Redes de Ordenadores e Internet

¿Qué es Internet?

Es una red de comunicación global que interconecta millones de redes.

- Host, routers y enlaces de comunicación.
- Protocolos TCP/IP Es una infraestructura que proporciona servicios a las aplicaciones distribuidas.
- E-mail, Web, aplicaciones P2P, juegos, VolP, streaming de vídeo, ...
- Internet proporciona dos tipos de servicios:
    - Fiable y orientado a conexión,
    - No fiable y no orientado a conexión.

¿Qué es una red?

Una red de ordenadores es una red de comunicación digital que permite a sus nodos compartir recursos y comunicarse.

- Nodo: host (PC, teléfono, servidores, ...) y hardware de red (routers, switchs, ...) Tipos de redes, cableadas o inalámbricas, según el canal de comunicación:
- Broadcast: canal de comunicación compartido. Posibilidad de múltiples destinatarios (broadcast o multicast). Redes pequeñas en general.
- Punto a punto: canales de comunicación dedicados para la comunicación entre dos máquinas.

Tipos de redes seguún su longitud

Redes de Área Local, LAN

- Medio compartido, 10 Mbps, 100 Mbps, 1 Gbps
- Incluyen las MAN (Metropolitan Area Network) Redes de Área Extendida, WAN
- Compuestas por circuitos de telecomunicación compartidos.

Tipos de tecnologías de red

Comunicación de circuitos: cuando dos nodos se quieren comunicar se establece una conexión terminal a terminal.

- Los recursos (buffers, ancho de banda, ...) necesarios se reservan a lo largo del recorrido.
- La reserva se mantiene durante la sesión. Computación de paquetes
- No hay reserva de recursos
- Los mensajes de la sesión utilizan los recursos bajo demanda, es decir, pueden tener que esperar para poder utilizar los recursos.

Redes de conmutación de paquetes

Los mensajes originales se dividen en paquetes para su transmisión. Estos paquetes se envían a través de enlaces y routers dentro de la red. Los routers utilizan una técnica llamada almacenamiento y reenvío, lo que significa que cada router debe recibir por completo un paquete antes de poder transmitir el primer bit hacia el

siguiente destino. Este proceso introduce un retardo conocido como retardo de almacenamiento y reenvío. Cada router dispone de un búfer en sus enlaces de salida, donde se almacenan temporalmente los paquetes que esperan ser enviados. Si el enlace está ocupado transmitiendo otro paquete, los nuevos paquetes deben esperar en la cola, lo que provoca un retardo adicional conocido como retardo de cola. En caso de que la cola esté llena, el router no puede almacenar más paquetes y se produce una pérdida de paquetes, lo que obliga a descartar alguno, generalmente el último en llegar.

Redes de datagramas: el envío de paquetes se realiza en base a la dirección de destino.

- No se mantiene información sobre el estado de las conexiones en los routers. **Redes de Circuito Virtual (CV):** el envío de paquetes se realiza en base al número de circuito virtual.
- Los conmutadores mantienen información del estado de las comunicaciones entrantes: interfaz de entrada - etiqueta de entrada - interfaz de salida - etiqueta de salida.

Tipos de retardo en las redes de conmutación de paquete

Retardo de procesamiento: tiempo requerido por el router para examinar la cabecera y determinar hacia donde seguir el paquete. **Retardo de cola**: tiempo de espera para ser transmitido en le buffer de salida. **Retardo de transmisión**: timepo para transmitir todos los bits del paquete al enlace. **Retardo de propagación**: tiempo necesario para propagarse desde el inicio del enlace hasta el final del enlace.

Redes de acceso y medio físico

El acceso a la red se divide en tres clases:

Acceso residencial, conecta sistemas terminales del hogar a la red a través de un ISP, Internet Service Protocol

- Banda ancha con fibra óptica.
- Sino, vía satélite, 4G, 5G, WiMax. Acceso de empresa, conecta sistemas terminales de una empresa u organización a la red.
- Se utilizan LANs para conectar el sistema terminal al router.
- Ethernet para redes cableadas y WiFi para redes inalámbricas. Acceso inalámbrico de área extensa: emplear la misma infraestructura inalámbrica de la red de telefonía móvil para enviar/recibir datos.

Medios de transmisión

Guiados

- Par trenzado: UDP, STP y FTP
- Cable coaxial
- Fibra óptica No guiados
- Canales de radio terrestres
- Canales de radio satélite

¿Qué es un protocolo?

Toda actividad en Internet que implica a dos o más entidades remotas que se cominican está gobernada por un protocolo. Un protocolo se puede ver como un proveedor de servicio -> Diferencia entre **Servicio y**

Protocolo:

- Las entidades utilizan los protocolos para implementar el servicio que ha sido solicitado por el usuario.
- **Independencia**: podía cambiarse el protocolo sin necesidad de que lo note el usuario. **Arquitectura de red**: conjunto de protocolos y capas que permiten la comunicación entre ordenadores. **Interfaz**: comunicación definida por un conjunto de primitivas y servicios que ocurre entre pares de capas adyacentes.

Arquitectura de red

Ventajas de la estructuración en nivel y protocolos:

- Un problema complejo se descompone en piezas pequeñas.
- Abstracción de los detalles de implementación.
- Compartición por multiples niveles superiores de los servicios de una capa inferior. Inconvenientes:
- Ocultación de información y principio de layering.
- Balance entre ocupación de información y rendimiento del sistema, una capa superior puede optimizar su rendimiento conociendo el funcionamiento de la capa inferior.

Modelo de referencia OSI

Un conjunto de protocolos es abierto si el diseño del protocolo es de dominio público y los cambios los gestiona una organización cuyos miembros y actividades están abiertos al público. Un sistema que implementa protocolos abiertos es un sistema abierto. International Organization for Standards, ISO, define un estándar para conectar sistemas abiertos, OSI. Ha tenido gran influencia en el diseño de pilas de protocolos

Nivel físico

Transmitir bits entre entidades conectadas físicamente. Estandarización, esquema de codificación para la representación de bits, sincronización a nivel de bit. No existe el concepto de paquete o trama.

Introducción a TCP/IP

La familia de procolos TCP/IP permite a ordenadores de todos los tamaños, de diferentes fabricantes, ejecutando sistemas operativos diferentes, comunicarse entre ellos. Es un proyecto financiado por el gobierno amerciano para investigar en redes de comunicación de paquetes, **ARPANET**.

Niveles y protocolos

Para interconectar dos o más redes, y crear una interred o internet, necesito un **router**: Hardware y software de propósito específico que permite conectar diferentes tipos de redes físicas. Implementa los niveles de red, enlace y físico. Los niveles de transporte y aplicación utilizan protocolos **extremo a extremo**. El nivel de red utiliza un protocolo **salto a salto** que se utiliza en los sistemas finales y en cada router. Hay otros tipos de dispositivos de interconexión de LANs: concentradores, puentes y conmutadores.

- Niveles físico y de enlace.

- Se basan en las direeciones del nivel de enlace, las direeciones MAC **Concentrador, hub**: repite cada trama recibida por sus puertos de entrada por el resto de puertos de salida.
- Sólo implementa nivel físico.
- La red se comporta como si fuese un único segmento LAN. **Conmutador, switch**: permite conectar distintos equipos para formar una LAN.
- Una trama de entrada es enviada, conmutada, sólo al equipo destino, usando la dirección MAC.
- Permite obtener una mayor velocidad efectiva. **Puente, bridge**: permite conectar distintos segmentos LAN. Una trama de entrada sólo es reenviada al segmento destino si es necesario.
- Puede realizar conversiones entre distintos protocolos de enlace.
- Realiza comprobación de errores.

Direcciones IP

Los dispositivos en Internet se identifican mediante **direcciones IP**. Cada **interfaz** debe tener una dirección IP, en relidad IPv4. Una dirección IP consta de 32 bits y se representa cada byte en decimal, separado por un punto. Por ejemplo 192.168.1.100

Clases de IP

Clase	Bits iniciales	Identificador	Rango de Direcciones
A	0	Id. red / Id. host	1.0.0.0 a 127.255.255.255
B	10	Id. red / Id. host	128.0.0.0 a 191.255.255.255
C	110	Id. red / Id. host	192.0.0.0 a 223.255.255.255
D	1110	Dirección de multicast	224.0.0.0 a 239.255.255.255
E	1111	Reservado	240.0.0.0 a 247.255.255.255

Para cada una de estas IPs, indica de qué clase es, su identificador de red y su identificador de host: 10.25.100.10 86.23.187.240 172.16.25.30 145.76.231.48 192.168.10.26 193.144.57.60

IPs públicas y privadas

Direcciones **IP públicas**: identifican unívocamente un dispositivo en Internet. Por ejemplo, 193.144.60.171
Direcciones **IP privadas**: exclusivamente para uso interno. Necesitamos algo que convierta las IPs privadas en públicas para comunicarse en Internet. Network Address Translation.

- Clase A: **10.0.0.0**, 1 red
- Clase B: **172.16.0.0 - 172.31.0.0**, 16 redes
- Clase C: **192.168.0.0 - 192.168.255.0**, 256 redes

Loopback

Se reserva la dirección IP tipo A 127.X.X.X para la interfaz de loopback. Normlamente será la dirección **127.0.0.1** y el nombre asociado es **localhost**. Pretende ser una interfaz a la que se envían los paquetes dirigidos a la misma máquina. Un datagrama cuyo destino sea la propia máquina no debe llegar físicamente a la red. Utilización de la interfaz de loopback:

- Todo paquete dirigido a la dirección de loopback aparece directamente como una entrada en la capa de red.
- Todo datagrama enviado a una dirección IP de la máquina se envía a la interfaz de loopback.
- Los datagramas de broadcast y multicast se copian a la interfaz de loopback y se envían a la red.

Direcciones IP

En IPv4 se definen tres tipos de direcciones:

- **unicast**: una dirección IP, una única máquina
- **broadcast**: una dirección IP, todas las máquinas en una red.
- **multicast**: una dirección IP, un grupo de máquinas denominado grupo multicast. Broadcasting y multicasting realizan una comunicación desde una máquina a un conjunto. Sólo es válido con UDP. Direcciones de broadcast: 255.255.255.255 e <Id. red> .255
- El paquete es recibido por todas las máquinas interesadas o no en el paquete. No se puede descartar hasta que la capa de transporte.
- Produce una fuerte sobrecarga cuando el broadcast es muy acusado. Suele estar filtrado en la mayoría de las máquinas. Este problema lo soluciona el multicast ya que para recibir estos paquetes la máquina tiene que estar suscrita a un grupo multicast. Si no está suscrito, los paquetes son directamente descartados por la interfaz de red. Algunas direcciones multicast (netstat -gn. Suele estar filtrado en la mayoría de las máquinas. Este problema lo soluciona el multicast ya que para recibir estos paquetes la máquina tiene que estar suscrita a un grupo multicast. Si no está suscrito, los paquetes son directamente descartados por la interfaz de red. Algunas direcciones multicast (netstat -gn):
- 224.0.0.1 = todas las máquinas de la red que soportan multicast.
- 224.0.0.2 = todos los routers de la red.

