Redes de área local y área extensa

Autores:

- Pablo Boronat Pérez (Universitat Jaume I)
- Miguel Pérez Francisco (Universitat Jaume I)
- David Rubert Viana (Universitat Jaume I)

Introducción

Los sistemas de comunicaciones se diseñan y programan en **capas** o niveles para que sean más fáciles de implementar, modificar y mantener.

Cada capa resuelve un conjunto de aspectos, ofrece servicios a la capa superior y usa servicios de la capa inferior.

Los **protocolos** son las reglas de comportamiento (establecimiento de comunicación, formato y tamaño de los mensajes, etc.) que permiten que un nivel en un ordenador comunique o sepa tratar con el mismo nivel en otro ordenador. Por ordenador nos referimos a cualquier aparato con el que comunicar.

Los niveles de los que hablamos son los siguientes: físico, enlace de datos, red, transporte, aplicación.

Aunque no es exacto, normalmente la red de área local se asocia con los niveles físico y de enlace de datos y las redes de área extensa con los niveles de red y de transporte.

Capas

Capa 2. Enlace de datos

La capa de enlace de datos es sobre la que se hace la comunicación en una **red de área local** (LAN). En nuestro caso Ethernet y Wifi. En esta capa se entregan mensajes entre tarjetas de red conectadas *en teoría* a un mismo medio físico. Ocurre que hay **puentes** que unen diferentes LAN formando una LAN mayor donde puede haber más de un medio físico, pero se trataría de la misma LAN trabajando en el segundo nivel (el nivel de enlace de datos). Por ejemplo, un punto de acceso doméstico que tiene wifi y puertos ethernet, normalmente está configurado de esta forma.

En la capa de enlace de datos las direcciones son las MAC (*Media Access Control*), de 48 bits, que vienen grabadas en la tarjeta de red por el fabricante. Cada MAC es única. Se expresan como 6 pares de cifras hexadecimales separados por “:”. Por ejemplo: direcciónHW 00:18:8B:DE:06:DB

*Spanning Tree Protocol*En algunas configuraciones se ven *puentes* (*bridge*) que unen varias tarjetas de red emulando un conmutador o *switch*. Por ejemplo, por defecto guifi.net crea un puente entre la primera ethernet y la primera tarjeta de radio de los routers. A este dispositivo le da el nombre de *wlan/lan*.

El puente wlan/lan que crea guifi.net no es imprescindible, pero tiene la ventaja de que la IP que se asigna a él es accesible tanto por radio como por cable. Normalmente esta IP es la que representa de modo general al router.

Otra cosa que se usa en los montajes son las **VLANs** (*Virtual LAN*). Las redes de área local virtuales se usan para dividir una red de área local.

Un ejemplo de uso de VLANs es separar el switch que lleva un router-punto de acceso doméstico con cuatro bocas ethernet, en redes separadas en la capa 2. De esta forma las difusiones de una VLAN no afectan a otras y el tráfico está completamente separado. Es decir, podemos dividir un switch en varios switches virtuales.

Capa 3. Red

La capa de red se encarga de enviar mensajes a través de Internet. Este proceso se compone de dos partes: el reenvío y el encaminamiento o enrutamiento.

En este nivel las direcciones son IPs. Existen dos tipos de direcciones IP. Las IP versión 4 (de 32 bits que todos conocemos) y las IP versión 6 (de 128 bits, que tendremos que tratar en breve).

En el estado actual vamos a tratar IPv4 porque es lo que de momento usamos en guifi.net.

El *reenvío* es el proceso mecánico de encontrar el dispositivo por el que enviar cada mensaje, utilizando una tabla de rutas.

El *enrutamiento* consiste en construir la tabla de rutas de forma dinámica, aprendiendo de los cambios que se producen en la red.

Cualquier ordenador tiene una tabla de rutas que funciona de forma estática. Un *router*, además, realiza enrutamiento o encaminamiento dinámico para adaptarse a los cambios que se produzcan.

Direcciones IPv4

Una dirección IPv4 es un vector de 32 bits. Para comprenderlo mejor, usamos una notación decimal en vez de binaria. Separamos los 32 bits en 4 octetos y cada uno lo pasamos a decimal. Por ejemplo 150.128.98.10 en realidad es la dirección:

10010110.10000000.01100010. 00001010 (los puntos no existen, solo se ponen para separar los grupos de 8 bits).

