicast
- ***11110*** Clase E Uso Futuro

IPv4 - Máscaras

Las máscaras de red determinan la cantidad de hosts en una red. Esto es, determinan un rango de direcciones IP que pertenecen a una red.

Ejemplos:

- La red 192.168.0.0 con máscara 255.255.255.0/24 comprende las direcciones IP desde 192.168.0.0 hasta 192.168.0.255.
- La red 10.12.124.0 con máscara de red 255.255.254.0 (/23) comprende las direcciones IP desde 10.12.124.0 hasta 10.12.12.255
- La red 186.137.0.0 con máscara 255.255.0.0 (/16) comprende las direcciones IP desde 186.137.0.0 hasta 186.137.255.25
- La primera y última dirección dentro de cada red tienen un uso especial:
- La primera representa a toda la red
- La última representa la dirección “broadcast”(difusión).

IPv4 - Funcionamiento de las Máscaras

Tenemos la dirección de un host 16.0.0.1 y la máscara 255.0.0.0 con su representación binaria.

Se trata de una dirección clase A ya que comienza con el bit 0 o está en el rango [1, 127].

Le aplicamos la máscara y obtenemos la dirección 16.0.0.0. Cuando decimos “le aplicamos” significa que realizamos una operación matemática entre la dirección IP y la Máscara.

Elemento	**.decimal**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**
Host	**16.0.0.1**	0001 0000	0000 0000	0000 0000	0000 0001
¿?					
Mask	**255.0.0.0**	1111 1111	0000 0000	0000 0000	0000 0000
=					
Red	**16.0.0.0**	0001 0000	0000 0000	0000 0000	0000 0000

IPv4 - Máscaras & AND

Operación AND

bit	op	bit	=	bit
0	&	0	=	0
1	&	0	=	0
0	&	1	=	0
1	&	1	=	1

Elemento	.decimal	Byte 1	Byte 2	Byte 3	Byte 4
Host	**16.0.0.1**	0001 0000	0000 0000	0000 0000	0000 0001
&					
Mask	**255.0.0.0**	1111 1111	0000 0000	0000 0000	0000 0000
=					
Red	**16.0.0.0**	0001 0000	0000 0000	0000 0000	0000 0000

Notemos que las máscaras comienzan con bits en 1 a la izquierda y crecen hacia la derecha con 1s hasta finalizar. Nunca se intercalan 0s.

La operación matemática usada es la operación AND, operación booleana que establece los valores de verdad **1** o falso **0**, analizando la combinación. El operador utilizado es el símbolo **&**.

Una forma sencilla es pensar en superponer la máscara sobre la dirección y suponer que los 1s son como huecos que me permiten ver el número que está debajo y los 0s no, y al no ver asumo 0.

Otra forma, la propuesta en la transparencia, es arrastrar cada bit de la IP hacia abajo y hacerlo pasar por la máscara como si esta última fuese una red. Los 1s dejan pasar lo que está, pero los 0s transforman todo a 0s.

Cualquiera de las maneras termina siendo sencilla con un poco de práctica.

IPv4 – Máscaras, ANDs y Decimales

Cuando se trata de las clases puras se puede operar en bloques, es decir, se puede resumir la segunda forma sencilla(apenas vista) directamente al número decimal.

Byte 1	.	Byte 2	.	Byte 3	.	Byte 4
16	.	0	.	0	.	1
&						
255	.	0	.	0	.	1
=						
16	.	0	.	0	.	0

Cantidad de Hosts

Dijimos que la máscara contiene 1s y 0s , que los 1s van de izquierda a derecha y la cantidad depende de la máscara del tipo de clase de dirección. Luego siguen 0s.

A estos 0s se los considera bits libres. La cantidad de direcciones para hosts la podemos calcular como 2 elevado a la cantidad de 0s de la máscara.

Vemos que la clase A tiene pocas redes, pero muchísimos hosts por cada una. Al contrario la clase C.