DNS

Nosotros usamos nombres para identificar las máquinas, pero TCP/IP usan direccipnes IP. **Domain Name System, DNS**, es el sistema que realiza la correspondencia. Importancia del DNS:

- Base de datos distribuida que almacena información sobre nombres de máquinas y direcciones IP.
- También proporciona de los servidores de correo electronico.
- Cada organización gestiona su propia base de datos, su propio servidor DNS.
- Los clientes consultan a su servidor DNS cada vez que necesitan averiguar una dirección IP.

Números de puerto

Una dirección IP identifica un ordenador pero, de todas las aplicaciones que hay en un ordenador, ¿Cómo sé con cuál me tengo que comunicar? Con los **números de puerto**. Un número de puerto es un número de 16 bits, 1 - 65535. TCP y UDP tienen puertos independientes. Los **servidores** usan puertos fijos y conocidos, **well known ports**, del 1 al 1023. Los **clientes** usan puertos **efímeros**: para cada servicio usan un puerto libre cualquiera y al finalizar el servicio se deja libre.

Introducción

Dos procesos en dos sistemas finales distintos se comunican intercambiando mensajes a traves de una red de computadores. Modelo **cliente-servido**

- Cliente envía peticiones al servidor.
- Servidor recibe las peticiones, las procesa y envía la respuesta. Modelo **Peer to peer**: los dos extremos realizan un servicio y solicitan servicios. Protocolo de nivel de aplicación:
- Definen el formato y el orden de intercambio de los mensajes.
- Acciones en la transmisión o recepción de mensajes.

Web

Es una aplciación/servicio más de Internet, combina cuatro ideas que no eran nuevas:

- Hipertexto: formato de la información que permite moverse de una parte a otra de un documento o entre documentos mediantes conexiones internas entre estos documentos, hiperenlaces o enlaces.
- Identificadores de recursos: permiten localizar un recurso en la red, URL o Uniform Resource Locator, URI o Uniform Resource Identifier)
- Modelo cliente-servidor
- Lenguaje de marcas: caracteres o códigos embebidos en texto que indican estructura, semántica o recomendaciones para su presentación. Componentes:
- **Página Web**: archivo HTML base + objetos
- **Navegador**: agente de usuario para la Web.
- **Servidor Web**: almacena objetos Web direccionables a través de una URL.
- **Protocolo HTTP**: permite comunicarse al servidor y al navegador.

URI

Identificador que permite acceder a un recurso web, por ejemplo, una página web. Estructura:
http://www.udc.es/lista.html?urlmenu=servizos/#Final

- **Esquema**: especifica el protocolo utilizado para acceder al recurso. Por ejemplo, http, ftp, https, ...
- **Parte jerárquica**:
    - Autoridad: nombre del servidor. Puede incluir el número de puerto o información del control del acceso (www.udc.es)
    - Ruta: para acceder al recurso. Similar a los directorio. (/lista.html)
- **Fragmento** (opcional): identifica una subdirección dentro de un recurso (#Final)
- **Consulta** (opcional): información adicional, normalmente variables y sus valores. Por ejemplo, para enviar los campos de un formulario. (?urlmenu=/servizos)

URL vs. URI

Normalmente denominamos URLs a todas las direcciones de recursos en Internet, aunque el URI es más completo que la URL. URL = URI - Fragmento.

HTTP

El Protocolo de Transferencia de Hipertexto, conocido como HTTP (por sus siglas en inglés, HyperText Transfer Protocol), es el protocolo utilizado para la comunicación entre clientes como los navegadores web y

servidores en la World Wide Web. Ha sido especificado en diferentes documentos RFC a lo largo del tiempo, entre ellos el RFC 1945 para HTTP/1.0, el RFC 2616 para HTTP/1.1, el RFC 7540 para HTTP/2 y el RFC 9114 para HTTP/3. A lo largo de sus versiones, HTTP ha mantenido una compatibilidad general hacia atrás, aunque HTTP/2 introduce cambios en la forma de transmisión que no son compatibles con versiones anteriores. HTTP utiliza el protocolo TCP como medio de transporte, lo que garantiza una conexión orientada y fiable, asegurando que cada mensaje HTTP enviado por el cliente o el servidor llegue a su destino sin modificaciones. Su funcionamiento se basa en un modelo de solicitud-respuesta en el que el cliente envía una petición HTTP al servidor, y este responde con una respuesta HTTP que puede incluir el recurso solicitado. Es importante señalar que HTTP es un protocolo sin estado, lo que significa que el servidor no conserva información sobre las solicitudes previas del cliente, y cada nueva petición se procesa de manera independiente.

Conexiones no persistentes

HTTP/1.0 usa conexiones no persistentes. Pasos en la petición de una URL

1. El cliente HTTP inicia la conexión TCP con el servidor www.fic.udc.es en el puerto 80.
2. El cliente HTTP envía al servidor el mensaje de petición solicitando el objeto/presentación.
3. El servidor HTTP recibe la petición, busca el objeto, lo encapsula en el mensaje HTTP de respuesta y lo envía.
4. El servidor finaliza la conexión TCP.
5. El cliente HTTP recibe la respuesta y finaliza la conexión TCP.
6. El cliente extrae el archivo del mensaje de repuesta, examina el archivo HTML y encuentra referencias a otros objetos HTML.
7. Para cada objeto, volver al paso 1. Dependiendo del navegador, las nuevas conexioens podrían ser en paralelo. Inconevenientes:

- Se necesita una conexión (buffers, variables, timeouts, ...) para cada objeto solicitado.
- Retardo de dos veces el RTT (Round-Trip Time): establecimiento de conexión + pretición y recepción del objeto. Por defecto, en HTTP/1.1 el servidor HTTP deja abierta la conexión TCP, esperando nuevas peticioens/respuestas.
- Sin pipeline: el cliente sólo envía una nueva petición cunado ha recibido la respuesta previa.
- Con pipeline: el cliente realiza una petición tan pronto encuentra una referencia a un objeto.

HTTP/2 y 3

HTTP/2 - RFC 7540 Se basa en el protocolo SPDY de Google, no cambia el protocolo: métodos, códigos de estado, ... son los mismos. Cambia la manera en la que se envían datos. Mejoras:

- Multiplexación total sobre una conexión TCP: descarga de objetos web asíncronamente. Soluciona el problema head-of-line (HOD) en HTTP/1.1
- Protocolo en formato binario y compresión de cabeceras.
- Server Push: el servidor puede enviar objetos no solicitados por el cliente para almacenar en caché. HTTP/3 - RFC 9114 Utiliza el protocolo QUIC de Google (Quick UDP Internet Connections), un protoclo de código abierto basado en UDP. Mejoras: mantiene multiplexación total, mejor latencia, control de flujo por stream e incluye Transport Layer Security, TLS, 1.3.

Petición HTTP

Línea de petición + línea en blanco: obligatorio Línea de petición: Método URL HTTP/Versión Método:

- GET: utilizado cuando el navegador solicita un objeto.
- HEAD: el servidor responde con un mensaje HTTP, pero sin incluir el objeto solicitado, sólo la cabecera.
- POST: incluye datos en el cuerpo de entidad, frente al GET que los codifica en la URL.
- PUT: permite a un usuario cargar un objeto en la ruta especificada.
- DELETE: permite borrar un objeto de un servidor Web. URL: objeto que se hace referencia Versión Host: especifica el host en el que reside el objeto User-agent: especifica el tipo de navegador que está haciendo la petición POST: utilizado comúnmente cuando un usuario rellena un formulario, el cuerpo de entidad contiene los datos introducidos por el usuario. GET: tambiñen soporta el envío de datos introducidos por el usuario. Se envían codificados en la URL real.

Respuesta HTTP

Línea de estado, compuesta por versión, código de estado + Frase. Códigos de estado agrupados en 5 tipos:

- Informativo: 1xx, por ejemplo, 100 Continue
- Éxito: 2xx, por ejemplo 200 OK
- Redirecciones: 3xx, por ejemplo, 301 Moved Permanently.
- Error del cliente: 4xx, por ejemplo, 404 Not Found.
- Error del servidor: 5xx, por ejemplo 500 Internal Server Error. Date: fecha y hora en la que se creó y envió la respuesta HTTP. Server: especifica el número de bytes del objeto enviado. Last-Modified: indica la fecha y hora en que el objeto fue creado o modificado por última vez. Content-Lenght: indica el número de bytes del objeto enviado. Content-Type: indica el tipo de objeto incluido en el cuerpo de entidad.
- La extensión del archivo no especifica formalmente el tipo de objeto.

Respuesta HTTP: Content-Type

Usa los tipos Multipurpose Internet Mail Extensions, MIME, para definir el tipo de contenido. **MIME**: estándar que indica el tipo de contenido. Gestionado por la IANA. Estructura básica: tipo/subtipo, por ejemplo text/html, image/gif ... Discreto: un único documento de un único tipo

- Aplicación: application/pdf, application/zip, application/octetstream
- audio: audio/mpeg
- image: image/jpeg, image/png, image/gif
- text: text/plain, text/html, text/csv
- video: video/mp4 **Multipart**: encapsula múltiples archivos, posiblemente de distintos tipos, en una única transacción

HTTP: Cookies

HTTP no tiene memoria, se utiliza un mecanismo que permite a un servidor web guardar información en el navegador, conocido como cookies.

