

Sistemas Informáticos

Unidad 11. Redes de computadoras



Autores: Sergi García, Alfredo Oltra

Actualizado Enero 2026



Licencia



Reconocimiento - No comercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Importante

Atención

Interesante

ÍNDICE

1. Introducción	3
2. Tipos de redes de computadoras	3
3. Arquitectura de redes	4
4. Capa de Internet	9
5. Componentes del hardware de red	15
5. Conectando computadoras a una red	20
7. Localización de recursos en la red	23
8. Seguridad	25
9. Acceso remoto	27
10. Recursos compartidos	28
11. Bibliografía	28

Unidad 11. Redes de computadoras

1. INTRODUCCIÓN

Sin duda, uno de los elementos que más ha impulsado la evolución de la informática en los últimos años ha sido la capacidad de conectar ordenadores entre sí para intercambiar información.

Toda comunicación entre dos partes se compone de una serie de elementos:

- **Mensaje:** la información que se desea transmitir.
- **Emisor/Receptor:** el dispositivo que envía o recibe el mensaje. Normalmente se denominan hosts.
- **Canal:** el medio a través del cual se transmite el mensaje.
- **Transductor:** dispositivo que convierte el mensaje en una señal transmisible. En los humanos serían las cuerdas vocales o los oídos; en un ordenador, la tarjeta de red.
- **Elementos accesorios:** elementos que mejoran la comunicación: un teléfono, una antena o, en redes informáticas, un hub, un router, un repetidor, etc.
- **Protocolos:** conjunto de reglas que controlan el flujo de datos y definen los parámetros físicos del sistema de comunicación. Por ejemplo, en una llamada telefónica:
 - Marcamos un número.
 - El teléfono suena.
 - Alguien descuelga.
 - Dice "hola".

2. TIPOS DE REDES DE COMPUTADORAS

Existen muchas formas de clasificar una red de computadoras, aunque las más comunes son:

2.1 Segundo área geográfica

- **LAN (Local Area Network):** Red de área local. Su característica principal es que la distancia entre equipos es reducida (desde una habitación hasta unos pocos kilómetros). Se utilizan ampliamente para conectar ordenadores personales y estaciones de trabajo en oficinas y empresas, con el fin de **compartir recursos** (impresoras, almacenamiento, etc.) y **intercambiar información**.
- **WAN (Wide Area Network):** Red de área amplia. Cubre distancias que pueden oscilar entre 100 y 1.000 kilómetros, proporcionando conectividad a varias ciudades o incluso a un país entero. Suelen ser implementadas por una empresa u organización para uso privado, o por un Proveedor de Servicios de Internet (ISP) para ofrecer conexión a sus clientes.

 **Importante:** Esta clasificación se puede ampliar con otros tipos, como:

- **MAN (Metropolitan Area Network)**
- **CAN (Campus Area Network):** dependiendo del tamaño y el alcance de la conectividad.

2.2 Segundo servicios

- **Cliente-servidor:** algunos equipos (clientes) solicitan servicios, mientras que otros (servidores) los proporcionan.
- **Peer to peer (P2P):** todos los equipos pueden actuar tanto como clientes como servidores.

2.3 Según tipo de comunicación

- **Simplex:** el canal permite la comunicación solo en una dirección. *Ejemplo: la emisión de radio.*
- **Half-duplex:** el canal permite comunicación en ambas direcciones, pero no simultáneamente. *Ejemplo: un walkie-talkie.*
- **Duplex (o full-duplex):** el canal permite comunicación en ambas direcciones al mismo tiempo. *Ejemplo: una llamada telefónica.*

3. ARQUITECTURA DE REDES

La arquitectura de red es un concepto que define todos los aspectos formales relacionados con la implementación de una red. Su estudio abarca la topología y los protocolos de comunicación.

3.1 Topología

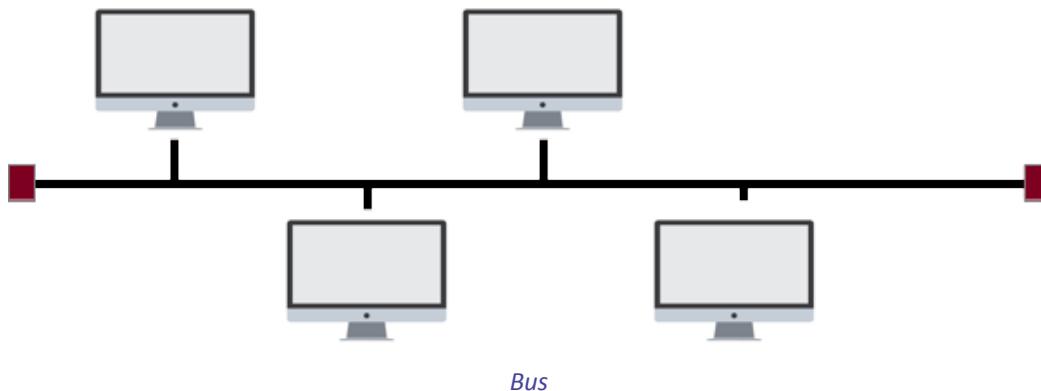
The topology refers to the physical form in which the network hosts are connected.

3.1.1 Topologías de redes cableadas (el medio es un cable)

Bus: Todos los ordenadores están conectados al mismo medio. Para evitar ecos (rebotes de la señal), se utilizan terminadores en los extremos.

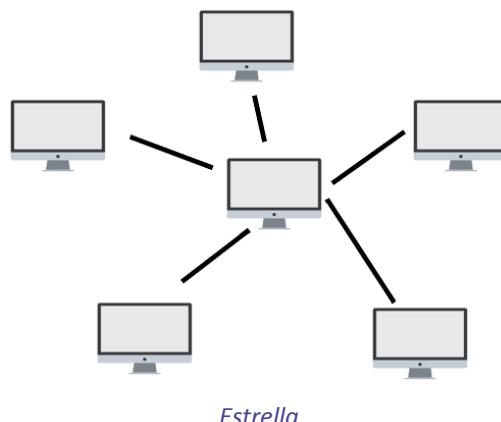
Inconvenientes:

- Si se rompe el cable, toda la red deja de funcionar.
- Su velocidad es baja, porque solo un equipo puede usar el medio a la vez.

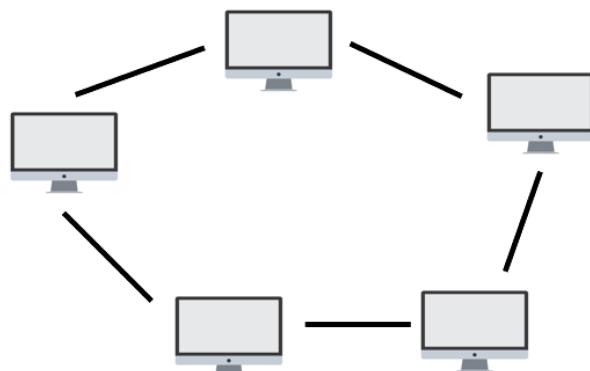


Estrella: Cada host se conecta a un **hub** central mediante enlaces punto a punto. Todo el tráfico pasa por el hub.

