I.E.S Alonso de Ercilla
CFGS Desarrollo de Aplicaciones Multiplataforma

SISTEMAS INFORMÁTICOS

Sistemas informáticos en red
Configuración y explotación

Castilla-La Mancha

IES Alonso de Ercilla
Puerta de Murcia, 13
45300 Ocaña
www.iesalonsodeercilla.com

UNIÓN EUROPEA
Fondo Social Europeo
El FSE invierte en tu futuro
Financiado como parte de la respuesta de la Unión a la pandemia de COVID-19

Las enseñanzas de Formación Profesional de Grado Superior: **Desarrollo de Aplicaciones Multiplataforma**, que durante el presente curo se está impartiendo en nuestro Centro, está siendo cofinanciado por el **Programa Operativo del Fondo Social Europeo de Castilla la Mancha, a través De los recursos adicionales REACT-UE**

¿Qué es INTERNET?

INTERNET es conocida como ***LA RED DE REDES***, es decir, que se encarga de conectar distintas redes entre sí (normalmente corresponden a servicios). Ejemplo de redes son:

- Operadoras (Movistar, vodafone, Orange, Jazztel, etc…)
- Motores de búsqueda (Google, Yahoo, Bing, etc…)
- Redes sociales (Facebook, Twitter, Instagram, etc…)
- Servicios en la nube (Drive, iCLoud, Dropbox, etc…)
- Servicios de Streaming (Netflix, HBO, Twitch, etc…)
- Institutos/Universidades (UCLM, URJC, IES Alonso de Ercilla, etc..)
- …

¿Qué es una IP? (Internet Protocol)

Para poder comunicarse dos equipos dentro de una red, ya sea un móvil, una tablet, una televisión, un ordenador, etc, es necesario disponer de algún mecanismo para saber dónde enviar/recibir esta información. Por ejemplo,cuando realizado un pedido en amazon, lo que hacemos es indicar un **nombre y apellidos, dirección, población, ciudad, cp,..**. para poder realizar la entrega, por lo que estamos indicando de **forma precisa** dónde queremos recibir nuestro paquete tan deseado.

De igual manera, cada dispositivo dispone de esa información para poder ser localizado de forma numérica.

Cuando indicamos una IP, lo que hacemos es referencia a un **NÚMERO ÚNICO** identificativo para un dispositivo dentro de la red de internet, por lo que si deseo comunicarme con alguien por Whatsapp, es necesario saber ambas IP’s (números únicos en todo internet).

IPv4 (Internet Protocol Version 4)

Existen dos versiones con para expresar una IP: v4 y v6, siendo la v4 (IP versión 4), la más utilizada hoy en día.

Ya hemos dicho que la IP es una dirección única para cada dispositivo. Lo que hace nuestro proveedor de servidor de internet, es asignarnos una de estas IP's. Dicha IP **no es fija**, y va cambiando dependiendo de las necesidades de la red de la operadora en ese momento.

El formato de una IPv4 utiliza 32 bits, desglosados en 4 bloques de 8 bits separados por puntos, de tal manera que cada bloque representa un número comprendido entre 0 y 255. es del tipo xxx.xxx.xxx.xxx

Por ejemplo: 192.168.1.100

¿Como podemos comunicarnos con un equipo de otra red?

En este ejemplo vamos a comunicarnos con un servidor de google

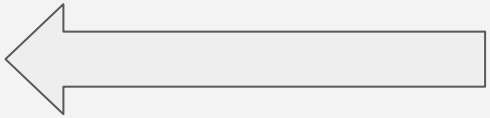
216.58.209.67

INTERNET

98.45.23.109

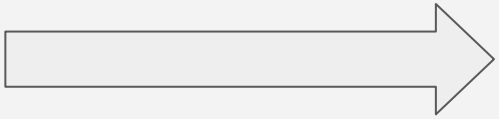
Realizamos una búsqueda en Google.
Para ello enviamos un paquete.

IP Origen: **98.45.23.109**
IP Destino: **216.58.209.67**



Se recibe el paquete, y se devuelve otro paquete de vuelta (el resultado de una búsqueda)

IP Origen: **216.58.209.67**
IP Destino: **98.45.23.109**



Recibimos el resultado de la búsqueda en nuestro equipo

En el ejemplo anterior hemos visto como es necesario indicar una IP de origen y una IP de destino para poder identificar a cada equipo, para poder enviar y recibir mensajes (paquetes)

Cuando nos conectamos desde nuestra casa con varios dispositivos, lo estamos haciendo con **la misma IP pública**, es decir, tu televisión, ordenador, teléfono, etc, tendrá la misma IP para cualquier petición que hagas a Google o cualquier otro servidor o red. Pero, ¿cada dispositivo no debe tener una IP pública única? Si, y es aquí donde entra en juego el **ROUTER**.

Todos los aparatos de nuestra casa se conectan a este dispositivo electrónico llamado router, el cual se encarga de enviar los paquetes hacia al exterior: hacer búsquedas en google, entrar a redes sociales, ver vídeos, subir tareas a Moodle, etc.

Es el router quien dispone de la **IP PÚBLICA**, asignando **IP PRIVADAS** a cada dispositivo conectado, por lo que nuestro móvil, televisión, etc, tendrá una **IP PRIVADA ÚNICA** asignada dentro de una **red de área local (LAN)** la cual es generada por el router.

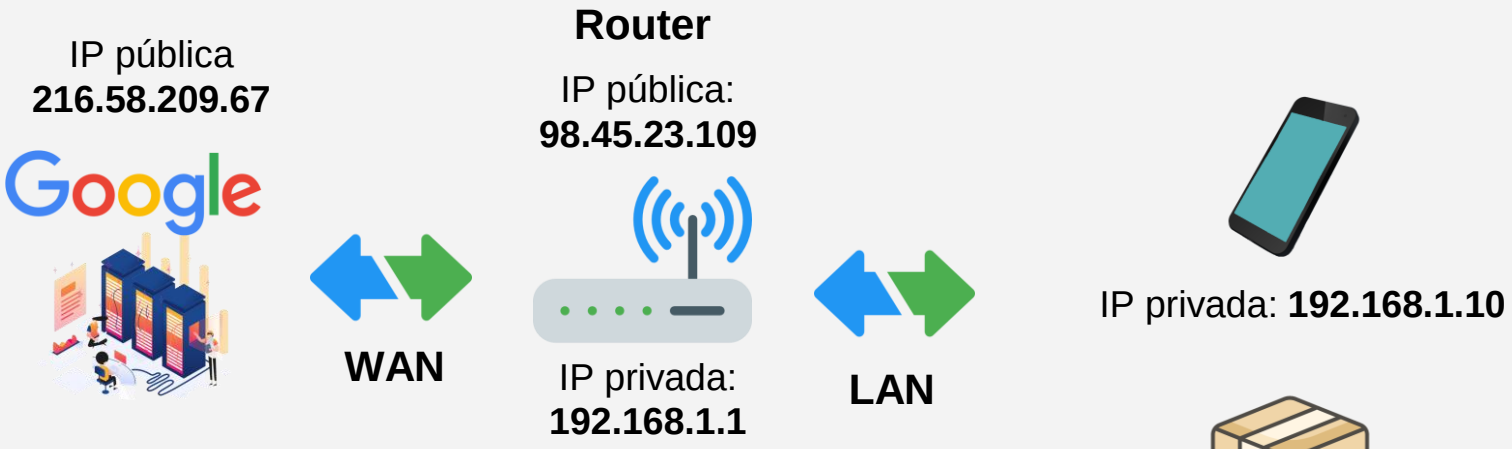
IP PRIVADAS

Cada equipo dentro de una LAN dispone de su propia IP privada, la cual es asignada por el router de forma automática, pero también la podemos asignar esta IP privada de forma manual. Normalmente estas IP's tienen el formato : 192.168.**0**.xxx - 192.168.**1**.xxx (cambia el 0 por el 1), en los ejemplos usaremos la 192.168.**1**.xxx. Recuerda que este formato de IP's puede cambiar.

Podremos encontrar que el router dentro de la LAN queda identificado por la IP privada 192.168.1.1. Esta IP es denominada **puerta de enlace** o **Gateway,** por lo que todos los equipos dentro de la red deberán hacer referencia a esta IP en su configuración. Un conjunto de equipos conectados a una LAN con esta puerta de enlace, dispondrán de IP's privada como por ejemplo: 192.168.1.**10**, 192.168.1.**11**, 192.168.1.**12**, …

Lo que se mantiene son los 3 primeros bloques de dígitos, cambiando el último bloque (hasta 255).

Ejemplo comunicación desde una red LAN.



Disponemos de 2 equipos conectados al router, cada uno con su IP privada.
El móvil quiere realizar una búsqueda en google. Para ello prepara un paquete con la IP de destino (**216.58.209.67**), y su IP privada (**192.168.1.10**).

Ejemplo comunicación desde una red LAN.