HTTP: GET condicional

La utilización de una caché reduce los retardos de recuperación de objetos y reduce el tráfico que circula por la red. Problema: la copia de un objeto en caché puede ser obsoleta. Solución: GET + If-Modificed-Since

- Solo devuelve el objeto si ha sido modificado después de la fecha indicada. Solicitar un objeto por primera vez: `GET /images/udc.git HTTP/1.1 User-agent: Mozilla/4.4` Recibir la respuesta del servidor: `HTTP/1.1 200 OK Date: Sat, 1 Jan 2000 12:00:15 GMT Server: Apache/1.3.0 (Unix) Last-Modified: Fri, 24 Dic 1999 13:03:32 GMT Content-Type: image/gif (datos)...` Pasado un tiempo, se vuelve a solicitar el mismo objeto: `GET /images/udc.git HTTP/1.1 User-agent: Mozilla/4.4 If-modificed-since: Fri, 24 Dic 1999 13:03:32 GMT` Si no se ha modificado, el servidor no envía el objeto de nuevo `HTTP/1.1 304 Not Modified Date: Wed, 5 Jan 2000 20:30:43 GMT Server: Apache/1.3.0 (Unix)`

Correo electrónico

Inventado por Ray Tomlinson en 1971, el correo electrónico es un medio asíncrono de comunicación, compuesto por: lectores de correo o agentes de usuario, servidores de correo y Simple Mail Transfer Protocol, SMTP

SMTP

Definido en el RFC 5321. Permite el intercambio de mensajes entre servidores de correo, el remitente actúa como cliente y el destinatario actúa como servidor. El cliente SMTP establece una conexión TCP con el puerto 25 del servidor SMTP, si el servidor está fuera de servicio se intentará más tarde. Se realiza la sincronización entre emisor y receptor, se indica la dirección de correo electrónico del remitente. El cliente envía el mensaje, este proceso se repite si hay más mensajes para el mismo servidor; y después se cierra la conexión TCP.

SMTP utiliza mensajes en formato ASCII, si el mensaje tiene caracteres no ASCII o binario, tiene que ser codificado. Es un protocolo de oferta, el cliente nvía al servidor, frente a HTTP que es un protocolo de demanda.

Proceso de envío de un mensaje de correo electrónico

1. El usuario, mediante su lector de correo, crea el mensaje, por ejemplo john.doe@udc.es
2. El lector de correo envía el mensaje al servidor de correo del emisor.
3. El servidor de correo, actuando como cliente SMTP, se conecta al servidor de correo del destinario.
4. El cliente SMTP envía el mensaje.
5. El servidor SMTP recibe el mensaje y lo almacena en el buzón del destinatario.
6. El destinatario utiliza su lector de correo para obtener el mensaje.

Formato correo electrónico

Un mensaje de correo electrónico consta de dos partes, cabecera y cuerpo, separadas por una línea en blanco.

- Cabecera: información sobre el corre.
- Cuerpo: el propio correo electrónico. Algunos campos de la cabecera son:
- **From**: sólo una por mensaje

- **To**: una o más por mensaje.
- **Cc** y **Bcc**
- **Subject**: tema del mensaje
- **Date**: fecha y hora en que el mensaje fue enviado.
- **Message-Id**: identificador de cada mensaje, insertado por el ordenador remitente.
- **Recieved**: información sobre el envío del mensaje, como las máuinas por las que pasño el mensaje.
- **Reply-To**: dirección a la que se debe responder.

MIME

Multipurpose Internet Mail Extensions Permite enviar contenidos distintos de texto ASCII en mensajes de correo electrónico, por ejemplo, idiomas con acentos, no lations, sin alfabetos o contenido binario. Sólo afecta a los agentes de usuario, ya que para SMTP es transparente. Campos MIME:

- MIME-Version
- Content-Description: cadena de texto que describe el contenido. Es necesaria para que el destinatario decida si descodificar y leer el mensaje
- Content-Id
- Content-Transfer-Encoding: indica la manera en que está codificado el cuerpo del mensaje.

Protocolos de acceso al correo

Post Office Procolv3, POP3 Protocolo de acceso al correo muy simple. Modo de operación en tres fases:

- Autorización: login y password
- Transacción: recuperar los mensajes, marcar para borrado y estadísticas de correo
- Actualización: cuando finaliza la sesión, el servidor de correo borra los mensajes marcados. Dos configuraciones del cliente POP3:
- Descargar y borrar
- Descargar y guardar Internet Mail Access Protocol, IMAP, permite crear y gestionar buzones remotos en el servidor de correo; IMAP asocia cada mensaje con un buzón, inicialmente al INBOX. Proporciona comandos para crear buzones, mover mensajes , buscar entre sesiones... Dispone de comandos para recuperar componentes de los mensajes.

Tema 4: Protocolos de nivel de aplicación II

DNS: Introducción

Domain Name System Nosotros utilizamos nombres para las maquinas, pero TCP/IP se comunican utilizando direcciones IP. DNS es el sistema que se encarga de hacer la correspondencia entre nombres de máquinas y direcciones IP, también proporciona información de los servidores de correo. Modelo cliente-servidor: Se implementa sobre UDP, aunque también puede utilizar TCP. Antes del DNS, se usaba un fichero de hosts.

Cliente DNS

DNS también es el protocolo que permite a los clientes y servidores comunicarse. **Cliente DNS**: cada máquina tiene un cliente DNS, cada vez que cualquier aplicación necesita averiguar una dirección IP, le pasa

la pregunta al cliente DNS, el cliente DNS le envía la conuslta a su servidor DNS, cuando obtiene la respuesta, se la pasa a la aplicación.

Servidor DNS

Cada red tiene un servidor DNS. El servidor recibe consultas DNS de clientes, averigua la dirección IP y la envía a los clientes. ¿Cómo averigua mi servidor DNS una dirección IP? El DNS es una base de datos distribuida, no hay un servidor que conozca todos los nombres y sus IPS, Hay múltiples servidores DNS organizados jerárquicamente.

Espacio de nombres DNS

- Estructura de nombres jerárquica en forma de arbol: Top-Level Domains, TLDs. .com, .net, .es, .uk Second-Level Domains, SLDs .ibm, .google, .udc, .usc Third-Level Domains .tic, .fic
- Nombre de dominio: www.fic.udc.es
    - No se distinguen mayúsculas y minusculas.
- FQDNs (fully qualified domain names): nombre de dominio completo, formalmente acabado en ".", si está imcompleto se "rellena" con nuestro dominio.

Servidores de nombres

Hay servidores DNS en cada nivel de la jerarquía de los nombres de dominio:

- **Distribuir** la carga entre los servidores de nombres.
- **Delegación** de la administración de los servidores de nombres. Servidores **raíz**
- Existen 13 servidores raíz (A-M), replicados por seguridad y fiabilidad.
- Conocen a todos los TLDs y delegan en ellos. Servidores TLD
- Cada dominio de primer nivel tiene su servidor TLD asociado.
- Delegan en servidores de 2º nivel la gestión de sub-dominios. Servidores DNS inferiores:
- Conocen a todos los equipos de su dominio.
- Conocen a los servidores DNS raíz.
- Ante una consulta, si no conoce una IP, le pregunta a un servidor raíz.

Funcionamiento DNS

Consultas recursivas: El servidor DNS hará todo el trabajo necesario para devolver la respuesta completa a la petición. Puede implicar múltiples transacciones del servidor con otros servidores DNS. No es obligatorio que los servidores DNS soporten este tipo de consultas. Consultas iterativas: Si el servidor DNS tiene la respuesta, entonces la devulve. En caso contrario, devolverá la información útil, pero no hará peticiones adicionales a otros servidores DNS. Los servidores raíz y TLD son no recursivos.

Caché DNS

Para reducir los mensajes DNS se utilizan cachés, cada par dirección IP - nombre que se resuelve se almacena en la caché, esta tiene un tiempo de vida, TTL, de varios días. También se almacenan las peticiones incorrectas. **Respuesta autoritativa**: responde directamnte el servidor DNS que "conoce" la información.

Servidor DNS de Forwarding

Servidores DNS de Forwarding: No es responsable de ninguna zona, no almacena información en disco. Sólo reenvía las consultas a otros servidores DNS. Almacena las respuestas en caché, implica una respuesta rápida para consultas frecuentes. Un router inalámbrico, lo normal es que incorpore un servidor DNS de forwarding:

- Reenvía las consultas al servidor DNS de mi ISP.
- Las consultas en caché se resulven en mi LAN, evito accesios a la red de ISP.

P2P

Los protocolos anteriores se basaban en el modelo cliente-servidor: el servidor proporciona un servicio y el cliente cosnumeese servicio. El modelo P2P, peer to peer, está compuesto por pares (peers) que realizan ambas funciones: consumir y proporcionar un servicio. Se basa en equipos de usuarios

- No son propiedad de un proveedor de servicio.
- Conectados intermitentemente
- Proporcionan acceso a una parte de sus recursos. Ventajas:
- Compartición de recursos.
- Gran tolerancia a fallos. Inconvenientes:
- Seguridad: acceso a los recursos de un equipo, aumento de las medidas de seguridad en los últimos años.
- Gran uso de ancho de banda, a veces restringidos por los ISPs

UDP

User datagram protocol. UDP es un protocolo de nivel de transporte, orientado a datagramas, y simple. Cada bloque de datos generado por la capa de aplicación produce un único datagrama UDP. UDP no garantiza que el datagrama alcance su destino. UDP multiplexa los datos de las aplicaciones y efectúa una comporbación de errores, pero no realiza control de flujo, control de congestión, retransmisión de datos perdidos, conexiones ni desconexiones. Se utiliza principalmente en los siguientes casos:

- Cuando el medio de transmisión es altamente fiable y sin congestión (LANs).
- Cuando la aplicación es en tiempo real y no se pueden esperar los ACKs.
- Cuando los mensajes se producen regularmente y no importa si se pierde alguno.
- Si se envía tráfico de broadcast o multicast.

Cabecera UDP

16 primeros bits: nº de puerto de origen (longitud UDP) 16 siguientes bits: nº de puerto destino, checksum UDP Los numeros de puerto identifican los procesos emisor y receptor, los números de puerto UDP son independientes de los de TCP. Longitud UDP = longitud de la cabecera UDP + longitud de datos. El valor mínimo es de 8 bytes. Es redundante la infromación de la cabecera IP. Checksum: se calcula sobre la cabecera UDP y los datos UDP. Antes era opcional, ahora es obligatorio por defecto.

