



Téléinformatique – Ch. 9

TCP / UDP

Vincent Magnin
vincent.magnin@hefr.ch

Objectifs

- Connaître la différence entre un service avec et sans connexion.
- Connaître l'objectif, les cas d'usages et les limitations des protocoles TCP et UDP.
- Connaître l'établissement et la fermeture d'une session TCP.
- Connaître les notions de « well-known port » et de socket.
- Connaître le principe d'architecture client / serveur.

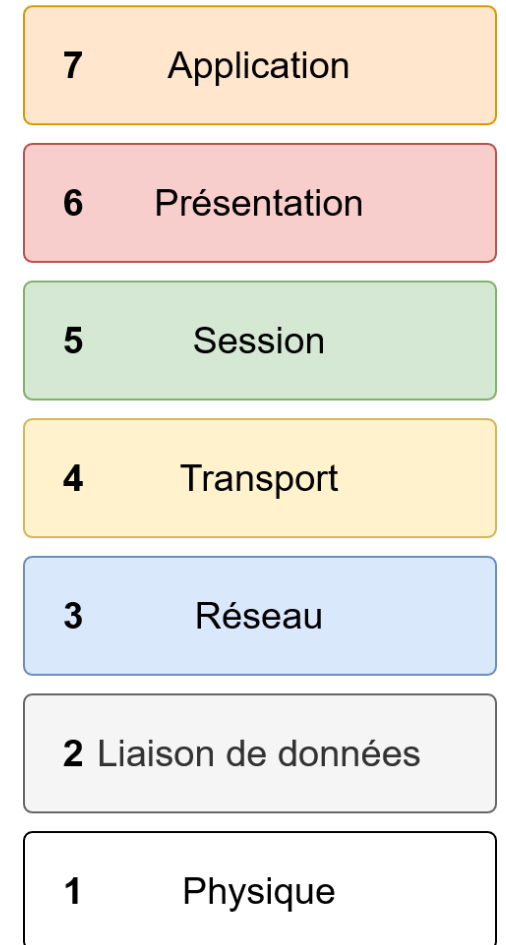
Couche 4 du modèle OSI

La couche « Transport » du modèle OSI sert à fournir un mode de communication de bout en bout pour les programmes applicatifs.

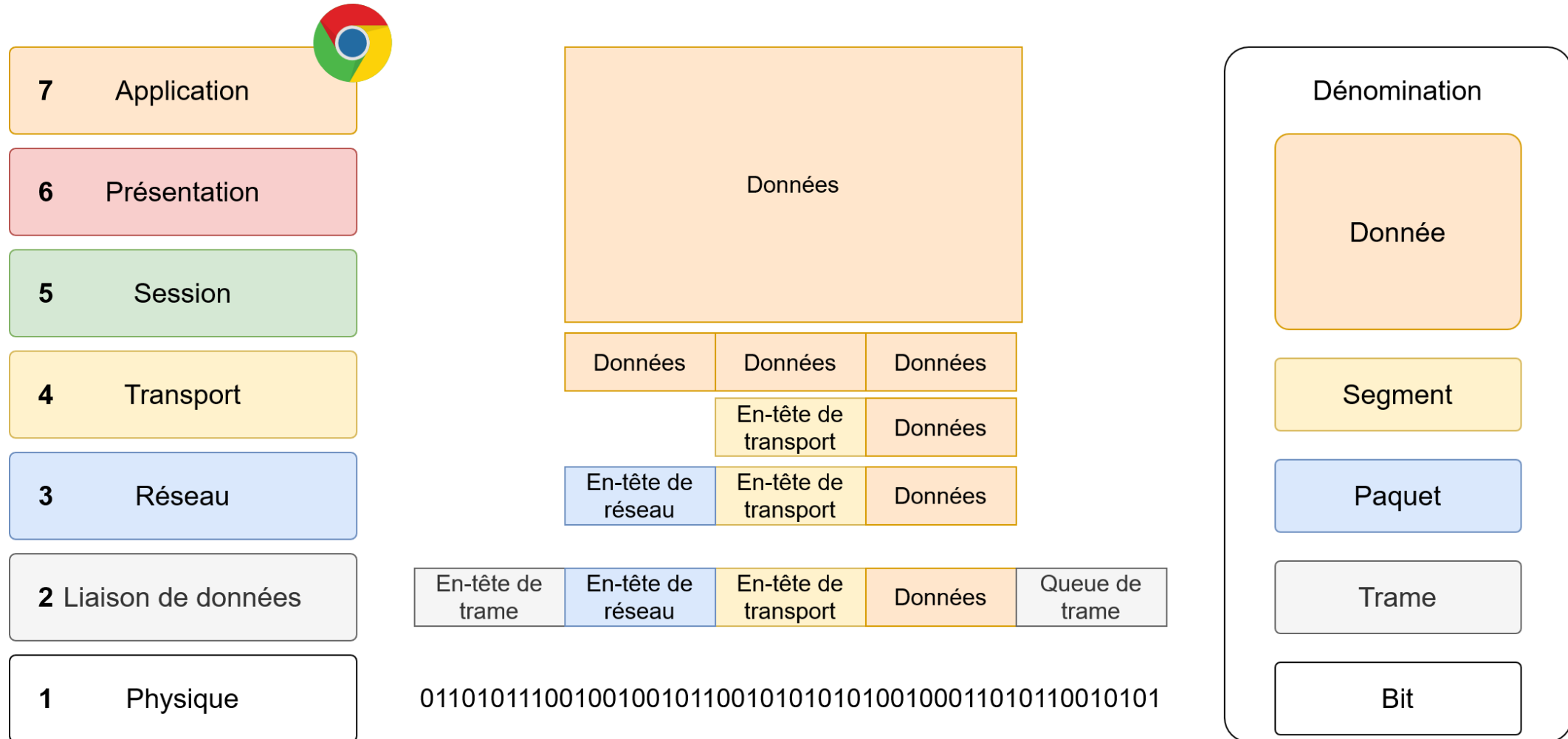
2 protocoles sont principalement utilisés : **TCP** et **UDP**.