IP pública
216.58.209.67

WAN

Router

IP pública:
98.45.23.109

IP privada:
192.168.1.1

LAN

IP privada: **192.168.1.10**

IP privada: **192.168.1.11**

IP Origen: ~~**192.168.1.10**~~

98.45.23.109
IP Destino: **216.58.209.67**

<< Cambia la IP privada, por la IP pública
Este dato queda almacenado en el Router para que cuando se reciba un paquete sepa a quién va dirigido.

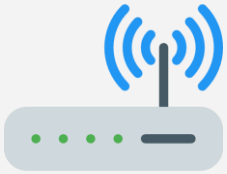
Ejemplo comunicación desde una red LAN.

Router

IP pública
216.58.209.67

IP pública:
98.45.23.109

WAN



IP privada:
192.168.1.1

LAN

IP privada: **192.168.1.10**

IP privada: **192.168.1.11**

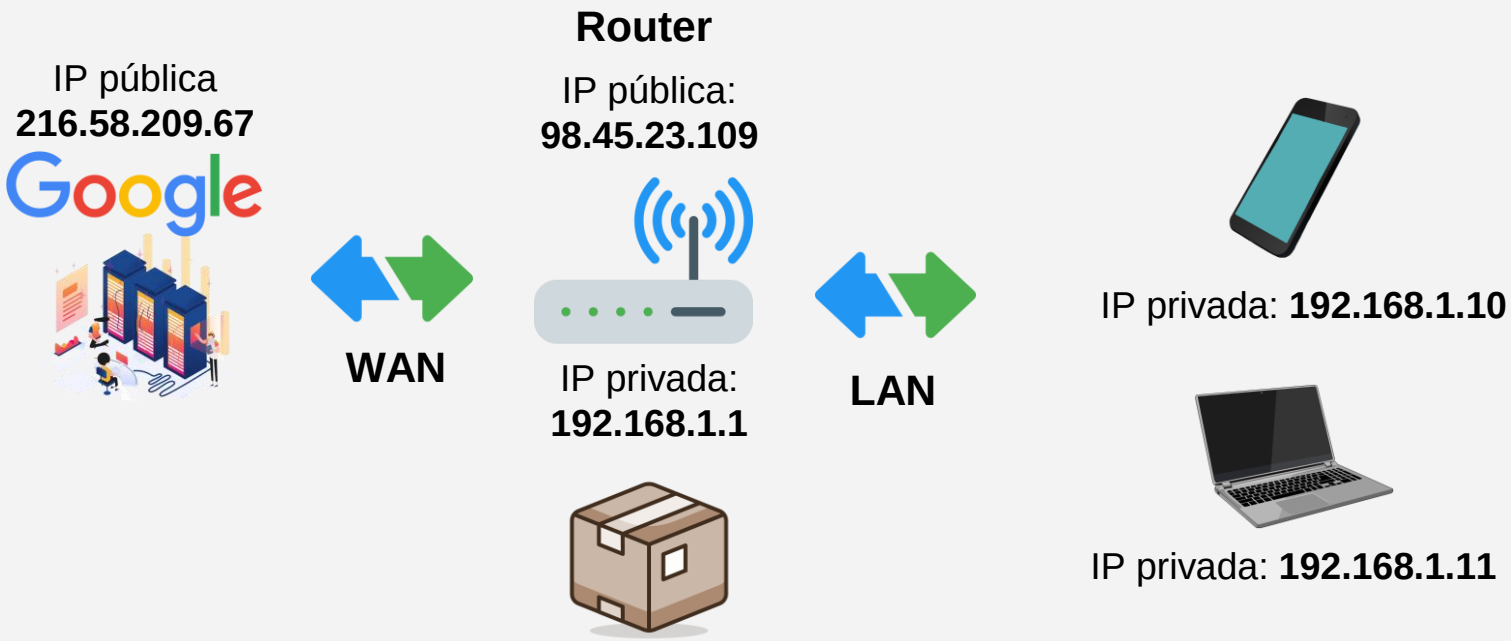
IP Origen: **98.45.23.109**
IP Destino: **216.58.209.67**

El paquete es recibido por el servidor de Google, el cual prepara un paquete de vuelta.

IP Origen: **216.58.209.67**
IP Destino: **98.45.23.109**

El paquete que envía Google tiene como IP de destino nuestra IP pública, la cual está asignada a nuestro router

Ejemplo comunicación desde una red LAN.



IP Origen: **216.58.209.67**
IP Destino: ~~**98.45.23.109**~~
_____
_____**192.168.1.10**

Nuestro router recibe el paquete, y cambia la IP de destino por la IP privada del dispositivo a quien corresponde la petición realizada (el móvil).

Ejemplo comunicación desde una red LAN.

IP pública
216.58.209.67

WAN

Router

IP pública:
98.45.23.109

IP privada:
192.168.1.1

LAN

IP privada: **192.168.1.10**

IP Origen: **216.58.209.67**
IP Destino: **192.168.1.10**

Finalmente nuestro móvil recibe el paquete desde Google >>

Si queremos realizar una petición desde nuestro ordenador, o cualquier otro dispositivo, el router actuaría de igual manera.

IP privada: **192.168.1.11**

Esta operación que realiza el router se conoce con el nombre de **NAT (Network Address Translation)**, y consiste en traducir las IP privadas en IP públicas para manejar todo el tráfico de nuestra LAN.

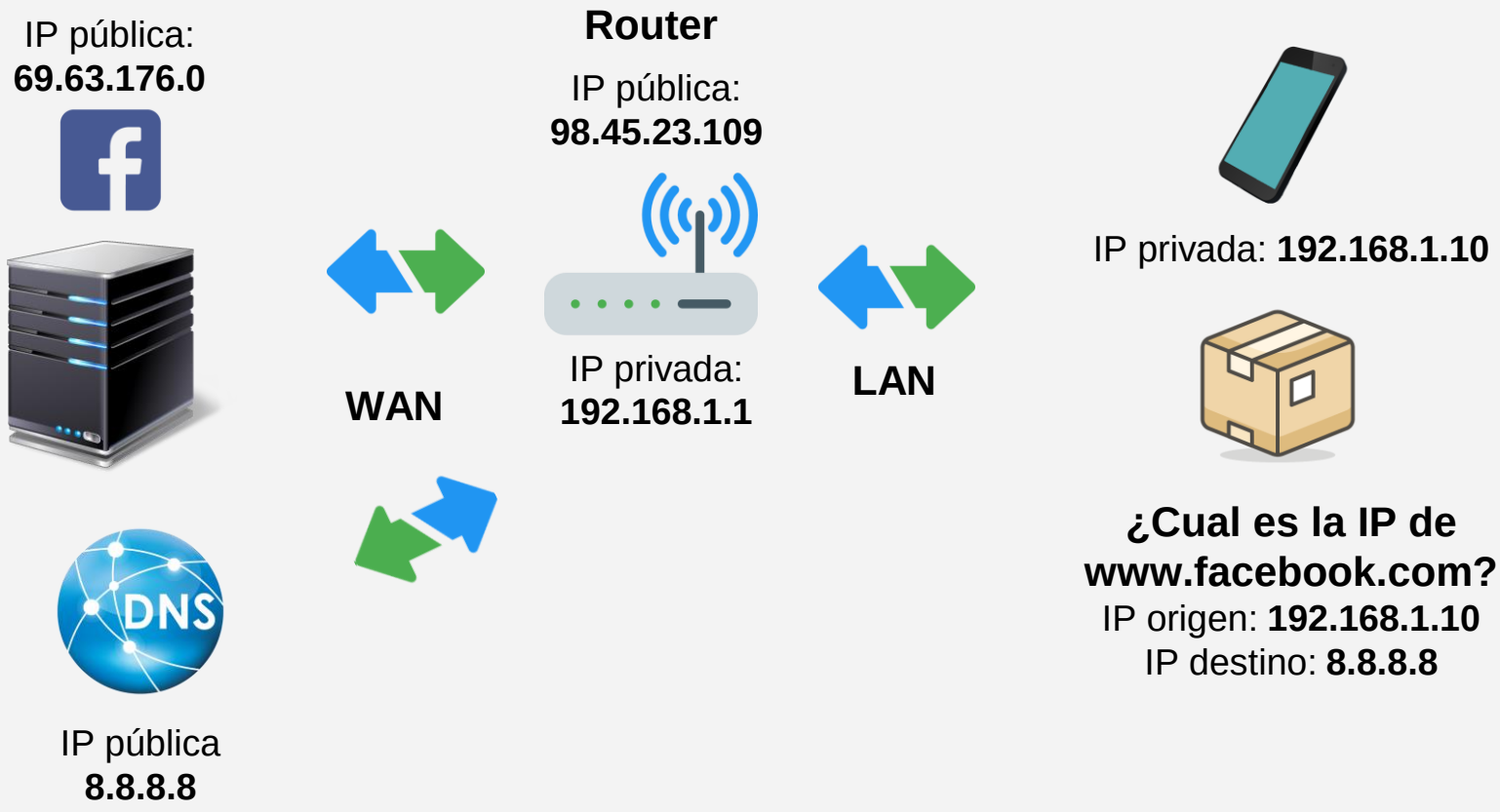
DNS (Domain Name System)

Hasta ahora sabemos que para poder comunicar a dos dispositivos dentro de internet es necesario disponer de una IP pública única en cada dispositivo. Pero, ¿no es algo complicado acordarse de cada IP pública para comunicarnos?,¿no es más sencillo utilizar nombres?. Efectivamente, y para ello se inventó los servidores **DNS**.

