Tema 4: Redes Conmutadas e Internet

Aclaración

Todo el contenido de estos apuntes viene de las diapositivas de la asignatura FR del profesor José Camacho Páez http://wdb.ugr.es/~josecamacho/ más las notas que he ido tomando en clase.

Contenido

1.Funcionalidades

Funciones y servicios en TCP/IP:

- Encaminamiento
- Conmutación
- Interconexión de redes
- En OSI: control de congestión

Ejemplos de protocolos de red:

- X.25
- IP

2. Comutación

Como la información navega por la red -> conmutación Hay tres estrategias:

- *Conmutación basada en circuitos*: se establece un circuito para la comunicación entre los dos finales. Siempre se mantiene ese circuito. Un

ejemplo es la red telefónica, solo su comunicación puede pasar por ahí.
- *Comunicación basada en paquetes*: la información se divide en paquetes y se envía por Internet. En el datagrama cada uno tiene su camino. En circuito virtual siempre siguen el mismo camino.

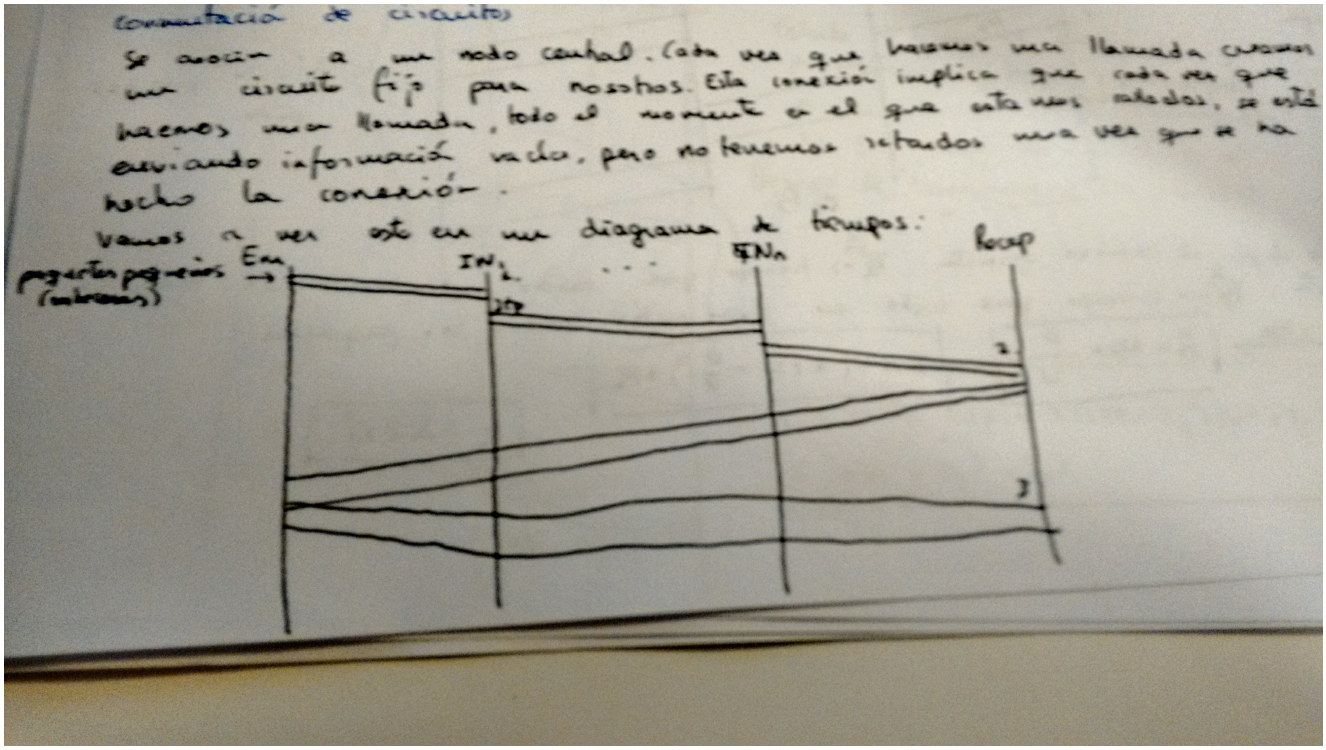
Conmutación de circuitos

Se asocia a un nodo central. Cada vez que hacemos una llamada creamos un circuito fijo para nosotros. Esta conexión implica que cada vez que hacemos una llamada, todo el momento en el que estamos callados, se está enviando información vacía, pero no tenemos retardos una vez que se ha hecho la conexión.

Pasos:

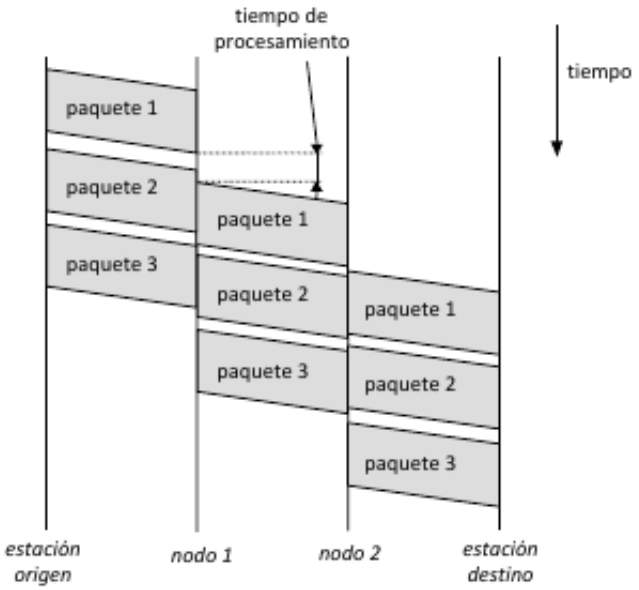
- *Conexión*
- *Transmisión*
- *Desconexión*

