

CC51C Comunicación de Datos Capa Red

1 Introducción

- Misión: ruteo, manejo de congestión y de errores.
- ahora el problema es conectar dos máquinas en distintas redes, interconectadas de algún modo.

2 IP

- End to end argument: inteligencia en las puntas
- en telefonía: inteligencia en la red
- IP sobre todas las cosas: cualquier red física debiera ser capaz de transmitir IP
- para interconectar dos redes: router (un computador que está en ambas redes y puede enviar datos de una a otra y decidir si corresponde)
- no existe concepto de conexión a este nivel
- requerimientos para IP:
 1. Espacio de nombres únicos en toda la inter-red
 2. independientes de la red. la idea es no exigir nada más que lo mínimo a las capas inferiores.
 3. traducibles a direcciones físicas
 4. ruteo de los datos en base al "nombre"IP
 5. paso de los datos por los routers sin alteración

3 Direccionamiento en IP

- espacio de direcciones de 32 bits
- esta dirección se divide en red/host
- notación "dotted decimal": 4 bytes en notación decimal (0-255), separados por puntos
- Clases:

Clase	bits	Rango de direcciones	Hosts / Red	# redes
A	0	1.0.0.0 → 126.0.0.0	$2^{24} - 2 = 16.777.214$	$2^7 - 2 = 126$
B	10	128.1.0.0 → 191.254.0.0	$2^{16} - 2 = 65.534$	$2^{14} - 2 = 16.382$
C	110	192.0.1.0 → 223.255.254.0	$2^8 - 2 = 254$	$2^{21} - 2 = 2.097.150$
D	1110	224.0.0.1 → 239.255.255.254	$2^{28} - 2 = 268.435.454$	-
E	11110	240.0.0.1 → 247.255.255.254	$2^{27} - 2 = 134.217.726$	-

La clase D corresponde a direcciones de Multicast y la clase E son direcciones reservadas. En estos dos casos, no hay una red sino sólo direcciones.

- direcciones reservadas y/o especiales:
 1. 127.0.0.0: red loopback (nunca debiera ir a una interfaz de red)
 2. todos los bits de host en 0: dirección de la red o bien *este* host
 3. todos los bits de red en 0: *esta* red
 4. todos los bits de host en 1: dirección de broadcast
 5. todos los bits de red en 1: todas las redes

Además, se han reservado las siguientes redes para que cualquier persona y/o institución las use en forma privada, pero no deben publicarse rutas desde ni hacia estas redes en la Internet:

- 10.0.0.0 (red clase A)
 - 172.16.0.0 → 172.31.0.0 (16 redes clase B)
 - 192.168.0.0 → 192.168.255.0 (256 redes clase C)
- Sub-redes: máscaras de red
Es posible dividir una red, y para ello existen 2 formas:
 1. Static subnetting: todas las sub-redes usan la misma máscara de red. Esta forma es la única soportada directamente y sin restricciones por el ruteo nativo de IP y por RIP (el protocolo de ruteo más usado). Es posible eso sí volver a dividir una sub-red en sub-sub-redes.
 2. Variable subnetting: las sub-redes pueden ser de distintos tamaños.

4 CIDR

- extendiendo el concepto de sub-redes: Super-redes. (1992)
 1. término de las direcciones clase B. La mitad estaban asignadas.
 2. término de todas las direcciones IP → IPv6.
 3. explosión en las tablas de rutas. Manejo en memoria y ancho de banda de los protocolos.
- Solución: CIDR (Classless Inter-Domain Routing).

- Toda dirección debe contar con máscara.
- backwards-compatibility: máscaras implícitas para redes clase A, B y C. Aumenta el problema 3, pero funciona.
- Ejemplo: las redes clases C 200.0.0.0 a la 200.0.3.0 pueden escribirse como 200.0.0.0/255.255.252.0 o como 200.0.0.0/22

5 Datagramas IP

5.1 Header IP

0	4	8	16	19	24	31
VERS	HLEN	SERVICE_TYPE	TOTAL_LENGTH			
IDENTIFICATION			FLAGS	FRAGMENT_OFFSET		
TIME_TO_LIVE		PROTOCOL	HEADER_CHECKSUM			
SOURCE_IP_ADDRESS						
DESTINATION_IP_ADDRESS						
IP_OPTIONS (IF ANY)					PADDING	
DATA						
...						

