ORIOL MARCO DEUMAL domingo, 20 de junio de 2021

APUNTS FINAL

Models

- **Model OCI**: Utilitzat per connexions a la xarxa. Es divideix en capes per facilitar la comunicació entre xarxes i permet estandaritzar connexions entre sí. Model conceptual no s'usa.
  - **Capa Física**: Es defineix com viatges els bits sobre un enllaç de dades conectant els nodes de la xarxa per formar finalment una xarxa.
  - **Capa Enllaç:** S'encarrega de transferir dades entre nodes que pertanyen al mateix segment de xarxa.
  - **Capa Red**: Responsable de l'reenviament de paquets i de l'enrutament a través de routers intermitjos.
  - **Capa Transporte:** Transfereix seqüències de longitud variable des d'una font a un host de destí, mentres es manté la qualitat de les funcions de servei.
  - **Capa Sessió:** Mecanisme per obrir, tancar i administrar una sessió entre els processos de la aplicació de l'usuari final.
  - **Capa Presentació:** Responsable del format I de l'entrega de la informació cap a la següent capa, per la seva posterior visualització o processament.
  - **Capa Aplicació:** Capa d'abstracció que especifica els protocols de comunicacions compartits i els mètodes usats a la xarxa de comunicacions.
- **TCP/IP:** Model actual. Dividit en:
  - **Interfaz de red**: Capa física i capa enllaç.
  - **Red**: Capa red.
  - **Transport**: Capa de transport.
  - **Aplicació de red**: Capa Sessió, Capa Presentació i Capa d'Aplicació.
- **Paradigma Client-Servidor:** Perquè el client accedeixi a un servei ha d'accedir a un servidor que estigui a N xarxes d'aquest servei, el servidor proporciona el servei.
  - **Servidor**: Ofereix servei o recurs en un Well-Known Port (<1024).
  - **Client**: Consumidor recurs, no sempre inicia la connexió.

  - Usat en **TCP** (servei confiable, comprova que els segments arribin bé I estiguin orientats a la connexió) I en **UDP**(servei no confiable, conection less, fa ús de datagrames).

- **Protocol IP:** usat a tot internet amb la funció de comunicar dispositius a través de les xarxes. 5 paquets 32 bits = 160 bits = 20 bytes +40 (opcions). Esq a dreta:
  - **Versión (0-3)**: **v4**(4 * 8 = 32 bits), **v6** (8 * 6 = 48 bits).
  - **Longitud(4-7):** Longitud capçalera.
  - **Tipus de servei(8-15): Best effort** (millor esforç per realitzar entrega paquets, no es pot assegurar QoS) **Stateless** (protocol sense estat, la connexió no manté l'estat de la connexió, cada connexió es tracta de forma independent) **Conectionless**(no orientat a la connexió, es poden enviar paquets sense tenir en compte si el dispositiu final està disponible o no).
  - **Longitud total (19-32):** Longitud total datagrama.
  - **Identificació(0-15):** id del paquet, s'usa quan hi ha fragmentació.
  - **Flags(16-18): 0** (no fragmentació) **1** (fragmentat) **2** (últim fragment).
  - **Desplaçament de fragment(19-32):** Posició dins el fragment segmentat.
  - **Temps de vida(0-7):** Nº màx de salts entre routers que es poden fer.
  - **Protocol(8-15)**: Indica el protocol de les capes superiors.
  - **Checksum(16-32):** Detecta paquets corruptes o danyats.
  - **Direcció IP Origen(0-31):** Origen paquet.
  - **Direcció IP Destí(0-31):** Destí paquet.
  - **Opcions(0-31)(40bytes):** Camp Opcions

-Fragmentació: Si el paquet que ens arriba es major que la MTU(Maximum Unit Transfer), s'ha de fragmentar.

  - **MTU Path Discovery:** Evita la fragmentació de paq IP, enviant un ICMP que no es pot fragmentar, sabent així el tamany màxim que podem atravessar.

- **Unicast:** Datagrama enviat a un sol destinatari.
- **Broadcast**: Datagrama enviat a tots els integrants de la xarxa.
- **Multicast:** Datagrama enviat a múltiples destinataris que poden estar a altres xarxes.

