Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Fonaments de Telemàtica

Apuntes

13/10/2011
Universitat Politècnica de Catalunya
Jordi Casademont Serra, Eduard Garcia Villegas, Rafael Vidal Ferré

Contenido

1. Procesos y funciones en las redes de telecomunicación

El poder establecer una comunicación telefónica, a pesar de que los dos extremos estén cada uno en un continente distinto a miles de kilómetros, o recibir fotografías desde un pequeño robot en la superficie de Marte (¡a centenares de millones de kilómetros!), plantea infinidad de problemas muy diversos que hacen que la tarea de llevar el Anillo a Mordor parezca tan complicado como el mecanismo de un Chupa-Chup: ¿Cómo superar tan largas distancias? ¿Cómo encontrar a un usuario en concreto dentro de una red con millones de usuarios? ¿Cómo asegurar que la información que se recibe es correcta y que no se ha perdido nada por el camino?... y un larguísimo y complicado etcétera.

En este capítulo se describen algunos de los procesos básicos de las redes de comunicaciones. O dicho de otro modo, los mecanismos que, sin que uno se dé cuenta, obran el "milagro" de que algo que está sucediendo en Tokio lo puedas ver desde el sofá de tu casa, o que puedas hablar por teléfono con tu abuela, que vive en Caracas, desde la cima de una montaña de los Pirineos.

1.1. Arquitecturas de protocolos: Como dijo Jack el Destripador, ... vamos por partes

Para poner un poco de orden, los **Ingenieros Telemáticos** agrupan todas las problemáticas antes mencionadas y diseñan las soluciones de una forma estructurada, en lo que se denomina una arquitectura de protocolos. Esto consiste en dotar a los elementos de la red de la capacidad de dialogar o negociar entre sí. Los protocolos son los lenguajes o idiomas que hablan estos elementos de red. Más concretamente, los protocolos son las reglas que describen la sintaxis y la semántica de estos lenguajes.

1.1.1. Un modelo teórico y un modelo práctico

Desde la aparición de los primeros ordenadores, se ha buscado que estos se comuniquen entre sí. Pero entonces, cada ordenador era una pieza única, y resultaba complicado hacer que dos ordenadores distintos llegaran a “entenderse”. A finales de los ’70 (del s. XX, se entiende), las redes de ordenadores ya estaban muy extendidas, aunque su presencia se limitaba a grandes oficinas, universidades, etc., y el problema se hizo más evidente: según si un ordenador era de uno u otro fabricante, o si tenía un sistema operativo u otro, no se podía comunicar más que con los de su marca y modelo. Entonces la Organización Internacional para la Estandarización (ISO) se planteó definir una arquitectura de protocolos que pudiera ser común a todos: la OSI (*Open System Interconnection*). Este sistema agrupa las funciones y protocolos de red

por *capas apiladas*, de manera que cada capa es responsable de una serie de funciones.

En las redes modernas, la información que envía un usuario es digitalizada (convertida en ceros y unos, o bits) y transmitida a "trozos", llamados paquetes. Pues bien, la comunicación con protocolos se consigue añadiendo algunos bits extras a la información de cada paquete. Esta información extra se conoce como *cabeceras (*Figura 1*)*. En transmisión, los datos del usuario fluyen de arriba abajo y cada capa añade las cabeceras con la información que necesita para lograr su cometido. A medida que se desciende por la pila de protocolos, estas cabeceras se van acumulando. En recepción, al ascender por la pila, cada capa lee y elimina las cabeceras correspondientes, hasta que al final queda sólo la información de usuario.

Figura 1: Formato de un paquete de datos.

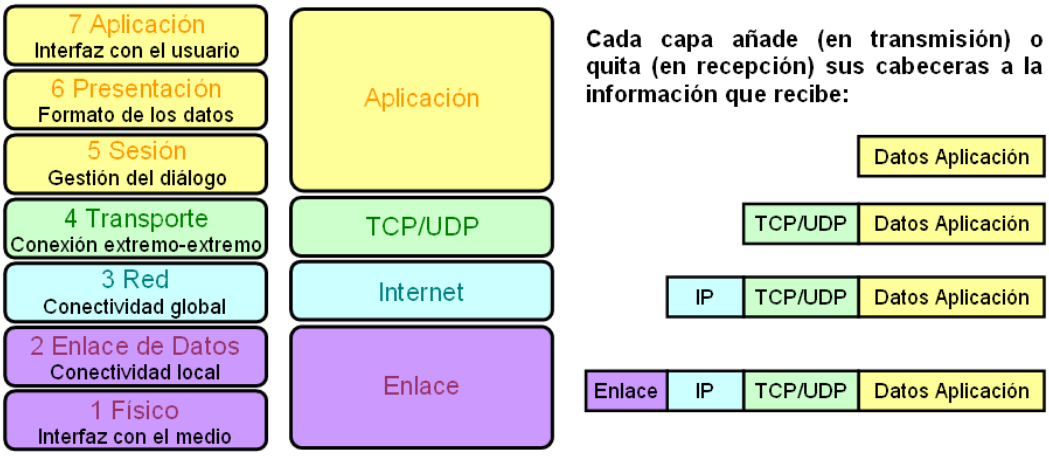

Figura 2: a) Pila OSI de ISO b) Pila TCP/IP.

Concretamente el modelo OSI define siete capas, tal y como se muestra en la Figura 2 a). Para entender el cometido de cada una veremos un ejemplo: la transmisión de un flujo de audio entre dos ordenadores lejanos. En primer lugar, la *capa de aplicación* (7) sirve como interfaz al usuario e inicia el envío tras detectar su petición. A continuación, la *capa de presentación* (6) establece el formato de los datos, es decir, la capa 6 del emisor tendrá que hacer saber a la capa 6 del receptor cómo leer los unos y ceros que le llegarán; los bits se tendrán que leer de una forma u otra dependiendo de si el audio está en formato mp3, wav, etc. La *capa de sesión* (5) gestiona el diálogo entre aplicaciones, es decir, cómo avisar a la máquina emisora para que empiece a enviar (play), cómo parar (stop), cómo reanudar el envío si este se ha interrumpido (pause), etc. La *capa de transporte* (4) se asegura que estos bits son recibidos por la aplicación

correcta dentro del ordenador, por ejemplo que no se entregan correos electrónicos a la aplicación de chat. La *capa de red* (3) permite localizar a la máquina destino; para ello en primer lugar es necesario identificarlo (con un número de teléfono, una dirección IP, etc.) y luego averiguar los posibles caminos que llevan a ese destino y escoger el más adecuado (pasando a través de diferentes enlaces y diferentes elementos de red intermedios). La *capa de enlace de datos* (2) se encarga de la transmisión de bits entre elementos de red conectados directamente, y esto incluye asegurarse de que los bits que se reciben son igual a los que se envían. Finalmente, la capa física (1), se limita a inyectar esos bits en el medio físico (aire, cable eléctrico, fibra óptica, etc.). En recepción, la capa 1 se encarga de detectar los bits que llegan.

El problema es que este esquema con tantas capas y una división de las tareas tan rígida, puede ser demasiado complicado y resulta poco práctico. Por eso, aunque la idea era buena, el modelo OSI se ha quedado en algo teórico. En realidad, el modelo que ha triunfado ha sido el de Internet (o, más exactamente, el TCP/IP). El modelo TCP/IP también es un modelo basado en capas, pero es algo más sencillo (Figura 2 b)). Por ejemplo, las capas 1 y 2 OSI se funden en una, y todos los mecanismos de las capas 5, 6 y 7 OSI conviven en la aplicación del modelo TCP/IP (firefox, e-mule, Messenger, etc. son aplicaciones del modelo TCP/IP). La capa de transporte es muy parecida a la capa TCP/IP. La capa 3 es igual a la capa Internet, donde reside el protocolo IP, que es precisamente el que da nombre a la red.

1.2. Digitalización de la información

Antiguamente, las redes de comunicaciones se limitaban a transportar un tipo concreto de información. Así que teníamos una red dedicada solamente a llevar tu voz al otro lado del mundo, si era necesario (red telefónica), otra red diferente para transportar texto (telegrafía), y el no va más: ¡imágenes en movimiento y con sonido (televisión)! Para cada una de ellas, se necesita toda una infraestructura muy particular, y los usuarios necesitan un dispositivo diferente. En la actualidad, esto puede sonar ya anticuado porque con un ordenador, o simplemente con un teléfono móvil, podemos transmitir y almacenar cualquier tipo de información. ¿Cómo se consigue eso? La respuesta es la **digitalización**. Mediante la digitalización, cualquier información (voz, música, imágenes, etc.) se transforma en un conjunto de ceros y unos (bits), simplificando su tratamiento. Precisamente, es este proceso de digitalización el que ha permitido la reciente revolución en las comunicaciones, o lo que de manera más romántica se conoce como la Era de la Información o la Revolución Digital.