- Para una red de clase A = 224
- Para una red de clase B = 216
- Para una red de clase C = 28

Cantidad de Hosts - Ejemplos

Ejemplo 1

- Dada 16.0.0.0 :
- Hosts de 16.0.0.1 a 16.255.255.254

Ejemplo 2

- Dada 200.10.10.0 :
- Hosts de 200.10.10.1 a 200.10.10.254

Broadcast

Mencionamos anteriormente que en el rango de direcciones posibles de hosts para una red, la última dirección es especial y no se pueden asignar a un host así como tampoco la dirección de red.

La dirección de host con todos 1s, la última para cada red, es una dirección de difusión (Broadcast).

Si un mensaje trae esta dirección está dirigido a todos los dispositivos de la red,

Ejemplo 1

- dirección clase A = 16.0.0.0
- Broadcast = 16.255.255.255

Ejemplo 2

- dirección clase C = 200.10.10.0
- Broadcast = 200.10.10.255

Preguntas de Repaso

- ¿Qué tipo de IP conoce? ¿Y clases?
- ¿Para qué se usan las IP privadas?
- ¿Qué son y para qué se usan las máscaras?
- ¿Cómo se relacionan las máscaras y las direcciones IP?
- ¿Cómo se calcula la cantidad de redes de cada clase? ¿Y la cantidad de hosts para cada red?
- ¿Cómo se calcula la dirección de broadcast de una red?

Nombre del Host

Complicado conocer de antemano las IP de los hosts o servidores con los que nos queremos conectar.

Para las máquinas números, para los humanos nombres.

Configuración: Nombre de la máquina

Durante el arranque del SO, se establece el nombre de la máquina al ejecutarse el comando hostname:

hostname nombre

El archivo /etc/hostname contiene el nombre del equipo que adopta el SO(Sistema Operativo) al iniciar el equipo.

Ejecutar man hostname para mas detalles.

Los nombres de las máquinas pueden ser con cualificación completa, o relativos al dominio local.

web.dominio.local ftp.dominio.local Son nombres de dominio completamente cualificados (FQDN *)

web o ftp → Son nombres locales de una máquina, el primer componente del nombre.

(*) Un FQDN (sigla en inglés de fully qualified domain name) es un nombre que incluye el nombre de la computadora y el nombre de dominio asociado a ese equipo. Por ejemplo, dada la computadora llamada «serv1» y el nombre de dominio «bar.com», el FQDN será «serv1.bar.com»; a su vez, un FQDN asociado a serv1 podría ser «post.serv1.bar.com».

Resolución de Nombres

Cuando nos queremos conectar con otra máquina usando internet casi siempre utilizamos un nombre que debe ser traducido a una dirección IP. Todos los sistemas operativos nos permiten configurar, mediante la modificación de ciertos archivos, cómo se debe realizar esta traducción.

Es sistema que se utiliza es el DNS que resulta ser una aplicación que usan otras aplicaciones para funcionar. Una consulta al DNS involucra una serie de pasos que pueden provocar cierta demora. Entonces normalmente se deja como última opción.

Primero buscamos en una base de datos local (un archivo) dentro del host, luego podemos hacerlo a nivel de la red local y por último utilizando el DNS.

Los equipos más conocidos o los más accedidos los colocamos a mano como en el archivo /etc/hosts de Linux, luego los que siguen en importancia de uso que pueden ser otros equipos dentro de la red con NIS (o similar) y el resto con DNS.

Para configurar el orden de búsqueda se utiliza el archivo /etc/nsswitch.conf que reemplaza al /etc/host.conf (puede aparecer en la configuración del sistema, pero no es tomado en cuenta).

El fichero nsswitch.conf permite al administrador de sistemas configurar una amplia variedad de diferentes bases de datos. (Limitaremos nuestra discusión a opciones que se refieran a la resolución de nombres de máquina y direcciones IP).

Las opciones posibles son:

- dns: Indica que se usa el DNS para resolver la dirección. Esto solo sirve para resolución de nodos, no de redes. Para ello se mira primero el fichero /etc/resolv.conf, que veremos después.

- files: Hace la búsqueda en un fichero local. Es decir, en /etc/hosts para los nodos, y en /etc/network para las redes.

- nis o nisplus: Usará el sistema NIS (sistema de información en red) para resolver nodos o redes.