76543210

7654

En una dirección IP se distinguen dos partes. Una parte (la de mayor peso) indica la red a la que pertenece esa dirección IP. La otra parte (de menor peso) indica una dirección concreta dentro de esa red. Por ejemplo, una dirección de

un ordenador.

La separación entre las dos partes de una dirección IP se indica con la **máscara**.

La *máscara* es otro vector de 32 bits. Tiene una parte de unos y el resto de ceros. La parte de unos es la que indica los bits de red de una dirección IP. La parte de ceros da una dirección dentro de esa red.

La máscara se escribe de dos maneras. Como 4 octetos expresados en decimal (por ejemplo 255.255.224.0) o simplemente diciendo los unos que contiene (/27). Por ejemplo una dirección de una red con esta máscara podría ser la siguiente:

10.228.132.101/27

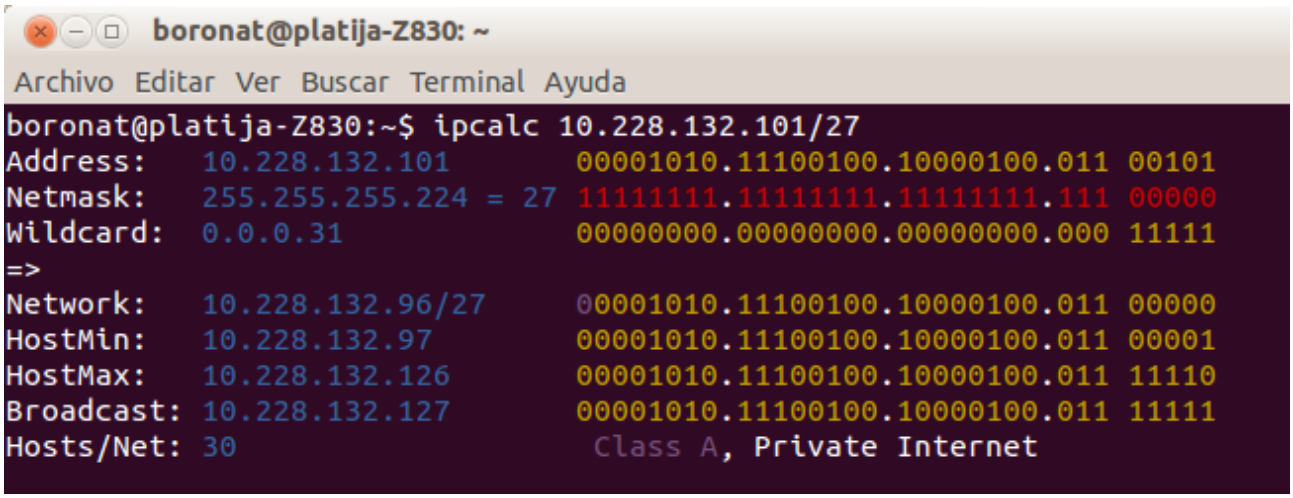
Si hacemos la operación lógica *and* entre una dirección IP y su máscara obtenemos la dirección de red.

dirección de red*dirección cero*

Los bits a cero de una máscara nos indica la cantidad de direcciones que contiene una red.

Por ejemplo, la red 10.228.132.96/29, tiene una máscara con 32-29=3 ceros. Por tanto, esta red puede repartir 2^3=8 direcciones. Por convenio, la primera y la última de estas direcciones no se asigna porque tienen un significado especial. La primera dirección (10.228.132.96) representa la IP de la red. La última dirección (10.228.132.103) es la **dirección de difusión** (*broadcast*). Es decir, que en realidad tenemos 2^3-2=6 direcciones aprovechables.

Hay calculadoras para hacer este tipo de comprobaciones. Por ejemplo la aplicación ipcalc (hay aplicaciones similares para todos los sistemas) calcula todos los datos de la red a partir de una IP.



Ejemplo de cálculo de parámetros de una red con ipcalc.

