

# Redes I

## Examen Integrador 06/2018

Emiliano Salvatori

Noviembre 2019

### 1. ¿Cuáles son las características de una Red de Circuitos Conmutados y una Red de Paquetes Conmutados?

Existen dos métodos fundamentales que permiten transportar los datos a través de una red de enlaces y conmutadores: la conmutación de circuitos y la conmutación de paquetes. En las **redes de conmutación de circuitos**, los recursos necesarios a lo largo de una ruta (buffers, velocidad de transmisión del enlace) que permiten establecer la comunicación entre los sistemas terminales están reservados durante el tiempo que dura la sesión entre dichos sistemas terminales. En las **redes de conmutación de paquetes**, estos recursos no están reservados; los mensajes de una sesión utilizan los recursos bajo petición y, en consecuencia, pueden tener que esperar (es decir, ponerse en cola) para poder acceder a un enlace de comunicaciones.

#### Conmutación de circuitos

La **conmutación de circuitos** es un tipo de conexión que realizan los diferentes nodos de una red para lograr un camino apropiado para conectar dos usuarios de una red de telecomunicaciones. A diferencia de lo que ocurre en la conmutación de paquetes, en este tipo de conmutación se establece un canal de comunicaciones dedicado entre dos estaciones. Se reservan recursos de transmisión y de conmutación de la red para su uso exclusivo en el circuito durante la conexión. Ésta es transparente: una vez establecida parece como si los dispositivos estuvieran realmente conectados.

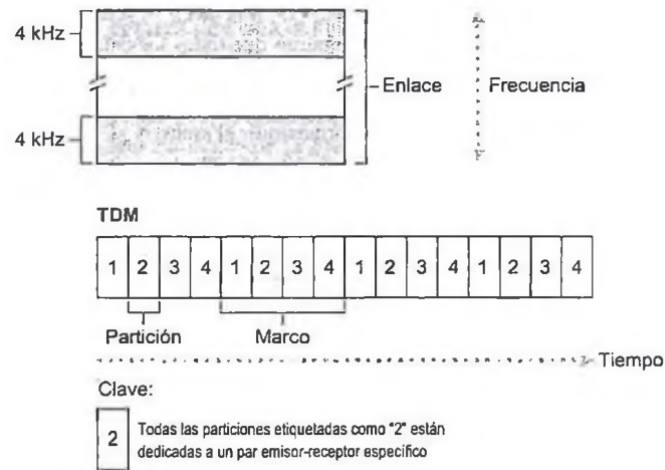
La comunicación por conmutación de circuitos implica tres fases: el establecimiento del circuito, la transferencia de datos y la desconexión del circuito. Una vez que el camino entre el origen y el destino queda fijado, queda reservado un ancho de banda fijo hasta que la comunicación se termine. Para comunicarse con otro destino, el origen debe primero finalizar la conexión establecida. Los nodos deben tener capacidad de conmutación y de canal suficiente como para gestionar la conexión solicitada; los conmutadores deben contar con la inteligencia necesaria para realizar estas reservas y establecer una ruta a través de la red.

En esta red, los cuatro conmutadores de circuitos están interconectados mediante cuatro enlaces. Cada uno de los enlaces tiene  $n$  circuitos, por lo que cada enlace puede dar soporte a  $n$  conexiones simultáneas. Cada uno de los hosts (por ejemplo, los PC y estaciones de trabajo) está conectado directamente a uno de los conmutadores. Cuando dos hosts desean comunicarse, la red establece una conexión terminal a terminal dedicada entre ellos. Por tanto, para que el host A envíe mensajes al host B, la red tiene que reservar en primer lugar un circuito en cada uno de los dos enlaces. Dado que cada enlace tiene  $n$  circuitos, para cada enlace utilizado por la conexión terminal a terminal, la conexión obtiene una fracción  $\frac{1}{n}$  del ancho de banda del enlace para el tiempo de duración de la conexión.

#### Multiplexación en Redes de Conmutación de Circuitos

Con **FDM**, el espectro de frecuencia de un enlace se reparte entre las conexiones establecidas a lo largo del enlace. Específicamente, el enlace dedica una banda de frecuencias a cada conexión durante el tiempo que ésta dure. En las redes telefónicas, esta banda de frecuencias normalmente tiene un ancho de 4 kHz (es decir, 4.000 hercios o 4.000 ciclos por segundo). El ancho de esta banda se denomina lógicamente ancho de banda. Las estaciones de radio FM también emplean la multiplexación FDM para compartir el espectro de frecuencias entre 88 MHz y 108 MHz, teniendo cada estación asignada una banda de frecuencias específica.

En un enlace **TDM**, el tiempo se divide en marcos de duración fija y cada marco se divide en un número fijo de particiones. Cuando la red establece una conexión a través de un enlace, la red dedica una partición de cada marco a dicha conexión. Estas particiones están dedicadas para uso exclusivo de dicha conexión con una partición disponible para utilizar (en cada marco) para transmitir los datos de la conexión.



## Conmutación de paquetes

Un paquete es un grupo de información que consta de dos partes: los datos propiamente dichos y la información de control, que indica la ruta a seguir a lo largo de la red hasta el destino del paquete. La **conmutación de paquetes** es un método de agrupar los datos transmitidos a través de una red digital en paquetes que se componen de un encabezado y una carga útil. Los datos en el encabezado son utilizados por el hardware de red para dirigir el paquete a su destino donde la carga útil es extraída y utilizada por el software de la aplicación. La conmutación de paquetes es la base principal de las comunicaciones de datos en redes informáticas de todo el mundo.

Las aplicaciones distribuidas intercambian mensajes para llevar a cabo sus tareas. Los mensajes pueden contener cualquier cosa que el diseñador del protocolo desee. Los mensajes pueden realizar una función de control (por ejemplo, los mensajes de saludo "Hola" del ejemplo anterior sobre establecimiento de la comunicación) o pueden contener datos, como por ejemplo un mensaje de correo electrónico, una imagen JPEG o un archivo de audio MP3. En las redes de computadoras modernas, el origen divide los mensajes largos en fragmentos de datos más pequeños que se conocen como paquetes. Entre el origen y el destino, cada uno de estos paquetes viaja a través de los enlaces de comunicaciones y de los conmutadores de paquetes (de los que existen dos tipos predominantes: los routers y los switches de la capa de enlace). Los paquetes se transmiten a través de cada enlace de comunicaciones a una velocidad igual a la velocidad de transmisión máxima del enlace.