Figura 1: Header del paquete IP

- Vers: versión del protocolo
- HLen: largo del header (en bloques de 32 bits). El tamaño máximo del header es por lo tanto $(2^4 - 1) = 15$ bloques de 32 bits = 60 bytes.
- Service type: (TOS) primeros 3 bits son la precedencia, luego vienen bits denominados D (low delay), T (high throughput), R (high reliability) y los últimos 2 no se usan.
- Total Length: largo total (header + datos) en octetos (bytes). Con 16 bits para codificarlo, el largo total máximo de un paquete es de $2^{16} - 1 = 65535$ bytes. El mínimo se define arbitrariamente en 8 bytes.
- Identification: número “único” para cada datagrama IP (ver fragmentación)

- Flags: 3 bits para fragmentación: el primero es reservado, el segundo es llamado *DF*: Don't Fragment, y el tercero es *MF*: More Fragments.
- Fragment offset: para fragmentación
- Time To Live (TTL): tiempo de vida en segundos que le queda al paquete. Cada router debe al menos decrementar este tiempo en 1. En la práctica siempre se hace eso, y por lo tanto no tiene mucho que ver con tiempo, sino con número de hops (saltos). Sirve para descartar paquetes en loop.
- Protocol: identifica el protocolo al cual se debe entregar los datos de este datagrama (ej: TCP, UDP, ICMP).
- Header Checksum: checksum de 16 bits sobre todo el header asumiendo que el checksum son 16 bits en 0. Es necesario calcularlo en cada hop, dado que se está modificando el TTL y por lo tanto el checksum cambia.
- Source y Destination IP Addresses: direcciones en 32 bits consecutivos, en network byte order (big endian, o sea, byte más significativo primero).
- Options: variadas opciones, no siempre están presentes.
- Padding: relleno para opciones que no llenan un bloque de 32 bits.

5.2 Opciones IP

- Opciones de ruteo: *Loose source routing* y *Strict source routing*:

En ambas opciones el origen especifica el camino que desea siga el paquete. El camino consta de una secuencia de números IP, y un puntero que diferencia la parte de la secuencia que ya se ha ruteado de la por rutear. En cada etapa se van modificando las direcciones IP de origen/destino del datagrama para simular una transmisión local del paquete. Además, cada router cambia la dirección suya de la lista por la dirección a través de la cual envía el datagrama al siguiente router (estas direcciones son necesariamente diferentes, ya que son redes distintas). Esta información se llama record route y es usada en el destino para enviar un paquete de vuelta pasando por los mismos routers. La diferencia entre *Loose* y *Strict* está en que en la primera modalidad se permiten varios hops entre cada router especificado, mientras que en ruteo estricto todas las direcciones deben ser accesadas directamente.

Muchos routers descartan este tipo de paquetes por ser una amenaza a la seguridad (ej: IP spoofing).

- Record Route:

Se le pide a cada router que agregue su dirección en el buffer reservado por el emisor dentro del header IP, en forma similar a como se hace en *source routing*, pero el camino es inicialmente nulo. Es necesario que el origen provea el espacio suficiente para que quepan todas las direcciones.

- Internet Time Stamp:
Cada router agrega un timestamp (fecha en segundos) al header. No sirve para mediciones de performance, dado que la resolución mínima es 1 segundo, y además los routers no necesariamente tienen sincronizados los relojes.

6 MTU y Fragmentación

Una red física tiene sus restricciones, entre las cuales se cuenta por ejemplo el máximo tamaño de un paquete de datos que acepta de capas superiores (como el MAX_DPACK que usamos en la capa datos). Como el header máximo es de 60 bytes, y el tamaño mínimo de los datos es de 8 bytes, IP no puede funcionar sobre un medio que provea tamaños de paquetes menores a 68 bytes, a menos que se proporcione un sistema de fragmentación transparente a IP.

Como el emisor no sabe desde un inicio cuál es el MTU mínimo de un camino, y porque los caminos pueden cambiar en el tiempo, es posible que llegue un paquete de largo x que debe ser enviado por una red de MTU $y < x$.