1.2.1. Analógico y digital

Una señal es una magnitud física que se puede medir y que varía con el tiempo. Por ejemplo, el sonido, es una onda de presión que se desplaza por un medio (normalmente el aire). La cantidad medible es la amplitud de esta vibración. Estas

señales son analógicas por naturaleza. La característica principal de una señal analógica es que esta cantidad medible puede tomar cualquier valor, en cualquier momento.

En cambio, una señal digital sólo puede tomar determinados valores en determinados momentos. La gráfica de la Figura 3 a) representa una señal analógica, y a su derecha se muestra una señal digital. Mientras que la señal analógica puede tomar cualquier valor real entre 0 y 10, por ejemplo, 1,235, 6,12, π, etc. Los valores que toma la digital están limitados a unos **dígitos** concretos. En este caso, la señal digital sólo puede tomar valores enteros (0,1, 2, 3, …10). Además, la señal digital no puede cambiar de valor en cualquier momento, sólo lo hace a intervalos fijados.

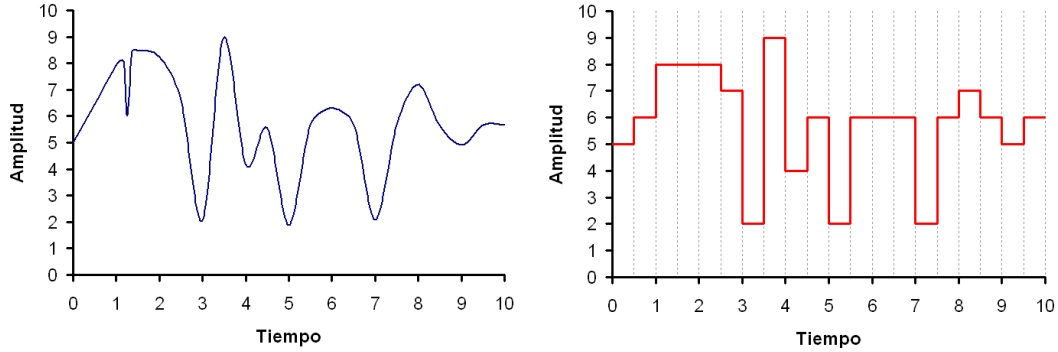

Figura 3: a) Señal analógica; b) Señal digitalizada.

¿Cómo funciona? Vamos a ver el proceso de digitalización a través de un ejemplo. La gráfica de la Figura 3 a) representa una oscilación producida por la voz humana. A mayor volumen de la voz, mayor amplitud, y cuanto más aguda es la voz (frecuencia más alta), más rápido varía la señal en el tiempo. A su derecha, tenemos esa misma señal una vez digitalizada. Un micrófono recoge estas variaciones de presión mediante una membrana, y las transforma en una señal eléctrica también analógica. Esta señal llega al conversor analógico/digital (ADC); a su salida tendremos una secuencia de bits, pero … ¿Cómo lo hace? El ADC sigue cuatro procesos básicos:

- **Muestreo**: cada cierto tiempo, se toma una instantánea (una “foto”) de la señal analógica. Es decir, se toman muestras de la señal. En el ejemplo de la gráfica de la Figura 3 a), en el instante 3, la amplitud tiene un valor de 2,156.
- **Retención**: la "foto" se mantiene fija durante el tiempo necesario para ser tratada en la siguiente fase (cuantificación).
- **Cuantificación**: el valor retenido de amplitud es medido con una escala que tiene unas pocas marcas. Se podría decir que en esta fase, se “redondea” el valor de la amplitud a la marca más cercana. Por ejemplo, la muestra tomada

en el instante 3 (2,156), cae entre los valores 2 y 3. El cuantificador decidirá 2, que es el nivel más cercano.

- **Codificación**: a cada posible valor que podemos encontrar a la salida del cuantificador se le asigna un valor binario. Así, 1 bit es la unidad mínima de información y puede representar sólo dos valores (0 ó 1). Con dos bits podemos representar cuatro valores, con 3 bits ocho, y en general, con *n* bits, 2^n valores. Por ejemplo, el valor cuantificado 2, se puede representar con 4 bits como la secuencia: 0010.

Ya en la misma Figura 3 b) se hace evidente que hemos perdido información por el camino. Si el muestreo es poco frecuente, (pocas muestras por segundo), las variaciones rápidas (frecuencias altas) de la señal analógica no se reflejarán en la digital. Esto ocurre, por ejemplo, entre los instantes 1 y 2 de la figura anterior. Para no perder esa información, el ingeniero Harry Nyquist estableció que la frecuencia de muestreo debe ser como mínimo el doble de la frecuencia máxima de la señal analógica. Por ejemplo, la frecuencia más alta que puede producir la voz humana llega típicamente a los 4KHz (¡si eres una soprano puedes llegar a los 9KHz!), por tanto, la frecuencia de muestreo debería ser de 8KHz (8.000 muestras por segundo). Además, si el cuantificador tiene pocos niveles, las variaciones pequeñas de amplitud de la señal analógica se perderán. Es como medir personas con una cinta métrica que marque los metros pero no los centímetros. Para mejorar este problema, se usa una escala con más niveles, lo que significa que se necesitan más bits por muestra. En la telefonía digital, el cuantificador puede dar a su salida 256 valores diferentes (8 bits cada muestra). Por tanto, para transportar voz humana digitalizada se necesita una red con una capacidad de 8k muestras/s x 8 bits/muestra = 64 Kbps.

1.2.2. Compresión: mp3 y mp4

Un sonido o un vídeo digitalizado, se guarda en un disco duro, en un DVD, se transmite por un móvil, etc. Pero esos recursos son caros (no vale lo mismo un *pendrive* de 500 MB que un disco duro de 160 GB) y por tanto, interesa reducir el tamaño de la información para que ocupe menos espacio (menos bits), o para que sea transmitida más rápidamente por el móvil (¡y que te cobren menos por la conexión!). Sin embargo, reducir el tamaño de esa información significa muchas veces perder calidad.

De todas formas se puede hacer sin que apenas se note la pérdida aprovechando que los sentidos (vista, oído) del ser humano son limitados e imperfectos. Este es el caso del popular mp3. Ya hemos visto que cuantas más muestras por segundo se use en la digitalización, frecuencias más altas se podrán detectar y más bits utilizaremos. Pero resulta que el oído humano normal, no puede oír sonidos agudos más allá de los 22 KHz, mientras que un perro llega a los 50 KHz, y un murciélago hasta los 100 KHz.

Entonces, dado que de momento, no se hacen iPods para perros, no tiene sentido una frecuencia de muestreo de más de 44 KHz (recordar lo que decía Nyquist). Adicionalmente, los inventores del mp3 se dieron cuenta que dentro del rango de frecuencias que un ser humano es capaz de oír, se da un fenómeno conocido como enmascaramiento. Cuando un tono (imagina un silbido muy agudo) suena muy fuerte, otro tono a menor volumen y muy cercano en frecuencia (otro silbido, un poco más agudo) no será percibido por un oído normal. Por tanto, no tiene sentido malgastar bits para codificar ese segundo tono, si total,… casi nadie lo iba a oír.

Cambiando ligeramente de tema... ¿Cómo se puede escribir una foto con unos y ceros? En primer lugar se divide la foto en pequeños puntos denominados píxeles (Figura 4). Cada uno de estos píxeles está pintado de un sólo color. Cada color se representa con una combinación de bits, de manera que cuantos más bits, más colores podremos representar. Lo más normal es usar 24 bits para representar el color (8 para la cantidad de rojo de la mezcla, 8 para el verde y 8 para el azul). Esto nos permite representar hasta 2^{24} colores (¡más de 16 millones de colores!). Si usamos muchos colores o píxeles muy pequeños, resultará que la imagen ocupará muchos bits, pero de no ser así, la imagen perderá calidad.