Estos servidores se encargan de traducir un **nombre de dominio, en una IP pública**. Por ejemplo, cuando queremos efectuar una conexión a facebook.com, lo que nuestro equipo hace es preguntar por el nombre de facebook.com a un servidor DNS, el cual tiene una tabla con la correspondencia entre el nombre facebook.com y la IP pública correspondiente.

Todos los proveedores de internet tienen sus propios servidores DNS.
Estos servidores pueden asignarse en la configuración del router, y en la configuración de nuestro equipo. Si no ponemos nada se elegirá la opción por defecto del proveedor de internet (Movistar, Orange, Jazztel, etc…)

Ejemplo de funcionamiento de una petición usando un DNS



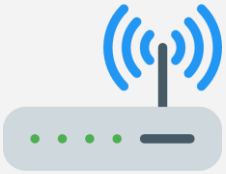
Para enviar un paquete a facebook, primero necesitamos conocer su IP pública. Para ello, realizamos una petición a un servidor DNS (8.8.8.8 en el ejemplo)

Ejemplo de funcionamiento de una petición usando un DNS

IP pública:
69.63.176.0

Router

IP pública:
98.45.23.109



IP privada:
192.168.1.1

WAN

LAN

IP privada: **192.168.1.10**

IP pública
8.8.8.8

¿Cual es la IP de www.facebook.com?
IP origen: **98.45.23.109**
IP destino: **8.8.8.8**

El paquete es recibido por el servidor DNS (previamente ha pasado por el NAT del router). El DNS buscará la IP que corresponde a facebook.com, y devolverá un paquete con la respuesta.

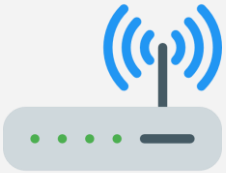
Ejemplo de funcionamiento de una petición usando un DNS

IP pública:
69.63.176.0

IP pública
8.8.8.8

Router

IP pública:
98.45.23.109



IP privada:
192.168.1.1

WAN

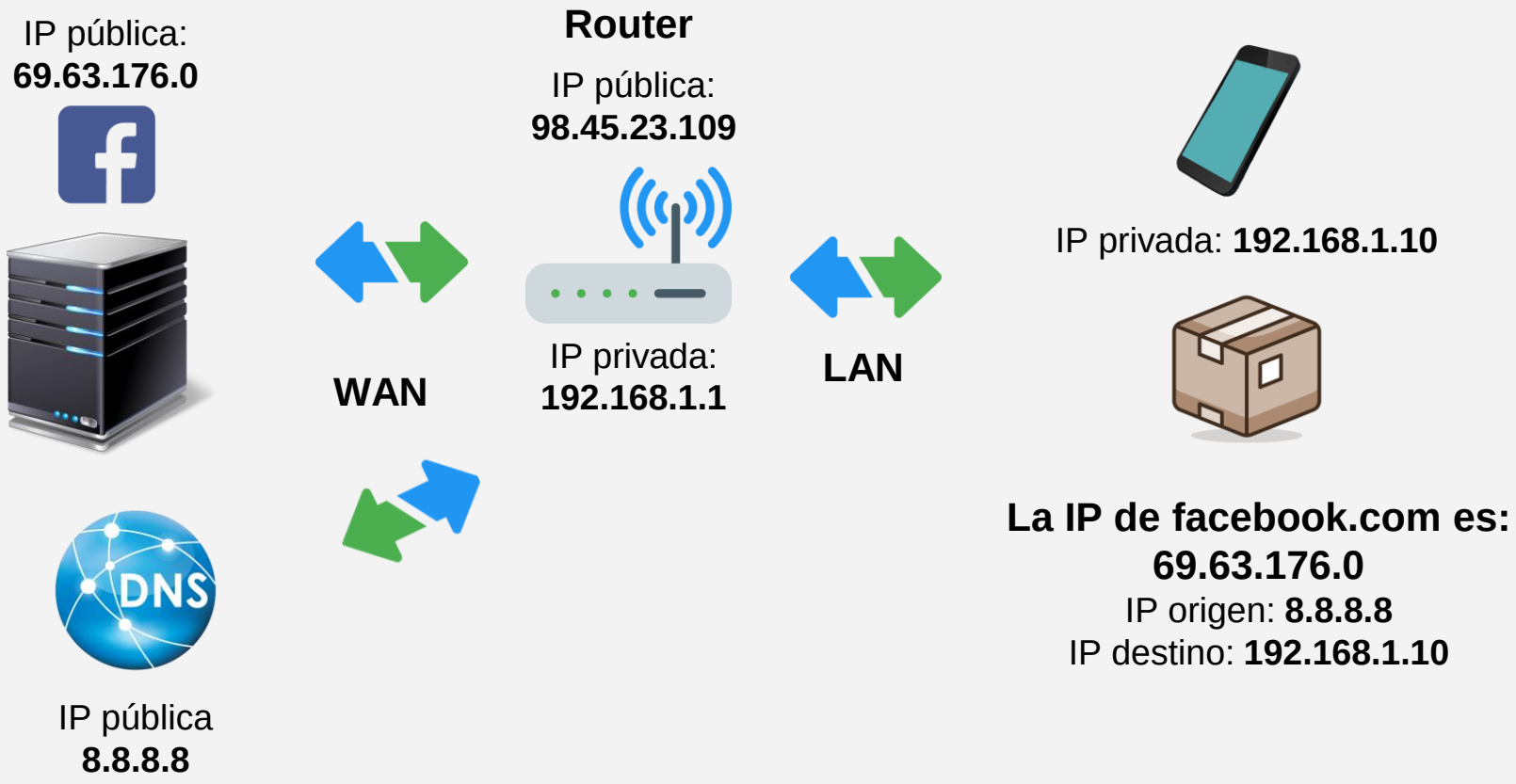
LAN

IP privada: **192.168.1.10**

El servidor DNS prepara el paquete con la dirección IP de facebook.com, y lo devuelve al router (gracias a la IP pública del router)

La IP de facebook.com es:
69.63.176.0
IP origen: **8.8.8.8**
IP destino: **98.45.23.109**

Ejemplo de funcionamiento de una petición usando un DNS



Nuestro móvil ya dispone de la IP pública de facebook, por lo que ya puede hacer una petición hasta esa IP, repitiendo los pasos vistos en el ejemplo de comunicación desde una red LAN.

Sobre DNS

La elección de un DNS lejos de ser algo arbitrario, es algo fundamental en nuestra configuración, ya que de ello depende:

- La **velocidad** de nuestras peticiones: una página web dispone de varias peticiones DNS dentro de la misma.
- La **seguridad y privacidad** de nuestros datos. Las peticiones son públicas, y las empresas pueden obtener hábitos de navegación.
- **Control de acceso** (parental, evitar publicidad, bloquear contenido, etc)

Podemos configurar un DNS:

- En el router (ya viene preconfigurado por el ISP).
- Cada dispositivo puede tener un servidor DNS.
- Un dispositivo específico dentro de nuestra LAN.

CLOUDFLARE **1.1.1.1**

Google **8.8.8.8**

quad9 **9.9.9.9**

Puertos

Un puerto es un número que se asigna a una aplicación o servicio que está siendo ejecutado dentro de un sistema informático.

Los servicios o aplicaciones al ser instalados utilizan siempre el mismo puerto (podemos indicar otro), de esta forma podemos conocer en todo momento a qué número hay que hacer referencia en caso de querer utilizar el servicio. El rango de puertos que puede ser asignado en un equipo es 0-65535, siendo 0-1023 los puertos denominados como “bien conocidos” y son usados en servicios específicos **(es importante no usar estos puertos en servicios propios).**

21

22

587

1433

80

9100

3306

445

25565

443

No confundir este número con el PID

Funcionamiento de los puertos. Sockets

Como ya sabemos, cuando queremos enviar un paquete desde un equipo a otro, necesitamos indicar la IP de origen y la IP de destino. Dado que es un servicio o aplicación quien se quiere comunicar con otro servicio o aplicación, es necesario indicar también el puerto de origen y el puerto de destino.

Por ejemplo cuando nosotros abrimos el Firefox, Discord, Drive, uTorrent, Teams, Outlook, o cualquier aplicación que se comunique con internet necesita de un puerto.