Subredes

En las tareas de administración de una red una muy típica es dividir Una red IP en **subredes lógicas**. Esta operación se puede ver de dos formas. Una es dividir una red IP en subredes lógicas de **igual tamaño**. La otra es dividirla en redes de **diferente tamaño** según las necesidades. En el primer caso todas las subredes tendrán la misma máscara y para el segundo, la máscara se irá

adaptado a la cantidad de direcciones que necesitemos en cada subred.

Desde fuera de la organización que tiene la red asignada no tiene porqué saberse si la red está dividida o no (es un decisión interna).

Para dividir una red en subredes de igual tamaño, se añaden a la máscara algunos bits de la parte de ordenador (estos bits serán los de mayor peso y diferenciarán una subred de otra). $2^{\text{bits añadidos}}$ será la cantidad de subredes que se podrán asignar. Los bits que quedan sin añadir a la máscara serán las direcciones que se pueden usar en cada subred.

- ¿Cuántos bits necesitamos añadir a la máscara? $\log_2(8) = 3$ (es decir, $2^3 = 8$)
- ¿Cuál será la máscara de las subredes? 26+3 = 29
- ¿Cuántos direcciones hay en cada subred? 32-29 = 3, por tanto $2^3 = 8$ direcciones; como no usaremos ni la primera ni la última, nos quedan 8-2 = 6 direcciones utilizables en cada subred.
- La primera subred: 10.90.74.01|000|000 = 10.90.74.64/29, dirección de difusión: 10.90.74.71/29
- La segunda subred: 10.90.74.01|001|000 = 10.90.74.72/29, dirección de difusión: 10.90.74.79/29
- La tercera subred: 10.90.74.01|010|000 = 10.90.74.80/29, dirección de difusión: 10.90.74.87/29
- La octava subred: 10.90.74.01|111|000 = 10.90.74.120/29, dirección de difusión: 10.90.74.127/29

Si se quiere dividir la red en subredes de diferente tamaño, según las necesidades, se adapta la máscara a lo que requiere cada subred. Nótese que las máscaras siempre serán mayores que la de la red inicial (más unos) porque en otro caso estaríamos saliéndonos de nuestra red. Los bits añadidos a la máscara se asignan a una red concreta y hay que anotar las combinaciones no usadas para asignaciones posteriores. Cada red asignada no puede tener los mismos bits en la parte de red que otra de las subredes porque eso significaría que se están solapando. Por otro lado las direcciones asignada a la red serán contiguas y usando la misma parte de red (indicada por la máscara) para todas las direcciones de cada subred.

7

prefijo

1

1

Este tipo de particiones se usa continuamente en guifi.net. Las direcciones públicas de guifi.net (son direcciones para redes *privadas* en Internet) parten de la red 10.0.0.0/8. Los enlaces troncales (normalmente punto a punto) entre nodos multirradio se hacen dentro de la 172.16.0.0/12. Tenemos la ventaja de que la web de guifi.net nos hace las asignaciones automáticamente, pero conviene entender el proceso para poder detectar errores o para hacer planificaciones de las direcciones.

Por ejemplo, cuando se crea una zona, se reserva una /24 dentro de la 10.0.0.0/8. Para cada antena de cobertura (acepta nodos cliente) se asigna una

/27 dentro de la /24. Los administradores suelen reservar /29 para direcciones de servidores o direcciones administrativas. Los enlaces punto a punto se hacen con /30 dentro de la 172.16.0.0/12. Todas estas redes o subredes deben ser disjuntas dentro de guifi.net.

Reenvío de mensajes

Es el proceso con el que los ordenadores y routers envían mensajes usando su tabla de rutas.

Las rutas están ordenadas por orden decreciente de máscara. Esto hace que las más específica estén arriba y las más generales abajo.

```
boronat@boga:~$ route -n
Tabla de rutas IP del núcleo
Destino         Pasarela        Genmask         Indic Métric Ref
Uso Interfaz
10.9.8.1        0.0.0.0         255.255.255.255 UH    0      0
0 tun1
192.168.10.0    0.0.0.0         255.255.255.0   U     1      0
0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0
0 eth0
0.0.0.0         192.168.10.1    0.0.0.0         UG    0      0
0 eth0
```

El proceso para elegir por qué interfaz se envía un mensaje IP y a quién se envía es el siguiente. Por orden y empezando por la primera ruta. Se hace IP-destino AND máscara. Si el resultado coincide con la columna destino, se usa esa línea (ya sabemos la interfaz de salida). En otro caso se sigue con la línea siguiente. Cuando se acierta una línea, si *Pasarela* tiene ceros quiere decir que es una red conectada. La trama de enlace de datos tendrá como MAC destino la correspondiente a la IP-destino. Si pasarela tiene una IP, entonces la MAC destino es la de la IP que hace de pasarela (el destino no está en una red conectada directamente y el mensaje se envía al siguiente salto).