La mayoría de los conmutadores de paquetes emplean el método de transmisión de almacenamiento y reenvío en las entradas de los enlaces. **Transmisión de almacenamiento y reenvío** significa que el conmutador tiene que recibir el paquete completo antes de poder comenzar a transmitir el primer bit del paquete al enlace de salida. Por tanto, los conmutadores de paquetes de almacenamiento y reenvío añaden un retardo de almacenamiento y reenvío en la entrada de cada enlace existente a lo largo de la ruta que debe seguir el paquete. Veamos el tiempo que se tarda en enviar un paquete de  $L$  bits desde un host a otro host en una red de conmutación de paquetes. Supongamos que existen  $Q$  enlaces entre los dos hosts, y que la velocidad en cada uno de ellos es igual a  $R$  bps. Suponemos que éste es el único paquete presente en la red. En primer lugar, el paquete tiene que enviarse a través del primer enlace que sale del host A, lo que consume un tiempo de  $\frac{L}{R}$  segundos. A continuación, tiene que ser transmitido por cada uno de los  $Q - 1$  enlaces restantes; es decir, se tiene que almacenar y reenviar  $Q - 1$  veces, añadiéndose cada vez un retardo de almacenamiento y reenvío de  $\frac{L}{R}$ . Por tanto, el retardo total es igual a  $\frac{Q \cdot R}{L}$ .

Cada conmutador de paquetes tiene varios enlaces conectados a él y para cada enlace conectado, el conmutador de paquetes dispone de un buffer de salida (también denominado cola de salida), que almacena los paquetes que el router enviará a través de dicho enlace. El buffer de salida desempeña un papel clave en la conmutación de paquetes. Si un paquete entrante tiene que ser transmitido a través de un enlace, pero se encuentra con que el enlace está ocupado transmitiendo otro paquete, el paquete entrante tendrá que esperar en el buffer de salida. Por tanto, además de los retardos de almacenamiento y reenvío, los paquetes se ven afectados por los retardos de cola del buffer de salida. Estos retardos son variables y dependen del nivel de congestión de la red. Puesto que la cantidad de espacio en el buffer es finita, un paquete entrante puede encontrarse con que el buffer está completamente lleno con otros paquetes que esperan a ser transmitidos. En este caso, se producirá una pérdida de paquetes, bien el paquete que acaba de llegar o uno que ya se encuentra en la cola será descartado.

## 2. Dado un paquete de longitud igual a 2000bytes, que se transmite a través de un enlace a una distancia de 2500km, siendo la velocidad de propagación de $2,5 \cdot 10^8$ m/s y la velocidad de transmisión de 2Mb/s

Suponiendo que los paquetes se transmiten de manera que el primero que llega es el primero que sale, lo que es una práctica común en las redes de conmutación de paquetes, nuestro paquete sólo puede ser transmitido después de que todos los paquetes que hayan llegado antes que él hayan sido transmitidos. Sea la longitud del paquete igual a  $L$  bits y la velocidad de transmisión del enlace del router A hasta el router B igual a  $R$  bits/segundo. Entonces, por ejemplo, para un enlace Ethernet a 10 Mbps, la velocidad es  $R = 10$  Mbps; para un enlace Ethernet a 100 Mbps, la velocidad será  $R = 100$  Mbps. Por lo que:

$$R_T = \frac{L}{R}$$

**Calcule el retardo de transmisión. ¿Depende este retardo de la longitud del paquete?**

Siendo  $R_T$  el **Retardo de Transmisión**,  $L$  la **Longitud del paquete**, y  $R$  la **Velocidad de Transmisión**, entonces se tiene:

$$R_T = \frac{2000b}{2048b/s} \approx 0,98s$$

En el **Retardo de Transmisión** depende directamente de la longitud del paquete que se esté tratando de transmitir, ya que el paquete será transmitido sólo después de que todos los paquetes que hayan llegado antes que él hayan sido transmitidos.

**Calcule el retardo de propagación. ¿Depende este retardo de la longitud del paquete? ¿y de la velocidad de transmisión?**

Una vez que un bit ha entrado en el enlace, tiene que propagarse hasta el router B. El tiempo necesario para propagarse desde el principio del enlace hasta el router B es el **retardo de propagación**. El bit se propaga a la velocidad de propagación del enlace. Esta velocidad depende del medio físico del enlace (es decir, que el medio sea cable de fibra óptica, cable de cobre de par trenzado, etc.) y es igual o menor a la velocidad de la luz.

El **Retardo de propagación** es igual a la distancia entre dos routers dividida entre la velocidad de propagación.

Siendo  $R_P$  el **Retardo de propagación**,  $D$  la **Distancia** entre dos routers y  $V_p$  la **Velocidad de Propagación** entonces se tiene:

$$R_P = \frac{D}{V_p} = \frac{2500000m}{2,5 \cdot 10^8 m/s} \approx 9259,25s$$

En el **Retardo de Propagación** NO importa la longitud del paquete, sino que lo que importa es el medio físico del enlace y la distancia entre los dos dispositivos (routers en este caso) que tienen que intercambiar los paquetes.

## 3. ¿Cuáles son las características principales de la Capa de Red?

La función de la capa de red es por tanto tremendamente simple: transporta paquetes desde un host emisor a un host receptor. En la realización de esta tarea podemos identificar dos importantes funciones de la capa de red:

- **Reenvío (forwarding)**: Cuando un paquete llega al enlace de entrada de un router, éste tiene que pasar el paquete al enlace de salida apropiado. Por ejemplo, un paquete que llega procedente de H1 al router R1 debe ser reenviado al siguiente router de la ruta hacia H2.

**Reenvío/Forwarding**: Tiene que ver con lo anterior, es mover paquetes desde una entrada del router a la salida del mismo. Cuando un paquete llega al enlace de entrada de un router, éste tiene que pasar el paquete al enlace de salida apropiado.