- campos del header:
 - Identification: para volver a juntar los fragmentos que correspondan al mismo datagrama.
 - Flags (RESERVADO, DON'T_FRAGMENT, MORE_FRAGMENTS)
 - Fragment Offset: 13 bits (número dado en múltiplos de 8 bytes = 64 bits)
- Fragmentación de fragmentos

7 Ruteo

Para posibilitar el ruteo, cada host y cada router maneja una tabla de rutas con la información necesaria de la red (su visión de la red). Todas las decisiones respecto de cómo y por dónde enviar un paquete se hacen en base a esta tabla de rutas. El ruteo propiamente tal se preocupa tanto de hacer llegar los paquetes al destino final como de mantener la información de las tablas de rutas de los routers. En general, las tablas de rutas de un host es estática y no necesita mantención.

El orden de la tabla de rutas es importante, sobre todo considerando CIDR. Es por esto que las redes más explícitas (con más bits para la red) deben ir primero, y cuando se encuentra un calce, se finaliza la búsqueda en la tabla.

Es importante notar que cada datagrama es independiente, y puede rutearse por caminos distintos, si cambia la tabla de rutas en el tiempo.

La tabla de rutas para el host H1 de la figura 2 sería la siguiente:

Net/mask	Gateway	Type
127.0.0.0/255.0.0.0	127.0.0.1	DIR
146.83.4.0/255.255.255.0	146.83.4.5	DIR
146.83.5.0/255.255.255.0	146.83.4.6	GW
(default) 0.0.0.0/0.0.0.0	146.83.4.1	GW

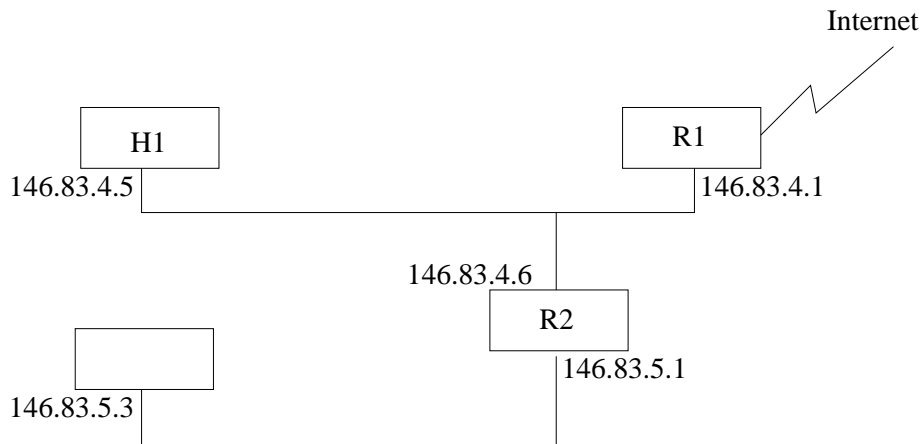


Figura 2: Ejemplo de red

El algoritmo básico se puede escribir como:

```
RouteIP(dgram, table)
{
    IPnet = getnet(dgram.destIP);
    Route = search(IPnet, table);
    if( Route.type == DIR )
        sendphys(dgram, dgram.destIP);
    else if (Route != NOT_FOUND)
        sendphys(dgram, Route.gateway);
    else {
        Route = search(default, table);
        if( Route != NOT_FOUND )
            sendphys(dgram, Route.gateway);
        else
            error(dgram, "Net Unreachable");
    }
}
```

7.1 Ruteo directo: ARP

El ruteo directo se refiere al caso en que dos hosts en la misma red IP quieren intercambiar información (no interviene ningún router). Los pasos a seguir son:

1. averiguar la dirección física correspondiente al host que tiene asignado el número IP de destino del paquete
2. encapsular el datagrama IP en un paquete físico
3. enviar el paquete físico a la dirección física del primer paso