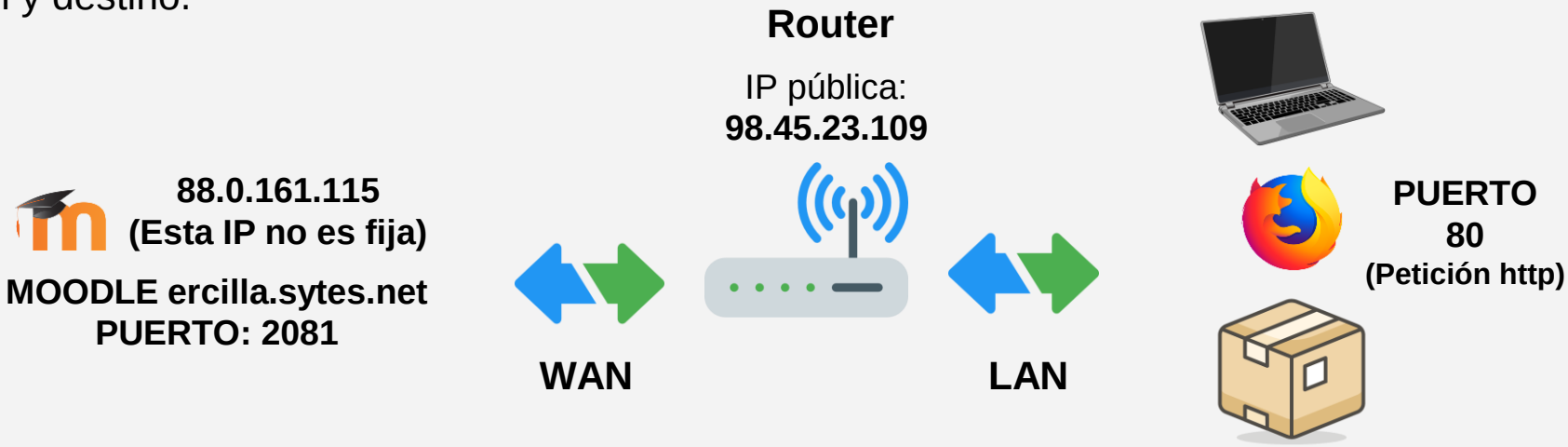
Como hemos visto en la anterior diapositiva, algunos servicios o aplicaciones tienen asignado un puerto definido, en caso contrario, es nuestro equipo (SO) quien asigna un número de puerto.

Funcionamiento de los puertos. Sockets

En este ejemplo vamos a obviar tanto el NAT como el DNS, por lo que trabajaremos con la IP pública del equipo, como la misma del router.

El ejemplo consiste en comunicarnos con nuestro Moodle del centro. Lo primero es abrir un navegador (por ejemplo Firefox). Al usar el protocolo http se utiliza el puerto por defecto 80. En caso deEl SO asigna un número de puerto al Firefox para poder realizar la comunicación.

Por tanto, al crear el paquete, además de poner las IP's es necesario poner los puertos de origen y destino.



NOTA: El puerto 2081 hace referencia al puerto de escucha de la máquina servidor Moodle, que a su vez redirige al puerto 80 (http), que es el puerto del servidor web (el que nos permite ver realmente el contenido del Moodle).

IP Origen: **98.45.23.109**
IP Destino: **88.0.161.115**
Puerto Origen: **80**
Puerto destino: **2081**

Funcionamiento de los puertos. Sockets

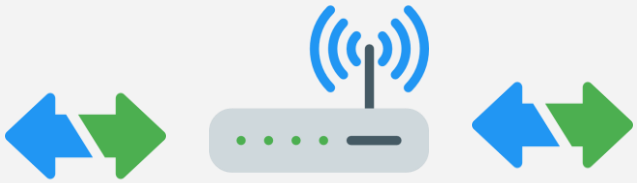
Una vez el servidor de Moodle recibe el paquete (realiza las operaciones pertinentes), y devuelve el paquete al destino, solo que ahora las IP's de origen son las de destino, y viceversa. Al disponer de las IP's y los puertos correspondientes, el paquete llegará a su destino sin problema.

88.0.161.115
(Esta IP no es fija)

MOODLE ercilla.sytes.net
PUERTO: 2081

Router

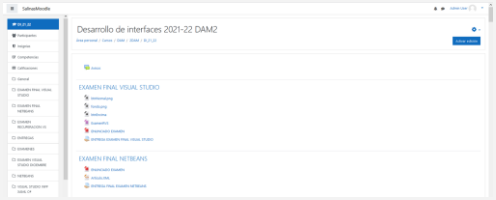
IP pública:
98.45.23.109



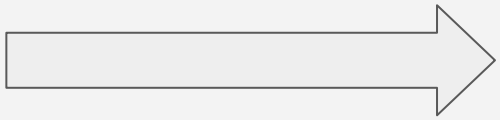
WAN

LAN

PUERTO
80



IP Origen: **88.0.161.115**
IP Destino: **98.45.23.109**
Puerto Origen: **2081**
Puerto destino: **80**

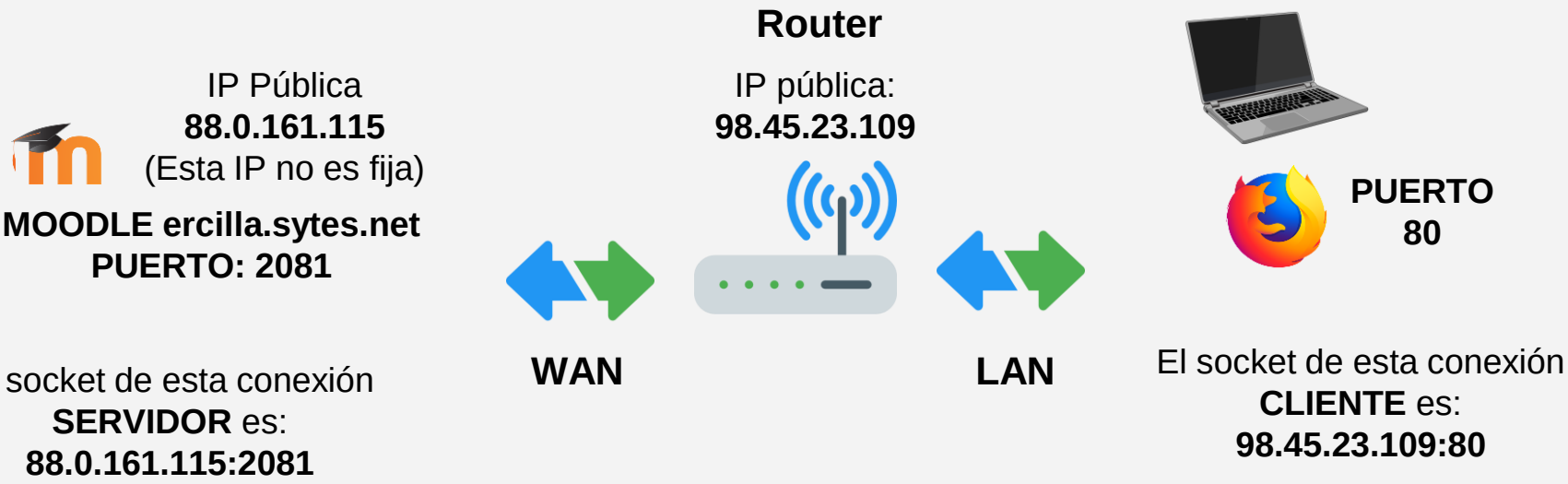


El router localiza al equipo de destino, y nuestro equipo deriva el paquete de destino a la aplicación Firefox gracias al número de puerto.

Se prepara el paquete para ser enviado de vuelta a nuestro equipo

Funcionamiento de los puertos. Sockets

A la unión de la IP con el puerto se conoce con el nombre de **SOCKET,** y tiene el siguiente formato **IP:PUERTO**



Este ejemplo sigue una estructura **CLIENTE - SERVIDOR**, donde el servidor corresponde con el **Moodle**, y el cliente es nuestro **Firefox**.

El cliente es quien inicia la comunicación (envía el primer paquete) a un destino, que es el servidor.

El servidor nunca inicia una comunicación por defecto, sino que se encuentra esperando una comunicación, es decir, en nuestro ejemplo Moodle nunca inicia una comunicación.

MODELO OSI (Open System Interconnection)

Al principio las comunicaciones se efectuaban de una forma muy caótica, ya que cada empresa fabricaba su propio Hardware, lo que implicaba que eran sistemas muy cerrados. Una tarjeta diseñada por una compañía no era compatible con una placa base desarrollada por otra. Esto suponía una gran limitación en las comunicaciones, por lo que era necesario adoptar un modelo común en que todos los fabricantes se pusieran de acuerdo para diseñar sus componentes acordes a una normativa común.