El orden en el que los servicios estén listados es el orden en el que serán interrogados para buscar un nombre. Es decir, los servicios son interrogados leyéndolos de izquierda a derecha, hasta encontrar la respuesta.

Nsswitch - Ejemplo

Configurando el archivo /etc/nsswitch.conf

```
# /etc/nsswitch.conf
#
# Ejemplo de configuración del nsswitch de GNU.
# En el paquete `libc6-doc' se documentan estos ficheros.
hosts:      dns files
networks:       files
```

Este ejemplo hace que el sistema busque los nodos, primero en el DNS y después en /etc/hosts, si no se encuentra. En cambio las redes se buscan exclusivamente en /etc/networks.

Hosts - Ejemplo

El archivo /etc/hosts es utilizado por el sistema operativo para guardar la correspondencia entre dominios de Internet y direcciones IP.

Es uno de los métodos que usa el sistema operativo para resolver nombres de dominios. Resuelve el localhost y la IP local.

Antiguamente, cuando no existía el servicio DNS, los archivos hosts eran los únicos que podían resolver nombres dominios. En la actualidad también puede ser usado para bloquear contenidos de Internet.

Contiene un registro por línea: una dirección IP, un nombre de máquina y de forma opcional, una lista de alias para esa máquina. Los campos se separan por tabuladores o espacios y el campo con la IP debe empezar en la primera columna.

Un archivo hosts, por defecto incluye solo la definición para localhost: 127.0.0.1 localhost

También se puede agregar la IP y e nombre de la máquina: 10.0.3.14 maq12

Correspondencia a una página web: 209.85.229.104 www.google.es

Dominios de Internet bloqueados: 255.255.255.0 www.paginabloqueada.com

```
# /etc/hosts
# definición de bucle local
127.0.0.1        localhost
#
172.16.1.1   web.dominio.local       web
172.16.1.2   gate.dominio.local      gate
#
172.16.2.1   mail.dominio.local      mail
72.16.2.2    host.dominio.local      host
```

Tanto el nombre con cualificación completa (oficial) como el nombre local se deben registrar en el archivo /etc/hosts, para ser referidos al resolver su dirección IP.

Resolv.conf

El archivo /etc/resolv.conf contiene las direcciones IP de las máquinas que pueden ofrecer servicios DNS a nuestro host.

La instrucción nameserver apunta a servidores DNS que puede utilizar el host para realizar sus resoluciones.

Otras dos opciones, domain y search, nos permiten usar nombres cortos (sin dominio) para máquinas que estén en nuestro dominio. Normalmente, para conectarnos a una máquina de la misma red, no queremos poner el dominio completo, sino su nombre. Por ejemplo, gauss en lugar de gauss.mathematics.groucho.edu

Para esto sirve la palabra domain. Nos permite especificar un dominio predeterminado que se añade a las peticiones cuando su búsqueda inicial falla. Por ejemplo, al buscar gauss y fallar el servidor de nombres buscándolo en Internet, le añade automáticamente su dominio predeterminado y ya sí puede resolverlo.

Tan pronto como hagamos referencia a una máquina que esté fuera del dominio tendremos que volver a teclear el dominio completo, si sólo queremos tipear parte del dominio de la máquina podemos usar la lista de búsqueda, que puede especificarse con la opción search. Esta lista especifica una lista de dominios donde resolver nombres cortos. Los elementos de la lista deben especificarse separándolos por espacios o tabuladores.

Las opciones search y domain son mutuamente excluyentes y no pueden aparecer más de una vez. Si ninguna de las dos se pone, el sistema intentará asignar a los nombres cortos el dominio de la máquina local. Si el nodo local no tiene dominio, se asumirá que el dominio predeterminado es el raíz.

Si decidimos poner una opción search en el archivo resolv.conf, habrá que ser cuidadosos con los dominios que añadimos a la lista para evitar búsquedas innecesarias a los servidores de nombres externos.

Ejemplo:

```
# /etc/resolv.conf
    # Nuestro dominio
    domain     vbrew.com
    #1 Nuestro servidor principal va a ser vlager:
    name server 172.16.1.1
```

DNS

DNS (Domain Name System) brevemente

Hay dos maneras de identificar un host en una red:

- Por su nombre, más accesible para las personas
- Por su IP (de longitud fija y estructura jerárquica), usado por los routers.