- **Enrutamiento (routing)**: La capa de red tiene que determinar la ruta o camino que deben seguir los paquetes a medida que fluyen de un emisor a un receptor. Los algoritmos que calculan estas rutas se

conocen como algoritmos de enrutamiento. Un algoritmo de enrutamiento debe determinar, por ejemplo, la ruta por la que fluirán los paquetes para ir de H1 a H2.

**Ruteo/Enrutamiento/Routing:** Determina una ruta de una punta a la otra, desde origen a destino. La capa de red tiene que determinar la ruta o camino que deben seguir los paquetes a medida que fluyen de un emisor a un receptor. Los algoritmos que calculan estas rutas se conocen como algoritmos de enrutamiento. Un algoritmo de enrutamiento debe determinar, por ejemplo, la ruta por la que fluirán los paquetes para ir de un Host situado en la ciudad/país A, hasta otro situado en la ciudad/país B.

## Ejemplo práctico de Reenvío y Enrutamiento

Un paralelismo con lo anterior se puede poner cuando un usuario quiere realizar un viaje en auto por el país. Decidir la ruta que se tomará para llegar a determinada provincia desde la casa del usuario sería el *enrutamiento*. En cambio, llegado a una ciudad (que sería para el ejemplo como ser un router), decidir por qué calle tomar para dar con la ruta que me llevará al próximo pueblo sería el *reenvío*.

El reenvío hace referencia a la acción local que realiza un router al transferir un paquete desde una interfaz de un enlace de entrada a una interfaz del enlace de salida apropiada. El enrutamiento hace referencia al proceso que realiza la red en conjunto para determinar las rutas terminal a terminal que los paquetes siguen desde el origen al destino.

Para saber la mejor ruta los routers corren algoritmos que van a determinar la mejor ruta para ir de un host a otro, una vez que termina el algoritmo genera una tabla de reenvío, esa tabla está conformada como si fuera un algoritmo de Dijkstra. Evalúa dándole determinado peso entre routers.

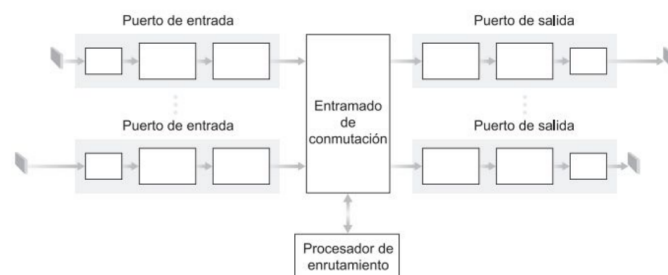
Todo router tiene una tabla de reenvío. Un router reenvía un paquete examinando el valor de un campo de la cabecera del paquete entrante y utilizando después ese valor para indexarlo dentro de la tabla de reenvío del router. El resultado de la tabla de reenvío indica a cuál de las interfaces del enlace de salida del router será reenviado el paquete. Dependiendo del protocolo de la capa de red, este valor de la cabecera del paquete podría ser la dirección de destino del paquete o una indicación de la conexión a la que pertenece el paquete.

Una vez que el router pasa el paquete a otro, se olvida del paquete que envió. Si cada router corre el mismo algoritmo proporcionado por el protocolo que se ejecute, todos llegan a la misma conclusión de que la mejor ruta es una, y en base a ello es que se genera la tabla de reenvío; la problemática es que todos corran el mismo algoritmo y saber cuál usar.

Cada router corre el algoritmo, genera la tabla, determina la mejor ruta, envía los paquetes; los algoritmos se corren cada determinado tiempo, para que se actualice la tabla de ruteo; y todo esto se denomina *Ruteo dinámico*. *Ruteo estático*: es cuando se define vía hardware por dónde debe salir un paquete determinado.

## 4. ¿Cuáles son las funciones de un router? Realice un diagrama de bloques de un Router y describa sus módulos.

La función principal de un router es la *transferencia real de paquetes desde los enlaces de entrada de un router a los apropiados enlaces de salida*.



En un router se pueden identificar cuatro componentes:

1. **Puertos de entrada:** El puerto de entrada realiza varias funciones. Lleva a cabo las funciones de la capa física (representadas por el recuadro situado más a la izquierda del puerto de entrada y el recuadro más a la derecha del puerto de salida en la Figura) consistentes en la terminación de un enlace físico de entrada a un router. Realiza las funciones de la capa de enlace de datos (representadas por los recuadros centrales de los puertos de entrada y de salida) necesarias para interoperar con las funciones de la capa de enlace de datos en el lado remoto del enlace de entrada. También realiza una función de búsqueda y reenvío (el recuadro más a la derecha del puerto de entrada y el recuadro más a la izquierda del puerto de salida)

de modo que un paquete reenviado dentro del entramado de conmutación del router emerge en el puerto de salida apropiado. Los paquetes de control (por ejemplo, paquetes que transportan la información del protocolo de enrutamiento) son reenviados desde un puerto de entrada al procesador de enrutamiento. En la práctica, suelen agruparse varios puertos en una única tarjeta de línea (line card) dentro del router.

2. **Entramado de conmutación:** El entramado de conmutación conecta los puertos de entrada del router a sus puertos de salida. Este entramado de conmutación está completamente contenido dentro del router.
3. **Puertos de salida:** Un puerto de salida almacena los paquetes que le han sido reenviados a través del entramado de conmutación y los transmite al enlace de salida. Así, el puerto de salida lleva a cabo la función inversa de la capa física y de la capa de enlace de datos que el puerto de entrada. Cuando un enlace es bidireccional (es decir, transporta tráfico en ambas direcciones), un puerto de salida del enlace normalmente estará emparejado con otro puerto de entrada de dicho enlace en la misma tarjeta de línea.
4. **Procesador de enrutamiento:** El procesador de enrutamiento ejecuta los protocolos de enrutamiento mantiene la información de enrutamiento y las tablas de reenvío y realiza funciones de gestión de red dentro del router.

