Cisco Networking Academy
Mind Wide Open

Chapitre 9 :
Couche transport

Présentation des réseaux v5.1

Plan du chapitre

9.0 Introduction

9.1 Protocoles de la couche transport

9.2 TCP et UDP

9.3 Résumé

Section 9.1 : Protocoles de couche transport

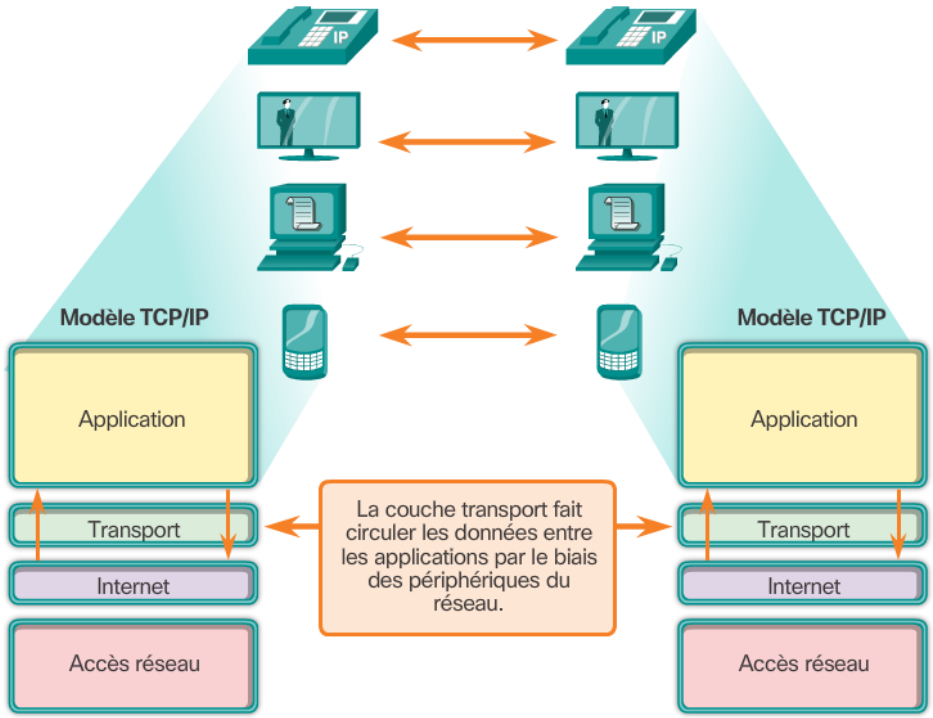
À la fin de cette section, vous saurez :

- Décrire le rôle de la couche transport dans la gestion du transport des données dans une communication de bout en bout
- Décrire les caractéristiques des protocoles TCP et UDP, y compris les numéros de port et leur utilisation

Rubrique 9.1.1 : Transmission des données

Rôle de la couche transport

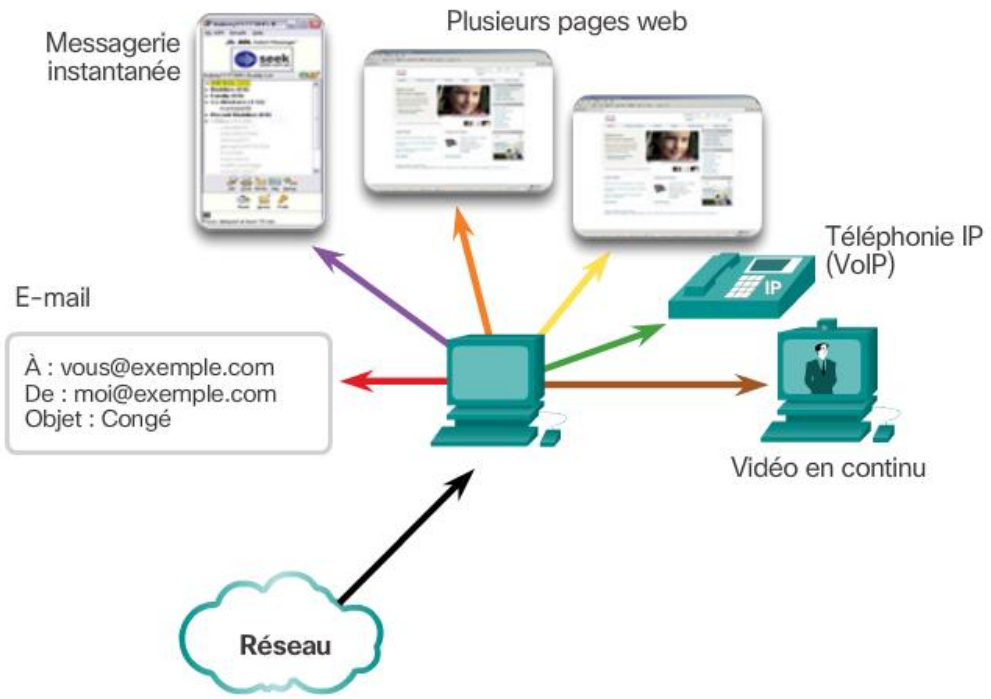
- La couche transport est chargée de l'établissement d'une session de communication temporaire entre deux applications et de l'acheminement des données entre ces deux applications.
- La couche transport offre les services suivants :
  - Prise en charge des flux de données orientés connexion
  - Fiabilité
  - Contrôle de flux
  - Multiplexage