TCP

TCP o Transmission Control Protocol es un protocolo que proporciona un servicio de envío de datagramas fiable y orientado a conexión. Al ser orientado a conexión, dos aplicaciones, generalmente en un modelo cliente-servidor, deben establecer una conexión TCP entre ellas antes de iniciar el intercambio de datos. Su fiabilidad garantiza que los datos se reciben correctamente y en el orden adecuado. Cabe destacar que TCP no admite broadcasting ni multicasting. Los paquetes en TCP se denominan segmentos y la comunicación que permite es full-duplex, es decir, bidireccional y simultánea. Entre sus principales funciones se encuentran el establecimiento y la terminación de conexiones, la gestión de buffers y el control de flujo eficiente, la multiplexación del nivel de aplicación mediante puertos, el intercambio de datos con las aplicaciones, el control de errores mediante la retransmisión de segmentos perdidos o erróneos y la eliminación de duplicados, así como el control de congestión en la red. Para implementar la fiabilidad TCP implementa lo siguiente:

- Divide datos de la aplicación en segmentos con la longitud más adecuada para la aplicación.
- Asocia un temporizador con los segmentos que envía. Si no recibe el ACK del destino a tiempo remite el segmento.
- Mantiene un checksum en la cabecera TCP para comprobar el segmento recibido. No se envía ACK si el segmento es incorrecto.
- El receptor TCP reordena los segmentos, si es necesario, para pasarlos ordenados a la aplicación (los segmentos se pueden desordenar un la transmisión).
- Descarta segmentos que se hayan podido duplicar.
- Proporciona control de flujo: un receptor TCP sólo deja transmitir al otro extremo segmentos que pueden almacenarse en su bugger de entrada, sin producirse desbordamientos.

Cabecera TCP

Nº de puerto origen y destino + dir. IP origen y destino de cabecera IP identifican unívocamente la conexión TCP. **Número de secuencia**: identifica ek nº de byte en el flujo de bytes TCP entre el emisor y el receptor que supone el primer byte de la sección de datos:

- Cuando se llega a 2^{32}-1 se comienza de nuevo por 0.
- Cuando se establece una conexión, se pone a 1 el flag SYN, y la máquina selecciona un Initial Sequence Number, ISN, para esa conexión. **Número de ACK**: indica el siguiente número de secuencia que el emisor del ACK espera recibir.
- Es el nº de secuencia + 1 del último byte recibido satisfactoriamente.
- TCP proporciona una comunicación "full-duplex" al nivel de aplicación, cada extremo mantiene su nº de secuencia.
- No existen ACK's negativos, pero sí selectivos, SACK. **Longitud de cabecera**: tamaño de la cabecera incluyendo opciones.
- Especifica el número de palabras de 32 bits.
- Valor máximo 60 bytes. **Flags**
- Congestion Window Reduced, CWR: el emisor reduce su velocidad de transmisión.
- ECE: el emisor confirma la recepción de un paquete con el flag Explicit Congestion Notification, ECN, activado.
- URG: puntero de urgencia válido.
- ACK: número de ACK válido, siempre activado una vez establecida la conexión.
- PSH: el receptor debe pasar estos datos a la aplicación lo antes posible, es una implementación poco fiable y poco usada.

- RST: reinicia la conexión.
- SYN: sincronizar números de secuencia para iniciar una conexión.
- FIN: el emixor finaliza el envío de datos. **Tamaño de ventana**: indica el nº de bytes, comenzando por el valor del campo de nº de acknowledge, que el receptor puede aceptar.
- Utilizado para establecer control de flujo.
- Máximo de 65.535, pero existe una opción de factor de escala para incrementar ese valor. **Checksum**: sobre todo el segmento TCP
- Es obligatorio, debe calcularlo el emisor y comprobarlo el receptor.
- El cálculo es similar al checksum de UDP. **Puntero de urgencia**: válido si el flag URG es 1.
- Indica un offset a añadir al nº de secuencia.
- Se utiliza para transmitir datos urgentes. **Opciones**: la más común es la opción de máximo tamaño de segmento (Maximum Segment Size). **Datos**: información enviada (opcional).

TCP: Establecimiento

Las conexiones las inicia, normalmente, el cliente. El servidor hace una apertura pasiva. Protocolo de establecimiento de conexión (**Three-Way Handshake**):

- El emisor envía un segmento SYN indicando el nº de secuencia inicial.
- El servidor responde con su propio segmento SYN que contiene el nº de secuencia inicial del servidor. También confirma (ACK) el SYN del cliente + 1 (los mensajes SYN consumen un nº de secuencia).
- El cliente confirma el SYN del servidor con un nº de ACK igual al ISN del servidor + 1. **Número de secuencia inicial, ISN**:
- Cada extremo sececciona su ISN al establecerse la conexión.
- Se obtiene pseudoaleatoriamente.
- Objeto: evitar que segmentos "antiguos", de otra conexión igual, se confundan con los actuales.

TCP: finalización

Se intercambian 4 segmentos para cerrar una conexión.

- Una conexión TCP es full-duplex y cada dirección se cierra independientemente.
- Cada extremo envía un FIN cuando ha realizado el envío de datos. El otro extremo puede continuar enviando datos. El extremo que envía el primer FIN realiza el cierre activo, y el otro extremo el cierre pasivo, cualquiera de los dos extremos puede empezar el cierre. Protocolo de finalización de conexión:
- El cliente finaliza la aplicación, el cliente TCP envía un FIN con el número de secuencia correspondiente.
- El servidor responde con un ACK del nº de secuencia + 1.
- A continución, el servidor envía un FIN.
- El cliente confirma la recepción del FIN, con un ACK del nº de secuencia recibido + 1.

TCP: MSS

Maximum Transmission Unit: número máximo de bytes de datos que puede enviar el nivel de enlace. Maximum Segment Size: indica el número máximo de bytes de datos que le conviene recibir a cada extremo, para evitar la fragmentación IP. Cuando se establece una conexión TCP, cada extremo anuncia el MSS que espera recibir:

- La opción MSS sólo aparece en un segmento SYN.

- Si no se declara, toma un valor por defecto.
- MSS no incluye las longitudes de cabecera IP y TCP. En general es preferible un MSS grande que amortice el coste de cabeceras. Pero también interesa evitar la fragmentación. No se realiza una negociación del MSS, el tamaño de segmento será el menor de los dos.

TCP: estados

Estado **TIME_WAIT**:

- TCP espera 2 veces el tiempo máximo de vida de un paquete en la red, por si se ha perdido el último ACK.
- Variable /proc/sys/net/ipv4/tcp_fin_timeout (segundos)
- Permite a TCP reenviar el ACK en caso de que se haya perdido, el otro extremo reenviará el FIN
- Mientras la conexión está en este estado, no se puede reutilizar el par de sockets de esa conexión, cualquier segmento retrasado recibido es descartado para garantizar que no aparecen reencarnaciones de segmentos en futuras conexiones. Estado **FIN_WAIT_2**:
- Permanecerá en este estado hasta recibir el FIN del otro estremo.
- El otro extremo está en el estado CLOSE_WAIT y debe esperar a que se cierre la aplicación.
- Para evitar una espera infinita, las implementaciones establecen un tiempo de espera, tras el cual pasa directamente al estado CLOSED.

TCP: Segmentos de Reset

Un segmento de Reset cuando se activa en la cabecera TCP el flag RST. Se activa el bit de Reset en una conexión TCP cuando el paquete que ha llegado no parece, en principio, estar relacioando con la conexión a la que está referido el paquete. Las causas de generar un paquete con ese bit para una conexión TCP pueden ser varias.

- Intento de conexión a un puerto no existente.
- Respuesta ante conexiones semi-abiertas.

Intercambio de datos TCP

Protocolo ARQ de parada y espera

Protocolo de **parada y espera**: el emisor no envía datos nuevos hasta confirmar que el receptor ha recibido correctamente los datos anteriores. También denominado de bit alternante. Obtiene un rendimiento muy bajo. Solución: enviar varios paquetes sin esperar a los mensajes de confirmación, procesamiento en cadena. ¿Qé necesito?

- Aumentar el tamaño de los números de secuencia, varios paquetes podrán estar en la red simultaneamente, sin confirmar.
- El emisor necesitará un buffer para almacenar los paquetes transmitidos pero no confirmados.
- El receptor necesitará un buffer para almacenar los paquetes recibidos correctamente, pero que la capa superior aún no puede procesar. Protocolos ARQ de procesamiento en cadena:
- Retroceder N (GBN - Go-Back-N)
- Repetición Selectiva (SR - Selective Repeat)

Protocolo ARQ retroceder N

El emisor puede transmitir varios paquetes sin esperar a que estén confirmados. Se establece un máximo de N paquetes enviados sin confirmación. Se suelen representar los paquetes que se pueden enviar como una ventana que se va desplazando:

- Cada vez que se recibe un ACK nuevo, se puede enviar otro paquete.
- Protocolo de **venta deslizante** El receptor no tiene buffer, Los números de secuencia son finitos, son circulares. Sólo utiliza ACKs positivos, pero son **acumulativos**.

Protocolo ARQ de repetición selectiva

Problema de retroceder N: un error en un paquete hace que se repitan otros (muchos) paquetes recibidos correctamente. Solución: que el emisor unicamente retransmita los paquetes erróneos, repetición selectiva.

- Los ACKs son individuales.
- Se utiliza una ventana, pero con algunos paquetes confirmados.
- El receptor necesita un buffer.
- Se utiliza un temporizador para cada paquete enviado.

Intercambio de datos TCP

En TCP se consideran dos tipos de tráfico de datos:

- **Interativo**: gran número de segmentos de pequeño tamaño.
- **No Interativo**: segmentos de gran tamaño, normalmente el máximo permitido por las limitaciones de la red. Para implementar la fiabilidad, se basa en el modelo ARQ retroceder N:
- Es un protocolo de ventana deslizante.
- Los ACKs son acumulativos y positivos. Con algunos matices:
- Cuando el receptor recibe un paquete fuera de orden no lo descarta, lo almacena en el buffer y envía un ACK del último paquete correcto.
- Retransmisión rápida: si el emisor recibe tres ACKs repetidos retransmitirá sólo el paquete siguiente al número de ACK.
- El emisor mantiene un temporizador por cada grupo de paquetes enviado. Al tiempo de espera antes de retransmitir se le denomina Retransmission Tiemout, **RTO**. En TCP el RTO se calcula a partir del Round-Trip Time, RTT, que se estima continuamente durante toda una conexión TCP.
- Cuando se envía un segmento se mide el tiempo que tarda en recibirse su ACK. También existe la opción de Timestamp, TSOPT: 10 bytes
- Campo Timestamp Value, TSval, el emisor indica el valor de su reloj en el momento de la transmisión.
- Campo Timestamp Echo Replay (TSecr): el receptor copia el TSval en el segmento de respuesta. Estimación de RTT con TSOPT:
- El emisor indica el TSval en los segmentos que envía.
- Al preparar la respuesta, ACK, el receptor copia el TSval en el campo TSecr.
- El emisor, al recibir el ACK, comprueba el reloj del sistema, le resta el TSecr y tiene la estimación del RTT.