## 5. ¿Para qué se utiliza el comando Tracert? ¿Cómo se utiliza?

El comando **tracert** es un comando del símbolo del sistema que se utiliza para mostrar varios detalles sobre la ruta que toma un paquete desde el equipo o dispositivo en el que se encuentra hasta el destino que se especifique.

Tracert determinará la ruta llevada a un destino. Para ello, envía mensajes de solicitud de eco del Protocolo de mensajes de control de Internet (ICMP) al destino. Al enviar el tráfico al destino, aumentará de manera creciente los valores del campo Tiempo de vida (TTL) para ayudar a encontrar la ruta a esa dirección de destino.

El comando **tracert**, o **tracert** es similar al comando **ping**, pero este proporciona información sobre la ruta que toma un paquete. Tracert envía paquetes a un destino, pidiendo a cada enrutador de Internet que responda cuando pase el paquete. Esto le mostrará los paquetes de ruta que toma cuando los envía entre su ubicación y un destino.

### Ejemplo

```
tracert 192.168.1.1
```

En el ejemplo anterior, el comando **tracert** se usaba para mostrar la ruta desde la computadora en red en la cual el comando **tracert** está siendo ejecutado por un dispositivo de red, en este caso, un router en una red local, que tiene asignada la dirección IP 192.168.1.1.

## 6. Detalle las características del protocolo IP. ¿A qué capa pertenece? ¿qué tipos de campos y/o servicios hay en el formato de un datagrama?

El protocolo de la **Capa de Red** de Internet es el **Protocolo IP (Internet Protocol)**. IP proporciona una comunicación lógica entre hosts. El modelo de servicio de IP es un servicio de entrega de mejor esfuerzo (best effort). Esto quiere decir que IP hace todo lo que puede por entregar los segmentos entre los hosts que se están comunicando, pero no garantiza la entrega. En particular, no garantiza la entrega de los segmentos, no garantiza que los segmentos se entreguen en orden y no garantiza la integridad de los datos contenidos en los segmentos. Por estas razones, se dice que IP es un servicio no fiable. Además, sabemos que todos los hosts tienen al menos una dirección de capa de red, que se conoce como dirección IP. Lo único que necesitamos saber es que todo host tiene una dirección IP asociada.

Los campos clave de los datagramas IPv4 son los siguientes:

- **Número de versión:** Estos 4 bits especifican la versión del protocolo IP del datagrama. A partir del número de versión el router puede determinar cómo interpretar el resto del datagrama IP. Las distintas versiones de IP utilizan distintos formatos de datagrama.
- **Longitud de cabecera:** Puesto que un datagrama IPv4 puede contener un número variable de opciones (las que se incluyen en la cabecera del datagrama IPv4), estos 4 bits son necesarios para determinar dónde comienzan realmente los datos del datagrama IP. La mayoría de los datagramas IP no contienen opciones, por lo que el datagrama IP típico tiene una cabecera de 20 bytes.

- **Tipo de servicio:** Los bits del tipo de servicio (TOS, Type of service) se incluyeron en la cabecera de IPv4 con el fin de poder diferenciar entre distintos tipos de datagramas IP (por ejemplo, datagramas que requieran en particular un bajo retardo, una alta tasa de transferencia o una entrega fiable). Por ejemplo, puede resultar útil diferenciar datagramas en tiempo real (como los utilizados en aplicaciones de telefonía IP) del tráfico que no es en tiempo real (como por ejemplo el tráfico FTP). El nivel específico de servicio que se proporcione es una política que determinará el administrador del router.
- **Longitud del datagrama:** Es la longitud total del datagrama IP (la cabecera más los datos) en bytes. Puesto que este campo tiene una longitud de 16 bits, el tamaño máximo teórico del datagrama IP es de 65.535 bytes. Sin embargo, los datagramas rara vez tienen una longitud mayor de 1.500 bytes.
- **Identificador, indicadores, desplazamiento de fragmentación:** Estos tres campos tienen que ver con lo que se denomina fragmentación IP. Es interesante comentar que la nueva versión de IP, IPv6, NO permite la fragmentación en los routers.
- **Tiempo de vida:** El campo Tiempo de vida (TTL, Time-To-Live) se incluye con el fin de garantizar que los datagramas no estarán eternamente en circulación a través de la red (debido, por ejemplo, a un bucle de enrutamiento de larga duración). Este campo se decrementa en una unidad cada vez que un router procesa un datagrama. Si el campo TTL alcanza el valor 0, el datagrama tiene que ser descartado.
- **Protocolo:** Este campo sólo se emplea cuando un datagrama IP alcanza su destino final. El valor de este campo indica el protocolo específico de la capa de transporte al que se pasarán los datos contenidos en ese datagrama IP. Por ejemplo, un valor de 6 indica que los datos se pasan a TCP, mientras que un valor igual a 17 indica que los datos se pasan a UDP. Puede obtener una lista de todos los valores posibles en [IANA Protocol Numbers 2009]. Observe que el número de protocolo especificado en el datagrama IP desempeña un papel análogo al del campo que almacena el número de puerto de un segmento de la capa de transporte. El número de protocolo es el elemento que enlaza las capas de red y de transporte, mientras que el número de puerto es el componente que enlaza las capas de transporte y de aplicación. En el Capítulo 5 veremos que la trama de la capa de enlace también contiene un campo especial que enlaza la capa de enlace con la capa de red.
- **Suma de comprobación de cabecera:** La suma de comprobación de cabecera ayuda a los routers a detectar errores de bit en un datagrama IP recibido. Esta suma de comprobación se calcula tratando cada pareja de 2 bytes de la cabecera como un número y sumando dichos números utilizando aritmética de complemento a 1. Como se ha visto en la Sección 3.3, el complemento a 1 de esta suma, conocida como suma de comprobación Internet, se almacena en el campo Suma de comprobación. Un router calcula la suma de comprobación de cabecera para cada datagrama IP recibido y detecta una condición de error si la suma de comprobación incluida en la cabecera del datagrama no coincide con la suma de comprobación calculada. Normalmente, los routers descartan los datagramas en los que se ha detectado que existe un error. Observe que la suma de comprobación tiene que volver a calcularse y almacenarse en cada router, ya que el campo TTL y, posiblemente, también el campo de opciones pueden cambiar. Una cuestión que suele plantearse en este punto es ¿por qué TCP/IP lleva a cabo una comprobación de errores tanto en la capa de transporte como en la capa de red? Existen varias razones para esta redundancia. En primer lugar, fíjese en que en la capa IP sólo se calcula la suma de comprobación para la cabecera IP, mientras que la suma de comprobación TCP/UDP se calcula sobre el segmento TCP/UDP completo. En segundo lugar, TCP/UDP e IP no necesariamente tienen que pertenecer a la misma pila de protocolos. En principio, TCP puede ejecutarse sobre un protocolo diferente (por ejemplo, ATM) e IP puede transportar datos que no se pasarán a TCP/UDP.
- **Direcciones IP de origen y de destino:** Cuando un origen crea un datagrama, inserta su dirección IP en el campo de dirección IP de origen e inserta la dirección del destino final en el campo de dirección IP de destino. A menudo el host de origen determina la dirección de destino mediante una búsqueda DNS.
- **Opciones:** El campo de opciones permite ampliar una cabecera IP. La idea original era que las opciones de cabecera rara vez se emplearan: de ahí la decisión de ahorrar recursos no incluyendo la información de los campos opcionales en la cabecera de todos los datagramas. Sin embargo, la mera existencia de opciones complica las cosas, ya que las cabeceras de datagrama pueden tener una longitud variable, por lo que no puede determinarse a priori dónde comenzará el campo de datos. Además, dado que algunos datagramas pueden requerir el procesamiento de opciones y otros no, la cantidad de tiempo necesario para procesar un datagrama IP en un router puede variar enormemente. Estas consideraciones cobran una particular importancia en el procesamiento IP realizado en los hosts y routers de altas prestaciones. Por estas razones y otras, las opciones IP fueron eliminadas en la cabecera de IPv6.