Responsabilités de la couche transport

Suivi des conversations individuelles

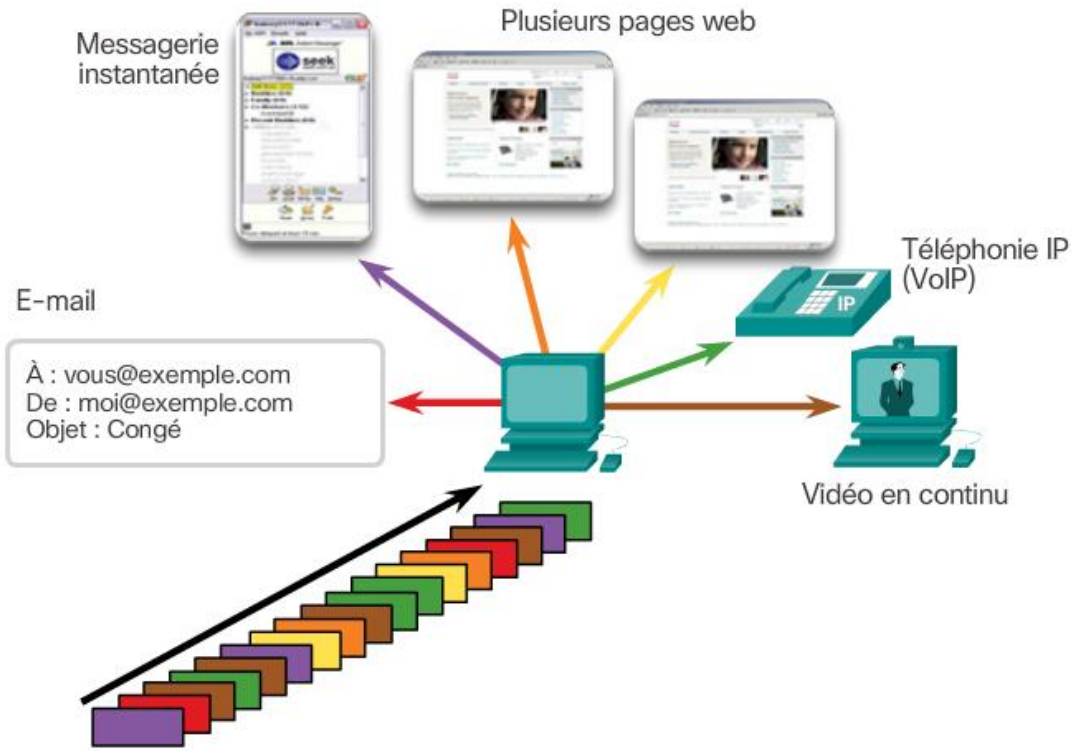
La couche transport suit séparément chaque conversation transmise entre une application source et une application de destination.