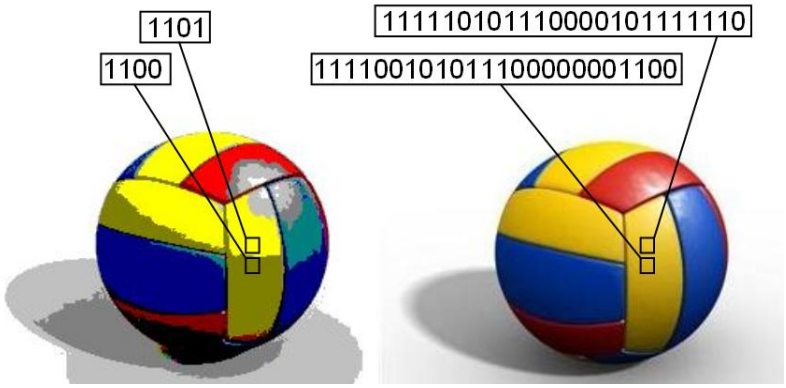

Figura 4: Digitalización de una imagen, con 4 bits ó 24 bits.

De nuevo intentaremos comprimir esta información, sin que el ojo humano aprecie una gran pérdida de calidad. Por ejemplo, una foto del cielo: un píxel cualquiera será muy parecido a sus vecinos (¡azul!). Las técnicas de compresión de imagen buscan ahorrar bits cuando píxeles cercanos tienen prácticamente el mismo color.

El vídeo, desde los hermanos Lumière, se basa en tomar muchas fotos seguidas para captar el movimiento. Luego, estas fotos, llamadas fotogramas, son reproducidas de manera que el ojo humano no distingue el cambio de una foto a la siguiente y le parece que el movimiento es continuo. Sin embargo, las fotos se tienen que pasar muy rápido (normalmente 25 fotogramas por segundo). Pero resulta que si se trata de una

imagen con poco movimiento, un fotograma es casi igual que el anterior, salvo alguna pequeña diferencia. Entonces, para reducir la cantidad de información, en lugar de almacenar los bits de cada foto completa, lo que se hace es digitalizar un fotograma entero de vez en cuando, y de los fotogramas siguientes, sólo se conserva la información del trocito de la imagen que se ha movido. Los nuevos formatos multimedia, como el mp4, usan una combinación de las dos técnicas para comprimir imagen en movimiento según el estándar MPEG: por un lado agrupan los píxeles cercanos y parecidos, y por el otro, sólo digitalizan por completo algunos fotogramas. Por ejemplo, una transmisión de fútbol, donde casi todos los píxeles son siempre verdes, será más fácil de comprimir que una película de Jackie Chan.

Existe también la compresión sin pérdidas, que consiste en reducir la cantidad de bits de la información digital, de manera que luego se puede recuperar la información tal y como era antes de la compresión. A esto se dedican utilidades como ZIP, ARJ, RAR, etc. Por ejemplo, para transmitir el texto: "*AAA BBBBB CC*"; se puede reducir un 33% si enviamos: "*3A 5B 2C"*, el receptor escribiría tres "as" seguidas, un espacio, cinco "bes", espacio, y dos "ces", recuperando así el mensaje original.

1.3. Bits por tierra, mar y aire: códigos de línea y modulaciones.

Ya hemos visto cómo la información se convierte en un flujo de bits mediante el proceso de digitalización. Recordar que un bit es la unidad mínima de información (0/1, Sí/No, Cara/Cruz, etc.). Dicho así queda muy bonito, en un plano casi filosófico, pero para una máquina, un bit debe ser algo más concreto. Por ejemplo, los bits almacenados en el disco duro de un ordenador son en realidad pequeños "imanes": los discos duros leen mediante una "brújula" la superficie de un material magnético (hierro, cobalto, etc.). Este material está dividido en porciones muy pequeñas (menores que una milésima parte de un milímetro) cargadas magnéticamente. Si esa pequeña porción de la superficie está cargada positivamente, tenemos almacenado un 1 y sino un 0.

Pero para enviar un bit de una máquina a otra a través de un cable o por radio ya no sirve la "brújula", debemos establecer la representación de los unos y los ceros en forma de valores de tensión, intensidad, pulsos de luz,… es lo que definen los **códigos de línea**. Otro problema es que las señales digitales no se transmiten muy bien a distancias largas, debemos adaptarlas mediante las denominadas **modulaciones**.

1.3.1. Códigos de línea

Cuando se transmiten señales digitales deben establecerse los valores de los dígitos binarios (el 0 y el 1). Lo normal sería pensar que el 0 podría tener un valor negativo de voltios, -5V por ejemplo y el 1 un valor positivo, 5V (codificación denominada No Retorno a Cero – NRZ, Figura 5-a). Pero estos códigos tienen problemas cuando hay

que enviar muchos bits seguidos iguales. Por ejemplo, si emisor y receptor no están perfectamente sincronizados (transmisor/receptor no escribe/lee exactamente al mismo ritmo), el receptor se puede llegar a confundir al no saber exactamente dónde acaba un bit y dónde empieza el siguiente (se pierde algún bit si el receptor es más lento o se lee el mismo bit dos veces si el lector va más rápido) En este caso, una solución adoptada por los ingenieros de telecomunicación es escoger bits que tengan un cambio de estado incorporado. El 0, durante la mitad de su duración vale -5V y luego cambia a 5V durante la otra mitad. El 1, durante la primera mitad vale 5V y durante la segunda -5V. A esta técnica se le llama codificación Manchester (Figura 5-b) y es la que se utilizaba en las primeras redes Ethernet, pero existen muchas otras.

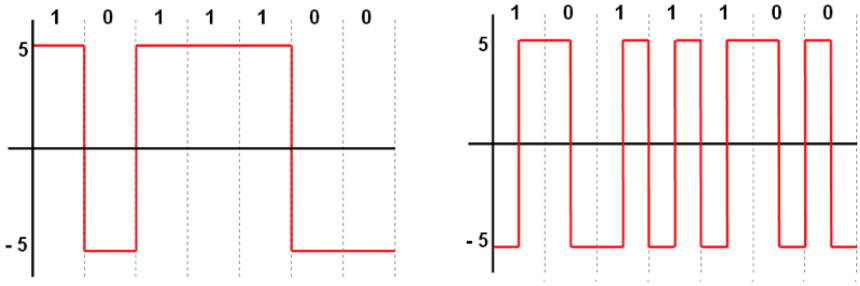

Figura 5: a) Codificación NRZ. b) Codificación Manchester.

1.3.2. Modulaciones

Cuando se trata de enviar esos bits por el aire, la cosa se complica. Usando cables, se pueden separar las transmisiones de redes diferentes, pero el aire es un medio compartido por muchos sistemas de comunicaciones. Los bits de la telefonía móvil van por el mismo aire que los bits de la televisión digital (TDT), igual que Wi-Fi, Bluetooth, etc. Para separar unas de otras se usan canales de frecuencia diferentes.

Además, las señales digitales no se transmiten bien a grandes distancias, en cambio las analógicas sí. Por ello se utilizan señales analógicas para transmitir datos digitales.

Las señales analógicas transmitidas son ondas electromagnéticas que tienen formas sinusoidales. Normalmente se representan con la fórmula $A \cdot \cos(\omega t+\varphi)$ que tiene tres parámetros: A es la amplitud (el voltaje), ω es la frecuencia y φ la fase. La frecuencia ya nos permite separar unas redes de otras, así por ejemplo, la TDT usa frecuencias alrededor de 800MHz, Wi-Fi 2400MHz, etc. La idea de modular en comunicaciones se basa en cambiar alguno de estos tres parámetros en función de los datos a la entrada (en nuestro caso, ceros y unos): modulación en amplitud (ASK), en frecuencia (FSK), o en fase (PSK). En la Figura 6 se muestran ejemplos de cada una. La señal ASK cambia su amplitud según los bits de entrada, multiplicando la información por la función coseno del generador. En FSK, un 0 se representa con la función seno normal y un 1 con una frecuencia ligeramente mayor. En PSK, un 1 se representa con un

salto de fase de π, mientras que un 0 no tiene cambio de fase. A partir de estas modulaciones básicas, se pueden obtener otras más complicadas, pero más rápidas, o más resistentes a las interferencias.

