

# Fundamentos de Redes



Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Doble Grado en Ingeniería Informática y Matemáticas  
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

# Fundamentos de Redes

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Irina Kuzyshyn Basarab  
José Juan Urrutia Milán

Granada, 2023-2024



# Índice general

|  |           |
|--|-----------|
| <b>1. Introducción a los fundamentos de redes</b>        | <b>5</b>  |
| 1.0.1. Introducción . . . . .                            | 5         |
| 1.1. Sistemas de comunicación y redes . . . . .          | 5         |
| 1.1.1. Motivación para usar redes . . . . .              | 6         |
| 1.1.2. Topologías de redes . . . . .                     | 7         |
| 1.1.3. Clasificación de redes . . . . .                  | 8         |
| 1.1.4. Nomenclatura típica en figuras (Iconos) . . . . . | 8         |
| 1.2. Diseño y estandarización de redes . . . . .         | 9         |
| 1.2.1. Modelo OSI vs TCP/IP . . . . .                    | 10        |
| 1.3. Terminología, conceptos y servicios . . . . .       | 10        |
| 1.3.1. Definiciones importantes . . . . .                | 11        |
| 1.3.2. Retardos en la comunicación . . . . .             | 12        |
| 1.3.3. Tipos de servicios . . . . .                      | 12        |
| 1.4. Internet: topología y direccionamiento . . . . .    | 13        |
| 1.4.1. Organización topológica . . . . .                 | 13        |
| 1.4.2. Red Iris . . . . .                                | 13        |
| 1.4.3. Direccionamiento por capas . . . . .              | 14        |
| <b>2. Relaciones de Problemas</b>                        | <b>15</b> |
| 2.1. Introducción . . . . .                              | 15        |



# 1. Introducción a los fundamentos de redes

## Objetivos

- Conocer y comprender los principios básicos de las comunicaciones.
- Entender el diseño funcional en capas de las redes y los conceptos y terminología fundamentales involucrados.
- Comprender desde un punto de vista teórico-conceptual el modelo de referencia OSI y su correspondencia con el modelo de capas usado en Internet.

### 1.0.1. Introducción

La arquitectura lógica de Internet está diseñada por capas. Veamos el modelo TCP/IP (también veremos el modelo OSI):

|                                   |
|-----------------------------------|
| Aplicación que hace uso de la red |
| Transporte ( <b>TCP/UDP</b> )     |
| Red ( <b>IP</b> )                 |
| Enlace                            |
| Física                            |

Tabla 1.1: Modelo de capas del protocolo TCP/IP

Las dos últimas capas están implementadas en Hardware y las tres primeras en Software, también llamado **NOS** (Network Operative System). En la asignatura veremos las capas altas.

## 1.1. Sistemas de comunicación y redes

**Definición 1.1** (Sistema de comunicación). Es una infraestructura (hw + sw) que permite el intercambio de información. Un sistema típico es el siguiente:

- Tenemos una fuente y un transmisor en un mismo equipo (que es el que va a mandar la información). La fuente genera la información y el transmisor adapta la información al medio.

- Después tenemos el canal de comunicación, el cual produce errores: ruidos, interferencias, diafonías (cuando hay muchos cables en paralelo juntos, puede suceder que la información de un cable se meta en otro).
- Al final tenemos un receptor y el destino (en un mismo equipo). El receptor adapta la información para el destino y éste espera los datos a recibir.

### 1.1.1. Motivación para usar redes

Para verlo, vamos a hablar de la primera red de comunicaciones que era una red de telefonía móvil. Cada usuario contaba con su línea de teléfono, que conectaba con una central de conmutación local, luego regional y luego nacional, la cual debía conectarse con la central local del destino. Se usaba conmutación de circuitos.

- Inicialmente se creaba un camino físico juntando cables, llamado circuito.
- Era ineficiente porque no se está hablando todo el tiempo, y los tiempos de silencio el circuito se desaprovecha.
- Era un problema de seguridad el mal funcionamiento de una central, pues dejaba a miles de teléfonos sin servicio.

Si ahora pensamos en ordenadores (o equipos más generales, móviles, PCs, portátiles, móviles...) en vez de móviles y cambiamos las centrales de conmutación por routers, contamos con muchísimos caminos para conectar dos ordenadores, haciendo más segura la red.

Ahora ya no tenemos un camino físico, sino que son los routers quienes deciden por dónde enviar los paquetes y en qué momento. Los routers tienen colas, lo que genera algo de retardo, pero tiene la ventaja de que se usa mucho mejor el canal y hay más seguridad, pues hay más caminos que solo uno.

De una red esperamos:

- Autonomía.
- Interconexión.
- Intercambio de información con eficacia y transparencia.