1. Ahora tiene que reservar recursos, debe ver si hay recursos disponibles. Esto es tiempo de procesamiento.
2. La comunicación ya es inmediata. El precio a pagar es que hemos dedicado todos esos recursos a la llamada. Por lo tanto necesitamos más cableado.
3. Todo es sin retardo, también la respuesta.



Conmutación de paquetes

La alternativa es la computación basada en datagramas. Es lo que utiliza IP. No exite la conexión. Todos los paquetes tienen retardo de procesamiento. Cada paquete lleva la dirección IP de destino. La conmutación por circuito es más rápida normalmente, pero por paquetes es más óptimo ocupar más cantidad de la red. Aquí no hay garantía de delay. Intentamos que el retardo por tiempo de procesamiento se minimice creando *circuitos digitales* por encima. Lo que reduce el tiempo de enrutamiento, ya que el nodo sabe a que circuito mandarlo. Pero de nuevo usamos un único camino y reducimos la robustez, ya que si se cae un enlace no puede elegir una nueva ruta.



3. El protocolo IP

Dentro del comportamiento de internet el protocolo IP es muy importante.

- Permite el direccionamiento en internet.
- La ide a de enviar los paquetes se un siitio a otro requiere el direccionamiento.
- Se parece a UDP
- La fragmentación(el tamaño máximo del paquete depende del hardware), para ello fragmentamos los paquetes.
- Retransmisión salto a salto entre hosts y routers.
- No orientado a conexión y no fiable: máximo esfuerzo.
- La unidad de datos(paquete) de IP se denomina datagrama.

Tenemos direcciones típicas de las IP. El nombre de dominio es con lo que trabajamos, pero a bajo nivel se trabaja con la IP. Cada dirección debe ser única, y le debemos decir al paquete el destino y como llegar.

Dos partes: subred y dispositivo

- *Máscara de red* -> sirve para determinar que parte de una dirección concreta indica a la subred y que parte al ordenador dentro de ella.

Si ponemos la máscara en binario, los 1 identifican a la subred y los 0 al ordenador dentro de ella.

➢ Dos partes: subred y dispositivo

a) Dirección IP ➔ 200.27.4.112 = 11001000.00011011.00000100.01110000

Máscara ➔ 255.255.255.0 = 11111111.11111111.11111111.00000000

b) 200.27.4.112/24

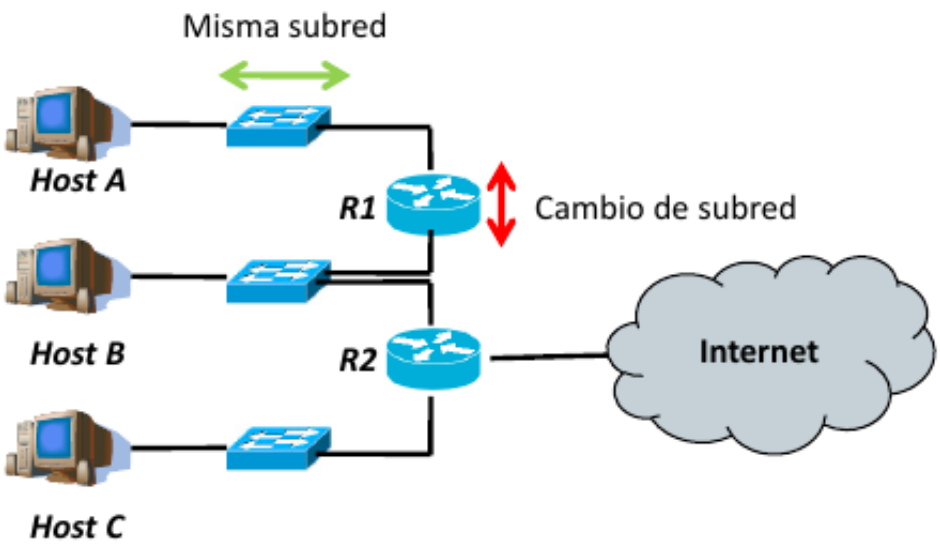
➢ Para obtener la dirección de la subred:

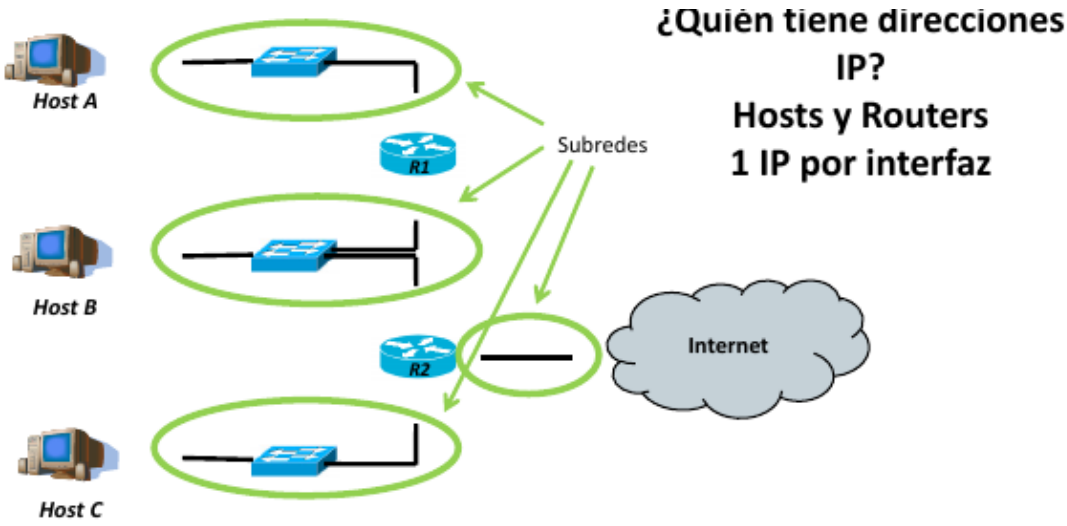
200.27.4.112 = 11001000.00011011.00000100.01110000
&
255.255.255.0 = 11111111.11111111.11111111.00000000

Subred ➔ 200.27.4.0 = 11001000.00011011.00000100.00000000

¿Qué es una subred?

El camino no hay que especificarlo a todos los ordenadores del mundo. En lugar se dice el camino para llegar a una subred, es lo que se mira para realizar el encaminamiento sobre eso. Se hace la operación AND entre la dirección IP y la máscara y así obtenemos la dirección de subred.





- ¿Cómo se elige la máscara? ➔ Según el número de dispositivos

Dirección IP ➔ 200.27.4.112 = 11001000.00011011.00000100.01110000

Máscara ➔ 255.255.255.0 = 11111111.11111111.11111111.00000000

dispositivos = $2^{\#\ ceros} - 2$

➔ ej. 8 ceros (/24) permite 254 dispositivos

➔ El -2 viene de que la primera y última son reservadas.

- 200.27.4.0 = 11001000.00011011.00000100.<u>00000000</u>

  ➔ Reservada (subred)

- 200.27.4.1 = 11001000.00011011.00000100.<u>00000001</u>

  ➔ Dispositivo #1

- ...

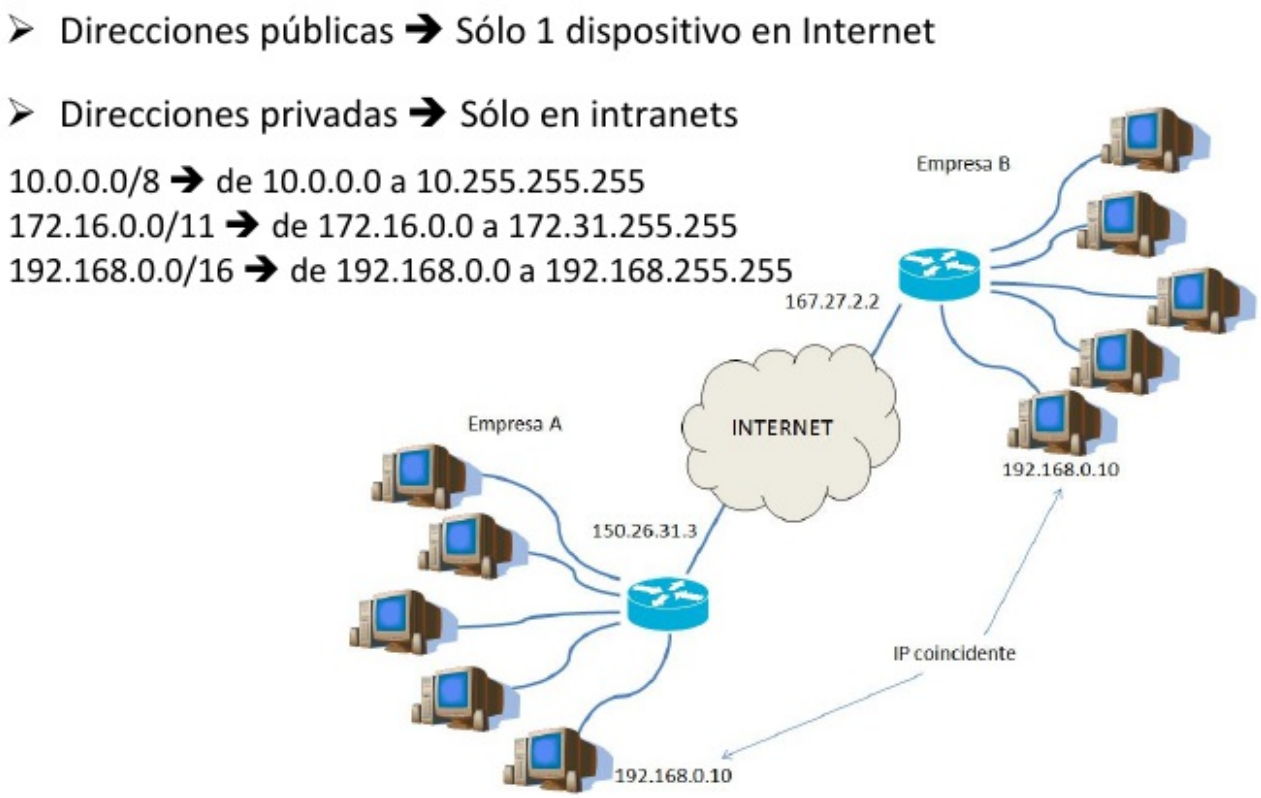
- 200.27.4.254 = 11001000.00011011.00000100.<u>11111110</u>