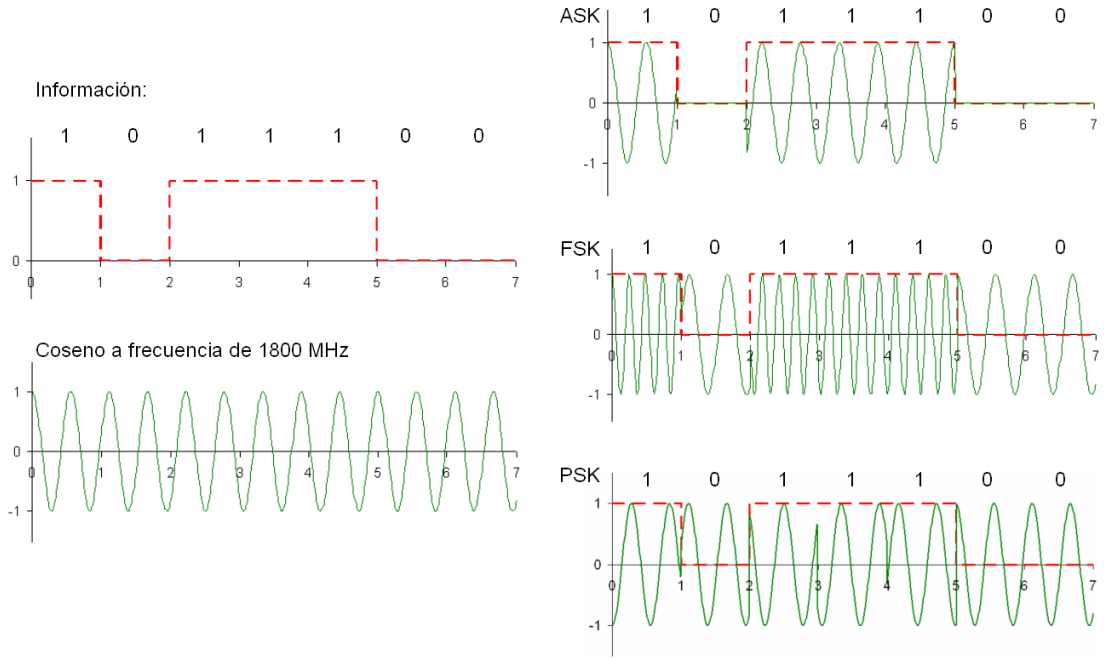

Figura 6: Modulaciones digitales básicas.

1.4. Cuando compartir es necesario: control de acceso al medio

En los apartados anteriores se han discutido los beneficios de trabajar con bits y como éstos pueden ser enviados por diferentes medios (cables o radio) mediante los códigos de línea y las modulaciones. Pero, ¿siempre podemos enviar? La respuesta a esta pregunta dependerá de si el nodo que la realiza utiliza un red en la que dispone de un enlace en propiedad (por ejemplo la red telefónica), o de si lo comparte con otros nodos (por ejemplo Ethernet, Wi-Fi, 2G/3G o Bluetooth).

Un medio dedicado permite que un nodo pueda enviar y/o recibir a voluntad. Por el contrario, un medio compartido exige que los nodos que lo comparten se pongan de acuerdo para utilizarlo de forma ordenada. Aunque ello sea más complicado, a veces no hay más remedio y además permite reducir costes.

Para compartir el medio es necesario establecer una serie de reglas que todos los nodos deben respetar. Es lo que se conoce como mecanismos de control del acceso al medio (MAC). Para comprender estas reglas, puede establecerse un paralelismo con lo que ocurre en una clase, en la que un profesor y sus estudiantes quieren discutir un tema. Si todo el mundo empieza a expresar su opinión sin fijarse en si otras personas también están hablando, se producen lo que se denominan colisiones: se

superpone la voz (señal) de más de una persona (nodo) y el resultado es que no se entiende nada (señal ininteligible). Es importante detectar que se ha producido una colisión porque quién expresa una opinión (fuente de la información) debe volver a expresarla (retransmitirla) pues ésta no se ha entendido.

De este ejemplo se puede extraer la necesidad de mecanismos para reducir la posibilidad de colisión, detectar las colisiones y finalmente retransmitir. Obsérvese como los 3 mecanismos son imprescindibles. Si no intentamos evitar las colisiones, podríamos estar hablando a la vez continuamente. Si no detectamos las colisiones, creeremos que los demás nos entienden cuando no es así, y finalmente, detectar la colisión y no retransmitir equivale a renunciar a que nos entiendan.

Un primer mecanismo para evitar las colisiones puede ser establecer turnos. A cada persona se le asigna un turno durante el cuál puede hablar por un tiempo predefinido. Este mecanismo evita las colisiones pero es poco flexible ya que asigna tiempo tanto a personas que quieren hablar como a personas que no. Una segunda opción es escuchar antes de hablar. Es decir, si nadie habla yo puedo hacerlo. La probabilidad de colisión se reduce pero no se elimina: dos personas después de un silencio (inactividad en el enlace) pueden intentar hablar a la vez. Este mecanismo, con diferentes variantes, es utilizado en las redes LAN Ethernet y WLAN Wi-Fi. Es mucho más flexible que el anterior, ya que permite que personas (nodos) que necesitan hablar más (i.e. enviar más información) lo hagan aprovechando que otros no lo hacen. Por contra, no asegura un mínimo de tiempo a nadie.

Cuando hay una colisión se debe retransmitir la información perdida, pero ¿cuándo se debe retransmitir? El instinto nos puede conducir a hacerlo de manera inmediata, para adelantarme a mis "competidores". Sin embargo, si todas las personas (nodos) que han "colisionado" siguen la misma premisa lo que va a pasar es que van a volver a colisionar. Para evitar este problema se crearon los algoritmos de *backoff*. Estos algoritmos definen un tiempo aleatorio que el nodo debe esperar para volver a realizar una retransmisión y un número máximo de reintentos. Siguiendo un algoritmo de este tipo, las personas que han "colisionado" al hablar tiran un dado y esperan para volver a hablar un tiempo igual al del valor que les ha salido. Puede suceder que algunos saquen el mismo valor y que la colisión se repita. En este caso se repite el proceso pero con un dado con más caras, de manera que la probabilidad que ambos saquen el mismo valor se reduce. El proceso se repite hasta un número máximo de veces en que, de alcanzarse, se da la comunicación por imposible.

Las dos soluciones comentadas, turnos y escuchar antes de hablar, son totalmente distribuidas. Todos los nodos que participan en la comunicación lo hacen como iguales y todos deben realizar las mismas tareas para poder compartir el enlace con éxito. Otra solución con una filosofía diferente es la pedir permiso para hablar. Esta opción exige que alguien centralice las peticiones. Por ejemplo los estudiantes pedirían permiso para hablar al profesor y sería éste quién indicaría qué estudiante puede

hacerlo. Esta es la filosofía que utilizan las redes celulares de 2G o 3G. Así cada vez que, por ejemplo, el usuario (estudiante) llama a alguien, el terminal realiza una petición de acceso a la red del operador (el profesor) que es contestada con una asignación de recursos adecuada. Obsérvese cómo no habría colisión en el momento de hablar, pero sí podría haberlo al pedir permiso (petición de acceso). La colisión sería detectada por el profesor (red del operador), que no dará permiso a ninguno de los estudiantes (i.e. no les asignará recursos) por lo que deberán volver a pedir permiso para hablar (i.e. retransmitirán la petición de acceso), siguiendo un algoritmo de *backoff*.

Finalmente, compartir el medio tiene otras implicaciones, que si bien menos complejas, también necesitan de sus reglas. En primer lugar, es necesario definir un sistema de identificación para saber a quién va dirigida la información y quién la envía para así poder contestarle si es necesario. Estos identificadores se denominan direcciones físicas, porque están asociados al hardware (por ejemplo, una tarjeta de red) y su valor esta fijado por el fabricante. Este el caso de las redes Ethernet o Wi-Fi.

1.5. Las redes de comunicaciones, un mundo imperfecto

No es suficiente con controlar el acceso al medio y saber transmitir información sobre él. Existen una serie de problemas que pueden producirse y que exigen de una serie de mecanismos de control. Estos mecanismos tienen como objetivo o bien prevenir la aparición de estos problemas o bien solucionarlos, minimizando su impacto en el peor de los casos. A continuación se explican su razón de ser y se describen las ideas básicas que los sustentan.

1.5.1. Control de errores de bit:

Puede suceder que lo que un nodo envía no sea igual a lo que reciba el nodo receptor. Esta situación es debida a que ningún medio de transmisión está libre de errores. Así puede ser que lo que en emisión era un '0', en recepción sea un '1' y viceversa. De cara a caracterizar el comportamiento de un enlace en términos de bits erróneos se define el parámetro BER (Bit Error Ratio), como la proporción de bits que se reciben incorrectamente. Valores típicos de BER pueden ser de entre 10^{-12} (un bit erróneo de cada billón) y 10^{-9} para un enlace de fibra óptica, mientras que para un enlace radio se puede llegar hasta valores de 10^{-2} (un bit erróneo de cada 100).