Por ejemplo, cuando queremos ver un vídeo en Youtube, el proceso que debe realizar el sistema es complejo ya que el vídeo puede verse en distintos dispositivos (ordenador, tablet, televisión, …), con sistemas operativos distintos (Android, iOS, Windows, Linux, …), en diferentes programas de navegación (Firefox, Chrome, Edge, …), con diferentes resoluciones de pantalla, con diferentes formas de transferencia de datos como puede ser por cable Ethernet, Wifi, Fibra, 5G, etc…

MODELO OSI

Para simplificar esta comunicación lo que hacemos es dividir esta comunicación en partes más pequeñas, cuya parte podría corresponderse con una capa, es decir, realizamos la abstracción de algo muy complejo, en partes más pequeñas manejables, donde cada capa

MODELO OSI

CAPA FÍSICA: Hace referencia al hardware, tipo y forma de conectores, categoría y materiales de los cables, potencia, tensión, intensidad de señal, forma de onda, banda de emisión, frecuencia.

CAPA ENLACE (Tarjeta de red, Switch): Encapsular datos en tramas, direccionamiento físico mediante MAC, verificar que los datos enviados/recibidos son correctos, control de colisiones.

CAPA DE RED (Router): Enrutar datos en la red (ruta más eficiente), direccionamiento lógico, asignación de IP, topología de la red, dispositivos conectados.

CAPA DE TRANSPORTE: Conexión de extremo a extremo (TCP, UDP, SCTP, SSL, TLS)

CAPA DE SESIÓN: Se establecen los mecanismos para controlar el diálogo entre las aplicaciones de los sistemas finales (como se van a comunicar ambas aplicaciones)

CAPA PRESENTACIÓN: Transformar y presentar los datos de forma correcta (representación y encriptación de los datos entre aplicaciones)

CAPA DE APLICACIÓN: Corresponde con los programas finales.

CAPA FÍSICA.

Clasificación de las redes

Las redes se pueden clasificar atendiendo a los siguientes criterios:

Según tamaño.

- **PAN (Red de área personal):** Bluetooth, Zigbee, NFC
- **LAN (Red de área local):** Red aplicada en hogares, oficinas, empresas, edificios, institutos, etc…
- **MAN (Red de área metropolitana):** Redes de extensión intermedia entre LAN y WAN. Ejemplo de estas redes pueden ser conexiones entre poblaciones próximas, campus universitarios.
- **WAN (Red de área extensa):** Redes de larga distancia (Internet, Redes bancarias, Redes militares, …)

- Según los medios empleados:
  - **Inalámbricas:** Bluetooth, WIFI, GPS, 5G, …
  - **Cableadas:** par trenzado de cobre, fibra óptica, …
  - **Mixtas:** utilizan medios inalámbricos y cableados.

CAPA FÍSICA:

Determina las especificaciones mecánicas, eléctricas y funcionales que establece y mantiene el enlace físico de transmisión. La trama, construida por bits, se traduce en señales eléctricas, electromagnéticas o pulsos de luz, hasta que llegan al receptor, donde se vuelven a convertir a bits.

Las redes cableadas son a día de hoy, las más fiables y rápidas debido a su conexión física mediante:

- cables de cobre (coaxial o par trenzado)
- fibra óptica.

Conector RJ45 para cable UTP

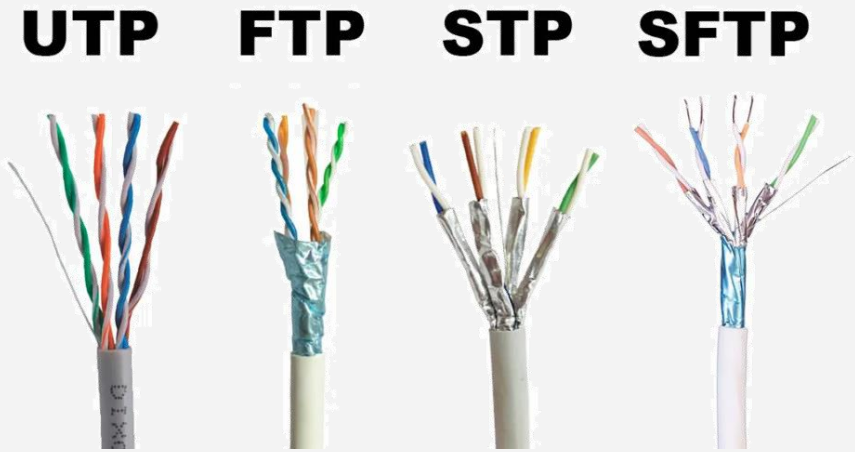
Conector RJ45 para cable apantallado

Conectores de fibra óptica

Cable de cobre de par trenzado. Redes cableadas.

El cable de par trenzado está formado por 8 cables de cobre aislados y entrelazados, identificados por el color individual de su cubierta. Los cables están entrelazados de la siguiente manera:

Azul - Blanco/azul
Naranja - Blanco/naranja
Verde - Blanco/verde
Marrón - Blanco/marrón



Apantallado

El apantallamiento o blindaje de los cables es un elemento fundamental que consigue eliminar posibles interferencias externas (ruido) en las comunicaciones de red.

El cable más sencillo es el tipo **UTP** (Unshielded Twisted Pair), el cual no dispone de ningún apantallamiento o blindaje.

Los cables con apantallamiento **FTP** (Foiled Twisted Pair), SPT o SFTP (es el más recomendado para realizar cualquier conexión de red.

CAPA FÍSICA:

Uno de los dispositivos que podemos encontrar en la capa física es el **HUB** o concentrador. Este dispositivo se encarga de emitir **en todos sus puertos** una trama **(broadcast)**, lo que significa que no puede redirigir el tráfico a un equipo concreto. Esto provoca un fallo de seguridad en la red y un tráfico innecesario en la misma, a parte de que se pueden producir colisiones de datos.

Por todo ello, estos dispositivos son desplazados por los Switches (capa de enlace), los cuales permiten conocer a qué dispositivo va dirigida una trama de datos.

Debemos tomar atención ya que ambos dispositivos son muy parecidos.

Concentrador HUB

Dispositivo Switch

Símbolo switch

CAPA ENLACE (Tarjeta de red - Switch).
Dentro de esta capa podemos encontrar estas subcapas:

- **LLC (Control de enlace lógico) 802.2.** Hace referencia a los mecanismos que se encargan de transformar las señales recibidas (por ejemplo una señal WIFI), a señales digitales dentro del sistema. Comprueba los tipos de protocolos de transmisión.
- **MAC (Control de acceso al medio):** se encarga del direccionamiento de los paquetes de entrada y salida. Supongamos que existen diferentes dispositivos conectados en la red, cada dispositivo dispone de una dirección MAC, (también conocida como dirección física), y es un número identificador que corresponde **de forma única** a una tarjeta o dispositivo de red. Por tanto, esta capa se encargaría de comprobar estas direcciones, y de agregar la dirección de origen y destino en cada una de las tramas que se transmiten (paquetes).

También se encarga de detectar y corregir errores de transmisión. En caso de haber problemas con los datos recibidos (datos corruptos), se intentan reparar mediante mecanismos. También se encarga de evitar colisiones entre paquetes.

CAPA DE ENLACE:

Una tarjeta de red (aún siendo hardware) pertenece a la capa 2 (capa de enlace), no a la capa 1 (capa física). Esta tarjeta almacena en una memoria no volátil (ROM) la dirección de memoria física (MAC), y consta de 48 bits, expresada en 18 dígitos hexadecimales.

Algunos términos que debes conocer:

- **ASIC:** Hardware dedicado con latencia baja.
- **CAM (Memoria de contenido direccionable):** Almacena la dirección física (MAC) de los diferentes dispositivos conectados en un puerto determinado.
- **Conmutación de capa 2:** Enviar tramas de un switch a otro.
- **Inundación:** Cuando el switch recibe una trama con una MAC no identificada, se envía dicha trama a todos los dispositivos conectados (excepto aquel que envió la trama).
- Un **dominio de difusión** hace referencia a un **switch**.
- Cada uno de los puertos de un switch crea un **dominio de colisión**.

CAPA DE RED (Router): También se hace referencia a esta capa como **enrutamiento**, y los elementos que podemos encontrar en esta capa son los routers, o los switches de capa 3 (o L3)**.**

Esta capa maneja los siguientes protocolos:

- OSPF
- **IP**
- IPSec
- **ARP**
- **NAT**
- ICMP
- …

Su objetivo es transportar el tráfico de datos entre dispositivos que **no están conectados localmente en un mismo dominio de difusión**, es decir, dos equipos de distintas redes (por ejemplo la red del centro con internet). El dispositivo que se suele encargar de este propósito es el router.

Símbolo router

Router WIFI

CAPA DE RED (Router):

Los routers disponen de su propio sistema operativo, por lo que necesitan de una CPU, memoria RAM, y ROM.

Tanto routers como hosts, utilizan **tablas de enrutamiento** para encaminar los paquetes a otros dispositivos de una red local o remota, principalmente 2 hosts que no se encuentren en una misma red local, ya que si se encuentran en la misma red es el switch el encargado de realizar esta comunicación. Por ello los routers suelen disponer de un switch de 4 puertos (normalmente), en su parte trasera.

