Capa 7 (model TCP/IP): Nivell d'aplicació

A la capa d'aplicació és on es gestiona la comunicació directa entre els programes executats als respectius extrems, fent servir en cada cas el protocol particular que entenguin ells. Aquesta capa engloba les tres darreres capes del nivell OSI perquè els programes que allà s'hi executen s'han d'encarregar i responsabilitzar de totes les tasques formalment assignades a aquestes tres capes: control de la sessió, codificació de la informació, compressió i formatatge de les dades transferides, etc.

Normalment, en la comunicació duta a aquest nivell un dels programes actua com a "client" i l'altre actua com a "servidor" (encara que no té per què ser així). El "client" és qui realitza peticions d'un recurs (per exemple, un navegador seria un client que demana pàgines web) i el "servidor" és qui comparteix aquest recurs (en aquest cas, seria l'ordinador a qui el navegador li ha demanat la pàgina web, el qual la tindrà guardada al seu disc dur). El "servidor" normalment roman permanentment a l'espera ("a l'escolta") per tal de rebre peticions de clients que li demanin un determinat recurs (moment en el qual l'ofereix si efectivament el té disponible) però el client només cal que estigui funcionant en el moment que necessiti demanar (i rebre) el recurs en qüestió.

A diferència del nivell de transport, on només existeixen uns pocs protocols (TCP, UDP, ICMP...), en el nivell d'aplicació existeixen molts protocols diferents perquè cadascun serveix per facilitar la comunicació entre un determinat tipus d'aplicacions. Dit d'una altra manera, depenent del tipus de tasca a realitzar pel servidor, aquest s´haurà de comunicar amb els clients corresponents utilitzant un determinat protocol, de manera que un client quan demani un recurs, aquest servidor concret el pugui entendre i respondre´l utilitzant el mateix protocol per tal de què el client també entengui la resposta (independentment del sistema operatiu respectiu subjacent). És per això que, a la pràctica, en aquesta capa trobem quasi tants protocols com tipus diferents de programes. Alguns dels més habituals, però, són:

HTTP : Protocol usat pels clients "web" (normalment, navegadors) per, en general, demanar a servidors "web" (Apache, Nginx, ...) la descàrrega de recursos web (pàgines, imatges, etc que el servidor tingui emmagatzemats). Aquest protocol ha de ser entès pels servidors "web" per tal de poder interpretar correctament la petició provinent dels client. Al seu torn, la resposta construïda "ad-hoc" pel servidor oferint el recurs demanat (o indicant algun altre tipus de missatge si això no fos possible) també és enviada fent servir el mateix protocol per tal de què aquesta sigui entesa correctament pel client.

HTTPS : Similar al HTTP però afegeix la capa d'encriptació **TLS** per tal d'aconseguir (mitjançant l'ús de "certificats") que la comunicació entre client i servidor es transmeti de forma segura i no pugui ser espiada ni modificada

> NOTA: La capa d'encriptació TLS formalment se situa entre la capa de transport i la d'aplicació com si fos una "falca". Permet dotar de seguretat (és a dir, de confidencialitat, integritat i autenticació gràcies a diverses tècniques criptogràfiques) a múltiples protocols de la capa d'aplicació que no la incorporen intrínsecament (perquè quan es van dissenyar fa molts anys no es va veure la necessitat). D'aquesta manera, podem tenir HTTPS=HTTP+TLS o SMTPS=SMTP+TLS o POPS=POP+TLS o LDAPS=LDAP+TLS, etc

DHCP : Protocol emprat pel client per fer una petició de tipus DHCP i pel servidor per retornar la resposta adient. Aquest tipus de peticions serveixen per demanar una configuració de xarxa (direcció IP, màscara, porta d'enllaç, servidors DNS a fer servir...). El servidor DHCP haurà d'entendre la petició i respondre convenientment al client (ja sigui assignant-li aquestes dades tal com demanava o bé denegant-se-la, entre altres opcions).