Responsabilités de la couche transport (suite)

Segmentation des données et reconstitution des segments

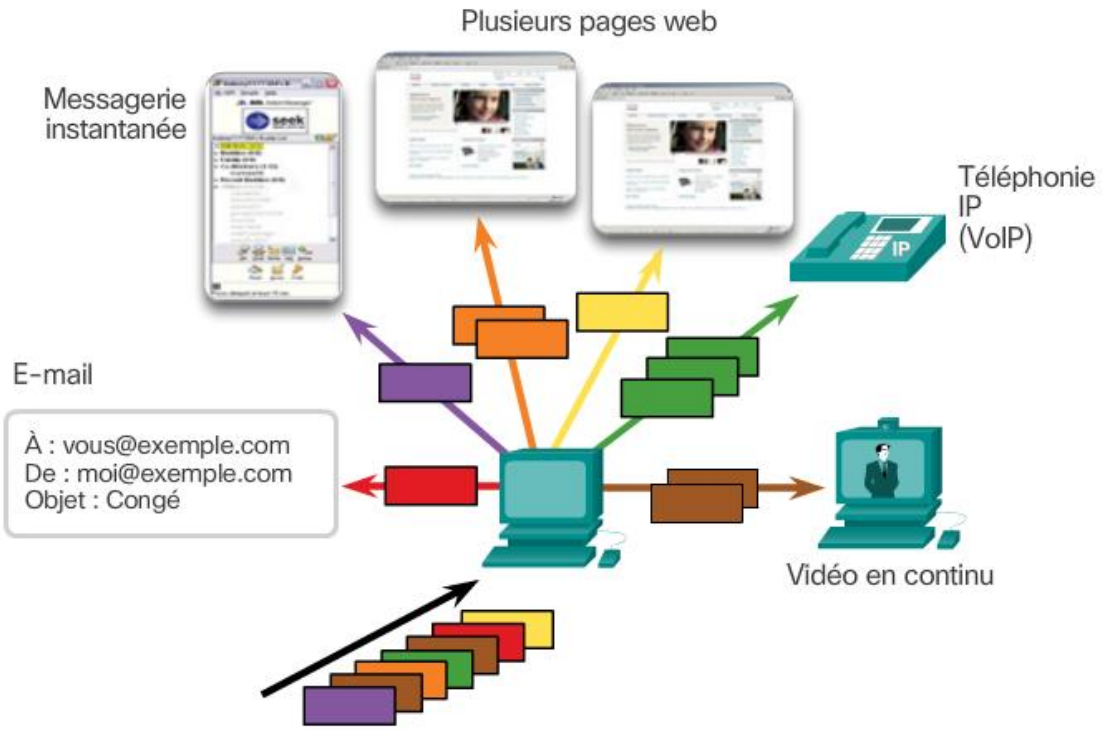
La couche transport divise les données en segments qui sont plus faciles à gérer et à transporter.



Responsabilités de la couche transport (suite)