  ➔ Dispositivo #254

- 200.27.4.255 = 11001000.00011011.00000100.<u>11111111</u>

  ➔ Reservada (difusión)

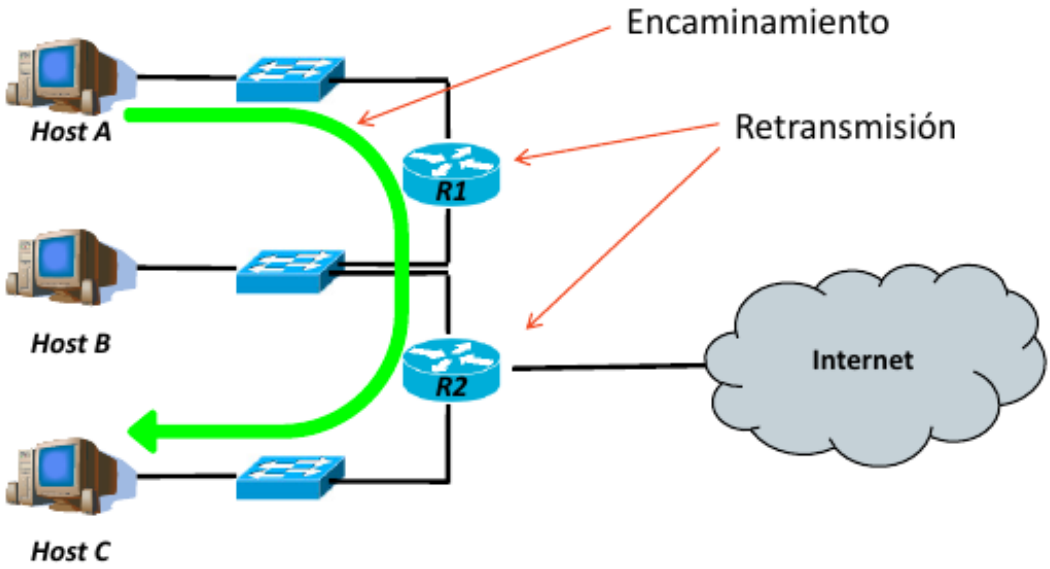
Todos los ordenadores estarán conectados a switches y ante los hosts se conectan con routers. El switch opera solo en capa de 2 y el router en cada red. Los dispositivos que no operan en capa de red, por lo que generan una frontera, pero los otros sí. Lo que me queda en verde son las subredes. Los host operan en capa de red ya que tienen dirección IP.



Las direcciones privadas si se pueden repetir dentro de una red. Todo ordenador para comunicarse en Internet debe tener una IP pública. Necesita esa IP pública para que sepan a donde enviarnos los paquetes. El router es el que nos da la IP pública. Este hace una operación NAT, lo que hace es que el router hace una traducción de la IP, con la pública. Cuando obtiene un paquete hace la operación inversa

El encaminamiento

- Llevar información (paquetes) de un origen a un destino en una red conmutada.
- Encaminamiento per sé(routing): decisión de rutas.
- Retransmisión(forwarding): operación básica en el dispositivo. Hacer los caminos de forma práctica.



Retranmisión salto-a-salto

Por donde se envia el paquete es equivalente a decir cómo se llega al siguiente nodo. Esto se hace en el nodo inicial, y en el resto de nodos.