**Definición 1.2** (Red). Sistema de comunicación con sistemas finales o terminales autónomos (con capacidad para procesar información) que facilita el intercambio eficaz y transparente de información. Concretamente tenemos:

- **Hosts:** sistemas finales o terminales autónomos. Los que transmiten y reciben datos.
- **Subred:** infraestructura para el transporte de información, formada por líneas de transmisión y nodos o elementos de conmutación: routers y switches.



En cuanto a medios de transmisión, originalmente se usaban cables de pares (pensado para transmitir 4kHz, la media de la voz humana), luego cables coaxiales, que mejoraron mucho; y fibra óptica que puede transmitir sin interferencias, por lo que es el mejor medio guiado existente. Los cables trenzados son para distancias más cortas, Ethernet por ejemplo.

### 1.1.2. Topologías de redes

**En bus:** es la más sencilla, pero como el medio es común, todos intentan acceder y se producen colisiones.

**En anillo:** un círculo en el que tenemos los distintos computadores. Es similar al bus pues el medio es compartido. Una versión habitual es **token ring**, testigo de anillo. Como el medio es común, se van pasando el testigo evitando así colisiones.

**En estrella:** todos están conectados a un centro principal, típicamente un switch.

A diferencia del bus, en este caso cada cable es independiente del resto. Si un PC pone algo en una toma el resto no lo escuchan. Cada línea tienen una cola para guardar a dónde enviar los paquetes y el switch tiene un procesador que coge los paquetes de dichas colas y los envían a las salidas. Es una topología mucho más segura por el hecho de no compartir el medio.

**En árbol:** típica en redes empresariales. Se suele estructurar en tres niveles:

- Primer nivel: red troncal.
- Segundo nivel: red de división.
- Tercer nivel: red de acceso.

Los equipos de primer y segundo nivel suelen ser switches. Pueden aparecer ciclos en el árbol. Ethernet no tiene ningún mecanismo para evitar que un paquete se mueva en círculo, lo que echa la red abajo.

El protocolo STP hace que cualquier topología quiten enlaces redundantes que forman bucles.

**Mallada:** todos los nodos están conectados entre sí por medios independientes. Es muy fiable, si se cae un enlace tienes más caminos para llegar a tu destino. Pero no es escalable, si metemos un nodo  $n$ -ésimo, hay que meter  $n - 1$  enlaces. Pero para redes pequeñas está bien. Dentro de una empresa la red troncal puede seguir esta topología para evitar caídas importantes.

**Híbrida:** se usa una mezcla de todas. Es la más utilizada.

**Definición 1.3** (CSMA/CD). Ethernet para compartir un medio común como un bus utiliza CSMA/CD: acceso múltiple sintiendo la portadora (Carrier Sense). CD viene de “detecta colisiones”: si lo que hay en el cable no es lo que ha puesto (si hay más ruido), da error.

**Definición 1.4** (CSMA/CA). Wifi usa este protocolo, no puede escuchar el medio, primero escucha que no haya nadie, envía el mensaje y recibe confirmación. Si no recibe confirmación, hay colisión.

### 1.1.3. Clasificación de redes

#### Según tamaño y extensión:

- LAN (Local Area Network). Red de área local, suele ser el mismo edificio.
- MAN (Metropolitan Area Network). Red de área metropolitana, para conectar un campus o una ciudad.
- WAN (Wide Area Network). Red de área extensa, redes disponibles en todo el país, como las redes telefónicas.
- PAN (Personal Area Network). Red de área personal, todo lo que puede tener una persona, relojes, portátiles, cascos. . .

#### Según tecnología de transmisión:

- Difusión: lo que pone alguien en el medio le llega a todos. HUB.
- Punto a punto: Sólo estoy unido a un nodo. Switch.

#### Según el tipo de transferencia de datos:

- Simple: solo transmite o recibe. Por ejemplo los TDT (para que una televisión analógica reciba señal digital).
- Half-duplex: transmite y recibe pero no simultáneamente. Por ejemplo el WIFI, aunque cambia muy rápido, por lo que no nos damos cuenta.
- Full-duplex: transmite y recibe simultáneamente. Por ejemplo Ethernet.