Clase	NetID	HostID	Rang
A	1	3	0.0.0.0 a 127.255.255.255
B	2	2	128.0.0.0 a 191.255.255.255
C	3	1	192.0.0.0 a 223.255.255.255
D (Multicast)	-	-	224.0.0.0 a 239.255.255.255
E (Reservades)	-	-	240.0.0.0 a 255.255.255.255

-Direcció de red: Identifica la red, tot el Host ID a 0: 192.168.1.0/24

-**Direcció de broadcast**: Envia missatges a tota la red, posant tot el hostID a 1: 192.168.1.255/24.

Classes Privades	Rang
Classe A	10.0.0.0 - 10.255.255.255
Classe B	172.16.0.0 - 172.31.255.255
Classe C (domèstic)	192.168.0.0 -192.168.255.255

———————————————-**TAULES D'ENRUTAMENT**———————————

- **Adquisició**: mecanisme o protocol usat. **S** (rutes estàtiques), **R** (rutes per RIP), **C** (connectades directament), **O** (Apreses per OSPF).
- **Destí i màscara:** Es defineix mitjançant la direcció IP i la seva màscara en format decimal. Si és 0.0.0.0 és la ruta per defecte.
- **Gateway:** Següent salt per arribar al destí. 0.0.0.0 —> connectats directament.
- **Interfaz:** Interfaç per la que es surt cap el destí.

————————————————**PROTOCOL ARP** ————————————————

Associar una IP amb una MAC, per poder realitzar correctament la comunicació local.

I. Enviem missatge ARP en broadcast (ARP request), preguntant qui té la direcció.

II. L'equip respon indicant la seva MAC(ARP reply), associant així la MAC a la IP X a la taula ARP.

- ARP funciona només a nivell local, en altres xarxes el router es qui realitza l'ARP.
- **Reverse ARP**: Associar MAC a IP (al revés).
- **ARP gratuit**: Serveix per detectar si una IP està duplicada.

———————————————PROTOCOL ICMP———————————————

Intercanviar missatges d'estat o d'error entre dues terminals. Un ICMP mai genera un altre ICMP.

- **Tipo:** Indica el tipus de missatge ICMP
- **Codi:** Junt amb el tipus, indica el ICMP que tractem.
- **Checksum**: Comprovació d'errors.
- **Dades ICMP**: Varia depenent del missatge.

———————————————PROTOCOL DHCP———————————————

Realitza la configuració inicial de la xarxa quan els equips entren a la xarxa. El **servidor** usa el port **67**. El **client** usa el port **68**. Viatja per **UDP.**

- **DHCPDISCOVER —>** Iniciem a la direcció 0.0.0.0, aquest envia missatge a broadcast, els hosts que no siguin servidor descartaran aquest missatge.
- **DHCPOFFER —>** El servidor respon a broadcast les dades de la configuració.
- **DHCPREQUEST —>** El client accepta la configuració oferta.
- **DHCPACK —>** El servidor envia un ACK per confirmar que s'ha fet correcte.

————-MECANISMO NAT(NETWORK ADDRESS TRANSLATION)————————

IP PRIVADA —> IP PÚBLICA

- **NAT estàtic:** Les direccions es mapejen una a una. Una IP pública s'associa a una privada.
- **NAT dinàmic:** Tenim una pool de direccions públiques que utilitzarem per mapejar de forma dinàmica segons la demanda de direccions que tinguem.
- **PAT(Port Adress Translation):** Més usat en cases, ens fa estalviar moltes direccions públiques, ja que només s'usa una direcció pública a la red per a que múlriples IP's privades puguin sortir, s'usa la direcció pública junt amb el port.

————————————-PROTOCOLS D'INTERNET———————————————

Internet està dividit en sistemes autonoms (AS) que són grups grans de xarxes que tenen una política interna que d'ha d'encamincar (ISP es un AS). Entre els AS, de forma interna es comuniquen mitjançant **OSPF**, que és usat per comunicar per IGP (Interior Gateway Protocol) i usa vector de distància. En canvi, a la comunicació externa usem **BGP**, que es comunica per EGP(External Gateway Protocol) i usa un vector de camí.

————————————————PROTOCOL RIP————————————————-

Utilitzat amb vector de distància. Envia missatges periodicament per actualitzar les taules d'enrutament de cada router. Aquest podrà actualitzar la informació mitjançant els missatges que li hagin arribat del router destí. La mètrica s'indica sent **1** el mínim**(** connectat directament**)** i **16** el màxim (infinit, enllaç caigut).

Per evitar bucles a la xarxa fem ús d'split horizon. Aquest, indica que un router no pot publicar una ruta per la mateixa interfaç per la que l'ha apresa.