Aquí es donde necesitamos un servicio de directorio que traduzca los nombres de host en direcciones IP. Éste servicio es el DNS.

DNS es:

Una base de datos distribuida implementada en una jerarquía de servidores DNS, y Un protocolo de la capa de aplicación que permite a los hosts consultar la base de datos distribuida.

Los servidores DNS suelen ser máquinas UNIX que ejecutan BIND (Berkely Internet Name Domain). El protocolo DNS se ejecuta sobre UDP y utiliza el puerto 53.

Ejemplo de funcionamiento:

- Suponga que una determinada aplicación (como por ejemplo un navegador web o un lector de correo), que se ejecuta en el host de un usuario, necesita traducir un nombre de host en una dirección IP. La aplicación invocará al lado del cliente de DNS, especificando el nombre de host del que necesita la correspondiente traducción. La aplicación DNS en el host del usuario entra en funcionamiento, enviando un mensaje de consulta a la red. Todos los mensajes de consulta y de respuesta DNS se envían dentro de datagramas UDP al puerto 53. Transcurrido un cierto retardo, del orden de milisegundos a segundos, el servicio DNS del host del usuario recibe un mensaje de respuesta DNS que proporciona la traducción deseada, la cual se pasa entonces a la aplicación que lo ha invocado.

nslookup

- Herramienta que consulta interactivamente los DNS de internet.
- Si el comando nslookup es ejecutado sin parámetros, de habilita el modo interactivo y nos muestra un prompt.
- Si el comando es invocado con un nombre de host, o de dominio, se obtiene una lista de las direcciones IP asociadas al nombre.
- Es útil para diagnosticar problemas de conexión y/o resolución de DNS

DHCP

Una vez que una organización ha obtenido un bloque de direcciones, puede asignar direcciones IP individuales a las interfaces de sus hosts y routers.

Las direcciones de host también se pueden configurar manualmente, pero frecuentemente ahora esta tarea se lleva cabo utilizando el Protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol) [RFC 2131].

DHCP permite a un host obtener (permite que se le asigne) automáticamente una dirección IP.

Un administrador de red puede configurar DHCP de modo que un host dado reciba la misma dirección IP cada vez que se conecte a la red, o un host puede ser asignado a una dirección IP temporal que será diferente cada vez que el host se conecte a la red.

Además de la asignación de direcciones IP a los hosts, DHCP también permite que un host obtenga información adicional, como por ejemplo su máscara de subred, la dirección del router del primer salto (gateway predeterminado) y la dirección de su servidor DNS local.

Por ejemplo, un estudiante que traslada una computadora portátil desde su casa a la biblioteca y luego a clase. Probablemente, en cada localización el estudiante se conectará a una subred y, por tanto, necesitará una nueva dirección IP en cada lugar.

DHCP está idealmente adaptado para estas situaciones, ya que existen muchos usuarios que van y vienen, y que necesitan direcciones sólo durante un periodo de tiempo limitado.

Del mismo modo, DHCP resulta útil en las redes de acceso de los ISP que trabajan en el mercado residencial. Considere por ejemplo un ISP residencial que tiene 2.000 clientes, pero no más de 400 clientes están en línea al mismo tiempo. En este caso, en lugar de necesitar un bloque de 2.048 direcciones, un servidor DHCP que asigne direcciones de forma dinámica sólo necesitará un bloque de 512 direcciones.

Preguntas de Repaso

- ¿Qué resulta mejor: usar direcciones IP o nombres?
- ¿Puede nombrar los archivos involucrados en la resolución de nombres en linux y decir que función cumple cada uno?
- ¿En que capa se implementa el sistema DNS?
- ¿Qué función cumple el servicio DHCP?

Subredes

La idea principal detrás de las subredes es administrar nuestra red local de una manera más sencilla.

Si se le asigna, por ejemplo, para la universidad una dirección clase A la tarea es cómo organizar y asignar las IP a los hosts. Podemos decir del 1 al 1000 son para informática, del 1001 al 2000 para ingeniería, y así sucesivamente.