Pero a la hora de caracterizar un medio de transmisión no sólo es importante conocer su BER sino también cuándo aparecen estos errores, lo que se conoce como su patrón. Así, en fibra óptica su patrón se considera totalmente aleatorio, es decir, algún bit muy de vez en cuando será erróneo. Por el contrario, en los enlaces de cobre, y sobretodo los enlaces radio, suelen producirse ráfagas de errores. El motivo es que en estos medios, las señales son vulnerables a las interferencias o sufren de problemas

de propagación, circunstancias que de producirse conllevan a errores en varios bits consecutivos.

Los errores de bit son inevitables. Ante esta realidad debemos detectar estos errores para luego corregirlos. A tal efecto se envían más bits de los estrictamente necesarios. Estos bits extras aparecen por la utilización de alguno de los tipos de técnicas de control de errores. Estos mecanismos serán mejores en función de la relación entre el número de bits extra que añaden y la cantidad de errores que detectan o corrigen.

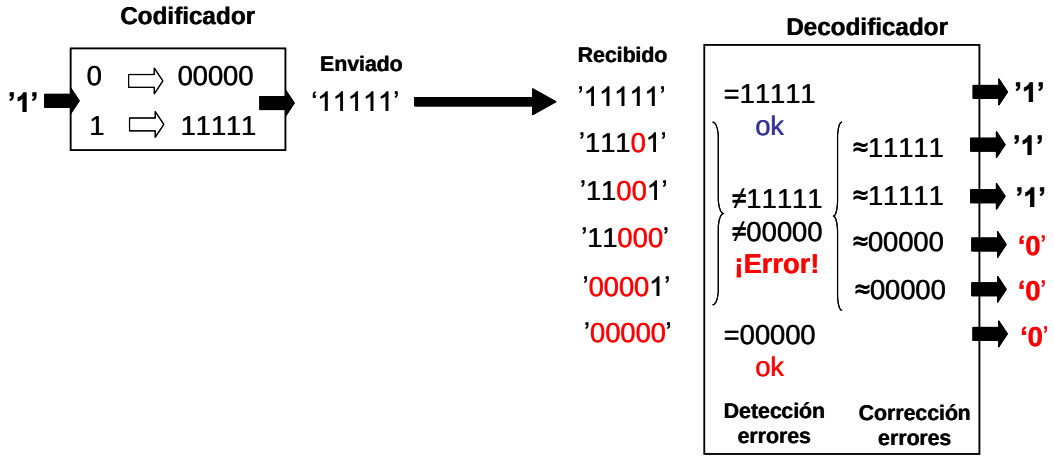

Figura 7: Ejemplo de codificación y decodificación con 6 posibles casos: 0 o más bits erróneos (bits erróneos en rojo).

Veamos un ejemplo ficticio de codificación (Figura 7). Supongamos que cada bit es enviado por quintuplicado, y el receptor primero detecta si hay error, y luego decide por mayoría cuál es el valor real del bit. Así, para enviar un '1' realmente enviamos '11111' y para un '0' enviamos '00000'. Si no se reciben todos los bits iguales, el receptor asumirá que se ha producido error y a continuación decide cuál es el valor correcto. Así, si enviamos '11111' y se recibe '11101', se detecta que ha ocurrido un error y se decide que se ha recibido un '1', lo cual es correcto. Pero si tenemos 3 errores y recibimos '10100', sabemos que ha ocurrido error, pero escogeremos un valor de '0', lo cual nos induce a una equivocación. Por lo tanto, esta técnica permite detectar hasta un máximo de 4 errores y corregir hasta un máximo de 2 errores.

Si bien no se trata de la codificación más eficiente del mundo porque estamos multiplicando por 5 los bits a enviar, permite sustentar dos ideas. La primera es que no existe codificación que detecte o corrija todos los errores. La segunda es que son necesarios más bits extra para corregir errores que para detectarlos.

Existen otras técnicas mucho más eficientes que sólo detectan errores, pero ¿de qué sirve saber que tengo información incorrecta, si no puedo corregirla? Aquí es donde echamos mano de dos conceptos nuevos: la confirmación y la retransmisión. El

receptor, en caso de que los datos lleguen correctamente, confirma con un paquete de control (denominado ACK, Acknowledgement) su recepción. En caso que el transmisor no reciba esta confirmación volverá a enviar los datos. Es lo que se conoce como un mecanismo de retransmisión automática (ARQ, *Automatic Repeat-reQuest*).

¿Y si retransmitir no es factible?

En algunos enlaces no es posible retransmitir, por ejemplo porque son unidireccionales, o no tiene sentido hacerlo por que la información, por ejemplo voz, nos llegaría con un retardo demasiado grande. Este es el caso de las redes por satélite. La mayoría de satélites de comunicaciones se encuentran en órbita Geoestacionaria, a una altura de 35.786 kilómetros. La gran distancia a la que se encuentran estos satélites supone un tiempo de propagación de más de 100 ms entre la Tierra y el satélite. Si la señal viaja de la Tierra al satélite y de éste a la Tierra para llegar al receptor y la confirmación debe realizar el camino contrario, deberemos esperar más de 400 ms a realizar una retransmisión, un tiempo demasiado grande para el caso de la voz. ¿Cuál es entonces la solución? Utilizar códigos que sí corrijan los errores, aunque se tengan que enviar muchos más bits.

1.5.2. Control de flujo: háblame más despacio, ¡qué no puedo escribir todo lo que me dices!

Con los mecanismos de control de errores aseguramos que la información llega correctamente al receptor. Pero, ¿qué sucede si el receptor no puede procesarla? Los nodos de la red, como las personas, tienen una capacidad de proceso limitada. Así puede suceder que tomando apuntes alguien nos hable tan rápido que no seamos capaces de escribir (procesar) todo lo que nos dice. Este caso trasladado al mundo de las redes podría producirse cuando me descargo una canción desde mi móvil. El servidor que me envía la canción será mucho más potente que mi móvil pero deberá "hablar" (enviar la canción) a un ritmo que el móvil le marque. Un caso totalmente diferente, pero de consecuencias similares, ocurre cuando muchas personas a la vez realizan preguntas a un profesor, que llega un momento en que ya no puede atender todas las preguntas porque no es capaz de acordarse de todas ellas. Esta situación es análoga a la que se produce cuando miles o millones de usuarios con máquinas modestas consultan a la vez un mismo servidor, por ejemplo Google. Aunque el servidor sea muy potente, para atender a la llegada de tanta información a la vez debe limitar la cantidad de información que permite enviar de una vez a cada usuario.

El mecanismo más básico de control de flujo es el mismo que utilizaban los carreteros para detener a los caballos o mulas que tiraban de los carros: gritar "Soooooo". La recepción de esta orden por parte del emisor para el envío de información hasta recibir un "Aaaarre". Pero, ¿qué sucede si la orden se pierde o no es entendida (debido a

errores de bit), o si la orden tarda mucho en llegar? En ambos casos el receptor puede verse "sepultado" por la cantidad de información recibida y que no ha podido procesar.

Otra filosofía totalmente opuesta es el denominado mecanismo de parada y espera. El emisor sólo puede enviar un trozo de información, y debe esperar a que el receptor le confirme su recepción (con un ACK) para enviar el siguiente trozo de información. Si se pierde algún paquete de datos o ha existido un error, el ACK no llegará y también servirá para darse cuenta de la situación, en este caso se volverá a enviar. Sería como decir una frase y esperar que quien me escucha asienta con la cabeza para seguir con la siguiente. Si no asiente es señal de que no nos ha entendido y se lo repetimos.

Ventanas que se deslizan:

El problema que presenta el método anterior es que si los retardos de transmisión entre los dos extremos son muy grandes, cuando el transmisor acaba de enviar un paquete tendrá que esperar mucho tiempo hasta recibir la confirmación y poder enviar el siguiente paquete. Frente a esta evidencia, aparece la ventana deslizante. La ventana tiene un tamaño que podría corresponder, por ejemplo, al número de palabras que puedo decir seguidas, sin esperar una confirmación (ACK) por parte de mi interlocutor. Por ejemplo supóngase que quiero decir la frase "Hola. Buenos días, te voy a explicar cómo funciona la ventana deslizante." y mi interlocutor me da una ventana de cinco palabras. Entonces podré decir de golpe "Hola. Buenos días, te voy" a continuación deberé esperar la recepción de una confirmación.

Cada confirmación me indica el número de paquete de datos que confirma, en nuestro ejemplo correspondería al número de la palabra dentro de la frase original. Si la confirmación llega y me indica que ha recibido "Hola" (confirmación de la palabra 1), entonces "Hola" sale de la ventana y puedo decir "a" pero deberé volver a esperar ya que la ventana ha vuelto a llegar a su máximo (Figura 8 a). Es decir, a medida que digo nuevas palabras, la ventana decrece y a medida que mi interlocutor asiente, la ventana crece.