Poison Reverse: Quan un enllaç cau, el router envia un update amb la red caiguda amb una mètrica de 16.

————————————————————ACL————————————————————

S'usen com a mètode de seguretat, podent filtrar els paquets que entren i surten d'una red. Regles.

Protocol || IP origen || Port Origen || IP destí || Port Destí || Acció

ssh —> 22 TCP

telnet —>23 TCP

FTP —> 20/21 TCP

DHCP —> 67/68 UDP

DNS —> 53 UDP

Https —> 443 TCP

Http —> 80 TCP

- Al final hem de fer un - - - - - Deny

————————————————-VPN I TÚNELS————————————————

VPN —> Red virtual privada, es crea mitjançant un túnel amb el que podem connectar-nos des de qualssevol ubicació a la xarxa de l'empresa.

Túnel —> Enllaç virtual que uneix dos routers, per poden enviar paquets s'ha de crear una xarxa entre ambdós.

IPinIP—> El túnel encapsularà el paquet IP dins un altre paquet IP amb les dades de la xarxa en el túnel. Per tant, tamany Nou = tamany del paquet + 20 bytes

————————————————————LAN'S————————————————————-

Sempre a la mateixa xarxa quan parlem de LAN.

- **LLC(IEEE 802.2):** El mateix en tots els estandards de MAC.

- **MAC(IEEE 802.X):** Distintos protocolos encargados de repartir el uso de la red local en el medio.

- **Colisió**: Quan dos dispositius intenten enviar dades a la vegada en una red eth.

- **Domini de Colisió:** Segment de la xarxa on pot haver-hi la colisió.

————————————————ETHERNET————————————————-

Protocol MAC Token Passing: Funciona mitjançant un token per evitar les colisions a la xarxa, només la terminal que tingui el token pot transmetre, per tant només pot viatjar un paquet a la vegada.

Random: No hi ha token, per tant existeix probabilitat de colisió, per evitar-ho s'usa el temps de back-off, el qual fa que esperis per tornar a reenviar la trama.

Preamble (8 bytes)	DESTINATION MAC address (6 bytes)	SOURCE MAC address(6 bytes)	FRAME TYPE (2 bytes)	Payload (46 to 1500 bytes)	CRC (4bytes)

Preamble: Usat per sincronitzar el rellotge.

MAC destino i MAC origen: identifiquen la direcció MAC de destí i origen.

Longitud: Indica la longitud de la trama.

Payload: Dades encapsulades dins la trama.

CRC: Càlcul de redundància i control d'errors.

10baseT = 10Mbps 100baseT = 100Mbps G(10G) = 10GBps

- **CSMA / CD:** Sistema usat a Ethernet per detectar colisions.
  - **CSMA —>** Abans d'iniciar una transmisió de dades, l'estació verifica si una altra estació està transmetent ja les dades.
  - **CD —>** L'estació també buscarà en cas que hi hagi dues transmisions iniciades a la vegada, en cas de detectar que una altra estació esta transmetent, s'espera un temps de back off en el que esperem per transmetre.
  - **Half-duplex:** permet transmetre.
  - **Full-duplex:** permet transmetre i rebre a la vegada.
- **Sol·lució al problema de les col·lisions:** Mitjançant:
  - **Bridges** (**ponts**): Un pont de red és un dispositiu que divideix una red en segments. Cada segment representa un domini de colisió independent, pel que es redueix el nombre de colisions a la xarxa. Cada domini té el seu propi ample de banda pel que també és millorarà el rendiment a la xarxa.
  - **Switch (connmutador)**: Dispositiu que connecta dispositius en una LAN (mateixa xarxa), un conmutador és essencialment un pont de red multiport i realitza les mateixes funcions que un pont tot i que a velocitats més ràpides i

amb més característiques. Cada port està ubicat en un domini de colisió independent, pel que pot executarse en full-duplex.

1. Si la **direcció origen** no és a la taula: s'agrega aquesta a la taula MAC del switch.
2. Si la **direcció destí** no és a la taula, o es una trama **broadcast,** o **multicast:** Es copia la trama en tots els buffers de transmissió dels altres ports.
3. Si es rep una trama dirigida a una altra estació des del mateix port: Es descarta aquesta.