Flujo de datos interactivo

Funcionamiento del ssh:

- Envío de la tecla pulsada por el cliente
- ACK de la tecla pulsada por el cliente
- Eco de la tecla desde el servidor
- ACK del eco **ACKs retardados**:
- Objetivo: enviar el ACK + eco en un único datagrama. TCP no envía el ACK inmediatamente al recibir el dato, sino que retarda la salida del ACK esperando un tiempo para ver si hay datos para enviarlos con el propio ACK. El tiempo de espera es de 200 mseg. No en valor absoluto, sino que se utiliza un reloj que da ticks cada 200 mseg. Tanto en el cliente como en el servidor se utilizan los acks retardados. El tráfico interactivo genera gran cantidad de paquetes de tamaño muy pequeño, denominados **tinygrams**
- En las redes de área local no presenta ningún problema.
- En las WAN supone una gran sobrecarga para la red. El **algortimo de Nagle** pretende resolver este problema, y se aplica en una conexión TCP de tráfico interactivo en una red de área extensa. *"Una conexión TCP puede tener un unico segmento pequeño que no haya sido confirmado. No se pueden enviar otros segmentos hasta recibir un ACK. En cambio, si esos datos se almacenan y son enviados por TCP al llegar el ACK"*. Esto es auto-ajustable: cuanto más rápido lleguen los ACKs más rápido se enviarán los datos. El algoritmo de Nagle convierte a TCP en un protocolo de parada y espera. En algunos casos puede no interesar, por lo que se puede controlar. setTcpNoDelay(boolean) de la clase Socket En este tipo de tráfico se generan "pocos" segmentos, pero de gran tamaño. El principal problema a resolver en este tipo de tráfico es el **control de flujo**: evitar que un emisor rápido sature a un receptor. TCP utiliza una ventana deslizante: permite al emisor enviar múltiples paquetes antes de parar y esperar por el ACK, lo que da una mayor rapidez a este tipo de tráfico. Con un protocolo de ventana deslizante no es necesario confirma todos los paquetes recibidos, sino que se pueden confirmar todos los paquetes simultaneamente.

Control de flujo

El envío de mensajes TCP no es determinista, depende de múltiples factores: la carga de la red, la carga del receptor y emisor, ... El emisor no tiene por qué enviar una ventana completa de datos. El receptor no tiene que esperar a que se llene la ventana para enviar un ACK.

Temp. de persistencia

En una conexión TCP, puede producirse una situación de interbloqueo cuando el emisor deja de enviar datos porque no recibe actualizaciones de la ventana de recepción, mientras que el receptor no puede notificar la liberación de espacio en su buffer. Para evitar este problema, TCP implementa un temporizador de persistencia, cuya función es asegurar que el emisor continúe verificando si el receptor ha abierto espacio en su ventana, incluso cuando no llegan notificaciones. Una vez activado, este temporizador hace que el emisor envíe segmentos especiales llamados sondas de ventana. Estas sondas consisten en segmentos muy pequeños, generalmente de un solo byte, y su propósito es comprobar si la ventana de recepción ha cambiado. Si el receptor ha liberado espacio, responderá con una confirmación (ACK) que indica el nuevo tamaño de la ventana. Si no hay espacio disponible, puede responder indicando que la ventana sigue cerrada o simplemente ignorar la sonda. El temporizador de persistencia emplea un mecanismo de retroceso exponencial, es decir, si no se recibe respuesta, el intervalo entre las sondas se incrementa progresivamente. Este proceso continúa hasta que el receptor abre la ventana y se reanuda la transmisión de datos, o bien hasta que las aplicaciones que utilizan la conexión TCP deciden cerrarla.

Control de congestión

El control de congestión de flujo evita que el emisor llegue a saturar al receptor. Esto es correcto en una LAN, sin embargo en un entorno con routers intermedios el control de flujo no es suficiente, los routers intermedios también se pueden saturar. Los routers no tienen nivel de transporte, por eso se utilizan controles de congestión. Algoritmo para evitar la congestión: Se establece el **umbral de inicio lento**.

- Por debajo del umbral, se aplica el algortimo de incio lento: "La velocidad a la que se inyectan paquetes nuevos a la red es la velocidad a la que se reciben ACKs del otro extremo"
- Por encima, se aplica un crecimiento suavizado. ¿Qué pasa si algo va mal?

1. ¿Qué es que algo vaya mal? Cuando se pierde un paquete. Dos posibilidades para "perder" un paquete: saturación en un router o un error en el paquete. Se asume que cuando se pierde un paquete es debido a que, al menos un router, está saturado.
2. ¿Cómo sé que algo va mal? Se utilizan dos indicadores para identificar un problema de congestión:

- Ha vencido un timeout de Retransmisión (Fast Recovery).
- Se han recibido ACKs duplicados (Fast Retransmit).

3. ¿Qué hago cuando algo va mal?

- Retransmitir.
- Y reducir la velocidad de transmisión. Todo esto se integra en el **algortimo para evitar la congestión**.

Temporizador de keepalive

Es una conexión TCP sin intercambio de datos, no se produce ningún intercambio de paquetes, problema en situaciones de fallos de uno de los extremos, normalmente el cliente. Solución: **temporizador de keepalive**, mantiene la conexión activa aunque no es parte del RFC de TCP. Permite liberar recursos al otro extremo, si realmente está desconectado. Funcionamiento: después de 2 horas de inactividad, el servidor enviará una sonda keepalive (segmento de un byte, correspondiente al último byte enviado). El otro extremo puede estar en cuatro estados: ok, caído, reiniciando o no alcanzable. Se controla con /proc/sys/net/ipv4/tcp_keepalive_time, tcp_keepalive_intvl, tcp_keepalive_probes.

Internet Protocol

Cabecera IP

TCP/IP usa la ordenación de bytes "big endian" (de izquierda a derecha) Si un equipo usa el formato "little endian" debe hacerse la conversión al transmitir y al recibir. **Versión** Versión actual de IP **Longitud de cabecera**: número de palabras de 32 bits de la cabecera, incluidas las opciones si las hubiera <= 60 bytes. **Tipo de servicio**: diseñado para Quality of Service, QoS, aunque nunca fue ampliamente usado. **Servicios diferenciados** (DS): campo de 6 bits utilizado para dar soporte a QoS mediante la técnica de DS. **Explicit Congestion Notification**, ECN: indicador de congestión o futura congestión en un router (2 bits). **Longitud total**: longitud total de datagrama IP en bytes. Longitud total - cabecera = tamaño datos. Campo de 16 bits: máximo tamaño de 65535 bytes. Se precisa este campo porque algunos protocolos del nivel inferior pueden no conocer de manera precisa el tamaño del datagrama encapsulado. **Identificación**: identifica unívocamente el datagrama IP enviado por una máquina. Normalmente se incrementa en una unidad cada

vez que se envía un datagrama. **Flags y offset de fragmentación**: campos para fragmentación. **Time To Live, TTL**: establece un tiempo máximo de vida para el datagrama. Previene bucles indefinidos por problemas de enrutamiento. Establece un límite en el número de routers por los que puede pasar un datagrama. Cada vez que el datagrama pasa por un router, se decrementa en una unidad el valor de este campo. Cuando vale 0 se descarta el datagrama y se notifica al remitente con un mensaje ICMP. **Protocolo**: usado por IP para demultiplexar. Permite identificar de qué protocolo de la capa de transporte son los datos enviados. **Checksum de cabecera**: sólo para la cabecera. **Dirección IP de origen y destino**: 32 bits cada una. **Opciones**: información opcional de longitud variable. Algunas opciones son: Registro de enrutamiento, cada router marca su hora y dirección IP, máximo de 9 routers. Timestamp, se registra la ruta y además pone una marca de tiempo en cada salto, máximo de 4 routers. Lista estricta de enrutamientos: la cabecera contiene la ruta paso a paso que debe seguir el datagrama, máximo 9 routers. Lista difusa de enrutamientos: la cabecera lleva una lista de routers por los que debe pasar el datagrama, pero puede pasar además por otros (máximo 9). NoOp: la longitud ha de ser múltiplo de 32 bits. Esta opción permite añadir bytes de relleno para cumplir esta condición.

Subredes

Una subred consiste en dividir una red en partes más pequeñas.

- Nivel jerárquico intermedio entre red y host.
- Utiliza unos bits de la parte del identificador de host para la subred.
- Organización jerárquica de la red, visión externa como una sola red, aunque internamente esté dividida en subredes.

Máscara de subred

Indica cuantos bits forman parte del identificador de red y subred, y cuantos forman el identificador host. Se ponen a 1 todos los bits correspondientes al identificador de red o subred. Se ponen a 0 todos los bits correspondientes al identificador de host. Cada máquina almacena su dirección IP y su máscara de subred.

Direcciones de subred

En cada subred hay dos **direcciones reservadas**, la dirección de subred y la de broadcast en la subred. **Dirección de subred**: identifica la subred, se calcula para cada subred poniendo a 0 el identificador de host; coincide con la primera IP del rando; es quivalente a realizar una operación AND entre la dirección IP y la máscara de subred. **Dirección de broadcast en la subred**: Se calcula poniendo todo a 1 el identificador de host. Coincide con la última IP del rango. Representa a todas las máquinas de la subred.

Máscaras de subred de tamaño variable

Fixed Lenght Subnet Masks (FLSM): todas las subredes usan la misma máscara, lo cual desprecia direcciones IP. Variable Lenght Subnet Masks (VLSM): cada subred usa la máscara óptima para su número de hosts.

- Ordenar las subredes de mayor a menor nº de hosts.
- Calcular la máscara para cada subred usando FLSM

DHCP

Dynamic Host Configuration Protocol: permite asignar direcciones IP dinámicas automáticamente a los hosts (**plug-and-play**):

- Las direcciones IP se asginan durante un tiempo limitado, después es necesario renovarlas.
- También incluye otros parámetros como máscaras de subred , router por defecto y servidores DNS. Se basa en el modelo cliente-servidor.
- Cliente DHCP: cualquier máquina "nueva" en la red que se esté iniciando y necesite una configuración de red.
- Servidor DHCP: granatiza que todas las direcciones IP son únicas durante su tiempo de vida. Métodos de asignación de direcciones
- **Estática** o **manual**: se asigna una dirección IP a una máquina concreta, en base a su dirección MAC. Evita que se conectien clientes no identificados.
- **Dinámica**: se utiliza un rango de direcciones IP y cada ordenador de la red está configurado para solicitar su dirección IP al iniciarse la interfaz. Permite la reutilización de las direcciones IP y facilita la instalación de nuevas máquinas en la red.
- **Automática**: similar al modo dinámico, pero un equipo siempre obtiene la misma IP.