On peut considérer que la couche 4 construit un « tunnel virtuel » entre 2 applications qui souhaitent communiquer. Les extrémités de ce tunnel s'appellent des **sockets**.

Modèle OSI



Encapsulation – Rappel



Protocoles TCP et UDP

Les protocoles **TCP** et **UDP** opèrent à la couche **4** du modèle OSI. On dit que ce sont des protocoles de **transport**. Leur objectif est d'assurer un transfert des données **de bout en bout** (d'application à d'application).

En guise de comparaison, le protocole IP s'occupe de transférer des paquets d'un hôte à un autre.

Les protocoles de couche 4 font la distinction entre les différentes applications sur un seul hôte grâce aux **ports**.

Ports

Il est nécessaire d'utiliser un système d'adressage de couche 4 pour identifier les processus qui communiquent. Pour cela, on utilise la notion de **ports**.

Un port est un numéro codé sur 16 bits permettant de distinguer les différentes applications sur une machine. Le couple **@IP:port (socket)** identifie de manière unique une application sur une machine.

On peut ainsi avoir plusieurs processus du même type qui sont exécutés sur la même machine et les différencier grâce à leur port.

Comme il y a 2 protocoles, on distingue 2 types de ports : les ports TCP et les ports UDP.

Ports (2)

Les ports 0 à 1023 sont les **well-known ports**. Ils sont réservés et associés à des applications particulières.

Quelques exemples de ports connus :

- 80/tcp : HTTP
- 443/tcp : HTTPS
- 53/udp : DNS

Les ports sont attribués par le système d'exploitation de chaque machine.

Structure client / serveur

La plupart des requêtes sur Internet s'effectue sur le modèle **client / serveur (C/S)**. Le principe est simple : un client initie une requête vers un serveur, puis reçoit la réponse de ce dernier.