Encaminamiento

Es el proceso por el que los routers encuentran nuevas redes y cómo alcanzarlas. Se trata de un proceso dinámico; los routers deben detectar y adaptarse a cambios en la red.

Un ordenador normalmente tiene una configuración estática de rutas. Contiene las redes a las que está conectado y alguna ruta añadida, como por ejemplo una puerta por defecto, la cual será la que se aplicará en último término si no encuentra ninguna ruta válida. Un router, además, normalmente tiene rutas *aprendidas* por el protocolo de encaminamiento dinámico.

Hay dos grandes tipos de encaminamiento: entre *sistemas autónomos (AS)*, encaminamiento (*EGP, Exterior Gateway Protocol*) y dentro de un sistema autónomo (*IGP, Interior Gateway Protocol*).

Un **sistema autómo** suele ser una red de una empresa o institución. Los AS se conectan e intercambian tráfico, formando Internet. Cada AS tiene un identificativo y unas redes asignadas para poder conectar e intercambiar

tráfico con otros.

El protocolo de facto usado para EGP es eBGP. Estos protocolos se rigen por acuerdos económicos o sociales más que para buscar la mayor eficiencia (las mejores rutas).

De momento guifi.net no está separada en sistemas autónomos.

Dentro de un AS se usan protocolos IGP que buscan la mayor eficiencia (las *mejores* rutas, que no siempre son las más cortas, porque una ruta más larga podría tener mayor ancho de banda con un retardo aceptable).

Hay varios tipos de protocolos IGP y muchas implementaciones. En guifi.net se usan principalmente OSPF y iBGP.

En algunas pequeñas zonas se usan protocolos de encaminamiento para *redes ad hoc* (a estas redes también se les llama *redes mesh*, pero esto crea confusión). En estas redes los nodos son a la vez clientes y routers para los demás nodos.

En la tabla de encaminamiento, según el origen de cada ruta (redes conectadas, rutas estáticas, rutas aprendidas por diferentes protocolos de encaminamiento) tienen diferente peso o se les otorga diferente confianza. Por ejemplo, normalmente las rutas estáticas son más de fiar que rutas aprendidas dinámicamente. El administrador puede modificar estos pesos, pero es raro hacerlo.

En las tablas de rutas se usa una mezcla de pesos o costes o *distancias administrativas* y valores o *métricas* proporcionadas por el protocolo de encaminamiento.

Los manuales de los routers deben indicar estas distancias. Un administrador las puede modificar.

http://en.wikipedia.org/wiki/Administrative_distance

En la tabla de encaminamiento puede haber varias rutas a un mismo destino. Se usará la de menor distancia administrativa. Si hay varias posibilidades con la misma distancia administrativa, se usará la que mejor métrica tenga.

Encaminamiento OSPF

Open Short Path First

Los pasos del este tipo de algoritmos de encaminamiento (*encaminamiento por estado de enlace*):

1. Descubrir a sus vecinos. Se envían mensajes *hello* y se indentifican.
2. Estiman el coste de cada enlace directo (las implementaciones actuales de OSPF lo hacen estáticamente según el tipo de interfaz).
3. Se difunden mensajes con los enlaces detectados (*mensajes de estado de enlace*).
4. Con los mensajes de estado de enlace se puede construir la topología de la red.
5. Cada router busca su ruta más corta a cada red destino con la topología construida.

Los routers comprueban periódicamente sus enlaces (con qué vecimos conectan directamente). Si se detectan cambios, entonces se hace una inundación con la nueva información. Los routers guardan esta información en una base de datos de estados de enlaces (comprobando que no es información rezagada, que se envían acuses de recepción, y otros mecanismos).