- **Datos (carga útil):** Finalmente, llegamos al último campo y más importante. En la mayoría de las circunstancias, el campo de datos del datagrama IP contiene el segmento de la capa de transporte (TCP o UDP) que va a entregarse al destino. Sin embargo, el campo de datos puede transportar otros tipos de datos, como por ejemplo mensajes ICMP.

## 7. Suponga un MTU de un cierto enlace está limitado a 1500bytes. Suponiendo una cabecera IP de 20 bytes ¿Cuántos datagramas se necesitarían para enviar un archivo MP3 de 5 millones de bytes?

Si el enlace está limitado a 1500 bytes y 20 de ellos serán de la cabecera IP, entonces se tiene:

- **Bytes de datos:** 1480
- **Bytes de Cabecera:** 20
- **Total de Bytes:** 1500

Si se requiere pasar cinco millones de bytes, se tiene  $L_a$  la **Longitud del archivo** y  $B_c$  la cantidad de **Bytes de datos**:

$$\frac{L_a}{B_d} = \frac{5000000}{1480} \approx 3378,37$$

Se debe quedar con la parte entera del resultado, es decir 3378 y se debe sacar el resto haciendo:

$$L_a - (C_p * B_d) = 5000000 - (3378 * 1480) = 560$$

Entonces se tienen la cantidad de paquetes:

- 3378 paquetes de 1500 bytes con 20 bytes de cabecera
- 1 paquete de 580 bytes, siendo 560 de datos + 20 de cabecera

La tabla de *offset* se generará de la siguiente forma:

- Paquete n° 1 - *offset*: 0
- Paquete n° 2 - *offset*: 1480
- Paquete n° 3 - *offset*: 2960
- ...
- Paquete n° 3379 - *offset*: 3378

**Respuesta:** Se tendrá que dividir el archivo en 3379 paquetes: 3378 de 1500 bytes y 1 de 580.

## 8. ¿Qué algoritmos de enrutamiento conoce? Detalle sus principios de funcionamiento

En términos generales, una forma de clasificar los algoritmos de enrutamiento es dependiendo de si son **globales** o **descentralizados**:

- **Un algoritmo de enrutamiento global** calcula la ruta de coste mínimo entre un origen y un destino utilizando el conocimiento global y completo acerca de la red. Es decir, el algoritmo toma como entradas la conectividad entre todos los nodos y todos los costes de enlace. Esto requiere por tanto que el algoritmo de alguna forma obtenga esta información antes de realizar realmente el cálculo. El cálculo en sí puede hacerse en un sitio (un algoritmo de enrutamiento global centralizado) o replicarse en varios sitios. La característica distintiva aquí, sin embargo, es que un algoritmo global dispone de toda la información acerca de la conectividad y de los costes de los enlaces. En la práctica, los algoritmos con información de estado global a menudo se denominan algoritmos de **Estado de Enlaces (LS, Link-State)**, ya que el algoritmo tiene que ser consciente del coste de cada enlace de la red.

- En un algoritmo de **enrutamiento descentralizado**, el cálculo de la ruta de coste mínimo se realiza de manera iterativa y distribuida. Ningún nodo tiene toda la información acerca del coste de todos los enlaces de la red. En lugar de ello, al principio, cada nodo sólo conoce los costes de sus propios enlaces directamente conectados. Después, a través de un proceso iterativo de cálculo e intercambio de información con sus nodos vecinos (es decir, los nodos que están en el otro extremo de los enlaces a los que él mismo está conectado), cada nodo calcula gradualmente la ruta de coste mínimo hacia un destino o conjunto de destinos. El algoritmo de enrutamiento descentralizado que estudiaremos se denomina algoritmo de **vector de distancias (DV, Distance-Vector)**, porque cada nodo mantiene un vector de estimaciones de los costes (distancias) a todos los demás nodos de la red.

### Algoritmo de enrutamiento por Vector de Distancias (DV)

Mientras que el algoritmo LS es un algoritmo que emplea información global, el algoritmo por Vector de Distancias (DV) es iterativo, asincrónico y distribuido.