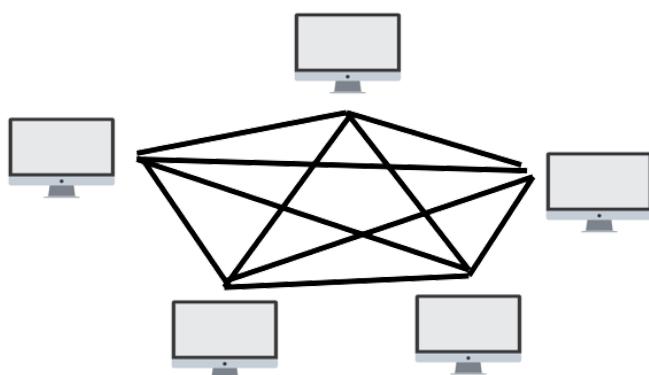
- **Ventajas:** diseño e implementación muy sencillos; es fácil añadir nuevos nodos.
- **Desventaja principal:** si falla el hub, **toda la red** queda inoperativa.



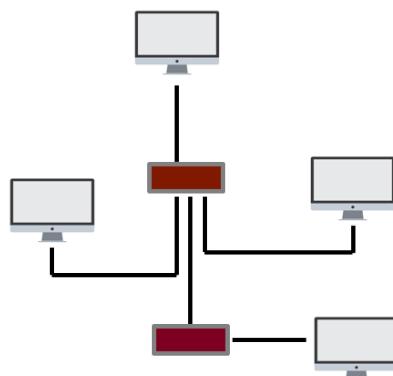
Anillo: Similar al bus, pero formando un circuito cerrado. Los datos se envían en una única dirección hasta llegar al host de destino.

*Anillo*

Malla: Cada ordenador está conectado punto a punto con todos los demás. Ventaja principal: si un nodo falla, los demás pueden seguir comunicándose sin problemas.

*Malla*

Árbol: Es una combinación entre la topología en estrella y en bus: varios segmentos de bus se conectan a un bus principal mediante hubs, routers o switches.

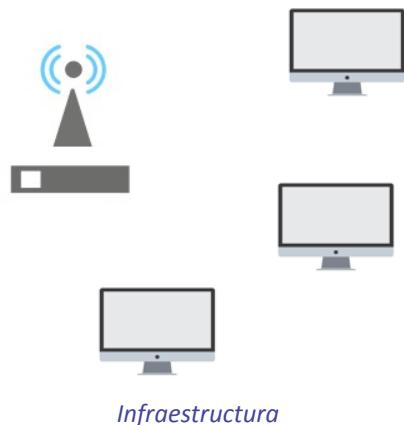
*Árbol*

3.1.2 Topologías de redes inalámbricas

Ad-hoc: No requieren un punto de acceso. Los dispositivos se comunican directamente entre sí, formando una red inalámbrica peer-to-peer.



Infraestructura: Utiliza un punto de acceso conectado a un segmento de cableado. Es la topología habitual en hogares y organizaciones.



3.3 Capas y protocolos

En los inicios de las redes, las soluciones para conectar ordenadores eran independientes: solo era posible interconectar equipos del mismo fabricante.

Con el tiempo, para lograr interoperabilidad, se decidió crear un estándar que permitiera que todas las máquinas pudieran comunicarse entre sí siempre que siguieran ese modelo.

La idea principal del estándar consiste en dividir el proceso de comunicación en fases pequeñas que, ejecutadas secuencialmente, permiten enviar y recibir mensajes.

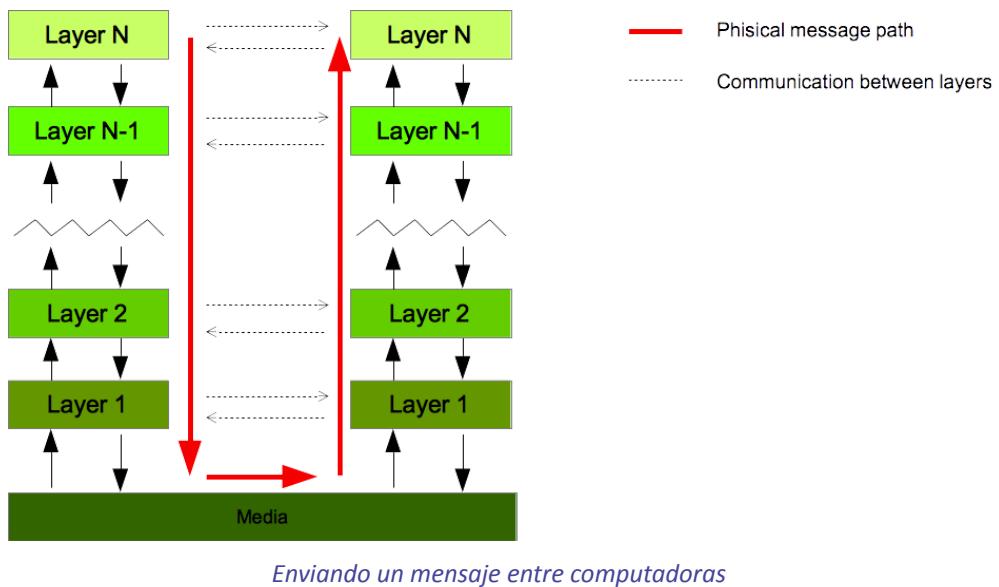
En el emisor, la ejecución en orden termina dejando el mensaje en el medio de transmisión.

En el receptor, estas fases se ejecutan en orden inverso para interpretar el mensaje.

Cada una de estas fases recibe el nombre de capa, y cada capa solo entiende la información procedente de su capa equivalente en el otro extremo.

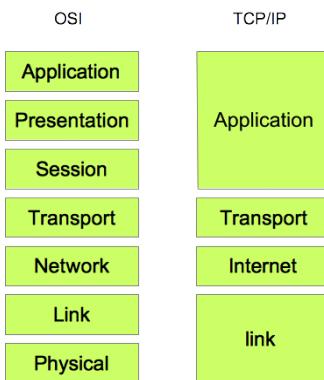
Para comunicarse, cada capa utiliza uno o varios protocolos, es decir, procedimientos que definen cómo enviar y recibir información.

 **Importante:** El conjunto de todas las capas con sus respectivos protocolos se denomina pila de protocolos (protocol stack).



El primer modelo adoptado fue el modelo OSI, con siete capas claramente diferenciadas.

Aunque se creó para estandarizar las comunicaciones, en la práctica ha sido reemplazado por un modelo más simple, utilizado hoy por la mayoría de dispositivos: el modelo TCP/IP, que agrupa varias de las capas del OSI y las reduce a cuatro.



OSI vs TCP/IP

3.3.1 Capa de enlace

Incluye los protocolos que usan las tarjetas de red para intercambiar datos. Los protocolos dependen de la tecnología utilizada y suelen estar almacenados en una memoria ROM de la tarjeta. Funciones destacadas:

- Convertir los bits en impulsos eléctricos o señales adecuadas para el medio.
- Gestionar el acceso al medio.
- Protocolos importantes:
 - Ethernet (en redes LAN).
 - DOCSIS (para redes de cable-módem).
 - xDSL (como ADSL).