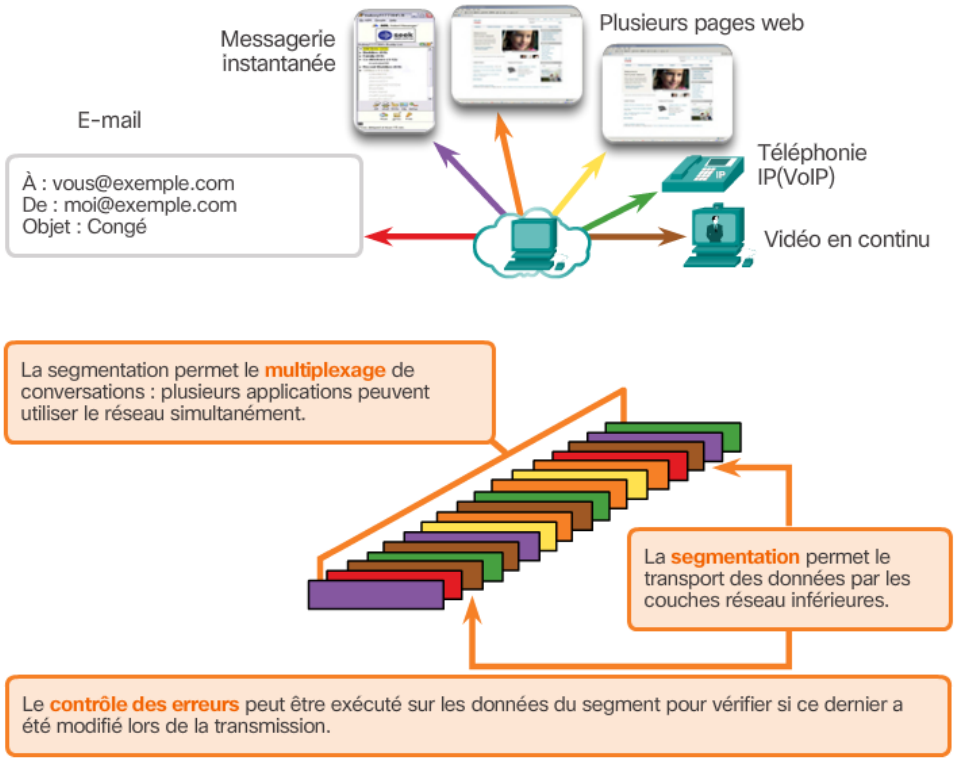
Identification des applications

La couche transport s'assure que même si plusieurs applications sont exécutées sur un périphérique, elles reçoivent toutes des données correctes.



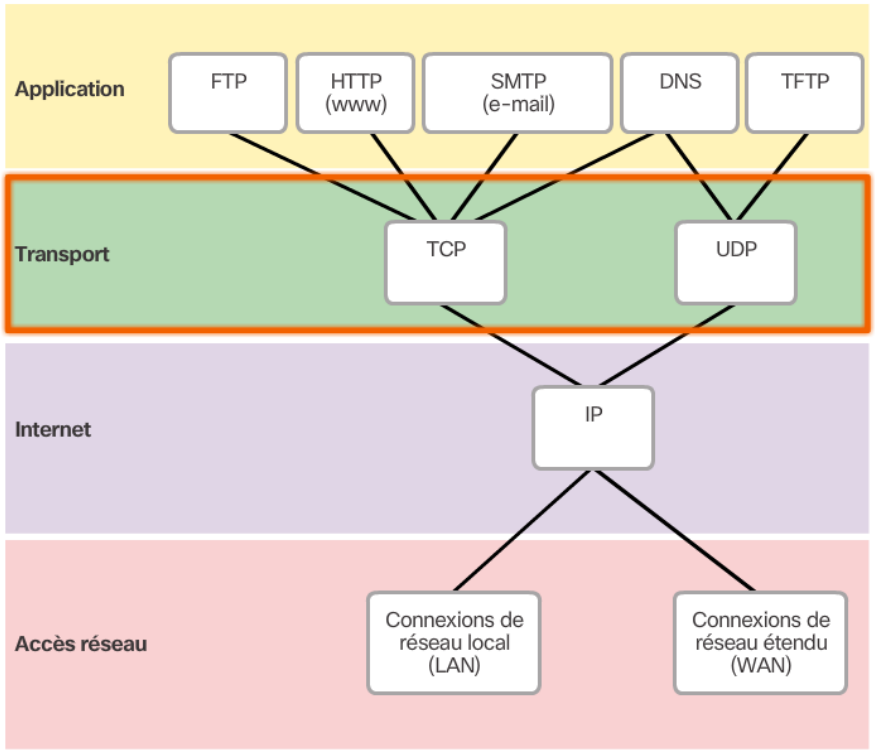
Multiplexage de conversations

- La segmentation des données en parties plus petites permet à plusieurs communications différentes, provenant de nombreux utilisateurs, d'être imbriquées (multiplexées) sur le même réseau.
- La couche transport ajoute un en-tête qui contient des données binaires pour identifier chaque segment de données et permettre à divers protocoles de couche transport d'exécuter différentes fonctions relatives à la gestion de la communication de données.