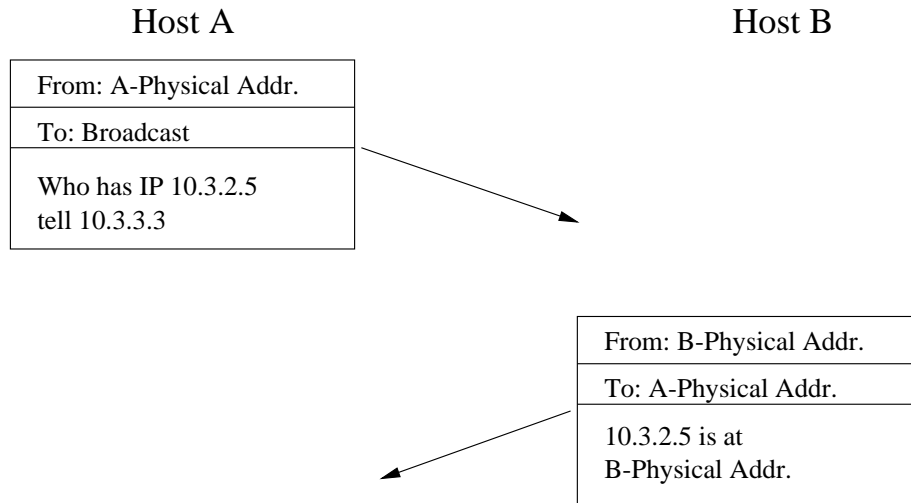


Figura 3: Esquema del ARP

Lo más complicado es la primera parte, y para resolver la traducción se utiliza muchas veces el protocolo *ARP* (Address Resolution Protocol), el cual es específico para la red. Para esto, se maneja un cache con un tiempo de expiración. Si el host de destino no se encuentra en el cache, se envía un broadcast preguntando por esa dirección (Figura 3). El host que tiene esa dirección IP responde y se agrega esa dirección física al cache.

El cache tiene un tiempo de expiración para permitir cambios en la red. Un host puede cambiarse a otra red o cambiar su número IP (ej: DHCP) y sería poco práctico tener que reiniciar todos los demás equipos para poder borrar la información antigua del cache.

En redes que no manejan broadcasting (ej: ATM) puede ser necesario tener un servidor de ARP con el cual cada host de la red se registra y luego hace las consultas necesarias.

7.2 Ruteo indirecto: propagación de las tablas de rutas

En el caso que un host que quiere enviar un datagrama a otro en una red distinta, le deberá entregar el datagrama a un router. Los routers usan exactamente el mismo algoritmo que los hosts para decidir cómo enviar el paquete, o sea que solamente deciden cuál va a ser el siguiente hop. Para poder tomar estas decisiones adecuadamente, se debe mantener la tabla de rutas de acuerdo a la realidad de la red, y para ello los routers se comunican entre sí y se actualizan la información.

7.2.1 Protocolos internos (IGP)

Los IGP o Interior Gateway Protocols son aquellos usados en una red que es administrada centralizadamente (hay una entidad que tiene control sobre toda la red). En este nivel, es posible conocer todos los caminos y calcular los óptimos. Los algoritmos más comunes son RIP (Routing Information Protocol) y OSPF (Open Shortest Path First), otros son IS-IS o E-IGRP. En general, estos protocolos no manejan políticas de ruteo, simplemente asumen que todo está conectado con todo.

- RIP:

A la información de la tabla de rutas se agrega una métrica, que se refiere a la cantidad de hops a la cual está el destino. Cada router comparte su tabla de rutas con todos sus vecinos directos. Si un router recibe una ruta a un destino que no conocía, la agrega a su tabla sumándole 1 a la métrica y definiendo el gateway como el router del cual recibió la información. Si un router recibe una mejor ruta hacia algún lugar desde un vecino, descarta su entrada antigua en la tabla de rutas (o mejor, la guarda más atrás por si llega a borrar la nueva), y agrega la nueva en su lugar. Este esquema se conoce como *Distance Vector Routing*.

RIP no es capaz de manejar bien las sub-redes, sobre todo las que tienen sub-redes de tamaño variable. Para superar esos problemas existe RIP2.