3.3.2 Capa de Internet

Los protocolos de esta capa buscan las rutas necesarias para que un paquete llegue a su nodo de destino. Una de sus funciones clave es definir cómo se identifican los nodos de la red. Para ello utiliza protocolos como IPv4 e IPv6.

3.3.3. Capa de transporte:

La capa de transporte divide la información procedente de la capa superior (aplicación) en **pequeños bloques llamados paquetes** y se encarga de **garantizar, en algunos casos, que lleguen correctamente al destino**.

Los protocolos más importantes de esta capa son:

- **TCP (Transmission Control Protocol)**: TCP es un protocolo **orientado a conexión** que proporciona una transmisión **fiable**, incluso cuando el canal de comunicación es poco fiable. Características principales:
 - Las aplicaciones que envían y reciben información se identifican mediante **números de puerto**.
 - Los servidores utilizan puertos fijos (por ejemplo, **80** para HTTP).
 - Los clientes suelen utilizar puertos temporales o aleatorios.
 - Cada paquete recibe un **número de secuencia** único. Cuando un ordenador envía un paquete TCP, el receptor devuelve un mensaje de confirmación (ACK) con el mismo número de secuencia.
 - Si no se recibe la confirmación en un tiempo predefinido, el emisor **asume que el paquete se ha perdido** y lo retransmite.
 - El receptor utiliza los números de secuencia para **ordenar los paquetes** y reconstruir el mensaje original en el orden correcto.
- **UDP (User Datagram Protocol)**: UDP es un protocolo no orientado a conexión, especialmente útil para el streaming y transmisiones en tiempo real.
 - Igual que TCP, usa números de puerto para identificar cliente y servidor.
 - A diferencia de TCP, el equipo emisor no espera confirmación de recepción.
 - El receptor ignora los paquetes que lleguen con un número de secuencia incorrecto.

 **Importante:** UDP se utiliza cuando la rapidez es más importante que la fiabilidad absoluta.

Por ejemplo: transmisión de audio y vídeo por Internet, videollamadas, videojuegos online, etc.

Perder un fotograma o un paquete de audio no genera un problema grave. Sin embargo, para descargar un archivo, perder un solo paquete lo corrompería: por eso no se utiliza UDP para transmisión de datos críticos.

3.3.4 Capa de aplicación

Gestiona la comunicación entre las **aplicaciones del usuario** (por ejemplo, un navegador web y un servidor web) ejecutadas en máquinas distintas.

Protocolos más conocidos:

- **HTTP**: solicitud de páginas web.
- **FTP**: carga y descarga de archivos.
- **SMTP, POP3, IMAP**: envío de correos entre servidores, descarga de correo a un cliente y lectura del correo directamente en el servidor.
- **DNS**: traduce URLs (como *google.com*) a direcciones IP (como *45.24.28.55*).
- **DHCP**: configura automáticamente las direcciones IP de los equipos en una red.

Ejemplo simplificado de comunicación entre un cliente y un servidor (petición web)

- En la capa de aplicación del cliente, se indica el protocolo (HTTP) y la URL solicitada.
- Esta información se divide en paquetes según el tamaño permitido por el estándar.
- En la capa de Internet, se añade a cada paquete la dirección IP del servidor.

- En la capa de enlace, los paquetes se convierten en impulsos eléctricos (u ondas, si es inalámbrico) y se envían por el medio.
- En el servidor, la capa de enlace recibe esos impulsos y los convierte en paquetes.
- La capa de Internet del servidor verifica si la dirección IP destino coincide con la del propio servidor.
- Los paquetes válidos pasan a la capa de transporte, donde se ordenan y se reconstruye el mensaje.
- Finalmente, en la capa de aplicación, el servidor procesa la petición y devuelve la página solicitada.

4. CAPA DE INTERNET

Aunque todas las capas son importantes, en este curso nos centraremos en la capa de Internet, donde se realizan las funciones de direccionamiento y encaminamiento de paquetes (datagramas).

4.1 Dirección IP

Cada tarjeta o conexión de una red tiene una **dirección IP única**.

Más exactamente: **cada interfaz de red** (tarjeta Ethernet, Wi-Fi, etc.) necesita su propia dirección IP. Un ordenador con cable y Wi-Fi tendrá, por tanto, **dos direcciones IP**.

4.2 Formato de una dirección IP (IPv4)

Una dirección IP es un número de 32 bits, agrupado en 4 bloques de 8 bits (octetos), separados por puntos. Ejemplo en binario:

10101010.10001101.11110000.00001111

o, en modo decimal

170.141.240.15

Cada octeto puede representar valores entre **0 y 255**, por lo que el rango total es desde 0.0.0.0 hasta 255.255.255.255

4.3 Clases de direcciones

Una dirección IP consta de dos elementos:

1. **ID de red** (network ID): identifica la red.
2. **ID de host** (host ID): identifica el equipo dentro de la red.

A diferencia de las direcciones postales, estas partes **no están separadas físicamente**, sino mezcladas en la dirección. Para separarlas, utilizamos la **máscara de red** (network mask).

La máscara de red es un número que contiene 1s en la parte de la dirección que corresponde a la red y 0s en la parte que corresponde al host.

11111111.11100000.00000000.00000000

Si realizamos una operación **AND** entre la dirección IP y la máscara de red, obtenemos el **identificador de red**:

10101010.10001101.11110000.00001111 -> IP (170.141.240.15)

AND 11111111.11100000.00000000.00000000 -> Máscara de red (255.224.0.0)

10101010.10000000.00000000.00000000 = 170.128.0.0 -> ID de red

Como todas las máscaras válidas son una secuencia de **1s seguidos de 0s**, es habitual expresarlas indicando solo cuántos bits son 1. Esta notación se llama **CIDR**.

Ejemplo: 170.141.240.15/11.

Para obtener la parte del host basta con aplicar **NOT** a la máscara y realizar otro **AND**:

10101010.10001101.11110000.00001111 -> IP (170.141.240.15)

AND 00000000.00011111.11111111.11111111 -> NOT máscara de red (0.31.255.255)

00000000.00001101.11110000.00001111 = 0.13.240.15 -> host ID

4.4 Clases de direcciones IP

Las direcciones IPv4 se dividen tradicionalmente en **5 clases**:

- **Clase A:**

Primer bit = 0 → rango 0.0.0.0 a 127.255.255.255

Máscara por defecto: 255.0.0.0 o /8

Ejemplo: 25.124.200.200

- **Clase B:**

Primeros dos bits = 10 → rango 128.0.0.0 a 191.255.255.255

Máscara por defecto: 255.255.0.0 o /16

Ejemplo: 165.124.200.200

- **Clase C:**

Primeros tres bits = 110 → rango 192.0.0.0 a 223.255.255.255

Máscara por defecto: 255.255.255.0 o /24

Ejemplo: 192.168.20.20

- **Clase D:**

Primeros cuatro bits = 1110 → 224.0.0.0 a 239.255.255.255

→ Para **multicast**

- **Clase E:**

Primeros cinco bits = 11110 → 240.0.0.0 a 247.255.255.255

→ Para **investigación**

! **Atención:** las clases D y E son especiales y no se utilizan para redes normales, son reservadas para multicasting e investigación.