- Es **distribuido** en el sentido de que cada nodo recibe información de uno o más de sus vecinos directamente conectados, realiza un cálculo y luego distribuye los resultados de su cálculo de vuelta a sus vecinos.
- Es **iterativo** porque este proceso continúa hasta que no hay disponible más información para ser intercambiada entre los vecinos. (Además, el algoritmo también finaliza por sí mismo, es decir, no existe ninguna señal que indique que los cálculos deberían detenerse; simplemente se detienen).
- El algoritmo es **asincrónico**, en el sentido de que no requiere que todos los nodos operen sincronizados entre sí.

Como tendremos oportunidad de ver, un algoritmo asincrónico, iterativo, distribuido y que finaliza por sí mismo es mucho más interesante y divertido que un algoritmo centralizado.

### Algoritmo de enrutamiento de Estado de Enlaces (LS)

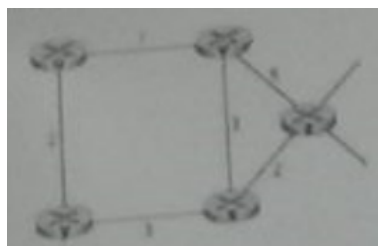
Recuerde que en un algoritmo de Estado de Enlaces, la topología de la red y el coste de todos los enlaces **son conocidos**; es decir, están disponibles como entradas para el algoritmo LS.

En la práctica, esto se consigue haciendo que cada nodo difunda paquetes del estado de los enlaces a todos los demás nodos de la red, con cada paquete de estado de enlace conteniendo las identidades y los costes de sus enlaces conectados. En la práctica (por ejemplo, con el protocolo de enrutamiento OSPF de Internet), esto suele conseguirse mediante un algoritmo de difusión de estado de enlaces.

El resultado de difundir la información de los nodos es que todos los nodos tienen una visión completa e idéntica de la red. Cada nodo puede entonces ejecutar el algoritmo LS y calcular el mismo conjunto de rutas de coste mínimo que cualquier otro nodo.

El algoritmo de enrutamiento de Estado de Enlaces que presentamos a continuación se conoce como algoritmo de Dijkstra, en honor a su inventor. El algoritmo de Dijkstra calcula la ruta de coste mínimo desde un nodo (el origen, al que denominaremos u) hasta todos los demás nodos de la red. El algoritmo de Dijkstra es iterativo y tiene la propiedad de que después de la k-ésima iteración del algoritmo, se conocen las rutas de coste mínimo hacia k nodos de destino y entre las rutas de coste mínimo a todos los nodos de destino, estas k rutas tendrán los k costes más pequeños.

9. Suponiendo que cada nodo conoce inicialmente los costos a cada uno de sus vecinos. Utilizando el algoritmo de Vector Distancias, especifique las entradas de la tabla de distancias para el Nodo n° 2





## 10. ¿Cuáles son los servicios que debe otorgar la Capa de Enlace?

Para la Capa de Enlace, nos resultará conveniente referirnos a los hosts y los routers simplemente como **nodos** ya que, no nos va a preocupar especialmente si un determinado nodo es un router o un host. También nos referimos a los canales de comunicación que conectan nodos adyacentes a lo largo de la ruta de comunicaciones con el nombre de enlaces. Para que un datagrama pueda ser transferido desde el host de origen al de destino, debe moverse a través de cada uno de los enlaces individuales que forman la ruta terminal a terminal.

Entre los posibles servicios que un protocolo de la capa de enlace puede ofrecer se incluyen:

- **Entramado:** Casi todos los protocolos de la capa de enlace encapsulan cada datagrama de la capa de red dentro de una trama de la capa de enlace antes de transmitirla a través del enlace. Una trama consta de un campo de datos, en el que se inserta el datagrama de la capa de red, y de una serie de campos de cabecera. (Una trama también puede incluir campos de cola; sin embargo, utilizaremos el término campos de cabecera para referirnos tanto a los de cabecera como a los de cola.) La estructura de la trama está especificada por el protocolo de la capa de enlace.
- **Acceso al enlace:** Un protocolo de control de acceso al medio (MAC, Médium Access Control) especifica las reglas que se utilizan para transmitir una trama a través del enlace. Para los enlaces punto a punto que tengan un único emisor en un extremo del enlace y un único receptor en el otro extremo, el protocolo MAC es muy simple (o no existe): el emisor puede enviar una trama siempre que el enlace esté inactivo. El caso más interesante es cuando hay varios nodos compartiendo un mismo enlace de difusión, en cuyo caso se presenta el denominado problema del acceso múltiple. En ese caso, el protocolo MAC sirve para coordinar la transmisión de las tramas de los múltiples nodos.
- **Entrega fiable:** Cuando un protocolo de la capa de enlace proporciona un servicio de entrega fiable, garantiza que va a transportar cada datagrama de la capa de red a través del enlace sin que se produzcan errores. Recuerde que ciertos protocolos de la capa de transporte (como TCP) también proporcionan un servicio de entrega fiable. De forma similar a los servicios de entrega fiable de la capa de transporte, el servicio de entrega fiable de la capa de enlace suele implementarse mediante reconocimientos y retransmisiones (véase la Sección 3.4). A menudo se utiliza un servicio de entrega fiable de la capa de enlace en aquellos enlaces que suelen presentar altas tasas de error, como por ejemplo en los enlaces inalámbricos, con el objetivo de corregir los errores localmente (en el enlace en el que se producen los errores), en lugar de obligar a que un protocolo de la capa de transporte o de la de aplicación realice una retransmisión de datos terminal a terminal. Sin embargo, la entrega fiable en la capa de enlace puede considerarse una sobrecarga innecesaria en aquellos enlaces que tengan una baja tasa de errores de bit, incluyendo los enlaces de fibra, los coaxiales y muchos enlaces de cobre de par trenzado. Por esta razón, muchos protocolos de la capa de enlace para enlaces cableados no proporcionan un servicio de entrega fiable.
- **Control de flujo:** Los nodos situados en cada extremo de un enlace tienen una capacidad limitada de almacenamiento en buffer de las tramas. Esto puede ser un problema cuando el nodo receptor puede recibir las tramas a más velocidad de la que puede procesarlas. Sin un control de flujo, el buffer del receptor puede desbordarse con lo que las tramas se perderían. De forma similar a lo que sucede en la capa de transporte, el protocolo de la capa de enlace puede proporcionar un mecanismo de control de flujo para evitar que el nodo emisor al otro lado del enlace abrume al nodo receptor situado en el otro extremo.
- **Detección de errores:** El hardware de la capa de enlace en un nodo receptor pudiera llegar a decidir, incorrectamente, que un bit contenido en una trama es cero cuando fue transmitido como un uno, y viceversa. Dichos errores de bit se introducen debido a la atenuación de las señales y al ruido electromagnético. Puesto que no existe ninguna necesidad de reenviar un datagrama que contenga un error, muchos protocolos de la capa de enlace proporcionan un mecanismo para detectar dichos errores de bit. Esto se lleva a cabo haciendo que el nodo transmisor incluya bits de detección de errores en la trama y que el nodo receptor realice una comprobación de errores. Recuerde de los Capítulos 3 y 4 que las capas de transporte y de red de Internet también ofrecen una forma limitada de detección de errores: la suma de comprobación de Internet. La detección de errores en la capa de enlace normalmente es más sofisticada y se implementa en hardware.
- **Corrección de errores:** La corrección de errores es similar a la detección de errores, salvo porque el receptor no sólo detecta si hay bits erróneos en la trama, sino que también determina exactamente en qué puntos de la trama se han producido los errores (y luego corrige esos errores). Algunos protocolos proporcionan corrección de errores en la capa de enlace sólo para la cabecera del paquete en lugar de para el paquete completo. Hablaremos de la detección y corrección de errores en la Sección 5.2.