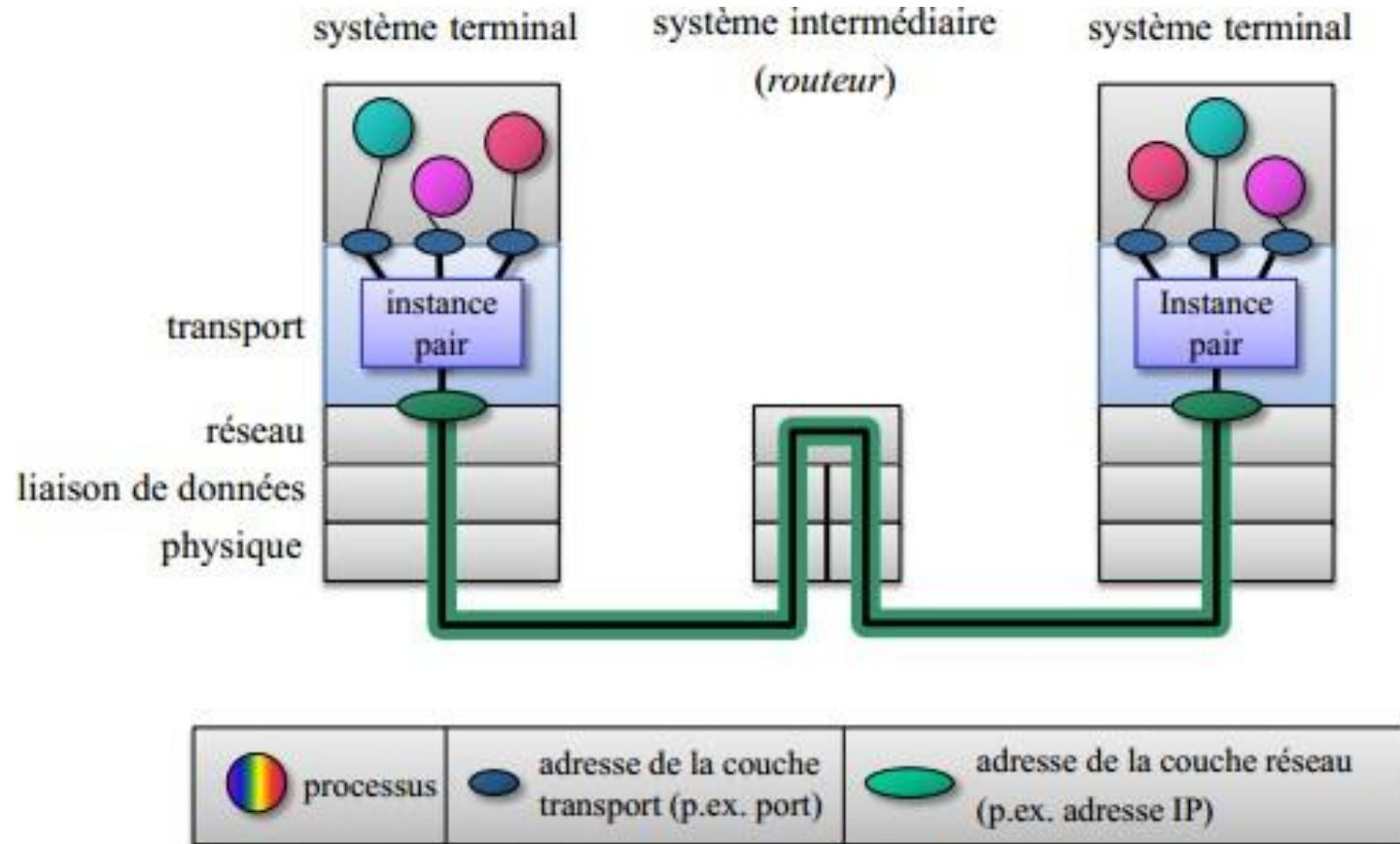
C'est par exemple l'architecture utilisée dans le cadre des protocoles DHCP, DNS, HTTP...

- DHCP : le client demande au serveur de lui attribuer une adresse IP.
- DNS : le client demande au serveur l'adresse IP liée au nom de domaine « heia-fr.ch ».
- HTTP : le client demande au serveur de lui envoyer le contenu de la page web « rts.ch/index.html ».

Il arrive souvent que **le client initie la connexion** avec un **port source aléatoire**, tandis que le port de destination peut appartenir aux well-known ports. Exemple :

- Un client initie une requête HTTP vers un serveur web. L'OS du client choisit un port disponible, par exemple 59671, et il sait que le serveur est prêt à accepter une connexion sur le port 80.

Vision globale



Connectionless / Connection oriented

Les échanges de couche 4 peuvent être réalisés de 2 manières :

1. Sans connexion (*connectionless*)
 - Envoi de datagrammes individuels
 - Service sans garantie contre
 - Les pertes de donnée
 - Les duplications
 - Les pertes de séquences
2. Orienté connexion (*connection oriented*)
 - Etablissement d'une connexion avant le transfert des données
 - Livraison correcte des données avec séquence des données garantie par le protocole

Protocole UDP

Le protocole **UDP** (**User Datagram Protocol**) est le plus simple des 2 protocoles. Il fournit un service minimal.