- Si no es fa ús d'VLAN's, un missatge de broadcast s'enviara per tots els ports d'un switch, l'unica forma de segmentar-ho és mitjançant un router.
- El switch realitza control de fluxe mijtançant els mecanismes: **Jabber Signal**(si és Half duplex) o **Pause Frames**(si és full duplex).
- Per dissenyar bé s'ha d'estar lliure de bucles, mitjançant el protocol **STP(Spanning Tree Protocol):** encarregat de conseguir una topologia lliure de bucles, deixant els possibles enllaços que poden causar algun conflicte com a enllaços de repostament en cas de que algún altre caigués.

———————————MECANISMO VLANS———————————

Permet crear dominis de broadcast logics que poden abarcar un switch o varis. Útil per reduir el tamany dels dominis de broadcast o per permettre que grups d'usuaris s'agrupin logicament sense necessitat d'estar ubicats en un mateix lloc. Cada port del switch indica a quina VLAN pertany.

———————————————————WIFI———————————————-

Protocol 802.11X, depenentde la lletra variarà la velocitat de connexió.

NOM	Desc	Velocitat	Any
802.11	Primera versión	1-2Mbit/s	1997
802.11b	Comercial,wifi	11Mbit/s	1999
802.11g	Wifi alta velocitat	54 Mbit/s	2003
802.11n	Més velocitat	Fins 600 Mbit/s	2009
802.11ac		Fins 1.3 Gbit/s	2013
802.11ax		Fins 10Gbit/s	2019

Per tractar problema de colisions s'usa **CSMA/CA,** aquest funciona de forma que sempre es fa ús del temps de backoff, d'aquesta forma assegura un sistema sense

colions. Es fa servir perquè en xarxes inalambriques és molt difícil trobar colisions degut a la diferència de magnitud en la transmisió i recepció respecte les xarxes no cablejades.

Frame Control (2)	Duration 2	Address 1	Address 2	Address 3	Seq Ctrl	Address 4	Payload 0 - 2312	FCS. 4

Problema Node Ocult —> Quan dos nodes estan en rang diferent de cobertura del AP, s'entra en conflicte quan s'envien dades a la vegada, provocant una colisió.

————————————PROTOCOLO TCP/UDP————————————

Pertanyen a la capa de transport, encarregats de crear canals de comunicació entre les aplicacions mitjançant els ports (identificadors de 16 bits) i mitjançant aquest, identifiquem l'aplicació. Els ports <1024 identifiquen serveis coneguts (**well-known ports**), la resta són **ports efímers** I identifiquen clients (**PAT**). Usen **client-servidor**.

- **UDP: No confiable** ja que no hi ha forma de recuperar un paquet si es perd, per tant, **no té recuperació d'errors** ja que no té cap mecanisme que utilitzant la secuencia de bytes pugui detectar un error, a més a més, **no funciona amb ACK** i per tant **no pot estar orientat a la connexió**, ja que no emplea cap sincronització entre l'origen i el destí. **No té control de flux** pel que els paquets poden arribar en qualssevol ordre. **No té buffer de transmissió** per tant no pot enviar bit a bit, sinó que envia **datagrames**.

Source Port(8 bits) (2bytes)	Destination Port (8 bits) (2bytes)
Lenght (8 bits) (2bytes)	Checksum (8 bits) (2bytes)

Source & destination Port: identifica l'aplicació d'origen i destí.

Lenght: tamany del datagrama UDP, inclou tbé el tamany de la cabecera.

Checksum: Control d'errors en quant a la capçalera.

————————————————-PROTOCOL ARQ————————————————

Usats per TCP per control de fluxe.

Control de fluxe: Mecanisme encarregat de regular l'excés de fluxe quan el buffer de lectura rep massa informació i es crea una sobrecarrega en el buffer.

Control de congestió: Mecanisme encarregat de regular la congestió a la xarxa, quan s'ofereix més tràfic del que la xarxa pot soportar.

Hi ha 3 protocols ARQ, aquests envien ACKS (segments que enviaran aquells que rebin un missatge per poder confirmar que aquest s'ha rebut correctament) dins el segments TCP. També tenen un altre camp que es RTO (temporitzador que serveix per saber si s'ha Perdut un segment, aquest es reinicia cada cop que rep un ACK).

- **Stop & Wait:** Emisor envia un segment al buffer de Tx i es relantitza la transmissió del segment. El receptor rep el segment I genera l'ACK. L'emisor rep l'ACK.
- **Go back N:** Usa ACK acumulativo, Si enviamos un ACK con numero de secuencia 10, estamos confirmando los segmentos del 1 al 10. En caso de haber un error en el envio o recibir una PDU fuera de orden, descarta todos los segmentos hasta que llegue el que tiene que llegar en orden. Si hi ha un RTO torna enrere fins al segmento i envia tots a partir d'aquell.
- **Selective Retransmission:** L'emisor només retransmetrà si hi ha hagut un RTO. Perquè la retransmisió selectiva funcioni, el receptor ha de ser capaç d'emmagatzemar el segement fora d'ordre.