- **Semiduplex y fullduplex:** Con la transmisión fullduplex, los nodos de ambos extremos de un enlace pueden transmitir paquetes al mismo tiempo. Sin embargo, con la transmisión semiduplex un mismo nodo no puede transmitir y recibir al mismo tiempo.

## 11. ¿En qué consiste el protocolo CSMA/CD?

El protocolo **CSMA/CD**: protocolo de acceso múltiple de Ethernet. El mismo, realiza lo siguiente:

1. Un adaptador puede comenzar a transmitir en cualquier instante; es decir, no existe el concepto de partición de tiempo.
2. Un adaptador nunca transmite una trama cuando detecta que algún otro adaptador está transmitiendo; es decir, utiliza un mecanismo de sondeo de portadora.
3. Un adaptador que está transmitiendo aborta su transmisión tan pronto como detecta que otro adaptador también está transmitiendo; es decir, utiliza un mecanismo de detección de colisiones.
4. Antes de intentar llevar a cabo una retransmisión, un adaptador espera un intervalo de tiempo aleatorio que normalmente es más pequeño que el tiempo que se tarda en transmitir una trama.

Estos mecanismos proporcionan a CSMA/CD un rendimiento mucho mejor que el del protocolo ALOHA con particiones en un entorno LAN. De hecho, si el retardo máximo de propagación entre estaciones es muy pequeño, la eficiencia de CSMA/CD puede aproximarse al 100 por ciento. Observe también que el segundo y tercer mecanismos de la lista anterior requieren que los adaptadores de Ethernet sean capaces de (1) detectar cuándo algún otro adaptador está transmitiendo y (2) detectar una colisión mientras están transmitiendo. Los adaptadores Ethernet realizan estas dos tareas midiendo los niveles de tensión antes y durante las transmisiones.

Dentro de un adaptador específico, el protocolo CSMA/CD opera de la siguiente forma:

1. El adaptador obtiene un datagrama de la capa de red, prepara una trama Ethernet y la coloca en un buffer del adaptador.
2. Si el adaptador detecta que el canal está inactivo (es decir, durante 96 periodos de bit el adaptador no recibe intensidad de señal procedente del canal), comienza a transmitir la trama. Si el adaptador detecta que el canal está ocupado, espera hasta comprobar que no hay intensidad de señal (más otros 96 periodos de bit) y luego comienza a transmitir la trama.
3. Mientras está transmitiendo, el adaptador monitoriza la presencia de señales procedentes de otros adaptadores. Si el adaptador transmite la trama completa sin detectar ninguna señal procedente de otros adaptadores, concluye que ha terminado su trabajo con esa trama.
4. Si el adaptador detecta intensidad de señal procedente de otros adaptadores mientras está transmitiendo, deja de transmitir su trama y transmite una señal de interferencia (jam) de 48 bits.
5. Después de abortar la transmisión de la trama (es decir, de transmitir la señal de interferencia), el adaptador entra en la fase de espera exponencial (backoff exponencial). Específicamente, a la hora de transmitir una determinada trama, después de experimentar la  $n$ -ésima colisión para esa trama, el adaptador selecciona un valor aleatorio para  $K$  del conjunto  $0, 1, 2, \dots, 2m - 1$ , donde  $m = \min(n, 10)$ . El adaptador espera entonces  $K \cdot 512$  periodos de bit y vuelve al Paso nº 2.

## 12. ¿Qué es un Switch? ¿Cómo funciona?

Los switches, son los encargados de la interconexión de equipos dentro de una misma red, o lo que es lo mismo, son los dispositivos que, junto al cableado, constituyen las redes de área local o LAN. Un switch o conmutador es un dispositivo de interconexión utilizado para conectar equipos en red formando lo que se conoce como una red de área local (LAN) y cuyas especificaciones técnicas siguen el estándar conocido como Ethernet.

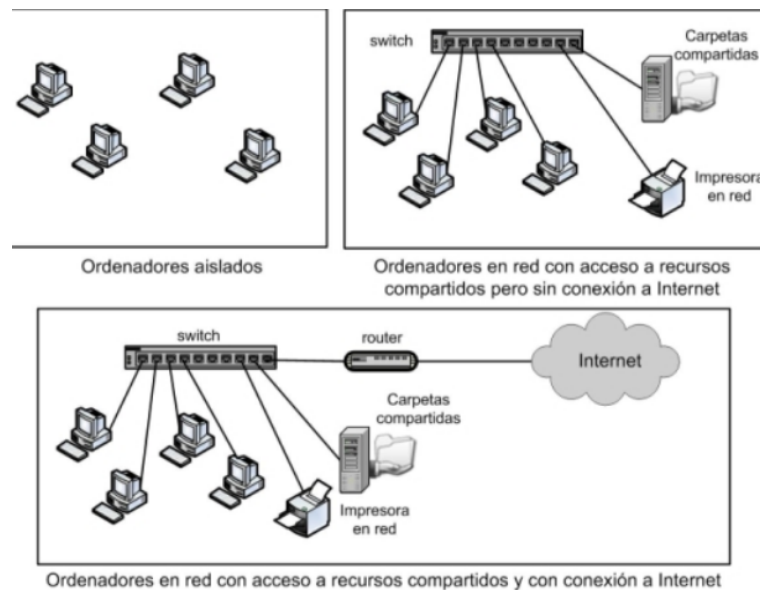
En realidad los switches no son los únicos elementos encargados de la interconexión de dispositivos en una red local. Los switches realizan esta función para medios cableados. Cuando la interconexión se realiza de forma **inalámbrica** el dispositivo encargado de ello se denomina **Punto de acceso inalámbrico**.