Si se quiere aplicar restricciones al funcionamiento se torna muy complicado. Además, se puede considerar que dentro de cada facultad existen diferentes grupos de usuarios que podrían agruparse dentro de una misma red, por ejemplo: alumnos, docentes, administrativos, etc.
Claramente el esquema de direcciones de clases básicas no funciona.

Entonces se utilizará subredes para solucionar estos problemas y quedará claro que solo es una cuestión de **ADMINISTRACIÓN INTERNA**.

El esquema va a cambiar en la red local, pero desde el exterior (resto de internet) se verá como el esquema básico de arranque.

Sabemos que una dirección IP se puede dividir en un parte para la red y una parte para el host.

La red de destino se obtiene a partir de la parte de red de la dirección IP.

Los hosts con números idénticos de red IP deben encontrarse en la misma red.

También tiene sentido proporcionar un esquema similar dentro de la red local, ya que ésta puede constar de un grupo de cientos de redes más reducidas, con las unidades más pequeñas haciendo de redes físicas como Ethernets.

IP permite subdividir una red IP en varias subredes.

La parte de red se extiende ahora para incluir algunos bits de la parte del host.

El número de bits que se interpreta como el número de subred viene dado por la llamada máscara de subred o máscara de red.

La máscara de subred también es un número de 32 bits, que especifica la máscara de bit para la parte de red de la dirección IP.

Subredes - Prefijo extendido

Clase B 135.146.0.0

Como es una clase B, los dos primeros bytes están bloqueados y siempre serán los mismos para todos los números internos. Esta es la causa de que lleguen los mensajes a la red desde el exterior.

Luego extendemos la máscara tomando un octeto y todas las variaciones que se produzcan acá determinarán subredes diferentes.

Las clave de las subredes está en la máscara de red. Extender la máscara es convertir algunos **0s** del sector de hosts en **1s** y, una vez convertidos, pasan al sector de red. **Siempre de izquierda a derecha**.

.decimal	Byte 1	Byte2	Byte3	Byte4
255.255.0.0	1111 1111	1111 1111	**0000 0000**	0000 0000
255.255.255.0	1111 1111	1111 1111	**1111 1111**	0000 0000

Subredes - Caso Práctico

En este ejemplo se asignó una dirección C 193.1.1.0 con máscara 255.255.255.0 (**/24** haciendo mención a la cantidad de bits en **1**).

Lo primero a considerar es cuantas subredes se requieren. Siempre conviene considerar algunas más por si luego se necesiten. A veces se está limitado por la cantidad requerida.

Lo siguiente es considerar la cantidad de host que se requieren. Si decimos que se necesitan 6 subredes la forma de calcular la cantidad de bits para identificar estas subredes tiene que ver con la base 2 que es binaria. Así que se necesitará **n** cantidad de bits. Y la cuenta como todo en binario es: + buscar un número **n** tal que **2** (base) elevado a ese número **n** sea mayor o igual al número de subredes requeridos.

2^n >= Subredes

Porque decimos mayor o igual, porque puede que no sea exacto, pero siempre debe alcanzar. En este caso es 3 y da 8. Sobran un par de direcciones de subred que se podrían utilizar como reserva.

Ahora se extiende la máscara:

Ejemplo:

Número de red 193.1.1.0/24

- ¿Cuántas subredes se requieren?
- ¿Cuántos Hosts por subred?
- Para el caso 6 subredes –¿hosts?
- 2n >= 6 → n=3 → 8 subredes
- Prefijo extendido /27 → máscara 255.255.255.224

En el ejemplo se ve que le "**roba**" a la parte de hosts. Esto es así ya que el número de red original no se puede tocar.

En este ejemplo se tomó o se extendió la máscara en 8 bits, quedando: 255.255.255.0.

Podemos notar que se pierden direcciones de hosts.

Si, pero no tantas como parece.

Subredes en Binario

Con 3 bits podemos diferenciar 8 subredes.

La red 0 (primera ya que la cuenta arranca en **0**, típico) coincide con la original y es **191.1.1.0** .