DNS**:** Protocol emprat pel client per fer una petició de tipus DNS i pel servidor per retornar la resposta adient. Aquest tipus de peticions serveixen per preguntar al servidor sobre quin és el nom de màquina associat a una determinada IP (o també al revés: per saber quina és la IP associada a un determinat nom). El servidor DNS haurà d'entendre la petició respondre convenientment al client (ja sigui oferint-li la informació demanada perquè l'hagi pogut obtenir d'alguna manera o bé reenviant la petició a un altre servidor DNS o bé no podent oferir cap dada, etc)

NOTA: Actualment també existeix implementacions de DNS sobre TLS (**DoT**) però també de DNS sobre HTTPS (**DoH**) per, en tot cas, dotar a les consultes i respostes DNS de seguretat (confidencialitat, autenticació i integritat)

NTP : Protocol usat per sincronitzar rellotges entre diferents ordinadors d'una xarxa.

SSH: Protocol usat per, mitjançant un client local, iniciar sessió segura en un terminal d'un servidor remot. Aquest protocol no fa servir TLS perquè ja incorpora "nativament" totes les funcionalitats segures necessàries

VNC: Protocol usat per, mitjançant un client local, iniciar sessió gràfica a l'escriptori d'un servidor remot. Un altre protocol similar però propietari (de Microsoft) és el **RDP**

SMTP/POP/IMAP : Protocols utilitzats durant l'enviament (SMTP) i recepció i accés a bústies (POP o IMAP) d'e-mails

NOTA: Encara que POP i IMAP "serveixen pel mateix", IMAP permet accedir a altres bústies més enllà de la bústia per defecte (l'anomenada "inbox" de correu entrant, que és l'única accessible a través de POP), podent així gestionar accessos a altres carpetes de missatges com ara els esborrats, enviats, etiquetats, els contactes, etc.

FTP**:** Protocol usat per "baixar" i "pujar" fitxers entre un client (el programa usat per l'usuari) i un servidor (que faria de magatzem remot). En desús a favor de ***WebDAV*** (una extensió d'HTTP que permet no només descarregar fitxers sinò també pujar-hi al servidor). D'altra banda hi ha un altre protocol dissenyat per a la transferència de fitxers entre màquines remotes però que és molt més senzill i elemental (no suporta autenticació, és unidireccional, etc) anomenat ***TFTP*** (la "T" ve de "Trivial"), el qual s'utilitza en entorns molt específics.

NFS**:** Protocol usat per compartir carpetes en una xarxa local de forma que el seu contingut estigui accessible, previ muntatge, a les màquines clients, i es pugui manipular de forma transparent com si fossin carpetes locals a cada client. Un protocol alternatiu, dissenyat per Microsoft, és ***SMB (CIFS)***

LDAP: Protocol usat per accedir remotament a dades emmagatzemades en forma d'arbre jeràrquic. Molt sovint utilitzat per mantenir una base de dades d'usuaris i contrasenyes centralitzada per tota la LAN

NOTA: Les especificacions oficials d´aquests protocols les estableix l'organització internacional IETF a través d'uns documents anomenats RFC (http://www.ietf.org/rfc.html). D'altra banda, es pot obtenir un llistat de la majoria de protocols de nivell d'aplicació existents a https://en.wikipedia.org/wiki/Category:Application_layer_protocols).

Tots els protocols de nivell d'aplicació (i, per tant, les aplicacions que els utilitzen, tant de tipus de client com servidor) estan basats en un dels dos protocols del nivell inferior existents: o bé en UDP o bé en TCP. Les aplicacions que utilitzen com a base UDP han sigut històricament relativament poques: les més rellevants són els clients/servidors DHCP, DNS, NTP i TFTP. Actualment, però, l'aparició de tecnologies d'àudio i vídeo en temps real, on els requisits de velocitat i baix retard s'imposen a la fiabilitat i ordre en la entrega, han fet que UDP sigui la base ideal per a RTP (protocol de nivell d'aplicació especialitzat en streaming), VoIP (veu sobre IP) i similars. De totes formes, TCP segueix sent el protocol de nivell de transport més utilitzat amb diferència ja que serveix de base per a la majoria de protocols del nivell d'aplicació, com HTTP (excepte la versió 3, però), POP/IMAP/SMTP, FTP, SSH, etc, etc. L'ampli ús de TCP és degut a què, tal com ja sabem, proporciona mecanismes que garanteixen una transmissió de dades entre clients i servidors sense errors, sense pèrdues, sense duplicacions i amb un control de fluix automàtic (quelcom que UDP no és capaç de fer).