Máscaras no coincidentes con clases

Puede llamar la atención que antes hemos puesto una máscara /11, que **no pertenece** a ninguna máscara por defecto de Clase A, B o C. Esto se debe a:

- La clasificación por clases es **histórica**. Técnicamente se puede usar cualquier máscara (siempre que sea una secuencia válida de 1s y 0s).
- En redes reales se crean **subredes** usando máscaras personalizadas para mejorar la eficiencia y la seguridad.

Veremos esto en **subnetting**.

4.5 Direcciones públicas y privadas

Si todas las direcciones IPv4 fueran públicas, el número total (256^4) sería insuficiente para todos los dispositivos de Internet.

Por ello se establecen dos tipos:

- **Direcciones públicas:** únicas en Internet, no se pueden repetir.
- **Direcciones privadas:** solo se usan dentro de redes internas; pueden repetirse en organizaciones distintas.

Rangos privados:

- **Clase A:** 10.0.0.0 – 10.255.255.255
- **Clase B:** 172.16.0.0 – 172.31.255.255
- **Clase C:** 192.168.0.0 – 192.168.255.255

Si usas direcciones internas en tu organización, debes elegir una de estas.

Si usas otra, tu red podría entrar en conflicto con direcciones públicas reales.

La coexistencia de direcciones privadas idénticas es posible gracias a técnicas como **NAT**, que veremos más adelante.

4.6 Direcciones especiales

Dentro de un segmento de red existe un conjunto de direcciones especiales que cumplen funciones específicas.

- **Loopback:** La dirección **127.0.0.1** se utiliza para realizar comprobaciones internas de la propia interfaz de red.

! **Atención:** Esta dirección está asociada al nombre **localhost**.

! **Atención:** Podrías pensar que probar la dirección IP asignada a la tarjeta de red es equivalente a usar la dirección de loopback, pero esto es **incorrecto**.

Si se utiliza la IP real, el paquete **sale** del equipo y **vuelve**.

En cambio, con la dirección loopback, las pruebas se realizan **internamente**, sin salir al exterior.

- **Broadcast:** Enviar un paquete a esta dirección hace que dicho paquete llegue a **todos los hosts de la red**. Se calcula poniendo a 1 todos los bits de la parte correspondiente al host.

Por ejemplo, la dirección de broadcast para la red **170.141.240.15/11** es:

170.141.240.15/11

→ 10101010.10001101.11110000.00001111

→ 10101010.10011111.11111111.11111111

→ 170.159.255.255 (dirección de broadcast)

- **Gateway (puerta de enlace):** No es una dirección fija como las anteriores, pero es fundamental. Indica la dirección del dispositivo que permite enviar paquetes **fuerza del segmento de red local**.

Normalmente se utiliza la **segunda dirección** de la red como puerta de enlace (la primera suele identificarse como la dirección de red). Ejemplo: **192.168.20.1**

4.7 Subnetting

Como se mencionó antes, el número de direcciones IP es finito y no muy elevado, por lo que es necesario ser eficiente en su asignación. Esto implica optimizar su uso y evitar desperdiciar direcciones.

Para ello se utiliza la técnica del **subnetting** (subnetado o subdivisión de redes).

La mejor forma de entender el proceso es con un ejemplo.

Supongamos que el proveedor de direcciones IP nos asigna la red **192.168.40.0/24** para nuestra organización.

Esta es una red privada de tipo C, con máscara **255.255.255.0**, lo que permite **256 direcciones** (de 0 a 255).

La organización tiene cinco departamentos, y cada uno dispone de **30 dispositivos** que requieren conexión.

Podríamos utilizar el rango completo de direcciones para todos los departamentos, pero es más eficiente **dividir la red en cinco subredes**, una por departamento.

Con ello:

- Optimizamos el uso de direcciones.
- Aislamos mejor los departamentos.
- Mejoramos la seguridad.
- Reducimos problemas de tráfico.

Para lograrlo, se utiliza **subnetting**.

El proceso consiste en añadir, además del ID de red y el ID de host, un nuevo elemento: el **ID de subred**, que se obtiene “**robando**” **bits** de la parte del host.

Proceso de subnetting

1. Calcular los bits necesarios para representar las 5 subredes

Necesitamos crear 5 subredes. Con **3 bits** obtenemos hasta **8 combinaciones** ($2^3 = 8$), suficientes.

2. Comprobar que quedan suficientes bits para los hosts

La máscara original es /24, lo que deja **8 bits** para hosts. Si usamos 3 bits para subredes, quedan **5 bits para hosts**.

Número de hosts por subred: $2^5 = 32$ **hosts**, que es mayor que los 30 necesarios → válido.

3. Calcular las direcciones de cada subred

Hacemos todas las combinaciones posibles de 0 y 1 con los 3 bits robados para la subred, dejando a 0 los bits de host:

Red (192.168.40) Subred Host Dirección de subred

11000000.10101000.00101000.00000000 → 192.168.40.0
 11000000.10101000.00101000.00100000 → 192.168.40.32
 11000000.10101000.00101000.01000000 → 192.168.40.64
 11000000.10101000.00101000.01100000 → 192.168.40.96
 11000000.10101000.00101000.10000000 → 192.168.40.128
 11000000.10101000.00101000.10100000 → 192.168.40.160
 11000000.10101000.00101000.11000000 → 192.168.40.192
 11000000.10101000.00101000.11100000 → 192.168.40.224

! **Atención:** Solo necesitamos las **primeras 5 subredes**.

4. Calcular la nueva máscara de subred

La máscara inicial era /24. Al añadir 3 bits para subredes:

$$24 + 3 = /27$$

Todas las subredes tendrán esta máscara.

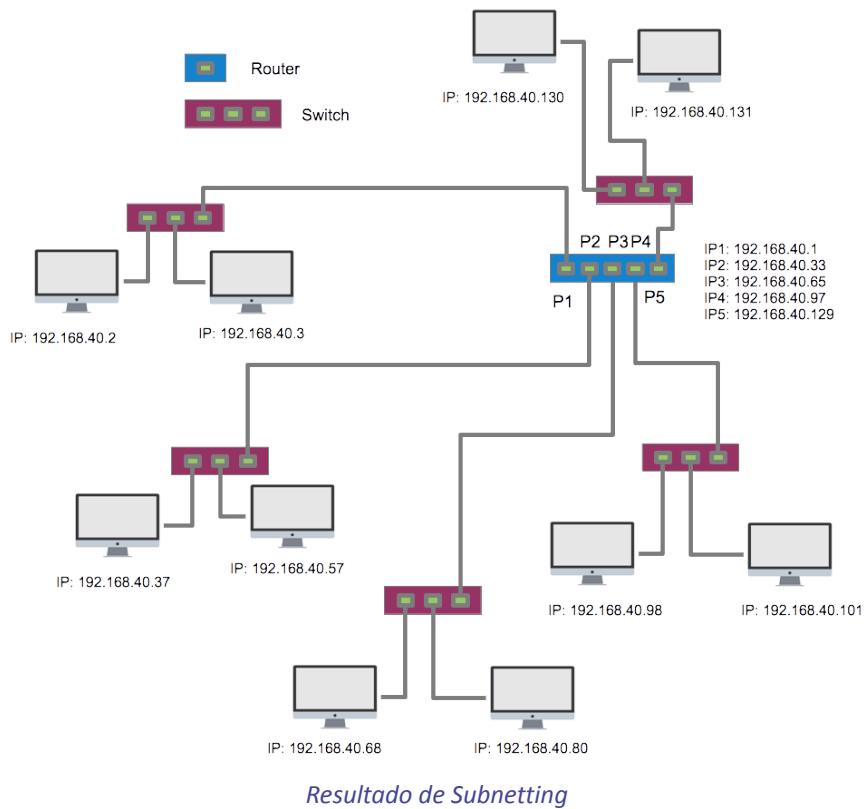
192.168.40.0/27
 192.168.40.32/27
 192.168.40.64/27
 192.168.40.96/27
 192.168.40.128/27

5. Calcular el rango de direcciones, broadcast y gateway

Con 5 bits para hosts → bloques de 32 direcciones.