———————————PROTOCOLS AMB FINESTRA———————————

Finestra òptima: finestra que permet conseguir la velocitat efectiva utilitzant el menor tamany de la pantalla. Molt petita (No aprofita el max de la vef), molt gran (si ja usa el max de la vef, no cal que es faci més gran.)

———————————————————TCP———————————————-——

Orientat a connexió, ja que abans de començar a transmetre la informació s'encarrega de realitzar una conexió mutua entre el client i el servidor (Three Way Handshaking), **Fiable**, ja que si es perd un segment es pot recuperar. Control de congestió, ja que ofereix una solució a la congestió (Quan en una red està circulant més informació de la que es pot aguantar). Funciona amb **ACK,** mantenint així un control sobre els segments**.**

Source Port(8)	Source port(8)	Destination Port (8)	DestinationPort(8)
Sequence number(8)	Sequence number(8)	Sequence number(8)	Sequence number(8)
Acknowledgment Number (8)	Acknowledgment Number (8)	Acknowledgment Number (8)	Acknowledgment Number (8)
Header Lenght(4)	RESERVED(6)	URG,ACK,PSH,RST,SYN,FIN (6)	Advertised window awnd (16bits)
Checksum(8)	Checksum(8)	Urgent Pointer(8)	Urgent Pointer(8)
Options(8)	Options(8)	Options(8)	Padding(8)

Nº sequencia: identifica el primer byte del camp de dades.

Nº de ACK: Conté el número del següent byte que estigui disposat a rebre.

URG—> Dades urgents, urgent pointer indica la quantitat d'aquestes que es troben.

ACK—> Reconeixer els paquets que el host ha rebut correctament.

PSH—> Indica al receptor que ha de processar els segments a mesura que són rebuts.

RST —> S'usa per acabar la conexió.

SYN —> S'usa quan establim conexions o en els processos TWH entre els dos hosts.

FIN —> S'usa per sol·licitar la terminació de la conexió, quan no hi ha més dades a enviar pel remitent. Allibera els recursos reservats i finalitza la conexió.

Advertised window—> quants bits composen la finestra de transmisió del protocol de control de fluxe.

Checksum: Per detectar errors

Opciones: S'usa afegint camps a la capçalera per **realitzar operacions** com **indicar el tamany màxim** del segment o indicar el **factor d'escalada.**

Padding: S'usa en TCP per garantitzar que finalitzi i totes les dades comencin a 32.

———————————-THREE-WAY HANDSHAKING———————————

1(SYN): Enviem un segment amb SYN que informa al servidor que volem iniciar una comunicació amb ell, i també li envia el número de seqüència. El client fa una trucada al sistema connect() per iniciar la conexió i servidor esta en listen().

2(SYN+ACK): El servidor respon enviant un ACK amb el número de seqüència que ha rebut+1 I un nombre aleatori com a nombre de seqüència. ACK és la resposta i SYN significa amb quin nombre de seqüència començaran els segments.

3(ACK): El client envia un ACK al servidor. El número de seqüència s'estableix com ACK+1, i l'ACK s'estableix en un més que el nombre de seqüència rebut.

Si tenim una finestra de 3000 i un factor d'escalat 3 —> 3000*2^3 = 24000 awnd.

———————————Slow Start/ Congestion avoidance———————————

Part de l'estrategia de control de congestió utilitzada per TCP, és bastant agressiu, ja que la finestra de congestió es duplica per cada ACK rebut.

————————-APLICACIONS DE XARXA————————————————

Aplicacions que donen un servei a través d'una xarxa que pot estar a nivell locar o en una altra xarxa diferent.

DNS(Domain name system)—> Traduïr IP's a noms entendibles per a les persones. Cada vez que enviemos una petición DNS iremos a un servidor que contendrá unos registros(**Resource Records**) donde estará la información de lo que estamos pidiendo. Cada servidor DNS estarà format per un conjunt d'**RR's.** Aquests poden ser **estàtics(**permanents) o "**cacheados"** (que desapareguin en un temps). Cada NS s'encarregarà del seu domini i tindrà com a mínim dos servidors, el principal i el de backup. Un NS pot delegar part de la seva zona a un altre NS, la zona delegada passa a ser la subzona.