### 1.1.4. Nomenclatura típica en figuras (Iconos)

**HUB:** es un concentrador: permite centralizar los nodos de una red de computadoras, se implementa mediante un bus.

**Bridge:** funciona como un switch, pero uniendo tecnologías distintas, funciona a nivel 2.

**Switch:** tiene muchas bocas y tiene LAN. Necesitan un router para conectarse a otro switch.

**Router:** tiene pocas bocas y lo que hace es conectar distintas redes.

**Cortafuegos:** bloquea el acceso no autorizado, permitiendo el autorizado.

**NAT:** traducción de direcciones de red.

**Switch multicapa:** (no entra) todas las bocas de un switch son de la misma LAN, pero esto a veces no nos interesa. Podemos hacer redes virtuales (VLAN) dividiendo un mismo switch en varias redes, permitiéndonos esto conectar dos switches distintos con la misma VLAN. Pero esto no nos permite movernos por distintas redes en el mismo switch, habría que pasar por un router, ya que necesitamos movernos a nivel de red para cambiar de red. Esto sí se puede hacer con un switch multicapa.

## 1.2. Diseño y estandarización de redes

La idea principal que se sigue al diseñar redes es solucionar los problemas en capas. Se estandarizan Modelos de Referencia (no son implementaciones, solo una referencia): definición de las capas y las funcionalidades de cada una. Los principios que se siguen son que las funcionalidades distintas tienen que estar en distintas capas y minimizar el flujo de información entre las capas.

A continuación detallamos los problemas que tiene que resolver una red por capas:

**A nivel físico:** hay que ver como transmitir los datos. Hay distintos tipos de codificaciones para enviar bits de información.

**Capa de enlace:** se encarga de los mecanismos de acceso al medio. Si hay un medio común, antes de transmitir datos tiene que asegurarse de que ningún equipo está transmitiendo. Suele seguir dos protocolos:

- MAC, control de acceso al medio.
- LLC, control de acceso lógico para las primeras retransmisiones. Si algún paquete llega mal, retransmite varias veces.

**Capa de red:** una vez llegado a este punto, se asume que no han habido colisiones en la comunicación. Esta capa se encarga principalmente de:

- El direccionamiento: saber dar una dirección y tener un identificador dentro de la red.
- El encaminamiento: saber cómo llegar al destino.

**Capa de transporte:** se encarga de recuperar los paquetes que en la capa de enlace no se ha podido, es la capa de la fiabilidad.

- Corrige errores.
- Gestiona la congestión.
- Control de flujos: si hay un receptor más lento que el emisor, debe decirle al emisor que disminuya la velocidad de emisión, para adecuarse a la del receptor.

Además se encarga de la multiplexación de datos: mediante puertos (los veremos más adelante) le indica al SO a qué aplicación corresponde cada paquete.

**Capa de aplicación:** los clientes y los servidores deben buscar alguna forma de comunicarse.

### 1.2.1. Modelo OSI vs TCP/IP

|              |
|--------------|
| Aplicación   |
| Presentación |
| Sesión       |
| Transporte   |
| Red          |
| Enlace       |
| Física       |

Tabla 1.2: Modelo OSI.

|                |
|----------------|
| Aplicación     |
| Transporte     |
| Red            |
| Red subyacente |

Tabla 1.3: Modelo TCP/IP.

El modelo OSI fue propuesto por la ISO y el TCP/IP por el Internet Engineering Task Force. Las tres primeras capas del modelo OSI se corresponden con la capa de aplicación, y las dos última con la red subyacente, que es la parte física. Esta última en el modelo TCP/IP depende un poco de la tecnología está implementada de una forma u otra, pero la comunicación con la capa de red no puede variar, pues la capa de red si está estandarizada.

- Las capas físicas solo se encargan de hacer la primera conexión.
- La capa de red, salto a salto se encarga de llegar al destino, usando routers y sus tablas de encaminamiento.
- Una vez hecho el encaminamiento, las capas superiores son solo de los extremos, son los computadores de los extremos los que se comunican.
- Por tanto, los computadores tienen las 5 capas (en el caso de TCP/IP) y los routers solo las 3 más bajas.

## 1.3. Terminología, conceptos y servicios

En la Figura 1.1 vemos el camino que siguen los datos que le manda un emisor a un receptor. Cada capa le pone una cabecera a lo que le llega de arriba. A este proceso se le llama encapsulamiento. La capa física por su parte manda señales eléctricas en vez de bits.

Por otra parte, cuando la información va subiendo por las capas del modelo, cada capa le quita la cabecera correspondiente y ve si tiene que hacer algo.