Subred	Rango	Broadcast	Gateway
192.168.40.0/27	192.168.40.0 – 192.168.40.31	192.168.40.31	192.168.40.1
192.168.40.32/27	192.168.40.32 – 192.168.40.63	192.168.40.63	192.168.40.33
192.168.40.64/27	192.168.40.64 – 192.168.40.95	192.168.40.95	192.168.40.65
192.168.40.96/27	192.168.40.96 – 192.168.40.127	192.168.40.127	192.168.40.97
192.168.40.128/27	192.168.40.128 – 192.168.40.159	192.168.40.159	192.168.40.129

En la figura se puede observar la topología resultante (en el ejemplo solo se dibujan dos hosts por subred para simplificar).



4.7 Enrutado

El **routing** (encaminamiento) es la técnica que permite enviar un paquete desde un ordenador a otro, incluso si pertenecen a redes distintas. El proceso lo realiza un **router**, utilizando las llamadas **tablas de encaminamiento**.

De forma simplificada, la tabla contiene:

- **Destino (target):** puede ser una IP, una red o la ruta por defecto (0.0.0.0)
- **Interfaz:** indica por qué puerto del router debe enviarse el paquete

Ejemplo de tabla del router de la figura anterior:

Target	Interface
192.168.40.0	P1
192.168.40.32	P2
192.168.40.64	P3
192.168.40.96	P4
192.168.40.128	P5

4.8 NAT

Dada la escasez de direcciones IPv4, la técnica **NAT (Network Address Translation)** permite que la información se transmita **entre una IP pública y varias IP privadas** de forma transparente para el usuario.

Gracias a NAT:

- Una única IP pública puede representar a **toda una red privada**.

- La red interna puede ser tan grande como se necesite.

El router añade información interna para identificar a qué equipo local pertenece cada conexión. Cuando llega la respuesta desde Internet, el router la entrega al host correspondiente.

4.9 IPv6

Aun con subnetting y NAT, el número de dispositivos sigue creciendo y las direcciones IPv4 se agotan. Para resolver este problema se diseñó **IPv6**, que utiliza **128 bits** para las direcciones.

Esto permite **2^{128} direcciones**, un número inmensamente mayor.

Formato:

- Se usan números **hexadecimales**.
- Las direcciones se agrupan en **bloques de 16 bits** (0000 a ffff) separados por ":".
- Los ceros iniciales de cada bloque pueden omitirse.
- Un conjunto continuo de ceros puede abreviarse con ":", pero solo **una vez** por dirección.

Ejemplo:

IPv6 completa:

de34:0000:0000:0000:045e:0000:0000:0ffa

Abreviada:

de34::45e:0:0:ffa

Al igual que en IPv4, la dirección incluye ID de red y de host, pero:

- Los **primeros 64 bits** son el **ID de red**
- Los **últimos 64 bits** son el **ID de host**.

Ejemplo:

Network ID: de34::

Host ID: 45e:0:0:ffa

5. COMPONENTES DEL HARDWARE DE RED

5.1 Tarjeta de interfaz de red (NIC)

La tarjeta de interfaz de red (NIC) es el componente hardware que conecta el ordenador con el medio de transmisión. Todos los dispositivos conectados a una red deben disponer, al menos, de una NIC.

Principalmente, existen dos tipos: cableada e inalámbrica.

Atención: La NIC trabaja en la capa de enlace.

Cada NIC se identifica físicamente mediante un ID llamado dirección MAC (Media Access Control), un número único asignado por el fabricante y que es independiente del protocolo de red utilizado.

Ejemplo: 00:35:AA:28:5F:69.

 **Importante:** Hoy en día la conectividad es tan esencial que prácticamente todos los dispositivos (ordenadores, móviles, impresoras, televisores...) tienen una NIC integrada.



Tarjeta de red (NIC)

5.2 Medio de transmisión

 **Atención:** los medios trabajan en la capa de enlace.

5.2.1 Medios guiados

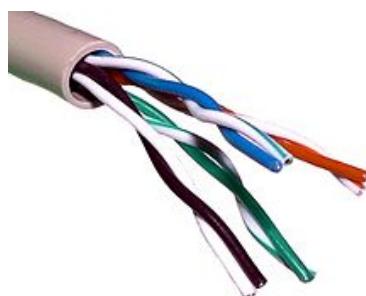
Son aquellos en los que el medio conduce físicamente la señal. Hay tres tipos principales: par trenzado, coaxial y fibra óptica.

Par trenzado

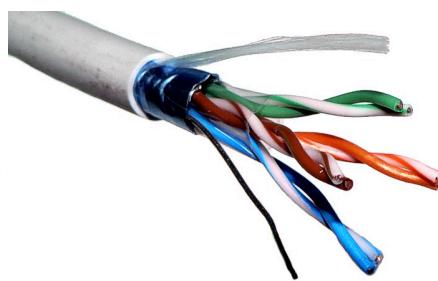
Está formado por dos hilos de cobre trenzados entre sí para reducir interferencias.

Existen tres variantes:

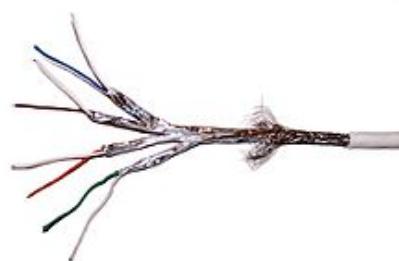
- UTP (Unshielded Twisted Pair): sin protección adicional contra interferencias.
- FTP (Foiled Twisted Pair): dispone de una única malla de protección que envuelve todos los pares.
- STP (Shielded Twisted Pair): cada par tiene su propia cubierta conductora conectada a tierra, ofreciendo mayor protección.



UTP



FTP



STP

 **Importante:** Actualmente existen variantes adicionales como S/UTP, S/FTP, S/STP o F/UTP, donde F indica apantallado por lámina (foiled) y S apantallado por malla (shielded).

Además, los cables se clasifican por **categorías (Cat1 a Cat7)**, según su calidad: mayor categoría implica mayor velocidad y distancia posibles, pero también mayor coste.

Los conectores asociados son:

- **RJ-11:** telefonía y xDSL
- **RJ-45:** redes Ethernet



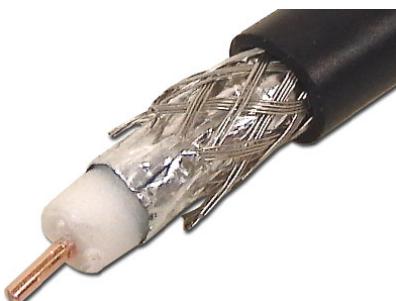
RJ-11 y RJ-45

Coaxial

El cable coaxial fue ampliamente utilizado en redes MAN, aunque cada vez se encuentra más en desuso debido a su sustitución por la fibra óptica.