Il propose un mode de communication bout en bout pour les programmes mais il est **non fiabilisé** et **non connecté**.

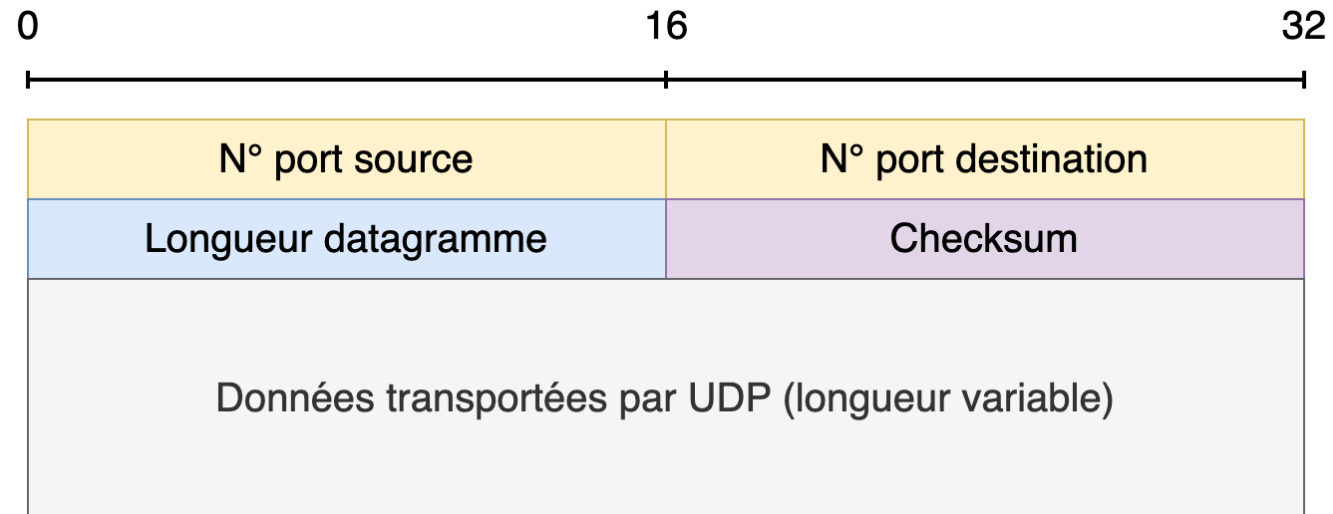
Il n'y a aucune « prise de contact » avec la machine distante avant d'envoyer les données. De plus, il n'y a aucune garantie que les données soient correctement acheminées. Les données sont appelées des **datagrammes UDP**, et ils sont tous indépendants les uns des autres.

Protocole UDP (2)

Lorsque l'on envoie des données avec UDP, on ne peut pas savoir si elles ont été correctement reçues.

Si plusieurs envois successifs sont faits vers le même destinataire, il est possible qu'ils soient reçus dans un ordre différent de celui d'envoi. UDP **ne remet pas en ordre** les datagrammes.

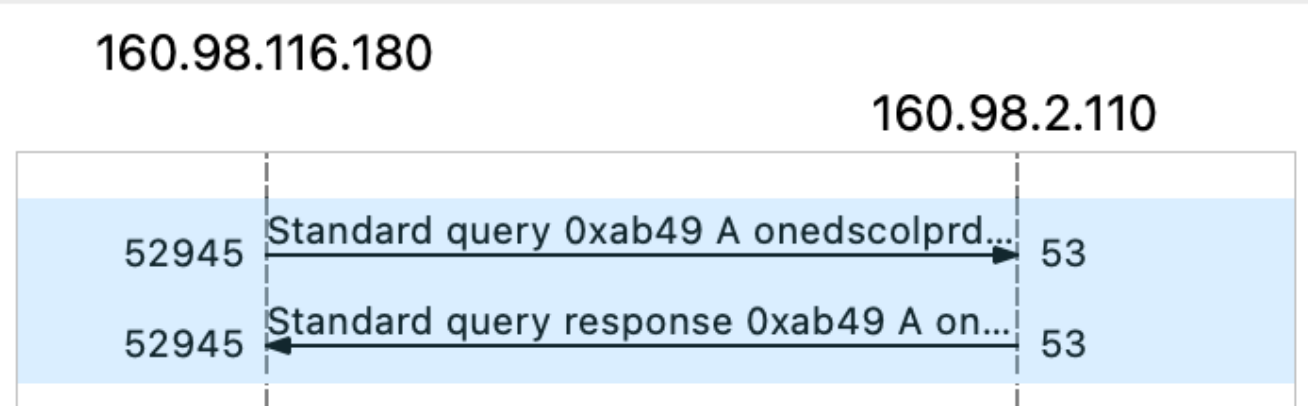
UDP est plus rapide que TCP, mais beaucoup moins fiable.