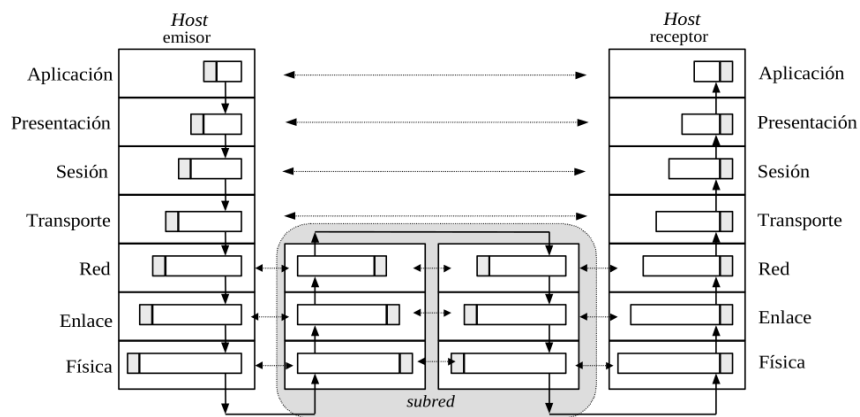


Figura 1.1: Comunicación real frente a comunicación virtual.

Las flechas continuas es por donde realmente pasa la información y las discontinuas representan la comunicación virtual u horizontal, que en breves definiremos.

Las capas del mismo tipo en distintos equipos hablan directamente en sentido abstracto por medio de otras capas. Aunque solo se habla entre las capas del mismo tipo se usan recursos que nos proporcionan otras capas adyacentes.

### 1.3.1. Definiciones importantes

**SDU:** unidad de datos de servicio. Son los datos de una determinada capa.

**PDU:** unidad de datos de paquete. SDU + cabecera.

**Comunicación real:** o vertical. Es por donde realmente va la información.

**Comunicación virtual:** u horizontal. Es la comunicación entre capas del mismo tipo en distinto equipo. No es una comunicación directa, sino a través de otras capas.

**Entidades pares:** entidades iguales, las que hablan entre sí.

**Protocolo:** cómo se comunican las entidades pares, los paquetes que se mandan.

**Interfaz:** cómo hablan las entidades adyacentes.

**Servicio:** lo que proporciona una capa a otra.

**Capa proveedora/usuario del servicio:** una capa es proveedora de la de arriba y usuaria de la de abajo.

**Pila de protocolos:** los protocolos de las distintas capas.

**Arquitectura de red:** modelo de referencia + pila de protocolos.

### 1.3.2. Retardos en la comunicación

Supongamos que queremos transmitir un paquete entre dos equipos (terminales), por medio de otro, un router. Veamos los retardos que tenemos en la comunicación.

- Para empezar, tenemos que saber el tamaño del paquete: en Ethernet un paquete típico son  $1500\text{Bytes} \cdot 8^{\text{bits}}/\text{Byte}$ . Luego el tiempo que se tarda en transmitir el paquete sale como resultado de dividirlo entre los bps. Esto es el **tiempo de transmisión** que depende exclusivamente de la tarjeta de red.
- Una vez transmitido, hay un retardo, que se corresponde con el **tiempo de propagación** ( $\text{distancia}/\text{velocidad de transmisión}$ ), que es el tiempo que ocurre entre que se escribe el primer bit y que llega el primer bit al siguiente equipo. La velocidad de transmisión inalámbrica es a la velocidad de la luz y la de un cable suele rondar  $2/3$  de la velocidad de la luz. En los cables se pierde un bit por cada millón de bits, y suele deberse a colas llenas.
- Cuando el paquete llega a un equipo intermedio, en nuestro caso el router; éste lo mete en una cola hasta que pueda procesarlo. El tiempo que está el paquete en dicha cola depende de la situación del equipo. Luego el tiempo que está el paquete en dicho equipo es el **tiempo de cola** más el **tiempo de procesamiento** (del orden de milisegundos) más el **tiempo de acceso al medio**.
- Ahora se envía el paquete al equipo destino, el tiempo de propagación, pese a calcularse de forma análoga al anterior, probablemente no coincida debido a distintas distancias, tarjetas de red. . . .

### 1.3.3. Tipos de servicios

Relacionado con el nivel de transporte, hay dos clasificaciones importantes: orientadas a conexión y a fiabilidad.

- Orientado a conexión (SOC): antes de mandar un paquete comprueba que el otro equipo esté encendido.
- Orientados a la fiabilidad: se asegura de que todo funcione bien (que todos los bits de un archivo estén bien). Además, si algo falla la conexión se termina.

Para tener un protocolo fiable, contamos con los siguientes mecanismos:

- Control de conexión. Ser fiable implica ser orientado a conexión.
- Control de errores.
- Control de congestión. Hablamos de congestión cuando las colas de los routers se llenan y empiezan a descartar paquetes.
- Control de flujo.
- Entrega ordenada. Si se envían muchos paquetes, estos deben llegar en orden.

Algunos protocolos que estudiaremos más adelante:

- TCP es un servicio orientado a conexión y fiable.
- UDP es un servicio no orientado a conexión y no es fiable. La finalidad de este protocolo es ser rápido.