Con OSPF se pueden crear zonas dentro de un mismo sistema autónomo. De esta forma se puede reducir la tabla de rutas de los routers que solo están en una zona (*sumarización de rutas*). La zona 0 (cero o *backbone*) es especial porque es una zona dorsal que debe conectar a las otras zonas. Habrá routers que están en la zona 0 y otras zonas. Todas las zonas deben conectar con la zona 0. La zona 0 es la que viene configurada por omisión. De momento en guifi.net no se están usando zonas OSPF.

*Sumarizaciónagregación*De momento en guifi.net no se están declarando zonas OSPF.

Aspectos que nos interesan:

- Redes que queremos publicar.
- Rutas estáticas que queremos publicar.
- Estado de los vecinos.
- Rutas aprendidas OSPF y costes.

La métrica usada en OSPF se basa en el ancho de banda de la cada interfaz de red. La mayoría de implementaciones usan 10^8/(anchoBanda bps). Cuanto mayor es el ancho de banda, menor es esta métrica y por tanto es más probable que ese enlace se use en las rutas OSPF. Por ejemplo, a partir de Fast Ethernet (100mbps) el valor es 1.

Esta forma de usar la métrica OSPF funciona bien pero es problemática en los enlaces inalámbricos. Dado que funciona de forma estática, si las interfaces de red son del mismo tipo, al final lo que cuenta es el número de saltos. Pero en nuestro caso no es suficiente porque los enlaces tienen diferente ancho de banda según el ruido, la distancia, el tipo de antena, etc. En algunos casos es necesario cambiar manualmente la métrica para forzar que no se utilice una ruta. Desgraciadamente coordinar este trabajo manualmente es complicado y puede acarrear otros inconvenientes.

Otro problema de OSPF es que cuando hay enlaces inestables, puede provocar inestabilidad en toda la red porque se hacen difusiones cuando se detecta un cambio, y los routers al recibir las actualizaciones de los enlaces, deben recalcular las rutas más cortas en la topología que poseen en su base de datos. Esto tiene un coste considerable.