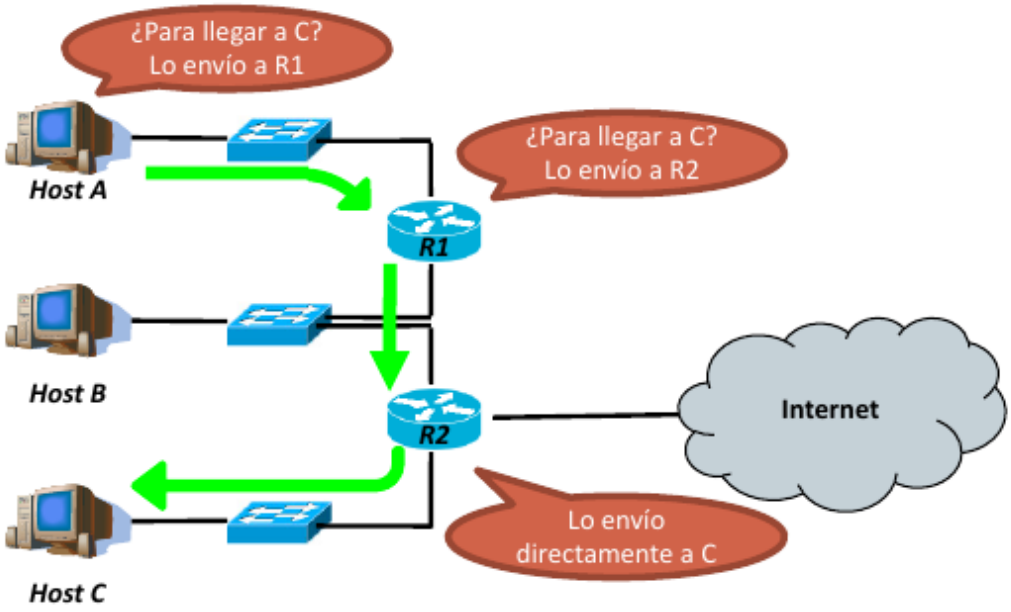


Tabla de encaminamiento

Para ello se utiliza la tabla de encaminamiento. La dirección de destino y la tabla de encaminamiento. Intentamos hacerla lo más pequeña posible. Siempre ponemos direcciones de subred y su máscara. Después va el siguiente nodo, si no tuviera iría "-". El host A y todos los nodos miran el paquete, la dirección de destino y la tabla de encaminamiento. Así sabes a que subred pertenecen, mirando la línea y viendo la dirección del siguiente nodo. Cuando aparece "-" se encarga la capa de enlace. Cada nodo tiene su tabla de encaminamiento.

El ordenador coge la dirección de destino y hace una and con la máscara. SI son

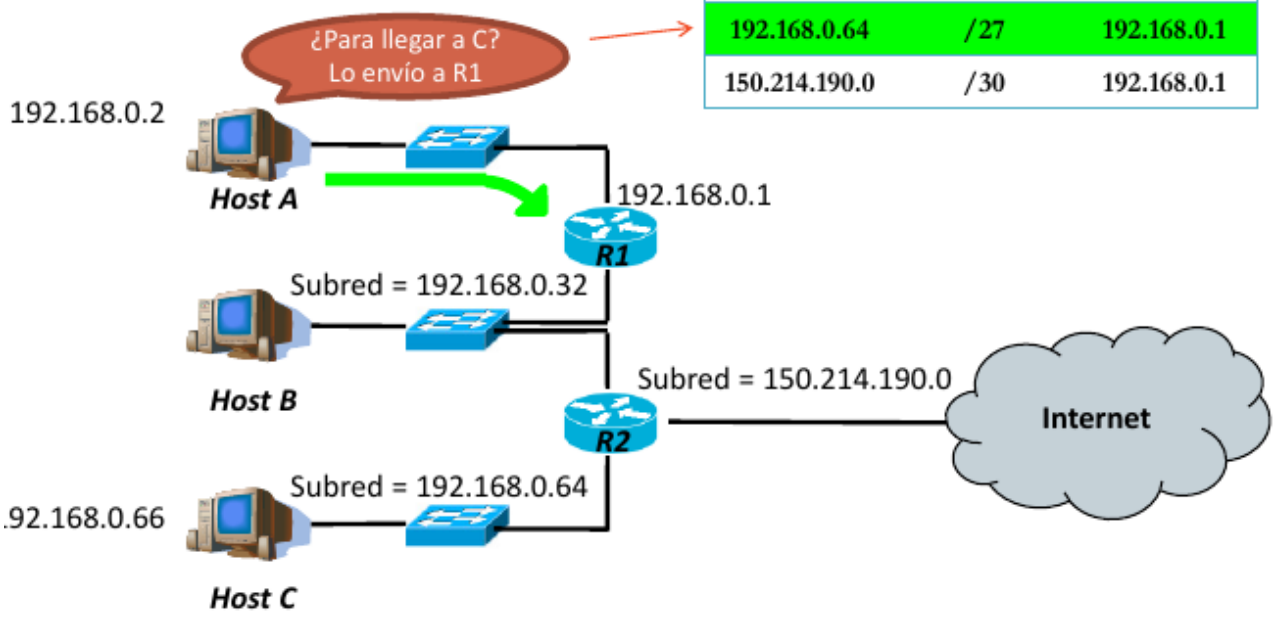
iguales, significa que es hacia donde tenemos que ir. Pueden ocurrir colisiones. Si hay una colisión se escoge la máscara más restrictiva. En la tabla hay que añadir la orden por defecto, que es por donde van a estar la mayoría de dispositivos en Internet. Con la máscara 0.0.0.0 siempre hago matching. La tabla con 2 entradas es la forma normal de configurarlo.

Los pasos al diseñar una tabla de encaminamiento son:

- Incorporar todas las redes directamente conectadas.
- Incorporar la entrada por defecto.
- Añadir todas las entradas adicionales necesarias.

➢ Tabla de encaminamiento:

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.32	/27	192.168.0.1
192.168.0.64	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1



➢ Tabla de encaminamiento:

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.32	/27	192.168.0.1
192.168.0.64	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

- ➢ Dirección de destino (DD):
  - ➢ 192.168.0.66
- ➢ Para cada entrada
  - ➢ DD & Máscara = A
  - ➢ ¿A = Dirección de destino?