Este deslizamiento es el que da nombre al mecanismo. También puede suceder que mi interlocutor se estrese y me pida que no le diga tantas cosas de golpe. Esta situación se representa en la Figura 8 b), en la que se modifica el tamaño máximo de la ventana a 3 en lugar de 5.

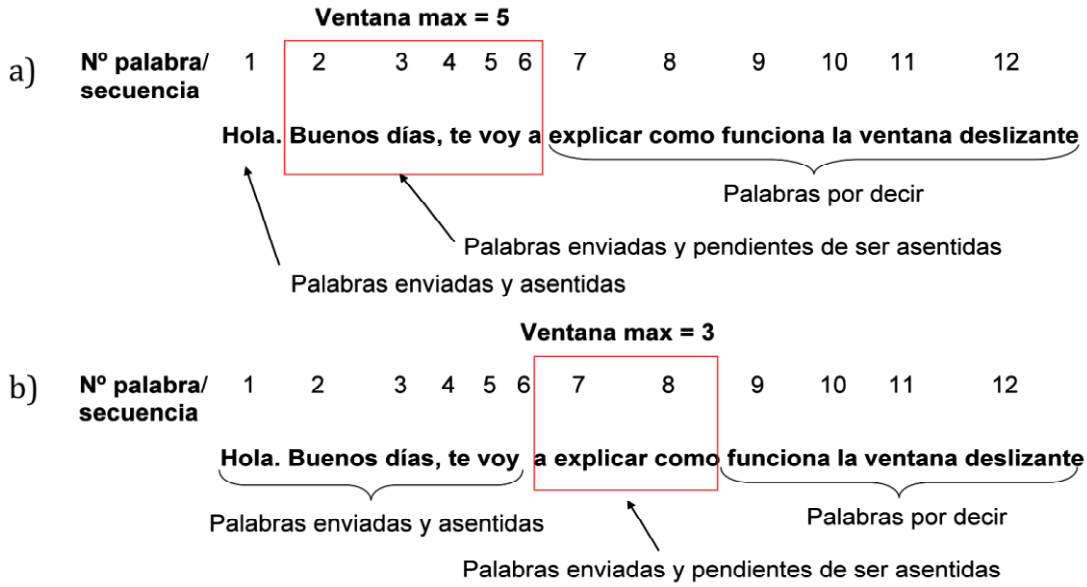

Figura 8: Ejemplo de ventana deslizante.

1.5.3. Control de congestión: quién me mandaría a mí coger el coche ...

Asegurar que el receptor ha recibido y procesado la información no es suficiente para asegurar que una red funciona bien, ya que no nos libra de uno de los problemas más grandes que afecta a todos sus usuarios: la congestión. Para entender lo que es congestión y sus funestas consecuencias basta con coger el coche a la vuelta de un fin de semana de verano para entrar en una capital. Colas interminables, que provocan el desespero de los conductores que llegan tarde a sus casas y que en algunos casos no llegan porque desisten en su intento.

En las redes de conmutación de paquetes sucede lo mismo: dependiendo de las circunstancias (tráfico que generan los usuarios) algunos de sus nodos pueden entrar en congestión. Esta se produce por sobrecarga de parte o todos sus elementos de interconexión, los conmutadores. En sus colas de entrada y salida se amontonan los paquetes a un ritmo más rápido del que puede atender. A partir de aquí, sucede como con los atascos: primero tardamos más tiempo en llegar y finalmente puede ser que ni tan sólo lleguemos (pérdida de paquetes por falta de memoria).

Llegar tarde es un problema si ese día vamos a trabajar o al médico. Lo mismo sucede si la información que se retrasa es una muestra de voz, o una imagen de un partido de fútbol en directo: puede que cuando se reciba no pueda reproducirse porque la secuencia a la que pertenecía ya ha sido visualizada.

¿Qué hacer? Es evidente que no se puede dejar el funcionamiento de la red al sentido común de sus usuarios, ya que como sucede con el tráfico en las carreteras por muchas recomendaciones que nos hagan (por ejemplo, salidas escalonadas) puede

ser que no hagamos caso. A partir de esta certeza se proponen dos soluciones: las preventivas y las reactivas. Las primeras intentan que la congestión no se produzca. Independientemente del estado de la red, limitan la cantidad de datos que el usuario puede enviar. Por ejemplo: de cada aula, el profesor sólo deja salir dos alumnos cada minuto, así a la salida del instituto no habrán aglomeraciones. Las segundas detectan la congestión para luego combatirla. Por ejemplo, al igual que la policía nos puede redirigir por una ruta alternativa, los conmutadores de paquetes pueden hacer lo mismo. No es el camino previsto pero esta menos colapsado. Otra medida es interactuar con los mecanismos de control de flujo limitando su funcionamiento, por ejemplo reduciendo temporalmente la medida de la ventana deslizante, lo que significa que el emisor va poder enviar menos información de golpe. Se trata de dar un respiro a los nodos intermedios que lo conectan con el receptor para que puedan vaciar sus colas.

Los cubos agujereados también sirven:

Sí, admitámoslo, los ingenieros telemáticos somos algo raros, pero en cualquier caso, muy prácticos. ¿Cómo limitar la cantidad de información que un usuario puede enviar por una red? Métela en un cubo y hazle un agujero. A mayor agujero, más información puede enviar el usuario y a la inversa. Pero, ¡cuidado! si echas demasiada agua (información) en el cubo te vas a mojar (desbordamiento).

Control de admisión: no es por tus zapatos, es que no cabe más gente:

Una manera muy eficiente de evitar la sobrecarga en una red es aplicar mecanismos de control de admisión. Todos hemos visto carteles en establecimientos públicos con el texto "Reservado el derecho a admisión". ¿Qué significa? Pues que quién regenta el establecimiento se reserva el derecho de no dejar entrar o de echar a quien quiera.

De la misma forma en una red, cuando alguien quiere conectarse a ella, o un usuario quiere establecer una nueva conexión, la red puede comprobar si tiene suficientes recursos libres para atender estas nuevas peticiones, y en caso contrario, desestimarlas. Otra vez en las carreteras encontramos ejemplos de control de admisión. No es raro ver mensajes en los carteles luminosos de autopistas o autovías indicando que está prohibido el tránsito de vehículos pesados durante determinados días y/o a determinadas horas. Normalmente se trata de franjas de tiempo en las que se prevé mucho tráfico, por ejemplo un puente, y lo que se busca es limitar la afluencia de ciertos usuarios en favor de otros. El objetivo final es asegurar que los que circulan lo puedan hacer sin problemas. Es decir, que no sufran congestión, y si es el caso, que se les garantice un determinado nivel de servicio.

1.6. Multiplexación y demultiplexación: juntos pero no revueltos

Si todos los caminos llevan a Roma, en las redes todos los caminos llevan al operador, es decir a la empresa que nos conecta a la red telefónica o Internet. Por sus redes, y más concretamente sus redes de troncales, circulan grandes cantidades de información. Para transportarla se utilizan mayoritariamente enlaces de fibra óptica que permiten alcanzar velocidades superiores a los Terabits por segundo (Tbps). Es decir, se puede transmitir por un enlace más de 10^{12} bits por segundo. Esto supone, por poner un ejemplo, nada más y nada menos que más de 15 millones y medio de llamadas de voz a la vez.

Además de esta gran capacidad, las redes de troncales necesitan mecanismos para compartirla ya que su finalidad es la de transportar información de un gran número de usuarios. Es decir, se trata de compartir la capacidad del enlace pero de una manera ordenada, que permita en cada momento situar los datos de un usuario para luego extraerlos correctamente. Son autopistas por las que circulan bits a toda velocidad (de aquí viene el concepto de autopistas de la información). El carril por el que circulan en un determinado tramo de la autopista determina si van salir en Zaragoza, en un enlace Barcelona-Madrid, o van a continuar hasta Madrid. Al mecanismo que realiza esta mezcla ordenada, es decir sitúa cada coche en su carril, se le llama multiplexación, y el que lo extrae para derivarlo hacia su destino demultiplexación.

Inicialmente la multiplexación de canales de comunicación se basaba en el uso de “carriles”. Cada carril se corresponde a una banda de frecuencias donde solamente tiene acceso una comunicación. Es lo que se conoce como multiplexación por división en frecuencia (FDM, Frequency-division multiplexing). Cuanto más ancho sea el carril más velocidad tendremos.