full

Encaminamiento iBGP

http://lacnic.net/documentos/lacnicxii/presentaciones/08_bgp.pdf revisar

http://en.wikipedia.org/wiki/Bgp

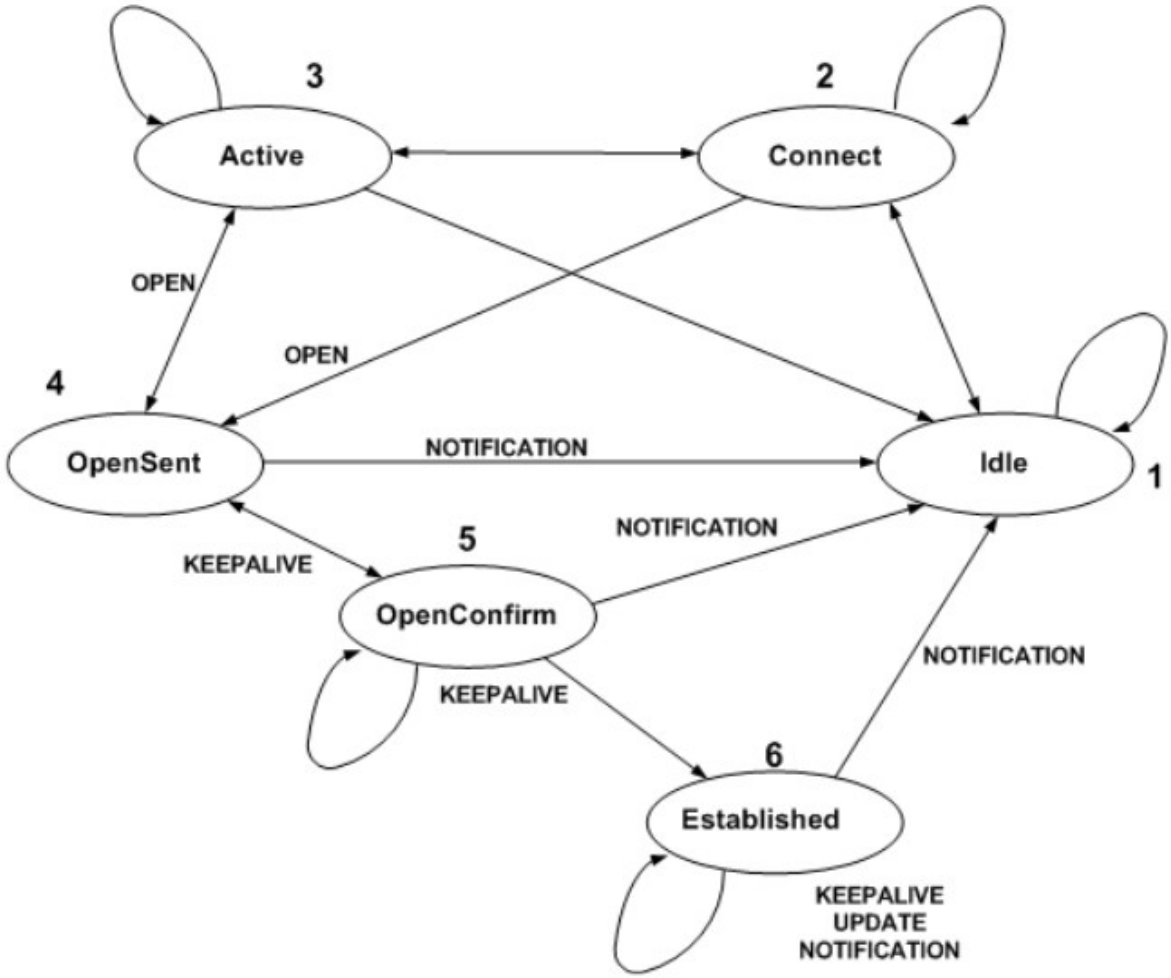
http://tools.ietf.org/html/rfc4271

BGP es el (único) protocolo que se usa para enrutamiento EGP (entre sistemas autónomos). En ese caso se le llama BGP de exterior o *eBGP*. Pero también se usa como protocolo de encaminamiento IGP (de interior de sistema autónomo). Entonces se le llama *iBGP*.

El funcionamiento de BGP se basa en que los routers encuentran rutas a los destinos y difunden a sus vecinos estas rutas. Es parecido a los protocolos de *vector de distancia* como RIP, pero en vez de indicar únicamente la distancia a cada destino conocido, se incluye la ruta. Eso evita algunos problemas de convergencia que tienen los protocolos de vector de distancia. Se le suele llamar protocolo de encaminamiento por *vector de rutas*.

*prefijos*Los vecinos de un router con BGP no se descubren automáticamente. Hay que establecer una conexión TCP con ellos.

Un router ejecutando BGP, y por cada vecino BGP, pasa por una serie de 6 posibles estados hasta que el funcionamiento se declara como establecido. Esto se resume en un autómata de estados finitos. Estos estados nos pueden servir para encontrar por qué dos routers vecinos no están colaborando en BGP (consultar los enlaces).



Si el estado está repetidamente entre *Connect* y *Active* y finalmente vuelve a *Idle* probablemente la conexión TCP no se establece. Puede haber un cortafuegos o la red no está funcionando en el enlace esperado. Si el estado está en *Established* la cosa funciona.

Una vez que un par de routers tienen una sesión BGP establecida intercambian varios tipos de mensajes.

- Mensajes *Keepalive* cada 60 segundos para indicar que siguen ahí y mantener la conexión TCP.
- Mensajes *Update*. Para indicar nuevas rutas.
- Mensajes *Notification*

Las rutas se describen en *NLRI* (*Network Layer Reachability Information*) dentro de los mensajes Update. En estos NLRIs se incluyen diferentes datos, además del prefijo de la red destino que se anuncia. Estos datos sirven para que el receptor de las rutas decida incluirlas en su tabla o guardarlas con un determinado peso. También se anuncian rutas que desaparecen o ya no son alcanzables (*withdrawals*).

Cuando se usa iBGP, todos los routers deber intercambiar rutas con todos los demás. Se puede usar una configuración en la todos envían a uno para limitar el número de conexiones TCP (*route reflector*). Es algo parecido como el *router designado* de OSPF cuando se conectan varios routers a una misma LAN.

Tracedump:

```
newBaseSize: 12pt
newBaseSizeInPt: 12
```