Protocole UDP (3)

UDP est le protocole très utilisé pour les applications temps réel (VoIP, vidéo-conférence...). Dans ce cas, on préfère souvent perdre un datagramme plutôt que d'attendre sa retransmission.

Exemple d'un échange UDP (requête et réponse DNS) :



Protocole TCP

Le protocole **TCP** (**Transmission Control Protocol**) est le deuxième protocole de couche 4. Il offre un service de communication de bout en bout pour les processus applicatifs.

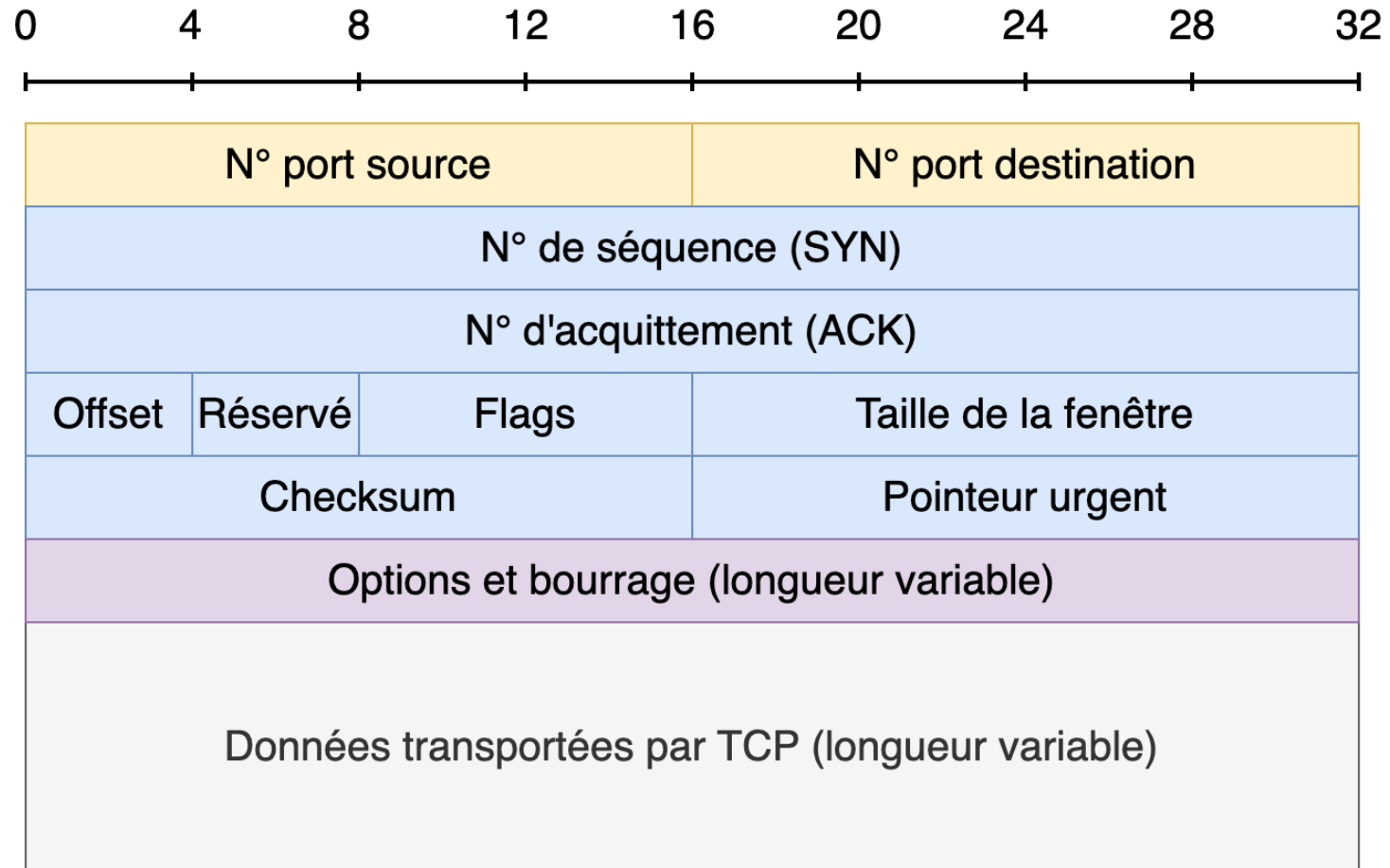
Les communications TCP sont :