Switch en un router

Los routers disponen de tres tipos de entradas en sus tablas de enrutamiento:

- **Conexiones locales.** Conectadas directamente por alguna interfaz del router.
- **Conexiones estáticas.** Establecidas manualmente por el administrador de la red.
- **Conexiones dinámicas.** Entradas que han sido aprendidas mediante algún algoritmo de enrutamiento. Estos algoritmos son utilizados por los routers para comunicarse e intercambiar entradas entre ellos. La mayoría de las entradas son de este tipo.

Protocolo DHCP.

Cuando nos conectamos a nuestro router ya sea por cable o por conexión inalámbrica, el router nos asigna de manera automática una dirección de red privada. Aquí es donde interviene el DHCP (Dynamic Host Configuration Protocol), el el cual asigna acorde a los valores establecidos, y a la máscara de red, la dirección IP para el host. Al ser una **asignación dinámica**, el número de IP cambiará cada vez que efectuemos una conexión en nuestro router.

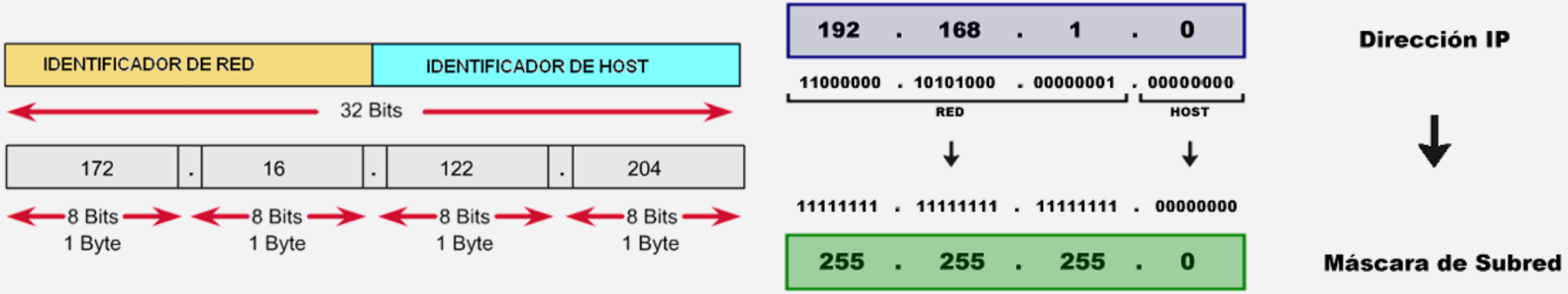
Los routers **SoHo** (Small office Home office) que son los que solemos disponer en nuestras casas, traen por defecto un pequeño servidor DHCP ya activado. Estos servidores permiten establecer el rango de direcciones asignables por el protocolo, es decir, podemos indicar a partir de que IP serán asignadas de forma automática.

De esta manera podemos disponer tanto equipos con IP's configuradas de forma automática, y de equipos que deben tener una **IP fija** dentro de la red, (ya que esto permite poder localizar de forma rápida un recurso o servicio). Por ejemplo, en la típica red clase C se podría indicar que se asignen de forma automática a partir de la 192.168.1.20, dejando las primeras IP para diferentes servicios dentro de la red.

Este protocolo aún siendo parte del router, **no es parte de la capa de red** ya que no es parte del enrutamiento. La asignación de IP's sucede a nivel de capa 7 (capa de aplicación).

CAPA DE RED (Router):

Como ya hemos visto, el protocolo IPv4 está formado por 4 bloques de 8 bits (32 bits en total) separados por puntos. De tal manera que cada bloque representa un número comprendido entre 0 y 255 (en decimal). Este protocolo necesita una máscara de red con el mismo formato que una dirección IP. De esta manera, se identifica la red a la que pertenece la dirección IP.



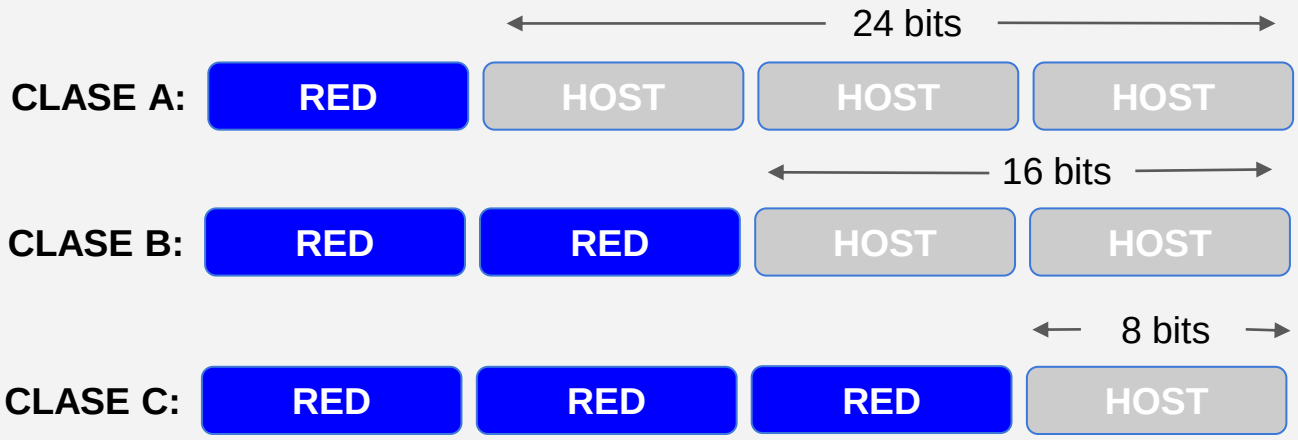
Intrínsecamente, la dirección IP se divide en una porción de red y una porción de host. La importancia de la máscara de red radica en que esta determina qué bits de la dirección IP se corresponden con la red a la que pertenece y qué bits especifica el host dentro de dicha red.

Otra forma de representar las máscaras de red es por el número de bits que se utiliza para red. Por ejemplo: 192.168.1.10 ***/24,*** siendo 24 los bits destinados a red.

CAPA DE RED (Router):

La cantidad de bits de red y de host depende de la clase a la que pertenece la dirección.

Tipo de Red	Rango de Red	Nº Bits para Red	Máscara de Subred
Clase A	0.0.0.0 - 127.255.255.255	8	255.0.0.0
Clase B	128.0.0.0 - 191.255.255.255	16	255.255.0.0
Clase C	192.0.0.0 - 223.255.255.255	24	255.255.255.0



Para obtener la clase de una red nos tenemos que fijar en el primer octeto, así por ejemplo una IP 192.168.1.39 será de clase C ya que se encuentra entre el rango de red de la tabla anterior (192 - 223). La IP 10.10.20.50 será clase A (0 - 127)

CAPA DE RED (Router):

Una máscara de red puede expresarse bien de forma de IP, o bien como prefijo de red. Por ejemplo, para una IP clase C 192.168.1.10 la cantidad de bits asignada para red es de 24. La expresión de la máscara de red es:

11111111.11111111.11111111.00000000

24 bits — 8 bits

255 . 255 . 255 . 0

Forma IP: **255.255.255.0** Forma prefijo: ***/24***

La IP se suele expresar de la forma 192.168.1.10/24, con lo que sabemos que los 8 bits en su totalidad son empleados para direcciones hosts (para equipos de la red), esto quiere decir que en una red podemos tener 2^8 IPs (256 IPs).

Es importante saber que la primera IP, hace referencia a la IP de red, y la última IP a la de broadcast, las cuales no pueden ser asignadas a ningún equipo.
(IPs desde 0 - 255) >> 256-2=254 equipos)

Para ver esto convertimos la parte dedicada a host de decimal a binario. Si ponemos a 1 todos los bits su conversión a decimal es:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

128+64+32+16+8 + 4+2+1 = **255**

CAPA DE RED (Router):
SUBREDES.

Dentro de una red de la clase que sea podemos aprovechar parte de los bits dedicados al hosts para crear otras redes. Si por ejemplo disponemos de la dirección de red 192.168.1.0 sabemos que los bits dedicados a host son 8, si queremos subredes a partir de esta red, debemos coger parte de estos bits dedicados a host, y reasignarlos a la parte de red.

Esto sería una máscara de red para una IP clase C:

11111111111111111111111100000000

24 bits — 8 bits

Vamos a utilizar 2 bits de host y los asignamos a la parte de red (**11**)

11111111111111111111111111000000

26 bits — 6 bits

Pasamos a tener 26 bits de red, y 6 bits de host. Esto influye en la máscara de red, ya que ahora deberemos usar un prefijo ***/26***, quedando la IP de máscara:

11111111.11111111.11111111.11000000

255 . 255 . 255 . 192

Al disponer de 2 bits de host para la red, podemos crear $2^2 = 4$ subredes.