Está formado por:

- Un conductor central, encargado de transportar la señal eléctrica.
- Un conductor externo o malla, que actúa como blindaje para proteger al conductor interno frente a interferencias.
- Un **material dieléctrico** que aísla ambos conductores.



Coaxial



Conector BNC

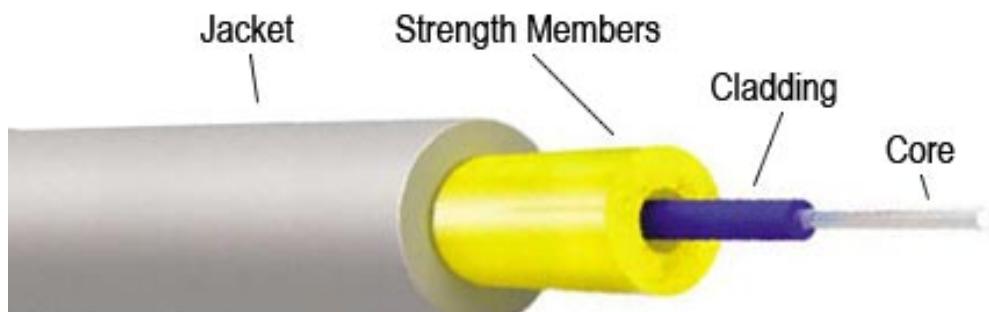
Conecotor habitual: BNC.

Fibra óptica

En la fibra óptica, la señal eléctrica se sustituye por pulsos de luz emitidos por un láser o LED, lo que las hace inmunes a las interferencias electromagnéticas.

Está compuesta por:

- Revestimiento externo (jacket): protege el cable y lo aísla de la luz exterior.
- Elementos de refuerzo (strength members): aumentan la resistencia mecánica y evitan la rotura de la fibra durante la instalación.
- Revestimiento (cladding): capa que refleja la luz hacia el núcleo.
- Núcleo (core): por donde viaja la señal luminosa.

*Fibra óptica*

Existen numerosos tipos de conectores: FDDI, LC, FC, SC, entre otros.

*Conector FC**Conector FDDI*

4.2.2 Medios no guiados

En los medios no guiados la información se transmite mediante ondas electromagnéticas que se propagan por el aire o el espacio.

- Su principal ventaja es la facilidad de instalación y su gran escalabilidad.
- Su inconveniente más destacado es que son muy sensibles a las condiciones atmosféricas y su velocidad suele ser inferior a la de los medios guiados.

Microondas

Son señales electromagnéticas de alta frecuencia.

Ejemplos típicos:

- Wi-Fi y Bluetooth, que operan alrededor de 2,4 GHz (y 5 GHz en algunas versiones).
- Emisiones de televisión, con frecuencias entre 50 MHz y 900 MHz.
- Telefonía móvil, que trabaja en bandas como 900 MHz, 1800 MHz o 1900 MHz.

Señales infrarrojas

Pensados para comunicaciones a muy corta distancia, y muy afectados por la luz ambiental. Un ejemplo clásico son los mandos a distancia de televisores y otros electrodomésticos.

5.3 Modem

Módem es el acrónimo de **MOdulador/DEModulador**.

Cuando una señal debe viajar largas distancias a través de un medio, es necesario modularla para que pueda transmitirse correctamente.

Cuando llega al receptor, debe realizarse el proceso inverso: demodularla.

En general, es el dispositivo que permite conectar nuestra red LAN con la WAN o MAN del proveedor de servicios.

Dependiendo del tipo de conexión, encontramos por ejemplo: módem de cable, módem ADSL, módem telefónico, etc.

! **Atención:** Trabaja en la capa de enlace del modelo OSI.

5.4 HUB

Un hub es un dispositivo con varios puertos cuya función es reenviar el paquete que recibe por uno de sus puertos hacia todos los demás, amplificando previamente la señal.

Aunque hoy en día han sido sustituidos por los switches, tuvieron un uso muy extendido porque permitían crear segmentos de red Ethernet.

Su mayor inconveniente es que, aunque el emisor sea un solo equipo, la información se envía a todos, lo que provoca: aumento del tráfico, disminución de la velocidad y que todas las tarjetas de red del segmento deban trabajar a la velocidad del dispositivo más lento.

5.5 Switch

Un switch es un hub inteligente.

El switch almacena una tabla con la dirección MAC del dispositivo conectado a cada puerto.

Cuando un equipo envía un paquete a otro, el switch solo reenvía dicho paquete al puerto de destino. Algunas de las ventajas frente al hub:

- Evita replicaciones innecesarias.
- Reduce el tráfico en la red.
- Mejora la velocidad.
- Permite que cada dispositivo funcione a diferentes velocidades.

Externamente, un hub y un switch pueden parecer idénticos; muchas veces solo se distinguen por la etiqueta.



HUB



Switch

! **Atención:** Aunque posee cierta “inteligencia”, sigue trabajando en la capa de enlace, ya que no modifica ni redirige paquetes en función de direcciones IP; simplemente abre o cierra el puerto según la dirección MAC.

4.6 Router

Un router es un dispositivo mucho más inteligente que un switch.

Trabaja en la capa de Internet (o capa de red en el modelo OSI), por lo que interpreta direcciones IP.

Gracias a esto, puede determinar la ruta que debe seguir un paquete para llegar a un equipo que no pertenece al segmento de red local.

Es decir, el router decide si un paquete debe enviarse a otro dispositivo dentro de la LAN o hacia redes externas, y lo enruta correctamente, añadiendo la información necesaria para permitir la respuesta del destinatario.

Para ello, un router cuenta al menos con dos interfaces de red (NIC), conectadas a redes distintas: una local y otra externa.



Router

Importante: La dirección MAC es un dato grabado de fábrica y no cambia.

La dirección IP, sin embargo, es asignada a la NIC según la red a la que se conecte, por lo que sí puede cambiar.

Importante: En entornos domésticos es muy común que el módem, el router, e incluso un switch, vengan integrados en un único dispositivo.

5. CONECTANDO COMPUTADORAS A UNA RED

El primer paso para poder utilizar dispositivos en una red es **conectarlos a ella**. Para ello, debemos conocer las **interfaces disponibles (NICs)** y **asignarles una dirección IP**.

Existen dos formas de asignar direcciones IP: **estática** y **dinámica**.

Asignación estática

En la asignación estática, la dirección IP de cada dispositivo debe configurarse manualmente.

Ventajas:

- La IP del dispositivo permanece fija en el tiempo.

Inconvenientes:

- Complica la incorporación de nuevos equipos.
- Aumenta la probabilidad de conflictos de direcciones si no se gestiona correctamente.

Asignación dinámica

La asignación dinámica se realiza mediante un servidor DHCP, que se encarga de gestionar la distribución de direcciones IP.

Cuando un dispositivo se conecta a la red, solicita una IP al servidor DHCP, que la asigna siguiendo determinadas reglas.