La información es transmitida cada cierto tiempo (30 segundos), y si la métrica es demasiado grande (> 16) o no se recibe nuevamente alguna ruta por más de un tiempo razonable, se marca el destino como inaccesible. La razón de esto radica en un problema que se conoce como “count to infinity” que se puede dar si alguna ruta se vuelve inaccesible, como se muestra en la figura 4.

Las métricas para los distintos routers hacia la red destino son:

Router	GW	Métrica
A	B	2
B	D	1
C	B	2
D	directo	0

Si se corta el enlace entre los routers B y D, la evolución de las rutas hacia la red destino en el tiempo es:

	t = 1	t = 2	t = 3	t = 4	...	t = 9	t = 10
D:	Direct 0	Dir 0	Dir 0	Dir 0	...	Dir 0	Dir 0
B:	Unreachable	C 3	C 4	C 5	...	C 11	C 12
C:	B 2	A 3	A 4	A 5	...	A 11	D 11
A:	B 2	C 3	C 4	C 5	...	C 11	C 12

Resultado: la convergencia es lenta. Existe una técnica llamada *split horizon* que consiste en nunca publicar una ruta hacia la interfaz desde la cual se aprendió. Con ese esquema, la misma situación de la figura 4 converge en 4 lapsos de tiempo. Una forma de mejorar esto aún más es publicar las rutas hacia las interfaces desde donde se aprendieron, pero con métrica infinita (*split horizon with poison*

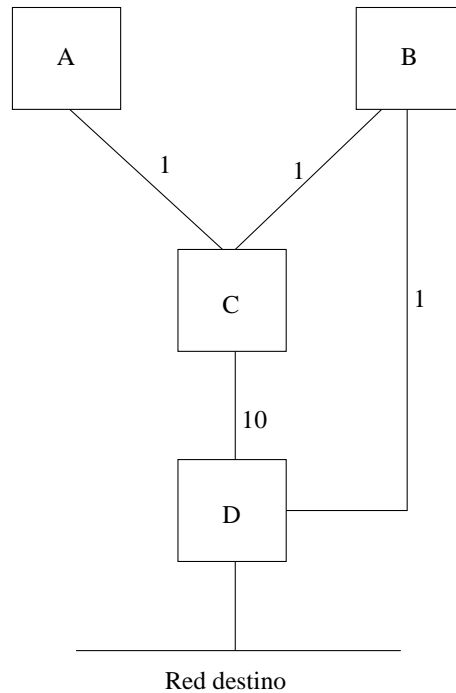


Figura 4: Ejemplo: count to infinity

reverse). Claro que esto último causa que las tablas de ruta que se intercambian sean más grandes.

Otro problema que tienen estos protocolos es que los hops en sí no significan mucho. Interesaría considerar costos arbitrarios para favorecer unas rutas sobre otras. La gran ventaja es que en RIP no es necesario configurar nada, porque se inicializa solo con la tabla de rutas estática de cada router, que contiene las redes a las cuales está conectado directamente.

- OSPF

Cada router conoce la topología completa de la red y calcula el camino óptimo para ir a cada punto de la red. Esto se hace siguiendo el algoritmo "Shortest Path First" (Dijkstra), creando un árbol con el nodo que lo calcula como raíz y los caminos más cortos a cada router dentro de la red. Con esto, puede garantizar rutas sin loops, pero obviamente requiere mucha más memoria y procesamiento que RIP. La información se intercambia en un algoritmo de inundación (flooding) cuando se dan los siguientes casos:

- un router descubre a un nuevo vecino
- el link con un vecino deja de funcionar
- el costo de un link cambia

- han pasado 30 minutos desde la última vez

Este tipo de protocolos se llaman *Link State Routing*, porque se envía información cada vez que cambia el estado de algún link. Una de las características de estos protocolos es que todos los routers tienen la misma información de la red, ya que se sincronizan apenas cambia algún detalle. Este protocolo permite particionar la red en áreas, mejorando la performance. El tamaño recomendado de un área es de unos 200 routers. Las áreas están unidas por un backbone por medio del cual se intercambia información inter-área. También soporta ruteo manejando TOS (type of service), load balancing y permite importar datos de los protocolos RIP y EGP.