CAPA DE RED (Router):
SUBREDES.

Hemos creado por tanto 4 subredes.

Como hemos cogido 2 bits de host, la primera corresponde con **00**

11000000.10101000.00000001.00000000
192 . 168 . 1 . 0

La segunda subred corresponde con **01**

11000000.10101000.00000001.01000000
192 . 168 . 1 . 64

La tercera subred corresponde con **10**

11000000.10101000.00000001.10000000
192 . 168 . 1 . 128

La cuarta subred corresponde con **11**

11000000.10101000.00000001.11000000
192 . 168 . 1 . 192

Siendo la máscara de red para todas ellas:

11111111.11111111.11111111.11000000
255 . 255 . 255 . 192

CAPA DE RED (Router):
SUBREDES.

Dentro de una red de la clase que sea podemos aprovechar parte de los bits dedicados al hosts para crear otras redes. Si por ejemplo disponemos de la dirección de red 192.168.1.0 sabemos que los bits dedicados a host son 8, si queremos subredes a partir de esta red, debemos coger parte de estos bits dedicados a host, y reasignarlos a la parte de red.

Esto sería una máscara de red para una IP clase C:

11111111111111111111111100000000

24 bits — 8 bits

Vamos a utilizar 2 bits de host y los asignamos a la parte de red (**11**)

11111111111111111111111111000000

26 bits — 6 bits

Pasamos a tener 26 bits de red, y 6 bits de host. Esto influye en la máscara de red, ya que ahora deberemos usar un prefijo ***/26***, quedando la IP de máscara:

11111111.11111111.11111111.11000000

255 . 255 . 255 . 192

Al disponer de 2 bits de host para la red, podemos crear $2^2 = 4$ subredes.

CAPA DE TRANSPORTE (Conexión de extremo a extremo):

Se establecen las reglas para la conexión entre 2 equipos, además de la fragmentación y reensamblaje de los paquetes.

Segmentación de paquetes: cuando por ejemplo queremos enviar una carpeta con archivos no lo podemos hacer en un sólo envío, se debe enviar en diferentes archivos, incluso esos archivos deberán ser fragmentados en diferentes partes:

Carpeta a enviar

Sacar el contenido

Dividir cada archivo (fragmentos), y enviar uno a uno.

Cuando estos archivos se reciben en el otro equipo, se debe volver a unir todas estas partes (**re-ensamblaje**)

CAPA DE TRANSPORTE (Conexión de extremo a extremo):

Para poder realizar esta operación, y que se lleve de forma correcta en ambos equipos, se deben establecer unas normas (protocolos), que sean iguales en ambos equipos, es decir, que el equipo receptor sepa unir todos los fragmentos sin equivocarse. Los protocolos disponibles en esta capa son **TCP** y **UDP.**

TCP (Transfer Control Protocol)

Consiste en un acuerdo estandarizado sobre el que se realiza la transmisión de datos. Su objetivo es **crear conexiones dentro de una red**, **garantizando que los datos serán entregados en su destino sin errores en el mismo orden que se transmitieron**. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de **puerto**.

Este protocolo dispone de un “acuse de recibo” lo que garantiza la recepción de todos los paquetes. Por tanto la capa de red (el router) se despreocupa de esta función centrándose en enviar simplemente las tramas.

Para entender esto pongamos el ejemplo anterior de la carpeta, pero esta vez asignaremos un número a cada fragmento, por lo que si el receptor detecta que falta un número en la trama se lo reclamará al emisor para que este vuelva a enviar el fragmento o fragmentos perdidos.

CAPA DE TRANSPORTE (Conexión de extremo a extremo):

UDP (User Datagram Protocol)

Consiste en enviar tramas o fragmentos de código sin establecer previamente una conexión (los datos en la cabecera de trama son suficientes para el destinatario), y tampoco realiza una comprobación de trama, por lo que si se pierde algún fragmento, este ya no será recuperado (no tiene acuse de recibo).

Cuando se fragmentan y envían los datos desde un equipo, estos no son numerados ya que no hay comprobación de datos recibidos. Lo que hace el equipo receptor es ir armando los datos acorde el orden de llegada, y sin importar si llegó completa o no. Esto lo que quiere decir es que a diferencia de TCP, el emisor y el receptor no mantienen un diálogo o conexión, por lo que este protocolo también es llamado **no orientado a la conexión**.

Este protocolo es ampliamente utilizado en eventos donde perder cierta cantidad puntual de información no es relevante, como es en la transmisión de video y de audio, ya que usar un protocolo TCP en estos casos provocan un retardo en la recepción de los datos.

El protocolo UDP por tanto, es mucho más rápido pero menos seguro que el protocolo TCP.

CAPA DE SESIÓN (Comunicación):

Se encarga de establecer, controlar y terminar las sesiones de comunicación entre las aplicaciones y equipos que se estén comunicando. Pongamos el ejemplo anterior de querer enviar una carpeta de archivos, básicamente se trata de preguntar al destino si está disponible en ese momento para recibir información, o si se debe poner a la cola para realizar ese proceso ya que por ejemplo, se encuentra realizando otras operaciones.

Otro ejemplo es cuando estamos navegando por internet y abrimos diferentes pestañas de navegación, **cada una de estas pestañas estaría iniciando una sesión**. Esto tiene su lógica ya que se está estableciendo una comunicación con otro equipo, lo que implica repetir cada uno de los mecanismos ya vistos, es decir, debe seguir todos los pasos de cada capa.

Estas comunicaciones pueden establecerse en modo **“full duplex”** o **“half duplex”**

Full duplex consiste en una comunicación bidireccional simultánea, esto quiere decir que la aplicación local puede estar recibiendo mensajes y a la vez puede estar enviandolos.

En **half duplex** la comunicación sigue siendo bidireccional, pero no simultánea.

CAPA DE PRESENTACIÓN (Conversión de datos):

Determina el formato de la información para transferir entre las aplicaciones emisora y receptora. Codifica los datos, pudiendo comprimirlos y cifrarlos.

Formato de los datos

En esta parte el sistema realiza una codificación de los datos a enviar, a lenguaje binario. Cuando nuestro sistema está recibiendo datos, realiza la conversión inversa de binario al formato que necesite.

Cifrado de datos

Se aplica un algoritmo de encriptación a la trama original de datos (mensaje original), con el objetivo de proteger los datos que van a ser enviados y que no sea legible a simple vista (texto plano).

Compresión de los datos

Consiste en reducir el tamaño de la trama que va a ser enviada. Al realizar el formato (convertir a binario) se buscan patrones de repetición con el objetivo de simplificar al máximo la trama que va a ser enviada. Por ejemplo, si se pretende enviar la frase: "*la **conexión** produce una nueva **conexión** dentro de otra **conexión**",* podemos ver que la palabra repetida es conexión. Si sustituimos esta palabra por c1, estamos reduciendo el tamaño de la frase a enviar. Básicamente la compresión consiste en algo parecido: "*la **c1** produce una nueva **c1** dentro de otra **c1**",*

CAPA DE APLICACIÓN (Manejo datos en los programas):

Actúa de interfaz entre el usuario y las propias aplicaciones: navegadores web, aplicaciones de transferencia de ficheros, correo electrónico, terminales de red, exploradores de archivos, etc.

El usuario de la red genera la información gracias a una aplicación, no gracias a la propia capa de aplicación. Esta capa maneja los datos generados por estas aplicaciones, con el objetivo de ser enviados a la capa posterior (capa de presentación). Por ejemplo cuando escribimos un correo electrónico y pulsamos el botón de enviar, se genera un evento el cual indica que se debe realizar una operación. Este evento pertenece a parte de esta capa de aplicación, el cual indica que se debe enviar un texto, archivos, etc, a una dirección o direcciones.