## 1.4. Internet: topología y direccionamiento

Internet tiene dos cosas importantes:

- Los protocolos de comunicación.
- Cómo se gestiona Internet. Las direcciones IP son únicas en todo el mundo (salvo algunas, ya lo veremos).

### 1.4.1. Organización topológica

Los operadores se establecen según la siguiente jerarquía:

- **Tier 3:** los más cercanos a los usuarios, ISPs (Internet Service Provider).
- **Tier 2:** necesitan pasar por una Tier 1 para llegar a toda Internet y ofrecen servicios de conectividad a operadores de Tier 3.
- **Tier 1:** los que componen la estructura troncal de Internet. Están todos comunicados entre sí y hay como mínimo en dos continentes.

Hay dos tipos de relaciones entre operadores:

- **Tránsito:** conexiones entre distintos tier. Por ejemplo un tier 3 paga a un tier superior para enviar datos.
- **Peering:** conexiones entre el mismo tier.

Antiguamente, para que un ISP de un país hable con otro del mismo país había que ir hasta EEUU por falta de recursos. Más tarde se pusieron puntos neutros en cada país para comunicar operadores dentro de un mismo país.

### 1.4.2. Red Iris

Es la red española para investigación. Todas las universidades públicas y centros de investigación están conectados a ella.

La red se divide según autonomía y por otro lado tiene conexiones externas con la red científica europea.

### 1.4.3. Direccionamiento por capas

- Enlace: depende de la tarjeta de red. Necesitamos conocer la dirección MAC del siguiente punto. Las direcciones MAC son de la siguiente forma: AA:BB:CC:DD:EE:FF. Son en teoría únicas en todo el mundo. Pero realmente se ponen aleatorias para evitar seguimientos (puesto que si sabemos la dirección MAC de una tarjeta podemos hacer seguimiento de paquetes).
- Red: direcciones IP: A.B.C.D. Las públicas son únicas en todo el mundo, las privadas no.
- Transporte: direccionamiento a través de puertos, que identifican a qué proceso va un determinado paquete.
- Aplicación: Nombres de dominio mediante DNS.



## 2. Relaciones de Problemas

### 2.1. Introducción

**Ejercicio 2.1.1.** Explique brevemente las funciones de cada una de las capas del modelo de comunicación de datos OSI.

El modelo de comunicación de datos OSI cuenta con 7 niveles o capas:

1. Capa física: Se encarga de la parte física de la transmisión de los datos. Encontramos distintas formas de codificar los bits para su envío.

Realiza funciones adicionales, como la codificación del canal.

2. Capa de enlace: Se encarga de los mecanismos de acceso al medio. En caso de haber un medio compartido por varios dispositivos, debe encargarse de no transmitir datos cuando otro medio lo está haciendo y de hacerlo cuando el canal se encuentre libre.

En esta capa nos encontramos con los protocolos MAC y LLC.

3. Capa de red: Se encarga principalmente del direccionamiento de equipos (saber dar una dirección y tener un identificador dentro de la red) y del encaminamiento de datos (saber cómo mandar los paquetes al destinatario).

4. Capa de transporte: Se encarga principalmente de la fiabilidad de las comunicaciones:

- Corrección de errores.
- Manejar el congestionamiento de la red.
- Controlar el flujo de datos (reducir velocidades si el receptor no es capaz de adecuar la velocidad de recibo a la de envío).
- Realizar la multiplexación de los datos (ya que un mismo equipo puede tener varias aplicaciones que estén recibiendo datos a la vez).

5. Capa de sesión.

6. Capa de presentación<sup>1</sup>.

7. Capa de aplicación: Se encarga de decidir qué datos envía a qué equipo, así como de interpretar los datos recibidos por otros equipos.

---

<sup>1</sup>No se han mencionado en clase las funcionalidades de las capas de presentación ni de sesión.

**Ejercicio 2.1.2.** Si la unidad de datos de protocolo en la capa de enlace se llama trama y la unidad de datos de protocolo en la capa de red se llama paquete, ¿son las tramas las que encapsulan los paquetes o son los paquetes los que encapsulan las tramas? Explicar la respuesta.

Son las tramas las que encapsulan a los paquetes, ya que son las capas inferiores (en este caso, la de enlace) las que encapsulan la información de las capas superiores (en este caso, la de red) para su envío.

De esta forma, los paquetes son la PDU (*Protocol Data Unit*) de la capa de red, que se convierte en el SDU (*Service Data Unit*) de la capa de enlace, la cual añade su cabecera al mismo convirtiéndolo en su PDU.