- Connectées : avant l'envoi des données, on prend contact avec l'extrémité distante.
- Fiabilisées : il y a un mécanisme d'acquittement qui permet d'acquitter tous les octets reçus. Si un octet (respectivement un segment TCP) a été mal reçu, on peut demander à le renvoyer.

Les étapes d'une connexion TCP sont :

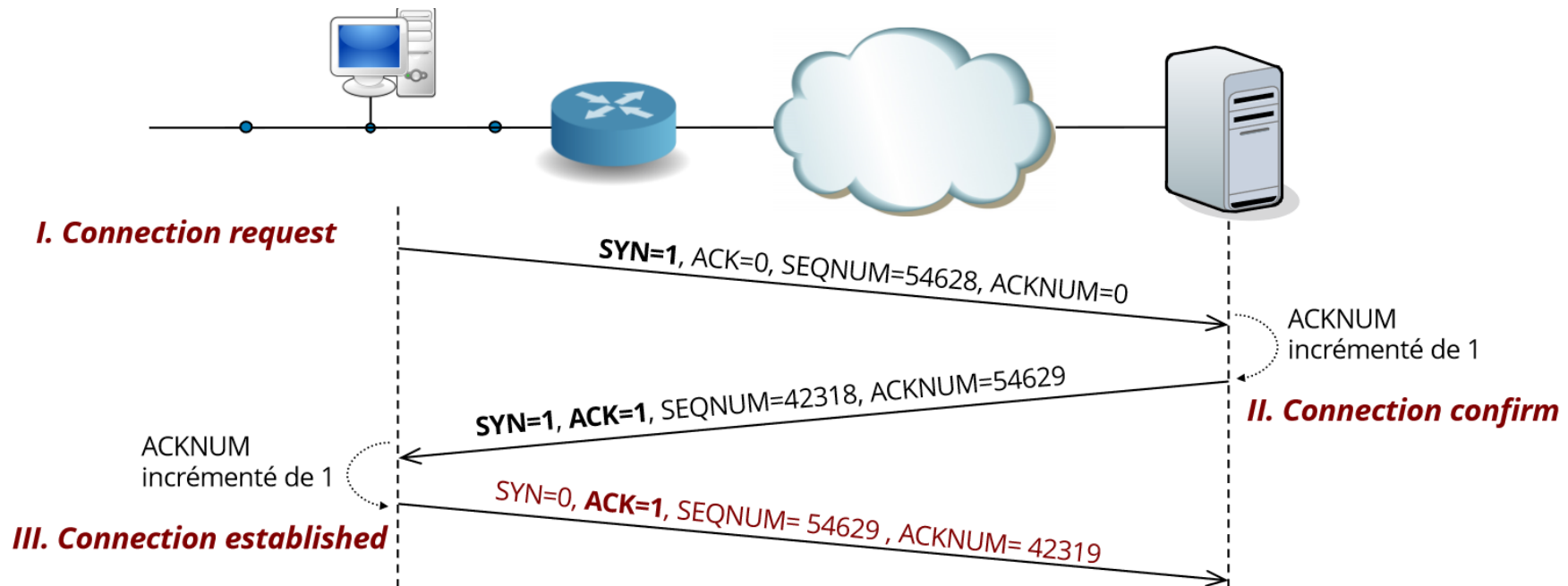
1. Etablissement de la connexion.
2. Transfert des données.
3. Libération de la connexion.