Elegir el Siguiente Nodo

- Tabla de encaminamiento:

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.32	/27	192.168.0.1
192.168.0.64	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

  - Dirección de destino (DD):
    - 192.168.0.66
  - Primera entrada
    - 192.168.0.66 & /27 = 11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
    - ¿192.168.0.64 = 192.168.0.0? NO
  - Segunda entrada
    - 192.168.0.66 & /27 = 11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
    - ¿192.168.0.64 = 192.168.0.32? NO

- Tabla de encaminamiento:

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.32	/27	192.168.0.1
192.168.0.64	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

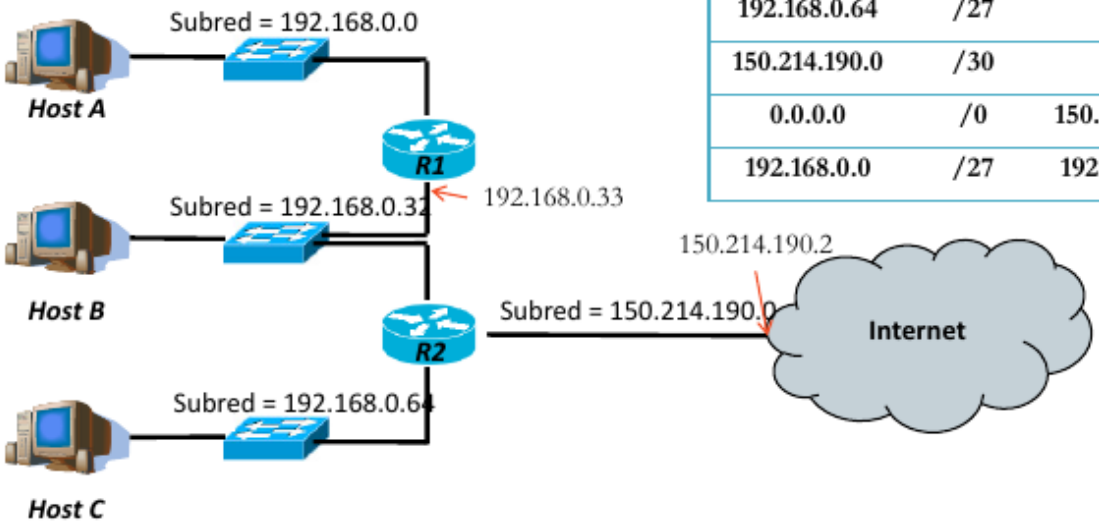
  - Dirección de destino (DD):
    - 192.168.0.66
  - Tercera entrada
    - 192.168.0.66 & /27 = 11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
    - ¿192.168.0.64 = 192.168.0.64? SÍ ➔ Siguiente Nodo = 192.168.0.1
  - Cuarta entrada
    - 192.168.0.66 & /30 = 11000000.10101000.00000000.01000**10** & /30 = 192.168.0.64
    - ¿192.168.0.64 = 150.214.190.0? NO
  - ¿Colisión? La de máscara más restrictiva (+ 1s)

i) Incorporar todas las redes directamente conectadas.

ii) Incorporar la entrada por defecto

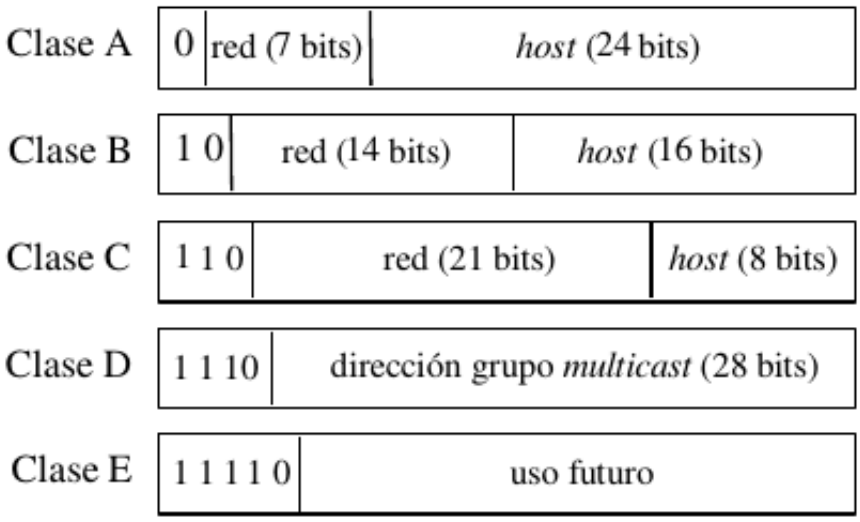
iii) Añadir todas las entradas adicionales necesarias.

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.32	/27	-
192.168.0.64	/27	-
150.214.190.0	/30	-
0.0.0.0	/0	150.214.190.2
192.168.0.0	/27	192.168.0.33



Direccionamiento basado en clases

Antes se usaba direccionamiento class-full, dependiendo de la clase pertenecia o no a un tipo de red. No existía la máscara de red.



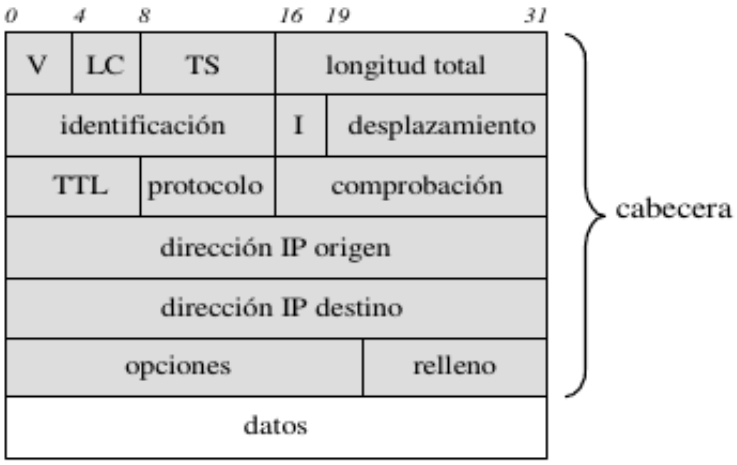
Formado de datagrama IP

- *V* -> versión de IP. La más extendida es IPv4, el futuro es IPv6. La diferencia es brutal, en IPv6 no hay una estructura fija para la cabecera. La IPv4 es más rígida. Está lo primero porque es fundamental para saber la sintáxis.
- *LC* -> longitud de la cabecera.
- *TS* -> se utiliza para informar de un contenido de mayor o menor calidad.
- *Longitud total* -> cabecera + datos. Los datos empiezan por TCP o UDP. Tiene 16 bits.