Per fer una resolució DNS, el client farà una petició recursiva, delegant així tot el treball sobre el servidor de DNS. El client realitza la petició al servidor i espera en resposta de la direcció ip que correspon al nom solicitat. El servidor DNS farà peticions iterativas a cada NS per poder arribar finalment a la direcció que busca. Els CDN són servidors mirall que serveixen per poder accedir més ràpidament a la informació, ja que accedim al servidors més propers que tenim, creant així un balanceig de carga ja que les peticions no acaban arriabant totes al mateix servidor.

Campo de clase: IN (internet) MX(servidor de correu)

SOA : camp que serveix per indicar el servidor de noms primarios del domini, correu del contacte administrador i camps de configuració.

NS indica un servidor de noms.

A —> IPv4 AAAA—> IPv6

CNAME indica un alias

Header(12bytes) El tipo de missatges que enviem.
Question(Variable) Que resolem?
Answer(Variable) Resposta
Authority(Variable) Autoritats que hem Hagut de contactar per poder arribar a aq resolució
Additional(Variable) Camp amb informació adicional que sol contenir les IP's de les autoritats.

Identification(16) perq es pugui relacionar amb la resposta de la pregunta	**Flags(16)** QR(indica si es preg o resp) AA(resposta autoritativa) RD(petició recursiva)
#Questions(16)	**#Answers(16)**
#Authorities(16)	**#Additional(16)**

SMTP —> Usat per enviar correus, mitjançant el port **25 per TCP.** Les direccions de correu es componen d'un nombre seguit d'una @ i el nombre de domini on s'allotja el servidor de correu. **MUA—> MAIL USER AGENT**, encarregat d'enviar el correu desde la màquina client al servidor de correus. **MTA —> MAIL TRANSFER AGENT,** encarregat de la comunicació entre servidors de correu. La comunicació es realitza amb el MUA enviant el correu al servidor MTA, aquest envia el correu al servidor destí i finalment se li entrega al client destí.

1) El client envia el correu al servidor de correus, un cop arribi aquest identifica la direcció a la que es vol enviar el correu, si es del mateix servidor, la redirigim i no cal DNS, en cas contrari, el servidor de correus realitza una resolució DNS buscant el servidor de correu destí, quan el tinguem, podrem enviar el correu. Es fa ús d'ASCII per codificar-se I funciona així—> HELO: per identificar el client SMTP || MAIL FROM: remitent de correu|| RCPT TO: Destinatari de correu|| DATA: Missatge de correu || QUIT: indica que s'ha acabat l'enviament del correu|| OK —> 250.

IMAP/POP3/HTTPS—> Protocols per accedir a correus.

- **POP3:** Servidor guarda els missatges de l'ususari fins que aquest es connecti, en aquell moment es descarreguen tots els missatges del servidor al equip de l'usuari i es borren del servidor.
- **IMAP:** No cal descarregar missatges i a més ens dona la possibilitat d'agrupar els correus mitjançant carpetes entre altres funcions.
- **HTTP:** Igual que IMAP per tractar missatges però s'accedeix des d'un client HTTP.

MIME

Mecanisme usat per poder realitzar intercanvis d'informació que no sigui només text en els correus electrònics. Podent enviar així audios, fotos,videos.. S'usa un delimitador (**boundary**) per delimitar cada part del missatge I d'aquesta forma tenim diferents tipus de fitxers a enviar.

HTTP

Protocol usat per realitzar la comunicació a la web per poder realitzar transferències o visualitzar la informació a la web. Basat en el paradigma **client/servidor.** Un client voldrà accedir a una web i aquesta estarà hospedada en un servidor que oferirà el servei web als client.

- **GET —>** Obtenir informació d'una pàgina web

- **Post —>** Poder pujar informació a una pàgina web.
- **Head —>** Mateix que el GET, però no retorna el cos, sinó les capçaleres.
- **Options —>** Per indicar les possibles opcions acceptades per la pàgina web.
- **Put** —> Insertar o reemplaçar un recurs de Xarxa o un objecte.
- **Delete** —> Borra un recurs de la pàgina web
- **Patch**—> Modifica un objecte existent
- **Trace**—> Funciona mitjançant ecos per comprobar coneció
- **Connect**—> Estableix un túnel, utilitzat per implementar SSL

HTML

Lenguaje de marcas basado en etiquetas/elementos/atributos/texto