DHCP: Funcionamiento

Modelo cliente-servidor basado en UDP: puerto 67 para el servidor y 68 para el cliente. Mensajes DHCP: el cliente incluye un identifcador de transacción en el mensaje de descubrimiento, que deberá ser repetido en los siguientes.

- **Discovery**: mensaje difundido en la red por el cliente para descubrir el/los servidores DHCP.
- **Offer**: mensaje que contiene la dirección IO que el servidor ofrece al cliente DHCP. Incluye la dirección MAC del cliente, la IP ofertada, la máscara, el tiempo de validez y la dirección IP que el servidor ofrece al cliente DHCP.
- **Request**: el cliente seleccionará una dirección de las ofertadas. En caso de existir varios servidores, se indica el servidor del que se acepta la oferta.
- **Acknowledgement**: el servidor confirma la solicitud del cliente y le indica cualquier otra información solicitada por el cliente. El cliente no tiene dirección IP y no conoce al servidor DHCP. Los mensajes DHCP tienen como destino la dirección de **broadcast** 255.255.255.255.

NAT: Direcciones privadas

Cuando contratamos una banda ancha, mi ISP me proporciona **una dirección IP**, pero ¿y si quiero conectar más de un dispositivo a internet? Direcciones IP públicas: identifican un dispositivo en Internet. **Direcciones IP privadas**: exclusivamente para uso interno.

- Los dispositivos de la red privada se pueden comunicar entre si con esas direcciones.
- Pero no se pueden comunicar con el exterior, Internet. Solución: NAT. Rangos de direcciones IP privadas:
- Clase A: 10.0.0.0 (1 red)
- Clase B: 172.16.0.0 a 172.31.0.0 (16 redes)
- Clase C: 192.168.0.0 - 192.168.255.0 (256 redes)

NAT

Network Address Translation: consiste en modificar la dirección IP origen y/o destino de un datagrama IP al pasar a través de un router o firewall. Permite a múltiples máquinas en una red privada acceder a Internet usando una única dirección IP pública. Surge debido a dos problemas: escasez de direcciones IP y escalabilidad del enrutamiento. También ofrece seguridad, no se admiten conexiones desde fuera. Tipos de NAT:

- **Port Addresss Translation, PAT**, o **Network Address Port Translation, NAPT**, múltiples máquinas comparten una única dirección IP pública, la traducción se realiza mapeando números de puerto.
- **Basic NAT**: solo se realiza el mapeo de direcciones IP. Cada dirección IP privada tiene asignada una dirección IP pública.
- **Carrier-grada NAT, CGNAT**: uso de redes privadas a nivel ISP para compartir un pool de IPs públicas por múltiples clientes.

Ventajas

Seguridad: no se permiten conexiones bidireccionales. Una máquina interna debe iniciar la conexión con una máquina de Internet, evita conexiones maliciosas desde el exterior. Solución para la escasez de direcciones IPv4:

- Utilizar direcciones IP públicas sólo para máquinas que requieran conexión bidireccional a Internet.
- Direcciones privadas para las máquinas que sólo se conectan a Internet.

Inconvenientes

No existe una conectividad extremo a extremo real.

- Se usan los números de puerto para direccionar hosts, no procesos.
- Los routers sólo deberían implementar hasta el nivel de red. Es un parche para la escasez de direcciones, cuando IPv6 soluciona el problema de raíz. Plantea problemas en las aplicaciones que requieren que se inicien conexiones desde el exterior. Problema de **NAT traversal**.

Enrutamiento

Tabla de enrutamiento

Contiene la información de la tabla de enrutamiento. Todo dispositivo conectado a Internet tiene su tabla de enrutamiento en memoria. Cada entrada de la tabla de enrutamiento contiene la siguiente información:

- Dirección IP de **destino**: puede ser un host (host ID != 0) o una dirección de red (host ID = 0)
- **Gateaway**: dirección IP del siguiente router, en caso de ser necesario.
- **Máscara** de subred.
- **Flags**:
  - Up (U): indica que esa entrada está activada.
  - Host (H): activado si la dirección IP de destino es de un host.
  - Gateaway (G): activado si es necesario pasar por un router para llegar al destino.
- Especificación de la **interfaz** de red a la que se debe pasar el datagrama para su envío.

Destino	Gateway	Máscara	Flags	Interfaz

Destino	Gateway	Máscara	Flags	Interfaz
10.51.1.0	0.0.0.0	255.255.255.0	U	eth0
0.0.0.0	10.51.1.1	0.0.0.0	UG	eth0

Algoritmo de enrutamiento

Algoritmo de enrutamiento: a partir de la IP de destino de un datagrama, busca la entrada correcta en la tabla para su enrutamiento.

- Establece la manera en que se busca en la tabla de enrutamiento.
- No importa el orden de las entradas en la tabla.

1. Para cada entrada de la tabla de enrutamiento, se aplica la Máscara a la IP de destino y el resultado se compara con la columna Destino. Si coinciden, la entrada es válida para ese destino.

- Si el destino está directamente conectado, Flag G desactivado, se envía directamente a la interfaz de salida.
- Si no, Flag G activado, es necesario pasar a través de un router. Se envía por la interfaz de salida indicada al router.
- En caso de empate entre varias entradas, se selecciona aquella con una máscara mayor (más unos). **Longest match prefix**.

2. Se busca en la tabla de enrutamiento una entrada "default". Si se encuentra, se envía el paquete al router indicado.

- En realidad, todas las IPs coincidirán con la entrada default.
- Pero siempre pierde el longest match prefix contra cualquier máscara.

3. Si ninguno de los pasos anteriores tiene éxito, se genera el error "Red inalcanzable". Ha sido imposible entregar el datagrama.

Enrutamiento estático

En **enrutamiento estático, las tablas de enrutamiento se mantienen mediante intervención humana. Válido para entornos reducidos y más o menos estables. Para las redes directamente conectadas:

- Cuando se crea una interfaz, manualmente o por DHCP, se crea automáticamente una entrada para la red o subred. Para las rutas indirectas:
- Se definen mediante el comando route.
- Cuando se obtiene el router por defecto, manualmente o por DHCP, route add default gw 10.51.1.1
- O cualquier otra ruta. route add -net 192.16.20.0 netmask 255.255.255.0 gw 192.168.0.2 En **enrutamiento dinámico**, los routers actualizan sus tablas de enrutamiento en función de los cambios de la red o de la carga de tráfico.

Enrutamineto CIDR

Classless Interdomain Routing Las direcciones IP de clase B se están agotando, por lo que se asignan direcciones clase C a sitios con demandas de redes tipo B, lo que provoca un aumento vertiginoso en el tamaño de las tablas de enrutamiento. CIDR, especificado en los RFC 1518 y 1519, también conocido como

superredes o supernetting, ayuda a prevenir este problema, aunque se considera una solución temporal. Las superredes consisten en agregar direcciones y se definen mediante máscaras aplicadas sobre el identificador de red. Por ejemplo, la red 194.10.160.0/20 (máscara 255.255.240.0) incluye las siguientes redes clase C: desde 194.10.160.0/24 hasta 194.10.175.0/24, lo que representa un total de 16 redes. El valor decimal 160 equivale en binario a 1010 0000 y 175 a 1010 1111, lo que indica que las superredes agrupan múltiples redes en un solo bloque a través de una máscara más general. A nivel visual, el identificador de red y el de host pueden dividirse en una estructura como esta: | Id. red | Id. host | | Superredes | Subredes | Por ejemplo, el RFC 1466 propone la siguiente división por zonas geográficas: | Europa | 194.0.0.0 - 195.255.255.255 | | Norteamérica | 196.0.0.0 - 197.255.255.255 | | Centro y Sudamérica | 198.0.0.0 - 199.255.255.255 | | Anillo Pacífico | 200.0.0.0 - 201.255.255.255 | | Otros | 202.0.0.0 - 203.255.255.255 | Las redes tipo C europeas serían las 194.0.0.0/7, máscara 254.0.0.0. Con una sola entrada en las tablas de enrutamiento, fera de Europa, se englobarian 131072 redes de tipo C. **Classless**: las clases de direcciones IP, A,B o C; no se tienen en cuenta. Se utiliza la dirección completa y máscaras de 32 bits. Enrutamiento basado en **longest match prefix**: en caso de dos entradas correctas en una tabla de enrutamiento se selecciona la máscara de "mayor longitud".

ICMP

El protocolo IP no cuenta con mecanismos para obtener información de diagnóstico, por lo que se utiliza ICMP para cubrir esa necesidad. ICMP se encarga de comunicar mensajes de error y otras condiciones que requieren atención. Existen dos tipos principales de mensajes: los de error y los de consulta. Los mensajes ICMP se transmiten dentro de datagramas IP, tal como se define en el RFC 792. Entre los mensajes ICMP más utilizados se encuentran los siguientes:

- La petición y respuesta de eco, que se usa comúnmente en el comando ping
- Los mensajes de destino inalcanzable, que incluyen:
- Puerto inalcanzable, usado principalmente por UDP cuando el destino no tiene un proceso escuchando en el puerto especificado
- Máquina o red inalcanzable, que es generado por un router cuando no puede entregar o reenviar un datagrama IP También se utilizan mensajes como Redirect, Fragmentación requerida y Tiempo excedido En la arquitectura de red, ICMP opera a nivel del protocolo IP, siendo invocado tanto por TCP como por UDP desde la capa de transporte, y apoyado por Ethernet en la capa de enlace

Ping

Packet InterNet Groper. Herramienta de diagnóstico que comprueba si un nodo de la red es alcanzable. El cliente envía ICMP echo request, el servidor responde con ICMP echo reply. El formato de los mensajes ICMP echo request y reply contiene un identificador, en UNIX es el identificador del proceso, y un número de secuncia, inicalmente 0, y se incrementa con cada echo request. Existen variedad de implementaciones.

Traceroute

Problemas del ping con registro de ruta:

- Falta de espacio en la cabecera IP
    - Registro de ruta: máximo 9 routers
    - Timestamp: máximo 4 routers, o 9 timestamps sin direcciones IP.

- No todos los routers soportan la opción de registro de ruta.
- No hay control sobre los relojes de los routers. Solución: **traceroute**: herramienta de diagnóstico que permite ver la ruta que sigue un datagrama hacia un destino. Se basa en datagramas UDP, el campo TTL de la cabecera IP y los mensajes de error ICMP Puerto inalcanzable y Tiempo excedido.
- Solo requiere que el protocolo UDP esté operativo en el destinatario.
- Cuando un router al decrementar el campo TTL obtiene 0, genera un mensaje de error ICMP Tiempo excedido.
- Cuando UDP recibe un datagrama para un puerto vacío genera un mensaje de error ICMP Puerto inalcanzable.