Los tres siguientes datos están orientados a la fragmentción de datagramas.

- *TTL* -> ("time to leave"). Tiempo que le damos al paquete para que haga su función. La implementación más extendida es un contador.
- *protocolo* -> lo que te dice que te vas a encontrar en datos.

Debajo tenemos las direcciones IP de origen y destino.



Fragmentación IPv4

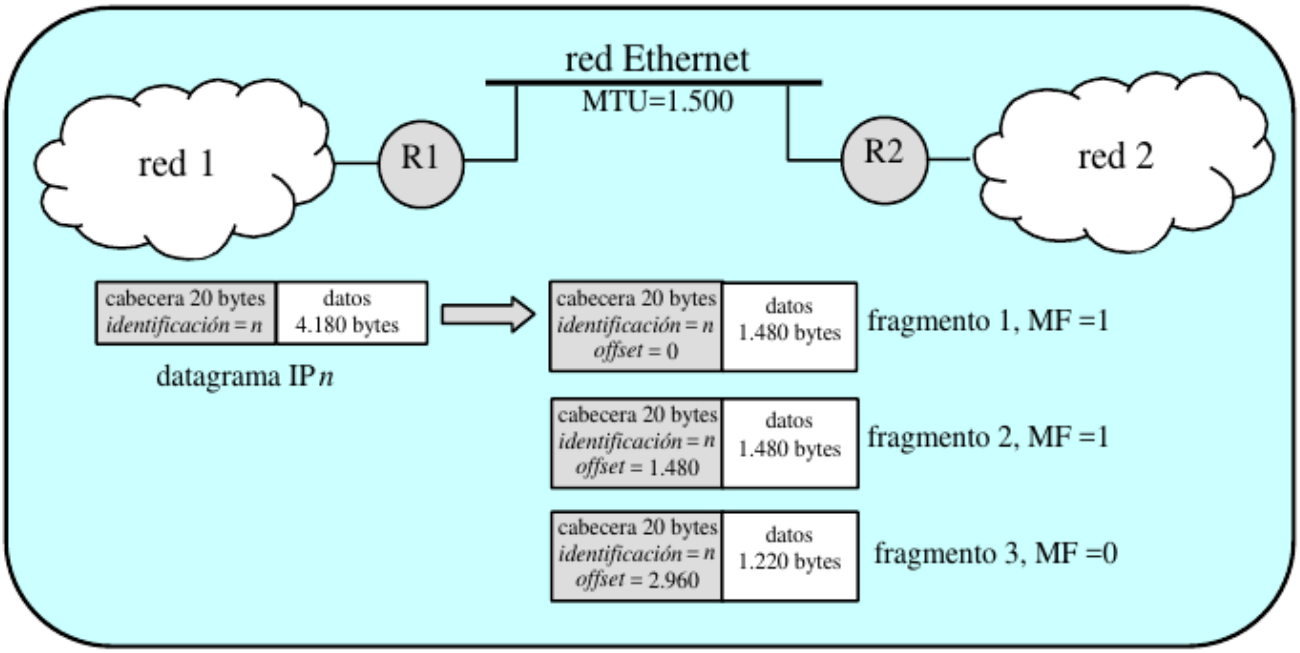
El tamaño máximo viene del tamaño de la cabecera. Para permitir que el datagrama llegue a su destino hay dos posibilidades:

- Fragmentamos en el router e entrada y lo enviamos, recomponiéndolo antes de llegar al router destino.
- O lo recomponemos al llegar al router destino.

En IPv4 se reensamblan en el destino final. Ya que ahí podemos estar seguros de que vana estar todos juntos.

Identificador es unívoco para el datagrama. Todos los fragmentos de un datagrama tienen el mismo.