Ventajas:

- Facilita la incorporación de nuevos dispositivos.
- Evita conflictos por asignación duplicada de direcciones IP.
- Permite una configuración rápida y automática.

Inconveniente:

- La IP asignada puede cambiar entre distintas conexiones.

Asignación dinámica en la práctica

Debido a su simplicidad, es el tipo de asignación más habitual en redes domésticas y corporativas.

La usamos, por ejemplo, al conectarnos con el móvil a una red Wi-Fi o al conectar nuestro ordenador al router de casa.

En la mayoría de los casos, el router actúa como servidor DHCP.

La asignación dinámica viene configurada por defecto en la mayoría de sistemas operativos utilizados como estaciones de trabajo, por lo que el usuario solo necesita conectar el equipo físicamente a la red.

6.1 Comandos relacionados con redes en sistemas Linux

En los sistemas Linux actuales, el conjunto de comandos recomendado para gestionar y consultar la red pertenece a la **suite ip**, que sustituye al antiguo ifconfig.

Para saber **qué interfaces existen** en nuestro equipo, utilizamos:

```
ip link show
```

Este comando muestra todas las interfaces, tanto **físicas** como **lógicas** (incluyendo la interfaz de loopback lo).

También se puede usar la versión abreviada:

```
ip link
```

Para ver las direcciones IP asignadas a cada interfaz, usamos:

```
ip addr show
```

o su forma abreviada:

```
ip a
```

Aquí veremos:

- Interfaces activas e inactivas
- Direcciones IPv4 e IPv6
- Estado de la interfaz (UP/DOWN)

Para **activar** una interfaz:

```
sudo ip link set <interfaz> up
```

Ejemplo:

```
sudo ip link set enp3s0 up
```

Para **desactivarla**:

```
sudo ip link set <interfaz> down
```

Asignar una IP estática (temporal)

Si queremos asignar una dirección IP manualmente de forma temporal:

```
sudo ip addr add 192.168.1.50/24 dev enp3s0
```

Para eliminarla:

```
sudo ip addr del 192.168.1.50/24 dev enp3s0
```

Consultar la puerta de enlace

```
ip route show
```

o simplemente:

```
ip r
```

Aquí aparecerá, entre otras rutas, la **default**, que indica la puerta de enlace usada por el sistema.

Interfaces en Linux: nomenclatura moderna

Los nombres de las interfaces en Linux ya no siguen el antiguo formato eth0, eth1 salvo en sistemas muy concretos. Actualmente suelen seguir el estándar *predictable network interface names*, por ejemplo:

- **enp3s0** → Ethernet, PCI bus 3, slot 0
- **wlp2s0** → Wi-Fi, PCI bus 2, slot 0
- **lo** → Loopback (interfaz lógica)

- **vboxnet0** → Interfaz virtual creada por VirtualBox
- **docker0, br0**, etc. → Interfaces virtuales creadas por contenedores o bridges

6.2 Sistemas Windows

Para ver cuántas interfaces de red tiene tu equipo:

```
ipconfig  
ipconfig /all
```

También puedes usar en Powershell:

```
Get-NetAdapter
```

7. LOCALIZACIÓN DE RECURSOS EN LA RED

Cada dispositivo conectado a una red IP tiene una dirección única.

Recordar esas direcciones numéricas para comunicarnos con otros equipos es complicado, por lo que se usa un sistema llamado resolución de nombres, que traduce nombres fáciles de leer a direcciones IP.

Por ejemplo: Es mucho más sencillo recordar www.google.com que 216.58.211.238.

7.1 Asignar nombre al equipo

Para poder acceder a un dispositivo por su nombre, primero debemos asignarle uno.

En Linux, el nombre del equipo se guarda en:

```
/etc/hostname
```

Lo podemos editar con:

```
sudo nano /etc/hostname
```

Y después realizar:

```
sudo systemctl restart systemd-hostnamed
```

o simplemente:

```
sudo reboot
```

En Windows (10/11) podemos ir a:

Configuración → Sistema → Información → Cambiar nombre del equipo

7.2 Relacionar nombres e IP localmente (archivo hosts)

Para relacionar manualmente IP ↔ nombre dentro de una red pequeña podemos usar el archivo hosts.

Formato: IP nombre_equipo

Ejemplo:

```
192.168.20.6 mortadelo-computer
192.168.20.7 filemon-computer
192.168.20.8 zipi-computer
192.168.20.9 zape-computer
192.168.20.10 carpanta-computer
```

En Linux esto lo podemos hacer en el fichero /etc/hosts y en Windows en C:\Windows\System32\drivers\etc\hosts

7.3 DNS - Resolución de nombres en Internet

El archivo hosts funciona bien en redes pequeñas, pero no sirve para Internet.

Para ello existen los servidores DNS, que traducen nombres de dominio públicos (como www.google.com) a direcciones IP.

Funcionan de forma jerárquica: si un servidor no sabe resolver un nombre, reenvía la solicitud a otro servidor superior.

Generalmente, los DNS se obtienen automáticamente por DHCP, pero podemos configurarlos manualmente.

Ejemplos de DNS públicos:

```
Google: 8.8.8.8 y 8.8.4.4
Cloudflare: 1.1.1.1
OpenDNS: 208.67.222.222
```

7.4 Configurar DNS en Linux

Desde hace años, muchas distribuciones ya no usan resolv.conf directamente, sino:

- systemd-resolved
- Netplan
- NetworkManager

Modificar /etc/resolv.conf suele ser temporal, porque lo sobrescriben automáticamente.

Alguna formas de configurarlo:

A) Ubuntu/Debian modernos con Netplan

Edita los ficheros en el directorio /etc/netplan/*.yaml

Ejemplo IP estática y DNS:

```
network:
  version: 2
  ethernets:
    enp3s0:
      dhcp4: false
      addresses: [192.168.20.5/24]
```

```
gateway4: 192.168.20.1
nameservers:
    addresses: [172.16.1.254, 8.8.8.8]
```

Aplicar:

```
sudo netplan apply
```

B) Con NetworkManager (Ubuntu Desktop, Fedora, Mint...)

Configurar DNS:

```
sudo nmcli connection modify enp3s0 ipv4.dns "172.16.1.254 8.8.8.8"
sudo nmcli connection modify enp3s0 ipv4.ignore-auto-dns yes
sudo nmcli connection up enp3s0
```

7.5 Configurar DNS en Windows

Ruta gráfica (Windows 10/11):

Configuración → Red e Internet → Cambiar opciones del adaptador → Propiedades → IPv4

Una vez ahí, elegir: “Usar las siguientes direcciones de servidor DNS”

Ejemplo:

Preferido: 8.8.8.8

Alternativo: 8.8.4.4

8. SEGURIDAD

8.1 Firewall (Cortafuegos) en Sistemas Linux

Un cortafuegos es un sistema (puede implementarse mediante software o hardware) que supervisa y controla el tráfico de red entrante y saliente. Se configura una serie de reglas de confianza y no confianza que se aplican a cada paquete de forma que se permita o no su paso dependiendo de si cumple alguna de esas reglas.