Si queremos insertar más carriles para tener más conversaciones, debemos hacerlos más estrechos. Sin embargo llega un momento que son tan estrechos que no son manejables. En este caso debemos buscar otra técnica, algo tan simple como tener carriles más anchos pero compartidos (esta técnica surgió más tarde porque es más complicada de implementar tecnológicamente). Para realizar esta compartición cada comunicación dispondrá del carril durante un cierto tiempo limitado que irá rotando por turnos. Es la multiplexación por división en tiempo (TDM, Time-division multiplexing). Piénsese en un caso en que dos empresas deben enviar información cada 10 ms por un enlace entre Madrid y Barcelona. Si el tiempo que esta información necesita para cubrir este enlace es de 5 ms, las dos empresas pueden compartir el enlace sin que se den cuenta. Si en lugar de dos empresas como estas quiero conectar a 10, deberé conseguir un enlace en el que se pueda correr cinco veces más rápido.

Finalmente está la multiplexación por división en longitud de onda (WDM, Wavelength-division multiplexing), que podríamos asimilar a superponer verticalmente diferentes autopistas. No necesitamos ocupar más superficie y ganamos en capacidad a base

construir un "piso" encima de otros. Inviable su realización con asfalto, sencillo sobre una fibra óptica, donde cada piso es una haz de luz de un color diferente.

Multiplexación estadística: el camello, si se agacha un poco, pasa por el agujero

Sea cual sea el mecanismo empleado, para que la multiplexación no sea un cuello de botella, la capacidad del enlace debe ser superior a la suma de las velocidades de las comunicaciones que confluyen en él. Por ejemplo si en un enlace confluye el tráfico de 100 ADSLs de 6 Mbps el enlace debería tener una capacidad de como mínimo 600 Mbps. Sin embargo, es de suponer que no todos los usuarios van a utilizar el ADSL a la vez, y quizás cuando lo utilicen no lo hagan a tope. El operador realiza estas suposiciones a partir de las estadísticas de tráfico de sus usuarios, y a partir de aquí decide que con un enlace de menor capacidad puede ser suficiente. Si la estimación es mala, difícilmente conseguiremos nuestros 6 Mbps, y esa es la diferencia entre una conexión de un operador "bueno" (por ejemplo que dispone de un enlace de 500 Mbps) con otro de "malo" (que dispone de un enlace de 200 Mbps).

1.7. Calidad de servicio: la importancia de ser puntual

Después de todo lo que hemos explicado, está claro que enviar información entre dos nodos/usuarios no es ni mucho menos un proceso simple. Pero que la información llegue no siempre es sinónimo de éxito en la comunicación. En algunos casos es crítico que el tiempo de entrega esté acotado. Piénsese lo que podría ser una conversación en que cada frase tarda uno o más segundos en llegar a su interlocutor o un juego en red en el que el disparo de uno de los jugadores no es percibido hasta mucho después por el jugador en el que impacta. En estas y otras situaciones se exige inmediatez en la comunicación. En otro caso quiero ver la televisión por Internet y necesito que mi operador me garantice una cierta velocidad porque sino se me va a cortar la imagen. En definitiva, que la red funcione no es suficiente sino que deben cumplirse unos mínimos en cuanto a velocidad, retardo, pérdidas. Es lo que se conoce como calidad de servicio y conseguirla es todo un reto.

La manera más simple de conseguir calidad es sobredimensionar. ¿Que necesitas un piso para vivir? Cómprate dos, que así seguro puedes meter todo lo que quieras dentro. El problema es evidente: el precio. Descartada esta opción, lo que queda es aguzar la inteligencia para repartir los recursos que forman una red, de manera que aquellos servicios o usuarios que necesiten y/o paguen por disfrutar de ciertos niveles de calidad, la obtengan. Algunas de las técnicas más empleadas son:

- Ingeniería de tráfico: el carril bus o el carril bici podrían ser un buen ejemplo, reservo un camino libre de problemas para un determinado servicio o cliente.

- Servicios diferenciados: no todos somos iguales, y ya sea porque se paga más, o un servicio se considera más prioritario, la información que genera pasa delante de la de otros. Sería como circular en un coche de policía con sirenas, o entrar de urgencias en un hospital con un traumatismo craneal pasando delante de otro paciente con una gripe.
- Servicios integrados: se pide a la red un determinado nivel de calidad y ésta dinámicamente y en función de los recursos disponibles, intenta darla. Imagínate que al salir de casa por la mañana pides llegar a tu instituto en 10 minutos. La "red" echa un vistazo al tráfico, mira por donde anda el próximo bus de tu línea y finalmente te recomienda la bici, aunque otro día te puede decir el bus. Pero no sólo eso, se va a asegurar mediante la guardia urbana que tienes el paso asegurado. Es complejo y sólo funciona sobre redes no muy grandes.

1.8. Ahorro de energía: autonomía y sostenibilidad

Para que una red funcione, los nodos que la forman deben estar alimentados. Teléfonos, PCs o equipos de interconexión (routers, puntos de acceso, centrales telefónicas,...), todos necesitan recibir alimentación de acuerdo a sus especificaciones. Pero de entre toda esta cantidad de dispositivos, existe un grupo en el que el problema de la alimentación adquiere otra dimensión. Se trata de los dispositivos móviles, ya sean teléfonos móviles, agendas, videoconsolas o portátiles. Todos necesitan de baterías para que cuando los utilicemos podamos movernos. Para estos dispositivos es crítico reducir su consumo porque esto se traduce en una mayor duración de sus baterías y por tanto en una mayor autonomía.

Este objetivo, el de aumentar la autonomía de los dispositivos móviles, ha sido y es el motor para desarrollar técnicas de ahorro de energía. Estas técnicas atañen tanto al hardware (CPU, memoria, pantalla,...), como al software (sistema operativo) del dispositivo, y también a la tecnologías radio que utilizan. Básicamente, se trata reducir la actividad del interfaz radio (por ejemplo la de una tarjeta de red Wi-Fi o un telefono 2G/3G). Así, en lugar de estar siempre activa, pasa a un estado durmiente en el que gasta menos batería pero del que sale a intervalos regulares para saber si alguien quiere conectar con el terminal, y poder seguir recibiendo una llamada o un SMS.

Sin embargo, en los últimos tiempos, el aumento del precio de la energía y la toma de conciencia sobre la necesidad de avanzar hacia un mundo sostenible, han situado el ahorro de energía como un objetivo clave en el desarrollo tecnológico de las redes de comunicaciones y uno de los motores de la investigación asociada a este desarrollo. A finales del primer decenio del siglo XXI, no es extraño encontrar el prefijo verde (*Green*) vinculado a las redes de comunicaciones como símbolo de su objetivo de

reducir el consumo, y que empresas como Google se planteen montar sus servidores en países fríos o en barcos en alta mar para facilitar su refrigeración.

Pero, ¿realmente consumen tanto las redes? Un estudio publicado en el 2008 sitúa el consumo de sólo los EEUU en 112 TWh, equivalente a 6 centrales nucleares, con una factura asociada de 15 billones de dólares. Ante tales "minucias", ya hay quien se dedica a calcular las emisiones de CO_2 asociadas a realizar una búsqueda con Google.

1.9. Seguridad:

Muchos siglos antes de la Era Digital, la seguridad de la información era ya considerada una cuestión importante. Julio Cesar utilizaba técnicas de *cifrado* o *encriptación* para evitar que los mensajes enviados a sus generales fueran descifrados en el caso de que los mensajeros fueran "interceptados" por el enemigo. Era una técnica bien sencilla, consistía en un alfabeto "desplazado": cada letra del mensaje es reemplazada por otra, a un cierto número de letras de distancia. El código de Cesar usaba un desplazamiento de 3: una *A* se convertía en una *D*, una *B* en una *E*, etc. Por ejemplo, una famosa frase de Julio Cesar – "Apresúrate despacio" – quedaría: "Dsuhvxudwh ghvsdflr"…irreconocible. Estos mecanismos siguieron mejorando a lo largo de la historia. Durante la II Guerra Mundial, por ejemplo, fueron muy conocidas las máquinas Enigma, una especie de máquina de escribir utilizada por los alemanes que cifraba automáticamente los mensajes. Los aliados lograron romper esos códigos con complejos análisis matemáticos y con la ayuda de los primeros ordenadores.