**Ejercicio 2.1.3.** Averigüe qué son los sistemas de representación de datos “*Little Endian*” y “*Big Endian*”. ¿Puede un host que utilice representación *Little Endian* interpretar mensajes de datos numéricos provenientes de un host que utilice representación *Big Endian* y viceversa? Discuta la respuesta.

Sí que puede, para ello, debe haber un determinado protocolo que permita indicar qué codificación llevan los datos en binario. De esta forma, en alguna parte de la cabecera de los paquetes enviados, debe haber un bit que indique si los valores numéricos que se envían estén en *Big Endian* o en *Little Endian*.

**Ejercicio 2.1.4.** Cuando se intercambia un fichero entre dos hosts se pueden seguir dos estrategias de confirmación. En la primera, el fichero se divide en paquetes que se confirman individualmente por el receptor, pero el fichero en conjunto no se confirma. En la segunda, los paquetes individuales no se confirman individualmente, es el fichero entero el que se confirma cuando llega completo. Discutir las dos opciones.

Suponiendo que enviamos  $n$  paquetes de datos, la primera forma envía de vuelta al emisor  $n$  paquetes de confirmación, uno por cada paquete. De la segunda forma, el receptor espera a unir todos los paquetes en un fichero completo (y a verificar que no se ha perdido ningún paquete del mismo) para enviar el mensaje de verificación.

De la segunda forma se envían menos mensajes de verificación al emisor, por lo que la posibilidad de congestión de red por paquetes de verificación es menor. Sin embargo, en caso de que un paquete no consiga llegar o llegue en mal estado, no será hasta el final del envío de todos los paquetes que el receptor no genere el mensaje al emisor, por lo que en caso de errores en la comunicación, hay un mayor tiempo en la comunicación, al tener que esperar a que el receptor tenga todos los paquetes. Además, en el caso de error, el receptor no informará de qué paquete ha llegado mal, por lo que deberá pedir al emisor que reenvíe todos los paquetes de nuevo.

Resumiendo, ambas estrategias de confirmación tienen sus pros y sus contras. Dependiendo de la situación (si queremos mayor velocidad en la comunicación o si queremos menor saturación de red), puede interesarnos una u otra.

**Ejercicio 2.1.5.** ¿Para qué sirve el programa *ping*? ¿y el programa *traceroute*?

El programa *ping* usa el protocolo ICMP para enviar un paquete a un equipo, el cual tratará de responder con un paquete de confirmación de recepción del primer paquete. Sirve para comprobar la conexión y el buen funcionamiento de la red existente entre dos equipos. También sirve para calcular empíricamente la latencia de la conexión.

Por otra parte, el programa *traceroute* sirve para consultar todos los nodos intermedios por los que pasan los paquetes que salen de un emisor y llegan a un receptor, junto con la latencia de cada salto. Se usan varios paquetes ICMP con un valor creciente del campo TTL (*Time To Live*) para que cada salto intermedio devuelva un paquete de error ICMP con el valor del TTL que ha recibido. De esta forma, el emisor puede saber cuántos saltos intermedios hay entre él y el receptor, así como la latencia de cada uno de ellos.

**Ejercicio 2.1.6.** ¿Qué protocolos de un paquete puede cambiar un router? ¿En qué circunstancias?

Puede cambiar el TTL (*time to life*) del paquete<sup>2</sup>.

**Ejercicio 2.1.7.** Averigue qué ISPs operan en España.

Algunos de los ISPs (*Internet Service Providers*) que operan en España son:

- Movistar.
- Vodafone.
- Orange.
- Jazztel.
- MásMóvil.
- Yoigo.
- Digi.

**Ejercicio 2.1.8.** ¿Qué es una aplicación cliente-servidor? ¿y una aplicación *peer-to-peer*?

Una aplicación cliente-servidor es una aplicación que depende de otra que probablemente esté en un equipo remoto (llamado servidor) para su funcionamiento.

Un ejemplo de aplicación cliente-servidor es una página web: tenemos aplicaciones que se ejecutan en local en cada equipo que accede a una determinada url. Dicha aplicación solicita datos a una aplicación que se encuentra en un equipo remoto, la cual proporciona datos (por ejemplo, accediendo a una base de datos) como respuesta a los datos solicitados por la aplicación que se ejecuta en cada equipo de forma local.

---

<sup>2</sup>Todavía no se ha mencionado en clase nada más acerca de esto.

Por otra parte, una aplicación *peer-to-peer*<sup>3</sup> es una aplicación que se distribuye entre varios equipos (que pueden estar muy lejanos entre sí) de forma que todas las aplicaciones tienen la misma relevancia en el buen funcionamiento del sistema.