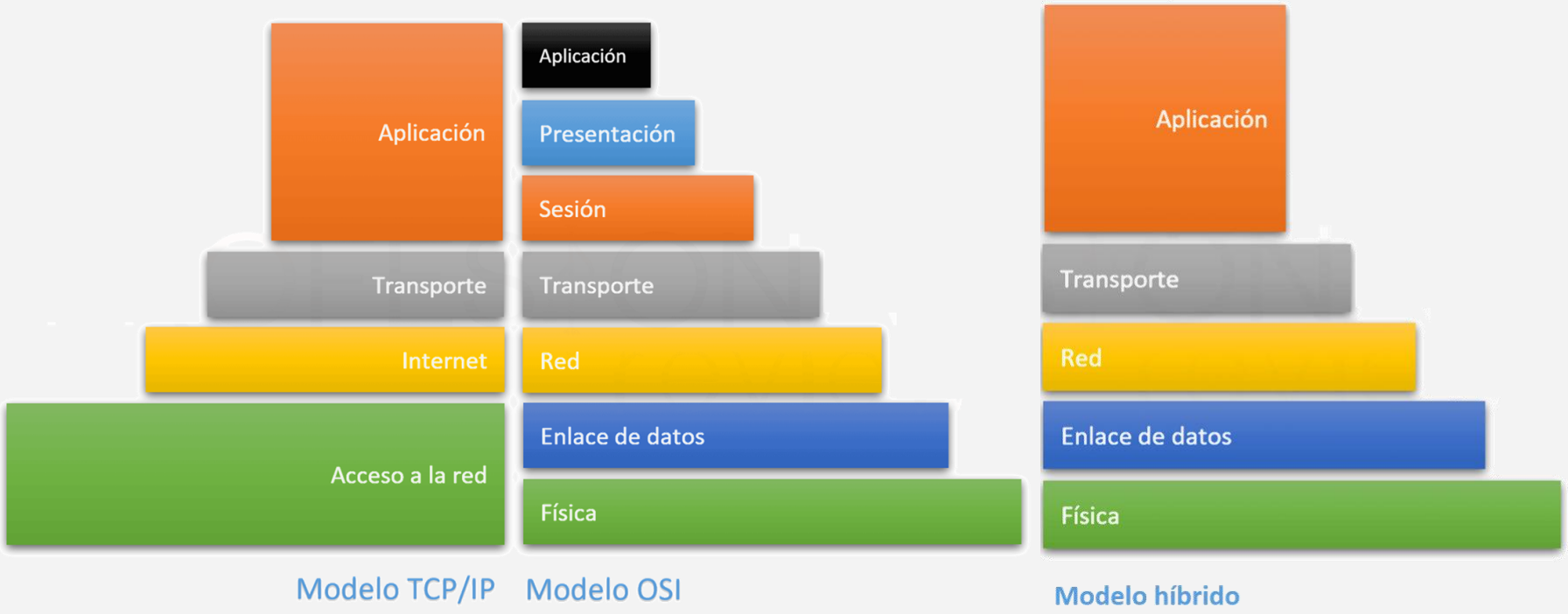
La capa de aplicación define los servicios que las aplicaciones necesitan para comunicarse, los cuales son los ya conocidos protocolos, es decir, el sistema de reglas bien definidas que indican el comportamiento de las comunicaciones.

Los protocolos más usados en esta capa son: HTTP, HTTPS, DHCP, DNS, SMTP, FTP, POP3, entre otros. Dependiendo del servicio que queremos utilizar deberemos usar un protocolo u otro. Por ejemplo, para navegar por una página web deberemos usar los protocolos web HTTP o HTTPS, para enviar un correo deberemos usar un protocolo POP/IMAP.

MODELO TCP/IP

El modelo TCP/IP es una variante del modelo OSI en el cual se agrupan en sólo 4 capas todas las etapas de comunicación. La parte de acceso a la red por tanto será la capa física junto con la capa de enlace, y la capa de aplicación engloba tanto la capa de sesión, presentación y aplicación. Esto quiere decir que realmente no se elimina la funcionalidad aquí estudiada, sino que se toma como parte de las capas.

Actualmente a lo que se tiende es a usar un modelo híbrido de los dos modelos.



EJEMPLO DE UNA CONEXIÓN USANDO EL MODELO HÍBRIDO OSI - TCP/IP

En este ejemplo mostraremos que sucede a nivel de trama de datos, cuando queremos compartir un archivo mediante un servidor FTP.

Disponemos de un archivo que por ejemplo sus datos ocupan 100 bytes. Situamos el archivo en nuestro explorador o nuestra aplicación para compartir por FTP. Aquí las acciones son llevadas a cabo en la capa superior (aplicación), y es el usuario quien inicia el evento.

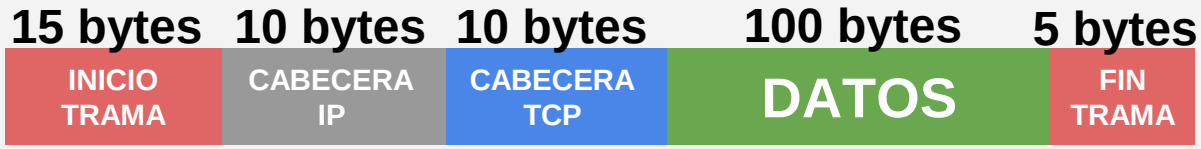
Lo siguiente es enviar este paquete de datos a la capa de transporte. La capa 6 presentación, y la capa 5 sesión están englobadas en esta capa 7 de aplicación para este modelo. En esta trama de datos se escribe una cabecera, en este caso TCP, especificando la dirección y puerto de destino entre otros datos. Como podemos ver esta cabecera ocupa un determinado espacio de datos.

EJEMPLO DE UNA CONEXIÓN USANDO EL MODELO HÍBRIDO OSI - TCP/IP

Desde la capa de transporte pasaría a la capa de red. En esta capa se añade la propia cabecera correspondiente a la capa de red, en este caso corresponde con la cabecera IP.

CAPA RED

El siguiente paso es enviar desde la capa de red a la capa de enlace donde se vuelve a poner una cabecera que indicaría el inicio de la trama, y otra parte de código también es añadida pero al final de la trama



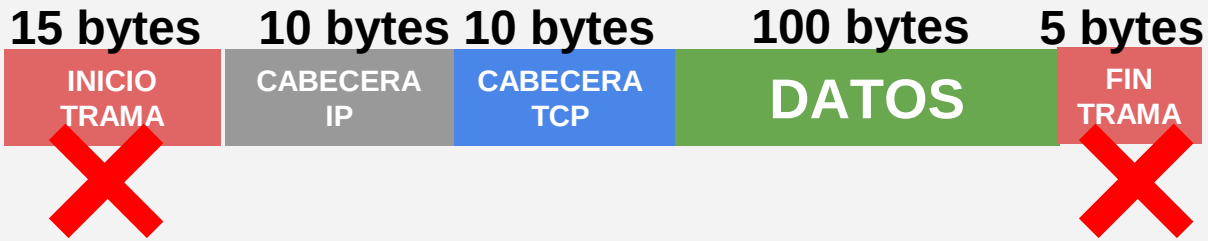
CAPA ENLACE

Como podemos ver, principalmente lo que se hace es ir añadiendo cabeceras o parte de código en la trama que queremos enviar. Un aspecto importante a tener en cuenta es ver cómo hemos pasado de 100 bytes del paquete inicial, a 140 bytes del paquete final (40% más de datos)

EJEMPLO DE UNA CONEXIÓN USANDO EL MODELO HÍBRIDO OSI - TCP/IP

Desde la capa de enlace debemos pasar a la capa física, momento en el cual nuestro paquete viajará hasta su lugar de destino por distintos caminos, entre ellos pueden ser: por cable ethernet, wifi, 5G, fibra, etc…, o combinaciones de ellas.

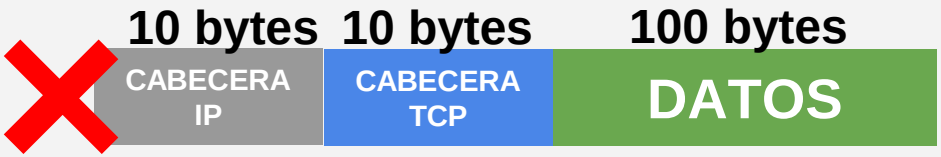
Desde la capa física del equipo emisor llegaría a la capa física del equipo receptor, y de allí a la capa de enlace del receptor que recibirá la trama. Lo que hace es quitar la parte de inicio y fin de la trama, y verifica si los datos recibidos han llegado bien.



Una vez se han verificado los datos, se pasa la trama a la capa de red.

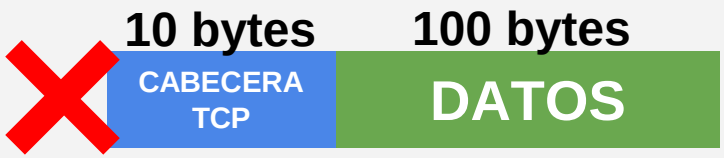
EJEMPLO DE UNA CONEXIÓN USANDO EL MODELO HÍBRIDO OSI - TCP/IP

Desde la capa de enlace debemos pasar a la capa física, donde de nuevo se elimina la cabecera IP quedándose con el resto de la trama.



CAPA DE RED

La capa de transporte al igual que el resto de capas, coge los datos que le interesa y elimina la cabecera en este caso la TCP para ser enviada a la capa de aplicación.



CAPA DE TRANSPORTE

En la capa de aplicación llegaron los 100 bytes iniciales enviados.

CAPA DE APLICACIÓN

EJEMPLO DE UNA CONEXIÓN USANDO EL MODELO HÍBRIDO OSI - TCP/IP

En este ejemplo hemos visto de forma muy resumida, cuál es el camino que sigue una trama de datos desde un equipo emisor a un equipo receptor, y como en cada una de las capas se añaden diferentes fragmentos de datos que son necesarios para poder identificar a qué capa corresponde en cada paso, **y qué protocolos se estarían empleando. Esto es fundamental para poder establecer una correcta comunicación.**