D'altra banda, tots els programes servidors d'un determinat tipus (HTTP, FTP, etc) solen utilitzar un mateix número de port estàndard ja conegut per escoltar i rebre-hi les peticions que arribin dels diferents clients remots (moltes vegades de forma concurrent). El repartiment oficial de números de port estàndard segons el tipus de servidor que hi hagi "al darrera" està gestionat per l'IANA i es pot consultar aquí http://www.iana.org/assignments/port-numbers (una còpia d'aquesta llista també es troba al fitxer "/etc/services"). Bàsicament, en aquesta llista s'estableix que hi ha tres tipus de ports:

-Reservats per processos del sistema (1-1023): Això vol dir que són ports importants que només poden ser "oberts" per programes executant-se com a "root". Entre ells, aquí trobem el port **80** (utilitzat pels servidors HTTP), el port **443** (utilitzat pels servidors HTTPS), els ports **25**, **110** i **143** (utilitzats pels servidors SMTP/POP/IMAP, respectivament), els ports **587, 995** i **993** (utilitzats pels servidors SMTPS/POPS/IMAPS, respectivament), el port **22** (utilitzat pels servidors SSH), el port **21** (utilitzat pels servidors FTP), el port **2049** (utilitzat pels servidors NFS), els port **389** i **636** (utilitzat pels servidors LDAP i LDAPS, respectivament), etc. Tots els ports anteriors són TCP (excepte en el cas de HTTP/Sv3) però també estan els UDP...concretament, els servidors DNS escolten al port **53** UDP (a excepció dels DoT, que en aquest cas escolten al **853** TCP), els servidors DHCP escolten al port **67** UDP, els servidors TFTP al port **69** UDP i els servidors NTP al port **123**, etc.

> NOTA:Aquests ports d'escolta "per defecte" es poden canviar per uns altres, si així es desitja, modificant la configuració del programa servidor en qüestió. De totes formes, això no sol fer-se perquè mantenir la configuració predeterminada permet que els clients de cada servei puguin estar preprogramats per accedir directament al port per defecte en qüestió de forma automàtica. Per exemple, si un usuari només indica a la barra d'adreces d'un navegador (recordem, client HTTP) el nom del servidor HTTP a què vol accedir (per exemple, "elpuig.xeill.net"), el navegador sempre es connectarà al port nº80 d'aquest servidor. En canvi, si l'usuari volgués emprar el navegador per accedir a un altre número de port diferent del nº80, hauria explicitar-lo d'alguna manera (en concret, si volgués accedir a l'port 1234 de la web de centre, a la barra d'adreces de navegador hauria escriure "elpuig.xeill.net:1234").

-Registrats per aplicacions comercials determinades (1024-49151): Per exemple, aquí trobem el port **3306** (utilitzat pel servidor MySQL), el port **5900** (utilitzat pels servidors VNC), el port 3**389** (utilitzat pels servidors RDP), etc.

-Lliures (49152-65535): Són els que utilitzen els clients, ja que és indiferent quin port fan servir en cada moment (de fet, l'elecció és aleatòria, de manera que, per establir una connexió, una vegada un client pot "agafar" un determinat port que estigui lliure -en realitat, li assigna el sistema operatiu- i un altra vegada pot "agafar" un altre).

Lògicament, si un programa client pretén accedir a un port tancat (és a dir, on no hi ha cap programa servidor escoltant), rebrà un error. També es rebrà un error si un tipus de client pretén accedir a un port vinculat a un servidor que no es correspon (és a dir, fer servir per exemple un navegador per accedir a un servidor DNS o un client DHCP per accedir a un servidor web, etc. .), ja que, insistim, tots dos extrems han de ser capaços de comunicar-se emprant el mateix protocol de nivell d'aplicació.