**Ejercicio 2.1.9.** Describa brevemente la diferencia entre un *switch*, *router* y un *hub*.

Para responder a la pregunta, usamos además información que hemos aprendido en el Tema 2:

- Un *switch* es un nodo en una red que permite conectar tantos equipos como deseemos (normalmente, estos tienen 48 bocas de entrada RJ45 en el caso de conectar los equipos por ethernet) a una red. Funcionan a nivel de enlace, luego no tienen una dirección IP asociada.
- Un *router* es un nodo en una red que permite conectar redes distintas entre sí. Para ello, disponen de distintas tarjetas de red, cada una asociada a una red que se encuentra conectada al router. Disponen por tanto de varias direcciones IP, una por cada red a la que se conecta. Funciona a nivel de red.

Presenta en su interior la tabla de enrutamientos, que permite el encaminamiento en la capa de red. Cuenta además con el NAT, que permite traducir direcciones de IP privadas a públicas y viceversa.

- Un *hub* es un nodo en una red que permite implementar la difusión. Se trata de un conjunto de bocas ethernet que internamente funcionan como un bus. Cada vez que un paquete se envía por una de las bocas, este es reenviado a todas las demás bocas, por lo que todos los equipos conectados al *hub* reciben el paquete.

**Ejercicio 2.1.10.** ¿Qué diferencia, en el contexto de una red de computadores, existe entre la tecnología de difusión y la tecnología punto-a-punto?

La tecnología de difusión permite enviar un paquete desde un equipo y hacer que este sea recibido por el resto de equipos que estén conectados a la misma red (o hacer llegar estos a equipos en distintas redes). Es cada dispositivo el que decide si el paquete es para él o no.

Por otra parte, la tecnología punto-a-punto permite el envío de paquetes desde un equipo a otro usando un medio directo, por lo que el destino está implícito desde que se envía el paquete. Esta tecnología es más rápida y segura, pero su escalabilidad es mucho menor.

**Ejercicio 2.1.11.** Un sistema tiene una jerarquía de protocolos de  $n$  capas. Las aplicaciones generan mensajes de  $M$  bytes de longitud. En cada capa se añade una cabecera de  $h$  bytes. ¿Qué fracción del ancho de banda de la red se llena con cabeceras? Aplique el resultado a una conexión a 512 kbps con tamaño de datos de 1500 bytes y 4 capas, cada una de las cuales añade 64 bytes cabecera. ¿Qué velocidad real de envío de datos resulta?

---

<sup>3</sup>En español, podemos pensar en “entre pares”.

Debemos sumar a los  $M$  bytes iniciales que proporcionan las aplicaciones  $n$  veces (la capa de aplicación también incluye una cabecera)  $h$  bytes, por lo que la longitud de los mensajes que de verdad se envían es de  $n \cdot h + M$  bytes. Por tanto, por cada  $n \cdot h + M$  bytes enviados,  $n \cdot h$  de ellos son de cabeceras:

$$\frac{n \cdot h}{n \cdot h + M} \cdot 100 \text{ \% de ancho de banda que se llena de cabeceras}$$

Si ahora partimos de 1500 bytes iniciales y añadimos 4 veces (una por capa) 64 bytes, estamos en realidad enviando mensajes de longitud:

$$4 \cdot 64 + 1500 = 256 + 1500 = 1756 \text{ bytes}$$

Por tanto, el ( $256/1756 = 0,145786$ ) 14.58 % de la red se emplea para enviar cabeceras, luego se aprovecha el ( $1 - 0,145786 = 0,854214$  %) 85.42 % de la red para el envío de datos.

Si enviamos paquetes a una velocidad de 512kbps, en realidad estaremos enviando datos a una velocidad real de:

$$0,854214 \cdot 512 = 437,357568 \text{ kbps}$$

**Ejercicio 2.1.12.** Clasifique como de *difusión* o *punto a punto* cada uno de los siguientes sistemas de transmisión:

1. Radio y TV: Difusión, ya que es un emisor (en este caso, una cadena de televisión o radio) que difunde paquetes a cualquiera que tenga sintonizado dicho canal.
2. Redes inalámbricas (WLAN): Difusión, ya que cualquier equipo puede conectarse a la red y recibir los paquetes que se envían de forma inalámbrica.
3. ADSL: Punto a punto, ya que la conexión se establece mediante un cable. Usa en medio único.
4. Redes de cable: Puede implementar ambas tecnologías, ya que puede ser punto a punto (si cada equipo tiene su propio cable) o de difusión (si todos los equipos comparten el mismo cable).
5. Comunicaciones móviles (por ejemplo, GSM, UMTS, ...): Difusión, ya que (de nuevo) cualquier equipo puede recibir los paquetes que se envían por la red.