Fragmentación IP

El nivel de enlace de la red impone un límite superior al tamaño de la trama que se puede transmitir: **Maximum Transmisions Unit, MUT**.

- Ethernet, 1500 bytes.
- Token Ring, 4440 bytes. Cuando el nivel de red (IP) recibe un datagrama, identifica la interfaz de red a utilizar y la interroga sobre su MTU:
- Compara la respuesta con la longitud del datagrama.
- Se hace fragmentación si la longitud del datagrama es mayor que el MTU. El **reensamblado** de datagramas IP fragmentados se produce cuando los fragmentos alcanzan el **destino final**:
- Lo hace IP en el destino.
- La fragmentación es transparente al nivel de transporte. En la cabecera IP se almacena la información relacionada con la fragmentación IP. **Identificación**: valor único para cada datagrama IP transmitido. Todos los fragmentos de un datagrama contienen el mismo valor. Flags:
- El primer bit está reservado.
- Bit **Don't Fragment, DF**: a 1 si se prohíbe fragmentar el datagrama IP.
- Bit **More Fragments, MF**: a 1 si hay más fragmentos a continuación. Se pone a 0 en el último fragmento. **Offset de fragmento**: desplazamiento en **múltiplos de 8 bytes** del fragmento desde el origen del datagrama original. **Longitud total**: se cambia la longitud total del datagrama por longitud total del fragmento. El tamaño de cada fragmento debe ser múltiplo de 8 bytes, excepto el último fragmento, por el campo offset de fragmento

Error ICMP

Error ICMP Unreacheble Error (Fragmentación Required)

- Mensaje de error utilizado por un router cuando tiene que fragmentar un datagrama IP pero tiene el flag DF activado.
- Incluye el MTU de la red que provocó el error y una copia de la cabecera del mensaje descartado.

Path MTU Discovery

Este mensaje de error es utilizado en un mecanismo denominado **Path MTU discovery** que permite averiguar el MTU mínimo durante una comunicación y reducir la fragmentación IP.

- Solo se implementa en el host origen.
- Path MTU: MTU mínimo en cualquier red en el camino entre dos hosts. Funcionamiento del Path MTU discovery:

1. Se habilita el bit DF en los datagramas enviados.
2. Si algún router en el camino necesita fragmentar, generará el mensaje ICMP Fragmentación requerida.
3. Si se recibe un mensaje ICMP Fragmentación requerida con el nuevo MTU:

- Si eran datos TPC, TPC debe reducir el tamaño del segmento, en base al nuevo MTU, y retransmitir.
- Sino, IP fragmenta los datagramas en base al nuevo MTU.

4. Como las rutas cambian dinámicamente, se puede probar un MTU mayor pasado cierto intervalo.

Limitaciones IPv4

Pocas direcciones, unas 4000 millones.

- Estructura de dos niveles, id de red y de host.
- Gran proliferación de redes. Crecimiento exponencial de Internet.
- Uso de TCP/IP en nuevas tecnologías. Móviles, tablets, TV...
- Múltiples IP por ordenador. Saturación del espacio de direcciones:
- Limita el crecimiento de Internet
- Enrutamiento ineficiente. Tablas de enritamiento muy grandes en la red troncal. Tiempos de respuesta grandes.
- Uso de Network Address Translation, NAT. Soporte inadecuado para aplicaciones con restricciones de calidad de servicio. No garantiza anchos de banda, tiempos de respuesta, seguridad. Se requieren mecanismos de seguridad en la capa de red:
- No fue diseñado para ser seguro. IPsec.
- Seguridad en los niveles superiores. TLS.

Características de IPv6

Espacio de direcciones ampliado y mecanismos de autoconfiguración: **Direcciones de 128 bits**, incremento en 2^{96}. Permite una arquitectura jerárquica de direcciones. Agregación de direcciones en el backbone. **Autoconfiguración** de los equipos. Mejora del multicast e introducción de las direcciones **anycast**. **Simplificación** del formato de la cabecera, tamaño fijo de 40 bytes, dos direcciones IP de 16 bytes y 6 campos más. Procesamiento más rápido y barato en los routers. Sporte mejorado de extensiones y opciones usando **Cabeceras de Extensión**. **Seguridad** intrínseca en el núcleo del protocolo: soporta autenticación y dispone de extensiones para la integridad y confidencialidad de los datos. Capacidad para **etiquetado de flujos**. Paquetes del mismo flujo de datos pueden ser etiquetados en origen, Calidad de Servicio, QoS.

Cabecera IPv6

Solo se requiere una cabecera. Se definen varias cabeceras de extensión opcionales:

- Cabecera de opciones salto-a-salto.
- Cabecera de encaminamiento.
- Cabecera de fragmentación
- Cabecera de las opciones para el destino
- Cabecera de autenticación

- Cabecera Encapsulating Security Security Payload (ESP) **Versión**, 4 bits que designan la versión del protocolo. **Clase de tráfico**: identifica diferentes clases o prioridades de paquetes. **Etiqueta de flujo**, 20 bits, permite diferenciar aquellos paquetes que requieren un tratamiento similar. Especialmente útil para tráfico multimedia y en tiempo real. Etiqueta de flujo + clase de tráfico: mecanismo potente de control de flujo y de asignación de prioridades diferenciadas según los tipos de servicios. **Longitud de carga**: longitud del paquete después de la cabecera IP. En IPv4, el campo longitud incluía la longitud de la cabecera + datos. En IPv6, no se considera la cabecera IPv6, y las cabeceras de extensión se consideran parte de la carga. Máximo tamaño de carga: 2^{16} = 64Kbytes. IPv6 permite la definición de Jumbogramas, paquetes de más de 64KB, que solo tienen sentido si el MTU de nivel de enlace es superior a 64 KB. **Cabecera siguiente**, 1 bit, identifica el tipo de cabecera que sigue a la cabecera IPv6. Las cabeceras deben ser procesadas en el orden riguroso en que aparecen. Las sucesivas cabeceras no son examinadas en cada nodo de la ruta, sino sólo en el nodo o nodos destino finales, excepto cuando se trata de la cabecera de opciones salto a salto. **Límite de saltos**, 1 byte, número de saltos permitidos. Análogo al campo TTL. **Dirección de origien**: 16 bytes **Dirección destino**: 16 bytes: normalmente, dirección IP del destino del paquete. Puede no ser el último destinatario del paquete, si está presente la cabecera de enrutamiento. Se eliminan 5 campos de la cabecera IPv4:
- Longitud cabecera: necesario en IPv4 al incluirse las opciones en la cabecera. Inútil en IPv6, cabecera fija de 40 bytes + cabeceras de extensiones.
- Identificación, flags y offset de fragmentación: necesarios para la fragmentación en IPv4. Si es necesaria, se realiza extremo a extremo (Path MTU discovery), utilizando la cabecera de extensión para fragmentación. **¡LOS ROUTERS NO SE FRAGMENTAN!**
- Checksum cabecera: eliminado para mejorar el rendimiento, así los routers no tienen que calcular y actualizar el checksum. Ya se realiza en el nivel de enlace, probablemente, y en el nivel de transporte. IP no es fiable.

Direccionamiento IPv6: Notación

Se representan mediante 8 bloques de 16 bits en hexadecimal, separados por ":"

- FEDC:BA98:7653:3210:FEDC:BA98:7654:3210
- FE80:0000:0000:0000:0202:B3FF:FE1E:8329 Se pueden eliminar los ceros por la izquierda en cada bloque
- FE80:0:0:0:202:B3FF:FE1E:8329 Se eliminan bloques consecutivos de ceros utilizando el carácter "::"
- FE80::202:B3FF:FE1E:8329 Solo pueden aparecer una vez en la dirección
- CAFF:CA01:0000:0056:0000:ABCD:EF12:1234
- CAFF:CA01::56:0:ABCD:EF12:1234
- CAFF:CA01:0:56::ABCD:EF12:1234

Direccionamiento IPv6: Tipos

Hay tres tipos de direcciones

- **Unicast**: identifica unívocamente **una** interfaz de un nodo IPv6. Un paquete dirigido a una dirección unicast se envía a la interfaz asociada a esa dirección.
- **Multicast**: identifica un grupo de interfaces IPv6. Procesado por **todos** los miembros del grupo, sustituye a las direcciones de broadcast. Prefijo **FFxx**/8.
- **Anycast**: se asgina a múltiples interfaces, típicamente en múltiples nodos. Enviado a sólo **una de esas** interfaces, normalmente la más próxima. Las direcciones IP se asginan a interfaces, como en

IPv4, cada interfaz necesita, al menos, una dirección unicast y puede tener asignadas múltiples direcciones de cualquier tipo. **Unicast global**: 2000::/3 - rango asignable actualmente, similares a las IPv4 públicas y enrutables en Internet. **Link-local**: FE80::/10, utilizadas en un mismo enlace local y limitada a un único enlace. **Local única**: FC00::/7 - FDFF::/7, similares a las IPv4 privadas. Se usan para direccionamiento dentro de un sitio o entre una cantidad limitada de sitios. **Loopback** ::1/128 y dirección sin especificar, ::/128.

IPv6: DNS e ICMPv5

DNS: se requieren unos cambios para resolver las peticiones de direcciones IPv6.

- Petición DNS IPv6: AAAA
- A partir de un nombre, obtendrá la dirección IPv6 asociada.
- dig @8.8.8.8 www.udc.es AAAA **ICMPv6** se define una nueva versión del protocolo.
- Reorganiza los tipos y códigos existentes, y define nuevos tipos.
- Incorpora funciones de Internet Group Managment Protocol, IGMP.
- Introduce el protocolo Neightbor Discovery Protocol, NDP.
- Incorpora funciones ARP. **Autoconfiguración**: dos mecanismos
- Stateless Address Autoconfiguration, SLAAC.
- DHCPv6.

Transición IPv4 a IPv6

IPv6 e IPv4 van a coexistir durante muchos años. Se han definido múltiples técnicas para la transición, que sae agrupan en tres categorías:

- Pila dual: permiten a IPv4 e IPv6 coexistir en los mismos dispositivos y redes. Soporte completo de las dos versiones de los protocolos en los nodos.
- Tunneling: permiten transportar tráfico IPv6 sobre infraestructuras IPv5 existentes. El tráfico IPv6 sobre infraestructuras IPv5 existentes. El tráfico IPv6 se encapsula en paquetes IPv4.
- NAT: permiten a los nodos IPv6 puros comunicarse con los nodos IPv4 puros. Traducen una dirección IPv6 en una dirección UPv4. Estas técnicas pueden y deben utilizarse de manera combinada.