Linux incluye un cortafuegos nativo integrado en el kernel. Actualmente, el sistema recomendado para gestionarlo es **nftables**, que sustituye a iptables como solución moderna y unificada para el filtrado de paquetes.

nftables se gestiona mediante el comando **nft** (puedes instalarlo con sudo apt install nftables -y). Este sistema utiliza **tablas**, **cadenas** y **reglas**, pero de una forma más eficiente y coherente que iptables.

- **Las tablas** definen el tipo de tráfico que se va a tratar (por ejemplo, filtrado).
- **Las cadenas** indican en qué punto del flujo de paquetes se aplican las reglas.
- **Las reglas** especifican qué hacer con los paquetes (aceptarlos, rechazarlos, etc.).

Una tabla típica de filtrado se crea dentro de la familia **inet**, que permite trabajar tanto con IPv4 como con IPv6.

Ejemplo de creación de una tabla y cadenas básicas:

```
sudo nft add table inet filter
sudo nft add chain inet filter input { type filter hook input priority 0 \; policy accept \; }
sudo nft add chain inet filter forward { type filter hook forward priority 0 \; policy accept \; }
sudo nft add chain inet filter output { type filter hook output priority 0 \; policy accept \; }
```

 **Importante:** En nuestro caso solo trabajaremos con la tabla de filtrado (filter). En nftables no es necesario indicar la tabla por defecto con una opción como -t, ya que se especifica directamente en cada comando.

Para listar la configuración actual de nftables:

```
sudo nft list ruleset
```

 **Importante:** Por defecto, si no se define ninguna regla restrictiva, la política de las cadenas suele ser **accept**, por lo que todos los paquetes son permitidos.

Una forma más segura de trabajar es cambiar la política por defecto de la cadena **input** a **drop**, al menos para los paquetes entrantes. Para hacerlo:

```
sudo nft add chain inet filter input { type filter hook input priority 0 \; policy drop \; }
```

A partir de este momento, el ordenador rechazará todos los paquetes entrantes que no coincidan con una regla explícita de aceptación.

Esta solución puede ser drástica, ya que la comunicación con el exterior puede quedar bloqueada si no se permiten paquetes de entrada. Por ello, es necesario añadir reglas que permitan el tráfico necesario según nuestras necesidades.

Algunos ejemplos de apertura de tráfico son:

Abrir una interfaz (loopback)

```
sudo nft add rule inet filter input iif lo accept
```

Esta regla permite el tráfico que entra por la interfaz de loopback (**lo**), imprescindible para el funcionamiento interno del sistema.

Por protocolo y puertos

```
sudo nft add rule inet filter input ct state established,related accept
```

Esta regla permite tráfico UDP que entra por la interfaz **eth1**, con puerto de origen 68 y puerto de destino 67 (típico del protocolo DHCP).

Por estado de la conexión

```
sudo nft add rule inet filter input ct state established,related accept
```

nftables puede controlar el estado de las conexiones mediante **connection tracking**.

En este caso, se permite el tráfico entrante que pertenece a conexiones ya establecidas (**ESTABLISHED**) o relacionadas con ellas (**RELATED**), es decir, respuestas a conexiones iniciadas previamente por nuestro equipo.

8.2 Firewall (Cortafuegos) en Sistemas Windows

El cortafuegos por defecto de los sistemas Windows es más limitado en comparación con nftables en Linux, pero para configuraciones sencillas puede resultar suficiente.

Puedes configurarlo usando la interfaz gráfica de Windows siguiendo este tutorial:

<https://www.youtube.com/watch?v=M4QhgpXZB6E>

9. ACCESO REMOTO

A nivel administrativo, una de las principales ventajas de las redes es la posibilidad de gestionar un ordenador de forma remota. Existen muchas opciones, pero nos centraremos en dos de las más utilizadas.

Importante: Una dirección IP puede ofrecer múltiples servicios. Para distinguir a qué servicio nos estamos conectando, cada uno utiliza un puerto diferente. Los puertos más comunes son:

- 80 para servidores web
- 22 para SSH
- 25 para SMTP

Más información en: [https://en.wikipedia.org/wiki/Port_\(computer_networking\)](https://en.wikipedia.org/wiki/Port_(computer_networking))

9.1 SSH

La primera opción histórica fue **telnet**, un programa que permitía conectarse por terminal a otro sistema indicando su IP y credenciales.

Sin embargo, telnet dejó de utilizarse debido a graves problemas de seguridad, especialmente porque la conexión no está cifrada y los datos (como contraseñas) pueden ser interceptados.

Para solucionar esto apareció la familia de herramientas del protocolo **Secure Shell (SSH)**, que permite el control remoto y la transferencia segura de archivos entre ordenadores.

El sistema consta de:

- Un **servidor SSH (sshd)** en el equipo al que queremos acceder.
- Un **cliente** (ssh, scp, etc.) en el equipo desde el que nos conectamos.

En el servidor necesitamos:

- Instalar el servidor SSH:

```
sudo apt-get install openssh-server
```

- Abrir el puerto correspondiente en el cortafuegos. El protocolo es TCP y el puerto por defecto es el 22:

```
sudo nft add rule inet filter input tcp dport 22 accept
```

En el cliente, instalar el cliente SSH:

```
sudo apt-get install openssh-client
```

Para conectarse desde el cliente:

```
ssh user@192.145.6.23
```

La primera vez aparecerá un aviso preguntando si confías en la huella digital del servidor remoto. Si aceptas, se guardará en el archivo oculto `.ssh/known_hosts`.

Si en el futuro la huella cambia, puede ser una señal de que alguien está interceptando la conexión.

Tras aceptar, se pedirá la contraseña y se iniciará la sesión remota.

9.2 TeamViewer

TeamViewer es una de las herramientas de terceros más utilizadas para la gestión remota. En este caso, la conexión no se realiza directamente entre los ordenadores, sino a través de los servidores de TeamViewer.

Su configuración es sencilla: basta con instalar el programa en ambos equipos. Cada uno se conectará a los servidores de TeamViewer y se mostrará un ID y una contraseña, que deberán introducirse para acceder remotamente.

Puedes ver un vídeo explicativo del proceso en el curso de Moodle.

10. RECURSOS COMPARTIDOS

A nivel de usuario, una de las grandes ventajas de las redes es la posibilidad de compartir recursos, especialmente archivos e impresoras. Existen distintas opciones, como NFS, pero en esta unidad trabajaremos con **SAMBA**, que permite compartir recursos entre sistemas Linux y Windows.

10.1 SAMBA

Actualmente, gran parte de la funcionalidad de SAMBA es transparente para el usuario final: un usuario de Linux puede abrir el explorador de archivos, buscar equipos Windows en la red y acceder a los recursos compartidos.

No obstante, en muchas situaciones no se dispone de entorno gráfico o se requiere una configuración más detallada.

En la plataforma se enlazan dos vídeos (<https://www.youtube.com/watch?v=zTuiwRSsIBw> y https://www.youtube.com/watch?v=p2r0kIB_ItE) sobre la instalación y configuración de SAMBA.

11. BIBLIOGRAFÍA

- Computer networks. S. Tanenbaum Andrew. Pearson. 2010

- Como funciona un servicio DHCP

<http://windowserver.wordpress.com/2013/09/20/cmo-funciona-el-servicio-dhcp-incluye-caracteristicas-de-red/>

- Computer networks. S. Tanenbaum Andrew. Pearson. 2010