**Ejercicio 2.1.13.** Clasifique los siguientes servicios como orientados a conexión/no orientados a conexión y confirmados/sin confirmación. Justifique la respuesta.

Recordamos que los medios orientados a conexión son aquellos que comprueban si el receptor está disponible antes de enviar la información, y que los confirmados son aquellos que confirman la recepción del mensaje.

1. Correo postal ordinario: No es orientado a conexión (ya que no comprobamos anteriormente que el destinatario esté disponible, simplemente enviamos la carta) y es sin confirmación, ya que tras enviar la carta nada nos garantiza que el destinatario nos responda, pese a pedirlo explícitamente.
2. Correo certificado: No es orientado a conexión por la misma razón que el correo normal. Sin embargo, es confirmado, ya que al recibir el destinatario la carta debe firmar para que el emisor sea consciente de que la carta ha sido recibida.
3. Envío y recepción de fax.
4. Conversación telefónica. Es orientado a conexión, puesto que no se puede establecer una llamada si la otra persona no coge el teléfono. Además, es confirmado, ya que la otra persona ha de responder para que se produzca una comunicación.
5. Domiciliación bancaria de recibos.
6. Solicitud de certificado de empadronamiento. Es no orientado a conexión, ya que no se comprueba si el destinatario está disponible antes de enviar la solicitud. Además, es con confirmación, ya que el ayuntamiento envía un documento que certifica que el solicitante está empadronado.

**Ejercicio 2.1.14.** ¿Cuál es el tiempo necesario en enviar un paquete de 1000 Bytes, incluidos 50 Bytes de cabecera, por un enlace de 100 Mbps y 10Km? ¿cuál es el tiempo mínimo desde que se envía hasta que se recibe confirmación? ¿qué relación hay entre este tiempo y los temporizadores en, por ejemplo, las capas de enlace y transporte?

En primer lugar, hemos de calcular el retardo de transmisión  $T_t$ , que es el tiempo que se tarda en enviar el paquete por el enlace. Para ello, tenemos que:

$$T_t = 10^3 \text{ B} \cdot \frac{8 \text{ b}}{1 \text{ B}} \cdot \frac{1 \text{ s}}{100 \cdot 10^6 \text{ b}} = 80 \cdot 10^{-6} \text{ s} = 80 \mu\text{s}$$

Por otra parte, el retardo de propagación  $T_p$  es el tiempo que se tarda en enviar el paquete por los 10Km de cable. Para ello, suponiendo que la velocidad de transmisión es  $2/3c = 2 \cdot 10^8 \text{ m/s}$ , tenemos que:

$$T_p = 10 \cdot 10^3 \text{ m} \cdot \frac{1 \text{ s}}{2 \cdot 10^8 \text{ m/s}} = 50 \cdot 10^{-6} \text{ s} = 50 \mu\text{s}$$

Por tanto, el tiempo total que se tarda en enviar el paquete es de  $T_t + T_p = 130 \mu\text{s}$ .

Veamos ahora el tiempo mínimo desde que se envía hasta que se recibe confirmación. Además de los tiempos anteriores, hemos de tener en cuenta el tiempo de procesamiento del paquete, el retardo de transmisión del paquete de confirmación y el retardo de propagación del paquete de confirmación. Tenemos que:

- No se proporciona información del tiempo de procesamiento del paquete. No obstante, en los dispositivos modernos, este retardo es de varios órdenes de magnitud menor que los otros, por lo que se puede considerar despreciable.

- El retardo de propagación del paquete de confirmación es el mismo, puesto que la distancia recorrida es la misma.
- EL retardo de transmisión sí difiere, puesto que el tamaño del paquete de confirmación difiere. Normalmente, estos solo incluyen una cabecera, por lo que este tiempo, notado por  $T_{ACK}$ , es:

$$T_{ACK} = 50 \text{ B} \cdot \frac{8 \text{ b}}{1 \text{ B}} \cdot \frac{1 \text{ s}}{100 \cdot 10^6 \text{ b}} = 4 \cdot 10^{-6} \text{ s} = 4 \mu\text{s}$$

Un temporizador de control de flujo en capa de enlace o de transporte debe ser suficientemente mayor a este tiempo mínimo para evitar un re-envío inmediato de paquetes ante cualquier eventualidad mínima en la red, como un retardo en las colas (mayor retardo de procesamiento) por un cierto nivel de congestión.

Por tanto, el tiempo total mínimo que se tarda en enviar el paquete y recibir confirmación es de:

$$T_{\text{total}} = T_t + T_p + T_{ACK} + T_p = 80 \mu\text{s} + 50 \mu\text{s} + 4 \mu\text{s} + 50 \mu\text{s} = 184 \mu\text{s}$$