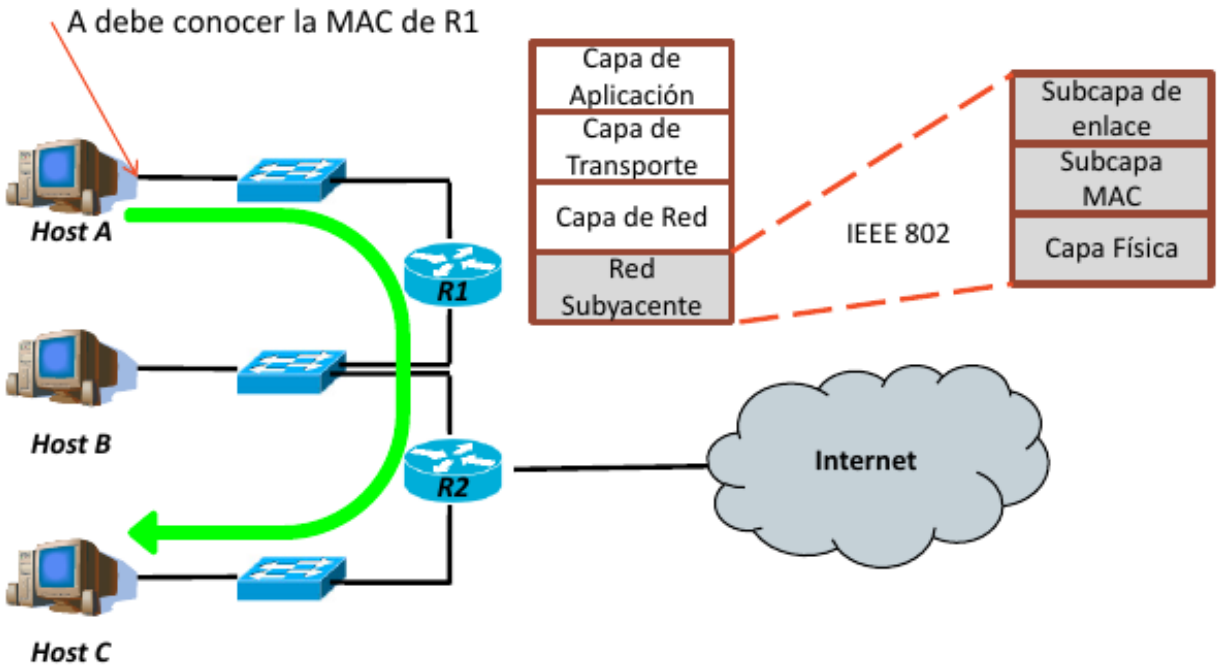
- *Indicadores (I)*:
    - "Don't fragment" -> que no se debe fragmentar. Si no puede pasar se tira.
    - "More fragments" -> destino al que ya le han llegado todos los fragmentos. El fragmento final lo tiene a 0. Espera a que se llenen sus huecos.
- *Desplazamiento*: nos dice a que hueco pertence.



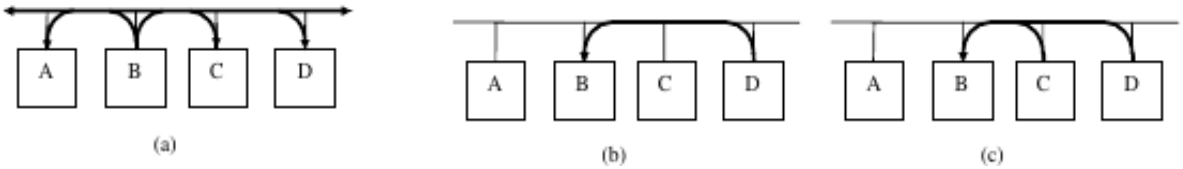
4. Asociación con Capa de Enlace: El protocolo ARP

Direcciones MAC

De alguna forma hay que pasar los paquetes a la capa de enlace(capa de enlace nodo a nodo). Para ello usa la dirección MAC, que identifica las tarjetas de red. Necesitamos un protocolo que traduzca de IP a MAC y viceversa, este es el protocolo **ARM**. El formato de la dirección MAC es en hexadecimal. El ARP directo consigue la MAC a partir de la IP. Esto no lo hace siempre, lo guarda en la caché.



Podemos obtener la MAC a partir de la IP (a y b). Y también obtener la IP a partir de MAC (a y c).



Ejemplo: B manda una petición ARP a la subred con un broadcast, diciendo quién tiene la dirección IP x. Y el que tiene esa dirección le responde con su dirección MAC.

Formato ARP

Se utiliza el formato en el que lo primero que se dice es el H (capa de enlace), P (capa de red). Se añade la MAC y la IP del emisor para que me responda y además ya de paso me de de alta en su caché.

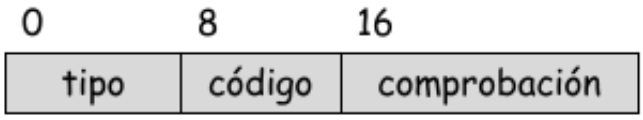
0	8	16	31
Htipo		Ptipo	
Hlen	Plen	Operación	
Hemisor (bytes 0-3)			
Hemisor (bytes 4-5)		Pemisor (bytes 0-1)	
Pemisor (bytes 2-3)		Hsol (bytes 0-1)	
Hsol (bytes 2-5)			
Psol (bytes 0-3)			

5. El protocolo ICMP

Es un protocolo de capa de red. Está encapsulado en TCP(y por lo tanto en IP), lo que significa que ICMP irá en el mismo sitio en el que va TCP(la cabecera va antes del otro protocolo).

Es un portocolo de gestión, informa de problemas. Siempre se manda el error al origen. El cheksum está en la cabeza y afecta a la cabecera más al contenido. COntiene una parte del paquete que ha causado el error.

- Cabecera de 32 bits
  - Tipo (8 bits): tipo de mensaje
  - Código (8 bits): subtipo de mensaje
  - Comprobación (16 bits)



- Mensajes ICMP:

Campo *tipo*	Mensaje ICMP
8/0	Solicitud/respuesta de eco
3	Destino inalcanzable
4	Ralentización del origen
5	Redireccionamiento
11	Tiempo de vida excedido
12	Problema de parámetros
13/14	Solicitud/respuesta de sello de tiempo
17/18	Solicitud/respuesta de máscara de red