En la actualidad las redes locales cableadas siguen el estándar Ethernet (prácticamente el 100 %) donde se utiliza una topología en estrella y donde el switch es el elemento central de dicha topología.

En las primeras versiones de Ethernet, la topología en estrella se implementaba con otro dispositivo conocido como hub. En la actualidad, los hubs se pueden considerar obsoletos. Y es importante tener en cuenta que,

aunque externamente son muy parecidos, los switches tienen prestaciones muy superiores a los hubs por lo que si aún encontramos alguna red que utilice un hub es muy recomendable sustituirlo por un switch.

La función básica de un switch es la de unir o conectar dispositivos en red. Es importante tener claro que un switch NO proporciona por si solo conectividad con otras redes, y obviamente, TAMPOCO proporciona conectividad con Internet. Para ello es necesario un router.



Como se observa en la figura, la existencia de la red local permite:

- **Compartir archivos:** Un equipo de la red habilita la compartición de archivos y el resto de equipos pueden acceder a dichos archivos a través de la red.
- **Compartir impresoras:** Todos los equipos de la red pueden utilizar la misma impresora.
- **Compartir la conexión a Internet:** Todos los equipos pueden acceder a Internet a través de router de acceso, que está conectado en la red.

## Características

### Puertos

Los puertos son los elementos del switch que permiten la conexión de otros dispositivos al mismo. Como por ejemplo un PC, portátil, un router, otro switch, una impresora y en general cualquier dispositivo que incluya una interfaz de red Ethernet. El número de puertos es una de las características básicas de los switches. Aquí existe un abanico bastante amplio, desde los pequeños switches de 4 puertos hasta switches troncales que admiten varios cientos de puertos.

El estándar Ethernet admite básicamente dos tipos de medios de transmisión cableados: el cable de par trenzado y el cable de fibra óptica. El conector utilizado para cada tipo lógicamente es diferente así que otro dato a tener en cuenta es de qué tipo son los puertos. Normalmente los switches básicos sólo disponen de puertos de cable de par trenzado (cuyo conector se conoce como RJ-45) y los más avanzados incluyen puertos de fibra óptica (el conector más frecuente aunque no el único es el de tipo SC).

### Velocidades

Dado que Ethernet permite varias velocidades y medios de transmisión, otra de las características destacables sobre los puertos de los switches es precisamente la velocidad a la que pueden trabajar sobre un determinado medio de transmisión.

### Conmutación

La función básica que realiza un switch se conoce como conmutación y consiste en transferir datos entre los diferentes dispositivos de la red. Para ello, los switches procesan la información contenida en las cabeceras de la trama Ethernet.

Sin entrar mucho en detalle en el funcionamiento de Ethernet podemos decir que Ethernet es una tecnología de transmisión de datos para redes locales cableadas que divide los datos que se tiene que transmitir en tramas y a cada trama se le añade una determinada información de control llamada cabecera. Dicha cabecera contiene la dirección MAC tanto del emisor como del receptor.

Los switches guardan en una tabla las direcciones MAC de todos los dispositivos conectados junto con el puerto en el que están conectados, de forma que cuando llega una trama al switch, dicha trama se envía al puerto correspondiente.

## Buffers

El elemento clave en los switches para llevar a cabo el proceso de conmutación son los buffers, que son zonas de memoria donde las tramas son almacenadas antes de ser reenviadas al puerto correspondiente. Esta característica además, permite al switch conectar puertos que trabajen a diferentes velocidades.

Los buffers pueden ser implementados en la salida de los puertos, en la entrada de los puertos o una combinación de ambos. Lo más habitual es implementarlos en la salida ya que es el modo más eficiente, consiguiéndose unos índices de eficacia cercanos al 98 %.

## Técnicas de Conmutación

Existen dos técnicas para llevar a cabo la transferencia de los datos entre puertos de un switch:

1. **Reenvío directo (cut-through):** En esta técnica, cuando un switch comienza a recibir datos por un puerto, no espera a leer la trama completa para reenviarla al puerto destino. En cuanto lee la dirección de destino de la trama MAC, comienza a transferir los datos al puerto destino.

Esta técnica proporciona unos tiempos de retardo bastante bajos, sin embargo, tiene como inconveniente que sólo puede usarse cuando las velocidades de todos los puertos son iguales.

2. **Almacenamiento y reenvío (Store and Forward):** En este caso, cuando un switch recibe datos por un puerto, almacena la trama completa en el buffer para luego reenviarla al puerto destino. La utilización de esta técnica permite realizar algunas comprobaciones de error antes de ser enviada al puerto de destino.

## Switches de Nivel 3 y Nivel 3/4

Los switches de gama alta utilizados en el troncal de redes Ethernet de mediana y gran envergadura suelen ofrecer capacidades de enrutamiento de paquetes IP. A este tipo de switches se le conoce como switches de nivel 3. Un switch de nivel 3 realiza todas las funciones de conmutación de un switch pero además proporciona funciones de enrutamiento IP. Esta característica es especialmente útil para switches que utilicen VLAN y necesiten comunicar algunas de sus redes LAN virtuales.

Además, pueden existir switches que ofrezcan características relacionadas con funciones del nivel 4, como control de puertos. A estos switches se le conoce como switches de nivel 3/4.