Protocole TCP (2)



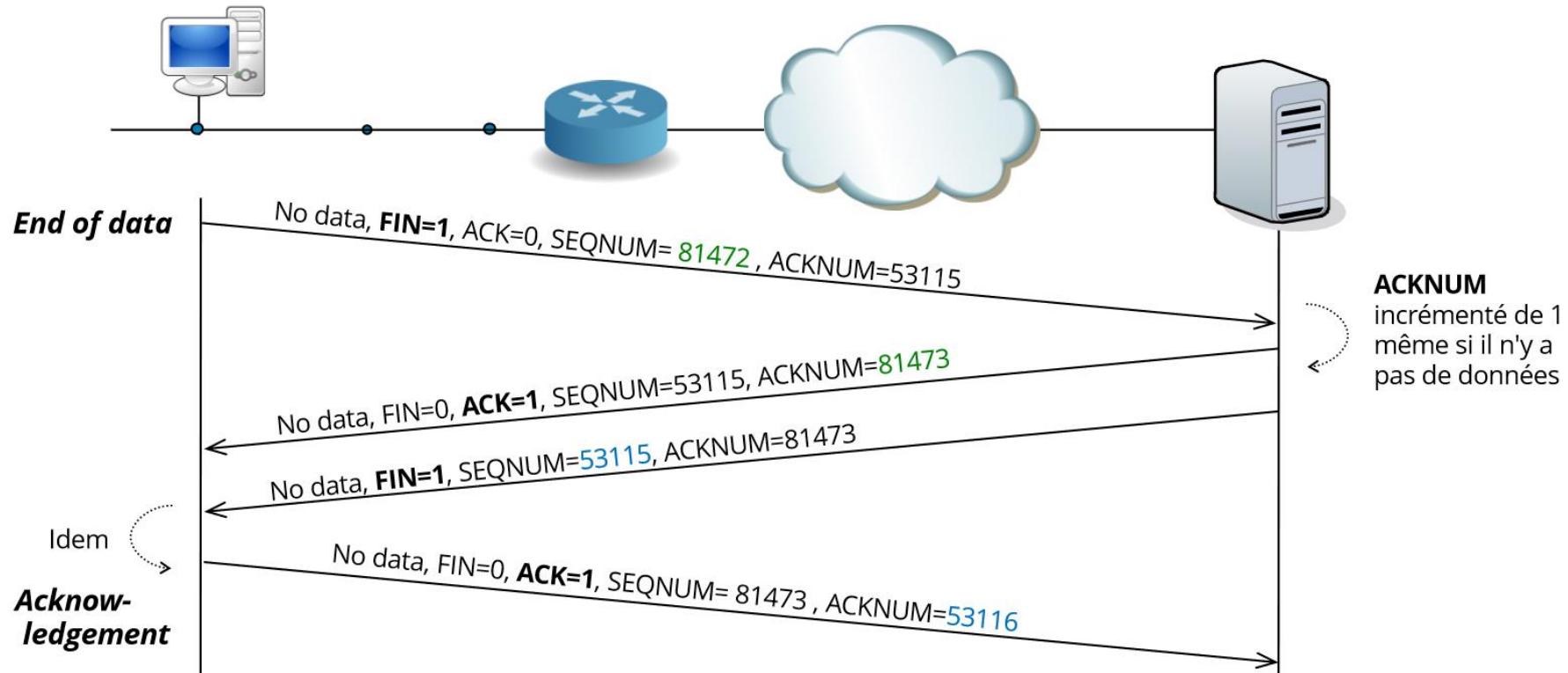
Protocole TCP (3)

TCP étant un protocole **connecté**, il est nécessaire d'établir une connexion avec le destinataire avant d'envoyer des données. Cet échange s'appelle le **three-way handshake**.



Protocole TCP (4)