Direcciones MAC

En internet, cada host tiene una dirección lógica IP. En las redes físicas, cada host tiene una dirección hardware, física o MAC. En la mayoría de redes LAN, una dirección MAC son 48 bits. por ejemplo, 1A:23:F9:CD:06:9B. Los primeros 24 bits son el Organizationally Unique Identifier, OUI, y los últimos 24 bits son asignados por el fabricante. La dirección de una tarjeta de red es única a nivel mundial. Al transmitir una trama, se indica la dirección MAC de destino. En una red de broadcast: Todos los nodos reciben la trama. La interfaz comprueba si la dirección de destino coincide con la suya, de coincidir se envía al nivel de red, por el contrario, se descarta. Dirección MAC de broadcast: FF:FF:FF:FF:FF:FF

ARP

¿Cómo se convierte/mapea una dirección lógica en una dirección MAC? Es decir, ¿cómo se convierte una dirección IP de 32 bits en una dirección Ethernet de 48 bits? Adress Resoluton Protocol, ARP, proporciona la correspondencia entre direcciones IP y direcciones MAC. ARP proporciona correspondencia dinámica, no

concierne al usuario ni al administrador de la red, entre distracciones IP y direcciones MAC usadas por distintas tecnologías de red.

ARP caché

El broadcast de los ARP Request es costoso ya que todos los receptores tienen que procesar este paquete. El caché ARP mantiene las conversaciones entre direcciones hardware. En un mensaje ARP Request, si la IP del emisor ya está en la cache, se actualiza con la dirección HW del emisor. El tiempo normal de vida es de 20 minutos.

Ethernet

Procolo de acceso aleatorio para canales de difusión. Se inventó a mediados de los 70 y se basaba en una topología en bus, con un cable **coaxial** conectado a todos los nodos. A mediados de los 90 se pasó a una topología en estrella basada en **concentradores** (hubs). Los equipos se conectaban con un cable de cobre de par trenzado al concentrador. A principios de la década de 2000, se cambió el concentrador po un **conmutador**, aportando una mayor velocidad efectiva. Va desde 10 Mbps hasta 10 Gbps hoy en día, todo sobre la misma trama Ethernet. Facilita la interconexión.

CSMA

En las redes LAN, y de radio, el retardo de programación entre las estaciones es mucho más pequeño que el tiempo de transmisión de las tramas:

- Cuando una estación transmite una trama, el resto lo saben casi instantaneamente.
- Si las estaciones pueden saber que otra estación está transmitiendo, esperan para evitar la colisión.
- Sólo habrá colosiones cuando dos estaciones empiecen a transmitir casi simultáneamente. Esta técnica se denomina de acceso múltiple sensible a la portada o Carrier Sense Multiple Access, CSMA. Una estación escucha al medio antes de transmitir, si está ocupado, espera, sino, transmite. Si dos estaciones intentan transmitir casi al mismo tiempo se produce una colisión, es necesario una confirmación del receptor que también debe competir por el cana. Tiempo de espera después de una colisión. CSMA 1-persistente: espera hasta que el canal esté libre y después transmite. Se produce colisión si hay dos o más estaciones esperando.

CSMA/CD

En CSMA, si colisionan dos tramas, el medio está inutilizado durante la transmisión de esas tramas. Continuar escuchando el canal mientras dura la transmisión (**Collision Detection**), no necesito recibir confirmación. Si el medio está libre, transmite, sino, continua escuchando hasta que esté libre, si se detecta una colisión durante la transmisión, se transmite una señal corta de alerta y se corta la transmisión. Se espera un tiempo aleatorio y después se intenta transmitir de nuevo.

- Tras cada colisión, sobre la misma trama, el tiempo de espera se dublica.
- Tras N intentos no se transmite más y se genera un mensaje de error.
- Si se congestiona el sistema, las estaciones deben esperar más y más para liberar al medio. El tiempo de detectar una colisión es <= dos veces el retardo de programación extremo a extremo. Una trama debe ser suficientemente larga para detectar la colisión antes de que acabe su transmisión.
- 2500 m. Aproximadamente 25msegs de propagación.

- Tº detección colisión = 25 msegs x 2 = 50 msegs
- Enviar 64 bytes a 10 Mbps -> 64 x 8 / 10 Mbps = 51.2msegs

Trama

8 bytes	6 bytes	6 bytes	2 bytes	>= 0 bytes	>= 0 bytes	4 bytes
Preámbulo	Destino	Origen	Tipo	Datos	Relleno	FCS

Preámbulo: patrón de 8 bytes, con 0's y 1's alternados, para sincronizar el emisor y el receptor:

- El último byte es 01010111.
- El receptor puede localizar el primer bit del resto de la trama. Dirección destino: puede ser una puede ser una dirección única, de grupo o global. Tipo: indica el tipo de protocolo utilizado en el campo de datos. En la cabecera IEEE 802.3 el campo Tipo indica la longitud (si <= 1500) o el tipo (si > 1535). Datos: máximo 15000 bytes. Relleno: bytes añadidos para garantizar que la técnica de detección de colisiones pueda operar correctamente, mínimo 46 bytes. Frame Check Sequence, FCS: cógido CRC de detección de errores, incluye todos los campos, excepto el preámbulo, el SFD y el FCS.

WiFi

Los sistemas inalámbricos se destacan por su movilidad y flexibilidad. Aunque no reemplazan completamente a las redes "tradicionales" (como aquellas basadas en servidores o dispositivos estáticos), ofrecen conectividad con ciertas limitaciones, especialmente en cuanto al ancho de banda. Uno de los principales estándares de redes inalámbricas es WiFi, basado en la norma IEEE 802.11 (más información). Además del WiFi, existen otros sistemas de transmisión inalámbrica, como los basados en redes móviles (GSM, GPRS, UMTS –3G–, 4G, 5G), así como tecnologías como Bluetooth o WiMAX. Elementos clave de una red inalámbrica:

- Red de infraestructura: Parte lógica del estándar 802.11 que permite enviar tramas a su destino. No está ligada a una tecnología específica, aunque normalmente se utiliza Ethernet. **Punto de acceso**: Dispositivo encargado de enviar y recibir tramas de los equipos inalámbricos conectados. **Medio inalámbrico**: Utiliza ondas de radiofrecuencia para la transmisión de datos. **Equipo inalámbrico**: Dispositivos con capacidad para conectarse a redes inalámbricas, como portátiles, tabletas o teléfonos móviles. **Basic Service Set, BSS**: grupo de estaciones que se comunican entre sí.
- BSS indpendiente, o ad-hoc: se cominican directamente.
    - Grupo reducido
    - Carácter temporal
- BSS infraestructura: usan un punto de acceso.
    - Comunicaciones entre estaciones móviles pasan por el pnto de acceso. Una estación se **asocia** a un punto de acceso.
    - Los puntos de acceso envían periódicamente una señal baliza.
    - Distancia de las estaciones al punto de acceso, no entre estaciones. **Extended Service Set, ESS**: asociación de BSSs. Se encadenan varias BSSs usando un backbone. **Service Set Identifier, SSID**: Identifica la red inalámbrica asociada a un punto de acceso. Un equipo móvil debe asociarse con un punto de acceso (PA). Los puntos de acceso envían periódicamente tramas **baliza** (MAC del PA + SSID).
- Exploración pasiva: el equipo espera a recibir tramas baliza.

- Exploración activa: el equipo solicita a los PA que se identifiquen. El equipo determina a que punto de acceso asociarse. Seguridad:
- Filtrado MAC
- Login y password, sobre un servidor de autentificación. Después, configuración IP por DHCP.

WiFi: CSMA/CA

Una vez asociado, el equipo móvil puede transmitir y recibir tramas del PA. Subcapa **MAC** del nivel de enlace. Pero, otra vez, tenemos el problema del acceso múltiple. ¿Por qué no CSMA/CD?

- Problema del **nodo oculto**, no todas las estaciones reciben todo. CSMA/CA:
- Cuando una estación empieza a transmitir, transmite la trama completa, haya o no colisión. Necesita un ACK para confirmar repetición. Solución al problema de los nodos ocultos: RTS/CTS.
- Cuando un emisor quiere transmitir, primero envía un Request To Send, RTS, indicando el tiempo total que necesita.
- Cuando el PA recibe el RTS, responde con un Clear To Send, CTS, indicando el tiempo restante que tiene reservado en el canal. El emisor sabe que tiene el canal disponible y el resto saben que el canal estará ocupado. Beneficios:
- Una trama sólo se enviará después de reservar el canal. Evita colisiones de nodos ocultos.
- Las colisiones se producen sobre las tramas RTS o CTS, que son tramas cortas. Desventajas: Introduce un retardo y consume recursos del canal. Es opcional-

WiFi: Seguridad

Las redes WiFi transmiten datos a través del aire, lo que las hace especialmente vulnerables a escuchas no autorizadas. Aunque esta vulnerabilidad también existe en redes cableadas, en el caso de las redes inalámbricas se requieren mecanismos de seguridad adicionales. Evolución de los mecanismos de seguridad:

- En un inicio se utilizaba WEP (Wired Equivalent Privacy), pero debido a sus debilidades, fue reemplazado por la familia WPA: WPA, WPA2 y WPA3. Principales protocolos de seguridad: **WEP (Wired Equivalent Privacy)**:
- Utiliza una clave estática, hoy considerada insegura y fácil de romper.
- Es eficiente en términos computacionales (usa claves de 64 a 128 bits).
- Fue exportable internacionalmente y su uso era opcional.
- Usaba el algoritmo RC4 para cifrado y CRC-32 para verificación de integridad. **WPA (WiFi Protected Access)**:
- Introduce TKIP (Temporal Key Integrity Protocol), que cambia las claves de cifrado de forma dinámica durante el uso.
- Incluye MIC (Message Integrity Check) para verificar la integridad de los mensajes. **WPA2**:
- Utiliza el estándar AES para cifrado y CCMP para garantizar la integridad.
- Es un método más seguro que WPA, utilizando claves de 128 bits. **WPA3**:

- Es el método más seguro actualmente, con claves de 192 bits.
- Ofrece mayor protección, incluso si se utilizan contraseñas simples.
- A diferencia de versiones anteriores, la contraseña inicial no se utiliza para derivar directamente las claves. Incluso si esta es descubierta, no se pueden recuperar las claves de sesión.