7.2.2 Protocolos externos

Un sistema de redes que está bajo una administración común y usa un protocolo interno de ruteo se llama Sistema Autónomo (AS), y se le asigna un identificador (ASN). El ruteo externo se preocupa de compartir información entre routers de distintos sistemas autónomos (qué redes tiene tal o cual sistema autónomo). Al final, igual las tablas de rutas son por red IP, solamente el intercambio de información es por sistemas autónomos.

- EGP (Exterior Gateway Protocol)

EGP se basa en mensajes periódicos del estilo "Hola/Te escucho", para determinar acceso a AS vecinos y pedir información acerca de sus redes. Cada router publica sólo las redes contenidas dentro de su sistema autónomo, es decir que no se publican rutas que pasan por más de un AS. La información que publica un router vía EGP debe haber sido recopilada mediante algún IGP.

- BGP-4 (Border Gateway Protocol)

BGP-4 es el protocolo de preferencia. Sus características son:

- soporta CIDR
- routers se comunican usando TCP
- la información que se intercambia es una ruta como una secuencia de ASNs.

Si existe más de un router usando BGP dentro de un AS, debe garantizarse que provean la misma información (esto se logra usando un IGP como OSPF, o bien comunicando a los routers del mismo AS mediante BGP). Es necesario que los routers manejen dos tablas independientes: la que contiene las redes internas del AS y las que se aprenden de otros AS.

El algoritmo funciona como sigue:

- en un principio, al establecer una sesión BGP, dos routers intercambian su tabla de rutas completa.
- la información que se guarda es el camino que ha recorrido la entrada.

- a partir de ese momento, sólo se intercambian *updates* a esas tablas.
- se propagan las tablas que se aprenden hacia otros vecinos.
- si me llega un camino que incluye mi AS, no lo considero (loop).

Los problemas que aún presentan los protocolos de ruteo externo son:

- route flapping
- el tamaño de la tabla de rutas
- la administración de los ASs
- errores → loops

8 ICMP (Internet Control Message Protocol)

Los paquetes ICMP son usados por IP para detectar y avisar de errores, y permitir la depuración en caso de problemas. Además, para evitar empeorar alguna situación complicada, nunca se generan errores por paquetes de errores.

8.1 Detección de ciclos

Para la detección de los ciclos, se usa el campo TTL del header IP. Cada router lo decreta en al menos 1 al procesarlo, y si llega a 0 se descarta el paquete, generando un paquete "ICMP Time exceeded".

ICMP es una parte integral de IP, y por lo tanto no puede existir IP sin ICMP. En cambio, otros protocolos como TCP son absolutamente independientes de IP. Es por esto que ICMP, a pesar de viajar encapsulado como dato en IP, forma parte del protocolo IP y no es parte de la capa transporte.

8.2 Ejemplos frecuentes

- *Echo Request / Echo Reply*: usado en programas como ping, para determinar si es posible llegar a un cierto host. Un host que recibe un echo request envía como respuesta un echo reply. Se incluye un número de secuencia para diferenciar varios paquetes.
- *Destination unreachable*: se envía cuando el paquete IP no puede llegar al host de destino, o bien llega pero el protocolo que se especifica no está activo (o el puerto no está siendo escuchado). En el paquete se especifica cuál de las condiciones anteriores (u otra) es la que produjo el error.
- *Redirect*: un router detectó que el camino pasa por un router que era accesible directamente desde el host de origen y por lo tanto en el futuro se le debiera pasar directamente el paquete a ese router.

8.3 Traceroute

Traceroute es una aplicación que trata de determinar el camino que sigue un paquete IP desde el origen hasta un determinado destino. Para ello, la aplicación envía un paquete UDP con un port inaccesible y un TTL seteado inicialmente en 1. El primer router debe disminuir el campo TTL y al encontrarlo en 0 debe eliminar el paquete y (aunque no es obligatorio) envía un paquete ICMP señalando el problema. Con esto el origen detecta cuál es el primer router al que se le entrega el datagrama. Para determinar el resto se sigue el mismo esquema, seteando el TTL inicialmente en n para determinar el n -ésimo router. Como el port de destino es inaccesible, al llegar al destinatario final se recibe un paquete ICMP de tipo *Destination unreachable*.