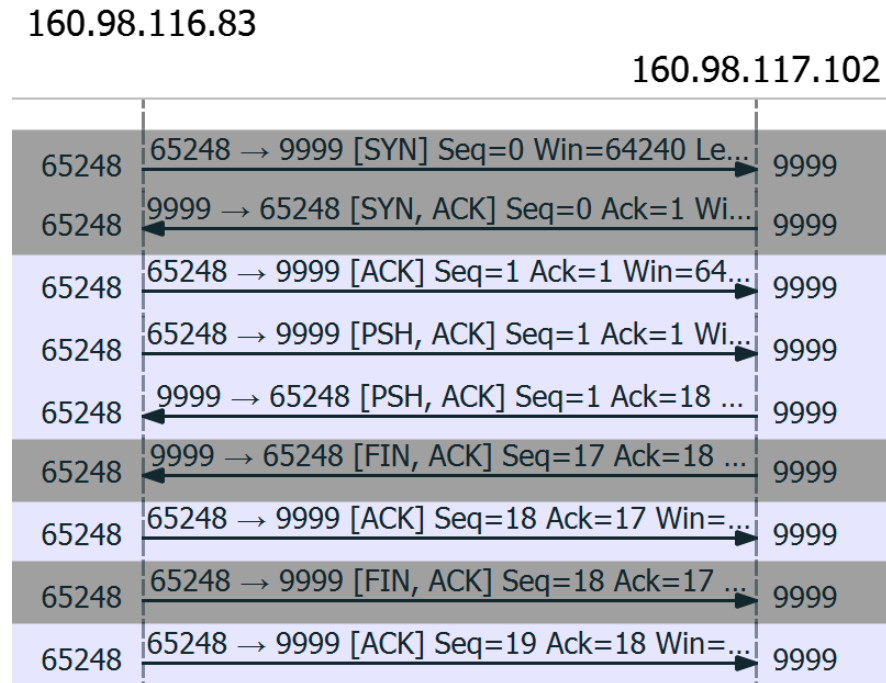
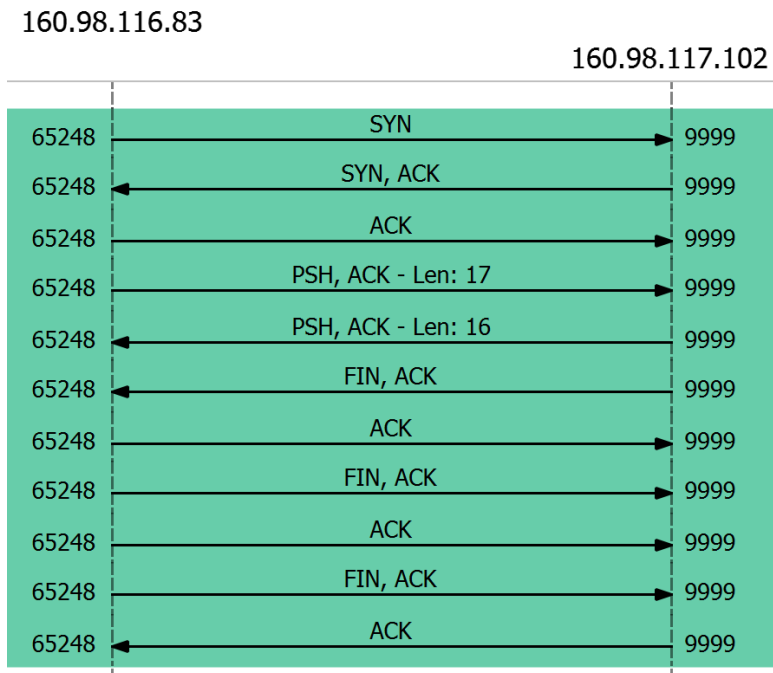
Après avoir transféré les données, la connexion TCP doit être terminée proprement. Cet échange s'appelle le **four-way handshake**.



Protocole TCP (5)

TCP est très utilisé dans de nombreux protocoles, lorsque la fiabilité de la transmission des données doit être assurée.

Exemple d'un échange complet TCP :



Protocole TCP (6)

Les fonctionnalités résumées de TCP sont les suivantes :

Transfert de données ordonné

- Le destinataire remet les données en séquence

Retransmission de paquets perdus

- Retransmission des données non acquittés d'après un timeout

Détection d'erreurs

- Checksum

Contrôle de flux

- Adaptation de la vitesse d'émission des données par rapport à la vitesse de réception du destinataire
- Exemple : Sliding Window

Contrôle de congestion

- Evite la saturation et le ralentissement du réseau
- Exemples : Slow Start, Congestion Avoidance, Fast Retransmit et Fast Recovery

Comparaison

Les protocoles sans connexion comme UDP fournissent un service plus rapide que TCP mais moins fiable. C'est idéal pour les applications où la rapidité (temps réel) est plus importante.

→ VoIP, conférence, etc.

Pour des transferts de données, il est préférable d'utiliser un mode de transmission sûr avec le protocole TCP.

Quelques exemples d'utilisation :

- TCP : HTTP, FTP, SMTP...
- UDP : streaming, DNS, SNMP...

Références

- Ancien cours « Téléinformatique » (G. Waeber, S. Paccard, Q. Vaucher, N. Wirth).
- Technologies de l'Internet avancé (Eric Chotin, IUT Annecy).
- Ancien cours « Téléinformatique » (M. Roch-Neirey).