Nótese que el número original de red no varía en ningún caso y es 193.1.1.x y es porque hacia afuera se mantiene para que los mensajes lleguen. Nadie del exterior necesita conocer el esquema interno.

Luego, la que sigue es la subred 1 o segunda pero **ojo**, no es la 191.1.1.1 o peor 191.1.1.2 , es la 191.1.1.32 porque el 001 está al comienzo del 4to byte. Los bloques de hosts son de a 32.

El número asignado es el 191.1.1.0/24 . Una red clase C.

Si se usa sin subredes hay 2^8 hosts para usar = 256.

Si se le aplica subredes, extendiendo la máscara en 3 bits se tiene 8 subredes.

En el archivo Ejemplo Completo de Subredes se puede ver la comparación del esquema original con el de subredes con colores diferentes. Poco práctico, pero se puede resumir sin perder información esencial en el archivo Ejemplo Reducido de Subredes.

Por último, en el archivo Ejemplo Reducido con Detalles de Subredes se pude ver lo mismo que el anterior pero las direcciones que están en rojo pertenecen a direcciones de red y las azules a direcciones de Broadcast.

En el esquema original solo dos direcciones no pueden asignarse a hosts, pero en el de subredes ya son 2 por cada subred(16).

La primera dirección de la primera subred coincide con la de la red original y la última de la última subred (Broadcast de esa subred) coincide con la dirección de Broadcast de la red original. Pero las otras 14 direcciones en la subred son direcciones que no se pueden asignar a host y se pierden.

Así, en el esquema original hay 254 direcciones para hosts contra las 240 del esquema de subredes. Estas últimas están distribuidas de a 30 por subred.

Como conclusión se puede decir que se pierden direcciones de hosts al aplicar esquemas de subredes, pero se gana en organización.

Recordar que el resto de internet no conoce el esquema de subredes.

Red Base: 11000001.00000001.00000001.00000000=193.1.1.0/24

Con 3 bits tenemos estas 8 subredes en binario. Los bits en negrita determinan los números de red de esas subredes:

- Subred 0: = 193.1.1.0/27 11000001.00000001.00000001.**000**00000
- Subred 1: = 193.1.1.32/27 11000001.00000001.00000001.**001**00000
- Subred 2: = 193.1.1.64/27 11000001.00000001.00000001.**010**00000
- Subred 3: = 193.1.1.96/27 11000001.00000001.00000001.**011**00000
- Subred 4: = 193.1.1.128/27 11000001.00000001.00000001.**100**00000
- Subred 5: = 193.1.1.160/27 11000001.00000001.00000001.**101**00000
- Subred 6: = 193.1.1.192/27 11000001.00000001.00000001.**110**00000
- Subred 7: = 193.1.1.224/27 11000001.00000001.00000001.**111**00000

Subredes – Números de Hosts

Un ejemplo que muestra como se calculan los hosts para cada subred. A los 3 primeros bytes que quedaban fijos, ahora se le suman los 3 bits del campo hosts y que también serán únicos y fijos solo que para cada subred.

La cantidad de hosts por subred se calcula como 25 por los 5 bits que se tienen. Eso da 32. Pero la primera dirección es la de red (subred en este caso) y la última, Broadcast. Entonces nos quedan 30 direcciones para asignar a hosts.

Números de hosts

5 bits → 32 host → 30 (n° red y Difusión)

Para calcular usamos los 5 bits libres del campo host.

Ejemplo para la:

- Subred 2: =193.1.1.64/24 11000001.00000001.00000001.01000000

- Host 1: = 193.1.1.65/27 11000001.00000001.00000001.010**00001**
- Host 2: = 193.1.1.66/27 11000001.00000001.00000001.010**00010**
- Host 3: = 193.1.1.67/27 11000001.00000001.00000001.010**00011**
- . . .
- Host 29: = 193.1.1.93/27 11000001.00000001.00000001.010**11101**
- Host 30:= 193.1.1.94/27 11000001.00000001.00000001.010**11110**

Subredes – Tabla Útil

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**	**Valor**
1	0	0	0	0	0	0	0	128
1	1	0	0	0	0	0	0	192
1	1	1	0	0	0	0	0	224
1	1	1	1	0	0	0	0	240
1	1	1	1	1	0	0	0	248
1	1	1	1	1	1	0	0	252
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

Caso de Estudio 1/4

Un estudio de caso como ejemplo usando una dirección IP clase B con máscara extendida en 8 bits resultando subred tipo clase C.

