

Encapsulation :

entête MAC	entête IP	data IP		enqueue MAC
		entête UDP	data UDP	
		entête TCP	data TCP	

UDP : user datagram protocol :

Protocole de couche 4 (transport), il utilise des datagrams, comme IP.
 Service sans connexion, qui ne garantit pas que les données arrivent à destination.
 UDP n'a pas d'acquittement, n'assure pas que les données sont dans l'ordre et non dupliquées. C'est la couche supérieure qui doit s'en charger.
 Par contre, UDP effectue un contrôle d'erreur (checksum).
 Applications qui utilisent UDP : RIP, DNS, TFTP, SNMP, IP tunnel, PC-NFS.

Format de l'entête de paquet UDP (chaque ligne comporte 32bits ou 4 octets):

octet 1	octet 2	octet 3	octet 4
port source		port destination	
long. (entête + data) en octets		checksum	

TCP : transport control protocol :

TCP établit et libère les connexions dont les applications ont besoin.
 Une application peut établir plusieurs connexions et plusieurs applications peuvent utiliser en même temps TCP.

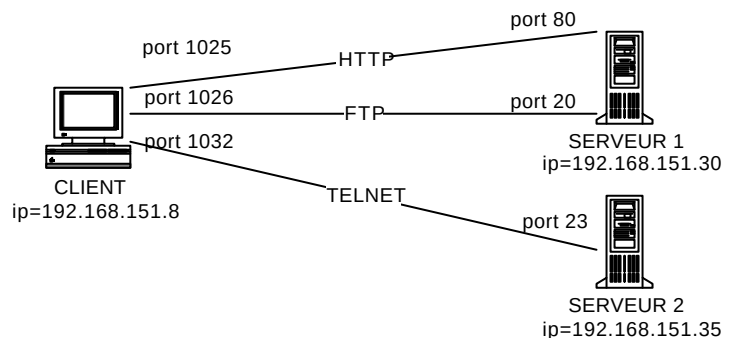
Quand un *client* sur un Host demande l'établissement d'une connexion à une application *serveur* sur un autre Host, cette connexion TCP est référencée de chaque côté par une adresse IP et un numéro de port (comparer le n° de port au n° de quai d'une gare).

Sur le serveur, le N° de port est fixé par l'application :

FTP (data)	20	tcp
FTP (control)	21	tcp
TELNET	23	tcp
TFTP(UDP)	69	udp
HTTP	80	tcp
SMTP	25	tcp
POP3	110	tcp

Sur le client, le port quelconque est décidé à la connexion (>1023).

Une connexion est identifiée par un socket (= adr. IP + n° de port).

**Segmentation :**

TCP segmente les données venant de l'application (la taille des segments est négociée à la connexion), y ajoute une entête gérant des compteurs en émission et réception.

Le client et le serveur possèdent 2 mémoires tampon où ils stockent les segments à émettre et les segments reçus. Les compteurs SEQ et ACK indiquent les adresses des emplacements où sont stockés ces segments.

Chacun peut envoyer des segments en anticipation (avant que le précédent ne soit acquitté), ce qui augmente le débit utile. Le destinataire contrôle le flux en indiquant la taille des tampons libres (fenêtre).

A la réception, TCP contrôle l'absence d'erreur, vérifie qu'il n'y a pas de données perdues, sinon demande la retransmission du seul segment perdu.

TCP remet dans l'ordre les segments et délivre les données originales à l'application.

Format de l'entête de paquet TCP (chaque ligne comporte 32bits ou 4 octets) :

octet 1	octet 2	octet 3	octet 4
port source		port destination	
n° de séquence SEQ			
n° d'acquittement ACK			
offset + 6 bits réservés + contrôles		fenêtre	
checksum		pointeur message urgent	
options éventuelles + bourrage			

n° de séquence	32 bits	indique le N° du 1er octet transmis dans le segment
n° d'acquittement	32 bits	indique le N° du prochain octet attendu par l'émetteur de ce message
offset (déplacement)	4 bits	taille en mots de 32 bits (ou groupe de 4 octets) de l'entête
contrôles	6 bits	URG : la valeur du champ "pointeur message urgent" est significative ACK : la valeur du champ "acquittement" peut être prise en compte PSH : les données doivent être immédiatement transmises à la couche supérieure RST : fermeture de la connexion à cause d'une erreur irrécupérable SYN : ouverture de la connexion FIN : fin de connexion (plus de data à émettre)
fenêtre	16 bits	nombre d'octets que le récepteur peut accepter
pointeur de message urgent	16 bits	indique les octets qui doivent être traités en priorité

ouverture de session TCP :

host A	SYN	ACK	n°SEQ	n°ACK	host B
demande de connexion →	1	0	x		
	1	1	y	x+1	← connexion acceptée
connexion établie →	0	1		y+1	
	0	1			← données
données →	0	1			

- les 3 premières étapes forment l'établissement de la communication
- chaque extrémité communique la valeur initiale (aléatoire) de son compteur SEQ à l'autre, les compteurs sont synchronisés après cette phase
- seul le 1^{er} paquet a le bit ACK à zéro, un firewall peut interdire les ouvertures de session TCP en filtrant les paquets dont le bit ACK=0

libération d'une connexion TCP :

host A	FIN	ACK	n°SEQ	n°ACK	host B
demande déconnexion →	1	1	x		
		1		x+1	← déconnexion acceptée
	1	1	y	x+1	← demande déconnexion
déconnexion acceptée →		1		y+1	

- le paquet avec le bit FIN à 1 indique une demande de déconnexion
- l'extrémité qui reçoit une demande de déconnexion l'acquiesce tout de suite, et puis émet à son tour une demande de déconnexion