A → D	J → M	R → U
B → E	K → N	S → V
C → F	L → Ñ	T → W
D → G	M → O	U → X
E → H	N → P	V → Y
F → I	Ñ → Q	W → Z
G → J	O → R	X → A
H → K	P → S	Y → B
I → L	Q → T	Z → C

Figura 9: Código de Cesar y máquina enigma

Hoy en día, cualquiera de esas técnicas sería muy poco útil para guardar ningún secreto. Actualmente se requieren mecanismos mucho más complicados. El código de Cesar o la máquina Enigma proporcionaban confidencialidad, pero hay otros conceptos sobre seguridad que son necesarios en las redes actuales:

- La **confidencialidad** busca que la información sea sólo accesible para aquellas personas autorizadas. Esto se consigue normalmente mediante la *criptografía*, o el arte de "esconder" la información. Como hacía Julio Cesar, sólo aquellos que conocían el método (alfabeto desplazado) y la clave de cifrado (desplazamiento de tres letras) podían acceder a la información, que permanecía oculta e irreconocible para todos los demás ¿Hpwlhpghv? Cuando la clave de cifrado es compartida por todos los interesados, se habla de un sistema de *cifrado de clave simétrica*, ya que la misma clave se usa para cifrar y para descifrar. De entre estos sistemas, uno de los más populares actualmente es el AES (*Advanced Encryption Standard*), usado por el gobierno de los EEUU para cifrar su documentación "clasificada", y también usado en redes Wi-Fi con WPA2.
- En otros casos puede que no importe que otros puedan leer el mensaje, pero sí que es vital poder estar seguro que nadie ha alterado mensaje por el camino (accidental o intencionadamente). Esta propiedad se conoce como **integridad** de los datos. Para asegurar la integridad se usan técnicas *checksum* o *hash*, como MD5 o SHA-1. A partir de los bits de una información digital, la función de *hash* proporciona un conjunto de bits mucho más reducido que en la información original, un resumen. Estos bits se añaden al final de los datos. El receptor de esa información hace la misma operación con los datos, obtiene un *hash* y lo compara con el *hash* que venía con el mensaje. Si se ha cambiado un solo bit, el resultado será totalmente diferente y el receptor detectará que los datos no son fiables.
- Además de integridad y confidencialidad, la seguridad se garantiza si se cumplen otras propiedades. La **disponibilidad** consiste en asegurar que la información está siempre disponible (para aquellos usuarios autorizados), eso incluye la defensa contra ataques de denegación de servicio (DoS), que consisten en sobrecargar una máquina o una red, de manera que se impide que los usuarios legítimos puedan acceder. El **no repudio** es la propiedad que evita que alguien pueda negar algo que efectivamente ha hecho. Finalmente, la **autenticidad** permite garantizar que una información proviene realmente de quien creemos que proviene.

Criptografía asimétrica

Antes ya se han mencionado las técnicas de clave simétrica (misma clave para cifrar y para descifrar), pero también hay *criptografía asimétrica*, donde se usan dos claves diferentes: una se denomina clave pública y la otra privada. Ambas están relacionadas. Si se cifra con la clave pública se debe descifrar con la privada, y si se cifra con la privada se debe descifrar con la pública. Si se intenta cifrar y descifrar con la misma no funciona.

Pero, ¿para qué nos complicamos la vida así? La idea es que nadie más que el destinatario pueda descifrar la información. Si utilizamos un sistema de clave simétrica, donde todos los usuarios conocen la clave para cifrar y descifrar, todos podrán ver el contenido de todos los mensajes (poco seguro). En cambio si utilizamos clave asimétrica, si alguien quiere enviarme algo a mí, cifra el contenido del mensaje con mi clave pública (que todos conocen), pero solamente yo, con mi clave privada, seré capaz de descifrar el mensaje. Si una tercera persona, que también conoce mi clave pública, intenta descifrar el mensaje no podrá, ya que como hemos dicho, el sistema no funciona si se intenta descifrar con la misma clave con la que se ha cifrado. Lo único que tengo que hacer es estar seguro de no decir a nadie mi clave privada.

Los sistemas de clave pública se usan también para la **firma digital** (ver Figura 10)**.** La firma digital sirve para estar seguro de que quien envía un mensaje es realmente quien dice ser. La idea se basa en que esta persona es la única que conoce su clave. Para firmar digitalmente un mensaje calculamos un resumen del mensaje (*hash*) y a continuación lo ciframos con nuestra clave privada, el resultado es la firma del documento. Quien quiera autenticar este mensaje, toma la firma del documento y con la clave pública del usuario que lo ha firmado se descifra el resumen. Por otra parte, con el documento recibido se calcula otra vez el resumen, si ambos coinciden querrá decir que quien lo ha firmado es realmente el dueño de la firma privada correspondiente, y como nadie más la conoce, tiene que ser la persona que dice que es (autenticidad y no repudio).

¿Qué pasa si otra persona pone al alcance de todo el mundo su clave pública y va diciendo que es la mía? Pues que esa persona podrá descifrar toda la información que me envíen a mí porque los demás creerán que aquella clave pública es realmente la mía, cuando no lo es. Además esa persona podrá firmar documentos digitalmente, y todo el mundo creerá que yo soy el autor. Para evitar esto, la clave pública suele estar acompañada de un **certificado digital**. Estos certificados provienen de sitios “de confianza” para todo el mundo, como la Fábrica Nacional de Moneda y Timbre, el Ministerio de Interior, etc. Cuando alguien quiere estar seguro de que una clave pública es auténtica, consulta el certificado y pregunta a esa entidad de confianza si la clave pública es realmente la que corresponde a ese usuario.

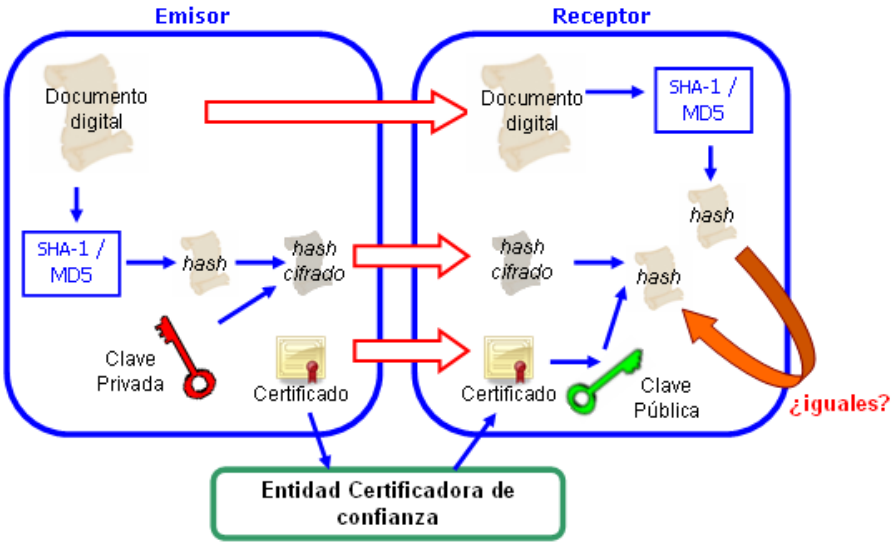

Figura 10: Proceso de comprobación de firma digital

Seguridad de las redes

También se necesita seguridad en las redes, no sólo para salvaguardar la información, sino sus recursos. Dicho de otra forma, se debe asegurar que un vecino malintencionado no usa tu router Wi-Fi, ni que otra persona cambia tus fotos del Facebook, etc. Esto se consigue mediante mecanismos **AAA** (Authentication, Access control, Accounting), que en inglés viene a ser: autenticación, control de acceso y contabilidad.

La primera A es la que se encarga de identificar a los usuarios (o máquinas) que quieren acceder a una red o a un servicio. Esto se suele hacer mediante un identificador de usuario (login) y una clave (password). Se supone que si conoces ambos datos, eres sin duda el auténtico usuario que dices ser. A veces con una clave basta, como en algunas redes Wi-Fi: cualquier usuario que conozca ese password, es un usuario legítimo.

Una vez te has identificado, mediante el control de acceso se establecen los permisos que tienes en esa red. Por ejemplo, un usuario autenticado de Facebook sólo tiene permiso para modificar su propio perfil, y no puede cambiar el de otros usuarios.

Finalmente, la contabilidad permite llevar la cuenta del uso que hace el usuario. Esta información se usa después para temas de facturación, planificación, u otros propósitos. Son estos mecanismos los que vigilan que tu buzón no ocupe más espacio del que te toca si tienes una cuenta de correo Gmail o Hotmail, o los que calculan cuánto tiempo has hablado por teléfono para cobrarte a fin de mes.