La red del campus GMU tiene un dirección IP de clase B 149.76.0.0 y su máscara de red es 255.255.0.0

Internamente, la red del campus tiene varias redes pequeñas, como las LANs de varios departamentos. De modo que el rango de direcciones IP se divide en 254 subredes; desde 149.76.1.0 hasta la 149.76.254.0.

Por ejemplo, la subred del Departamento de Física Teórica es 149.76.12.0. La dorsal (backbone) del campus es una red por derecho propio, y se le ha asignado 149.76.1.0.

Estas subredes comparten el mismo número de red, mientras que el tercer octeto se usa para distinguirlas entre sí.

Utilizarán así una máscara de subred de 255.255.255.0

Caso de Estudio 2/4

Más abajo se muestra como 148.76.12.4 se interpreta de forma distinta cuando la dirección viene dada como una red de clase B ordinaria y cuando se usa como subred: 1.7

Clase B

Red		Host	
149	76	12	4

Clase B con subredes

Red		Sub	Host
149	76	12	4

Caso de Estudio 3/4

Continuando con el ejemplo, se verán 4 redes pertenecientes al campus de GMU. Se describe cada subred y a algunas de las máquinas:

Departamento de matemáticas 149.76.4.0/24

Nombre	IP	Mask	Gw
Gauss	149.76.4.23	/24	149.76.4.1
Erdos	149.76.4.17	/24	149.76.4.1
Sophus	149.76.4.1	/24	149.76.1.2

Departamento de Física Teórica 149.76.12.0/24

Nombre	IP	Mask	Gw
Quark	149.76.12.4	/24	149.76.12.1
Niels	149.76.12.1	/24	149.76.1.2

Centro de Cómputo de GMU

Nombre	IP	Mask	Gw
Gcc1	149.76.2.1	/24	x.x.x.x

Backbone del campus

Nombre	IP	Mask	Gw
Sophus	149.76.1.1	/24	149.76.1.2
Niels	149.76.1.12	/24	149.76.1.2
Gcc1	149.76.2.1	/24	x.x.x.x

El Backbone es la subred que interconecta al resto de las subredes. Las máquinas que pertenecen a esta subred en este ejemplo, también pertenecen a otras subredes. Estas funcionan como Gateways o Routers y por lo tanto deben tener una dirección IP por cada subred conectada (normalmente una por cada interface).

Podrían existir otras máquinas en el backbone, como por ejemplo servidores, que son de común acceso por el resto de los hosts.

Los routers Sophus y Niels tienen como Gw default a Gcc1 y seguramente, este último, debe tener uno que lo comunique con Internet expresado como x.x.x.x (no importante ahora).

Caso de Estudio 4/4

Se analiza como un paquete viaja de un host basado el el ejemplo anterior. Esta explicación se volverá a ver nuevamente y con más detalles.

Cabe aclarar que el ejemplo está tomado en una red ethernet pero se puede extender a cualquier otro tipo de red.

Suponiendo que ya se conocen las direcciones IPs y las MACs. Existen dos casos:

- 1.- Ambos hosts se encuentran en la misma red (subred): en este caso solo debemos enviarle el paquete al destino utilizando la MAC que lo identifica.

- 2.- Los hosts se encuentran en redes diferentes: en este caso no podemos enviarlo directamente, pero existe alguien a quien se lo podemos enviar para que lo resuelva que está en la red y se llama router.

Se mencionó anteriormente que los routers tienen más de una red conectada y por lo tanto una IP por cada conexión. También que las IP están asociadas a una interface.

Las tablas de ruteo son la herramienta fundamental para que los paquetes encuentren su destino en la red. Todos, tanto los hosts como los routers la tienen. Sin embargo para los hosts, en general, es muy simple y posee solo dos entradas o lineas: la red conectada y el resto.

Recordar que los parámetros que se debe configurar en un host para que pueda comunicarse son: + IP + Máscara + GW + DNS

Nota:el DNS no lo tendremos en cuenta en los laboratorios porque por una razón de simplicidad utilizaremos el archivo /etc/hosts.

Los Datos son:

Dato	**Erdos**	**Gauss**
IP	149.76.4.17	149.76.4.23
MASK	255.255.255.0	255.255.255.0
GW	149.76.4.1	149.76.4.1

Tabla de ruteo de Erdos

RED	**MASK**	**GW**	**Interface**
149.76.4.0	255.255.255.0	-	eth0
default	0.0.0.0	149.76.4.1	eth0

Tiene dos entradas(filas). La primera con información sobre la red a la cual está conectado y la segunda cualquier otra red.

Casos de envío de mensajes:

1) Erdos le manda un mensaje a Gauss. Como tiene su IP que es 149.76.4.23 la compara con la primer entrada en la tabla de ruteo. Le aplica la máscara de la segunda columna:

IP Gauss	**149.76.4.23**
&	
MASK	255.255.255.0
=========	==========
Red	149.76.4.0

Luego compara la red resultado que le dió con la red de la columna 1. Como son iguales significa que están en la misma red (por eso la casilla GW está vacía). Entonces solo tiene que enviárselo a Gauss a su MAC por la interface eth0 (más adelante veremos como se averiguan las direcciones MACs).

2) Erdos le envía un mensaje a Quark cuya IP tiene y es 149.76.12.4. El proceso es similar al caso anterior. Compara la IP con la primer entrada en la tabla de ruteo. Le aplica la máscara de la segunda columna:

IP Quark	**149.76.12.4**
&	
MASK	255.255.255.0
=========	==========
Red	149.76.12.0

Luego compara la red resultado que le dió con la red de la columna 1. En este caso son diferentes entonces salta a la entrada siguiente. Le aplica la máscara 0.0.0.0 que en este caso es un comodín y resulta que es la entrada correcta (default - todo lo que no coincide con alguna entrada anterior se resuelve acá). Se fija en la casilla GW que tiene la dirección del router (default para Erdos) y le envía el mensaje a la dirección del MAC del router Sophus por la interface eth0 (la única que tiene).

Sophus recibe el mensaje analiza su tabla de ruteo y decide enviárselo por el Backbone a Gcc1, su GW default.

Gcc1 decide que debe mandar el mensaje a Niels y se lo envía a su MAC por el Backbone.

Niels recibe el mensaje y determina que la IP destino pertenece a su red interna y entonces lo entrega enviándolo a la MAC de Quark.

La tabla de ruteo de un router es un poco más compleja que la de un host pero el funcionamiento es igual. Siempre se comienza con la primer entrada, luego la que sigue y así hasta que se resuelve. Por eso todo lo que no se puede resolver se lo mandamos a alguien (GW default). Hasta los routers tienen esta opción. Si no se puede resolver (seguro no tiene la opción default) en mensaje es descartado. .

Hay que aclarar que cuando se dice mayor máscara de red significa mayor cantidad de bits en 1.

Existen cuatro tipos de entradas en la tabla de ruteo: + 1.- Las redes directamente conectadas. + 2.- Redes configuradas en forma estática. + 3.- Redes configuradas en forma dinámicas. + 4.- Defecto.

Se verá con mas detalle en una próxima teoría.

Preguntas de repaso

- ¿Cuál es la idea detrás de las implementación de subredes?
- ¿Porqué se dice que se introduce un nivel más de jerarquía en las subredes?
- ¿Porqué desde el exterior no deben conocer el esquema de subredes en una red local?
- ¿Qué significa extender la máscara de red a máscara de subred? ¿Cómo se hace este proceso?

Referencias

- Guía de Administrador de redes Linux
- Capítulo 2 y 5
- Notas de esta presentación
- Página de la materia en pedco.uncoma.edu.ar
- Kurose-Ross Computer Networking A Top-down Approach Featuring the Internet Third !!br0ken!!
- Redes de Computadoras, 5ta Edición - James F. Kurose & Keith W. Ross