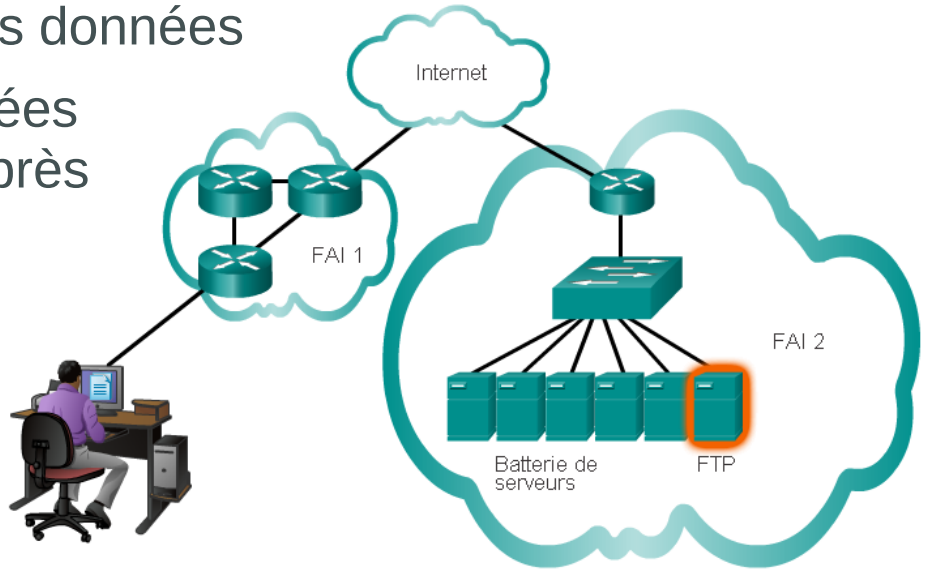
Fiabilité de la couche transport

- La couche transport est également responsable de la gestion de la fiabilité.
- Certaines applications ne nécessitent pas de fiabilité. Les besoins de la couche transport varient d'une application à une autre.
- La suite TCP/IP fournit deux protocoles de couche transport : TCP (Transmission Control Protocol) et UDP (User Datagram Protocol).
- Le protocole IP utilise ces protocoles de transport pour permettre aux hôtes de communiquer et de transmettre des données.
- TCP est considéré comme un protocole de couche transport fiable, riche en fonctionnalités et qui permet de confirmer la remise des données de paquet.
- En revanche, le protocole UDP est un protocole de couche transport très simple qui ne permet pas de garantir la fiabilité.

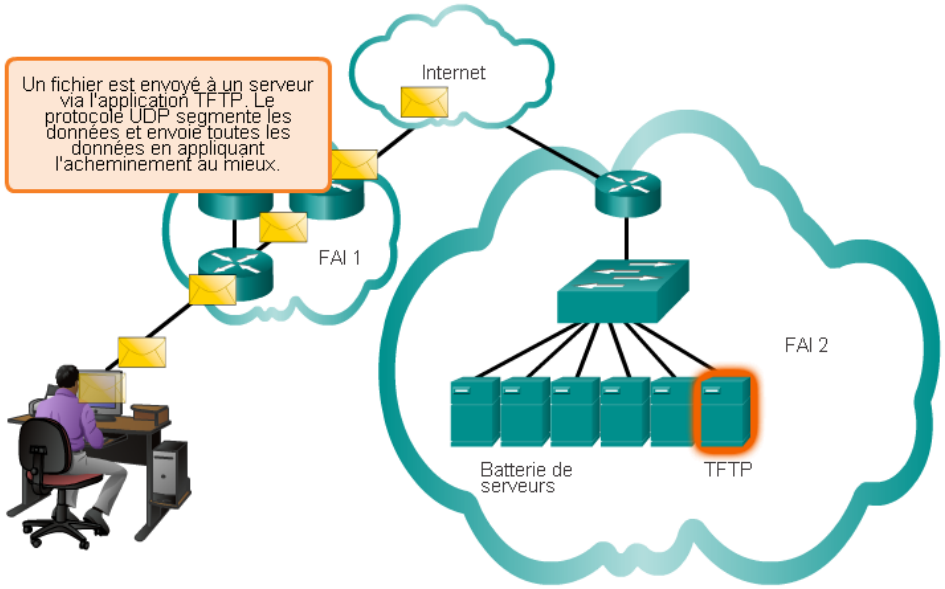


TCP

- La transmission de données via le protocole TCP est fiable dans la mesure où ce dernier prend en charge la confirmation de remise des paquets.
- Trois opérations simples garantissent la fiabilité avec TCP :
  - Le décompte et le suivi des segments de données transmis vers un hôte particulier depuis une application donnée
  - les accusés de réception des données
  - La retransmission des données sans accusé de réception après un certain laps de temps

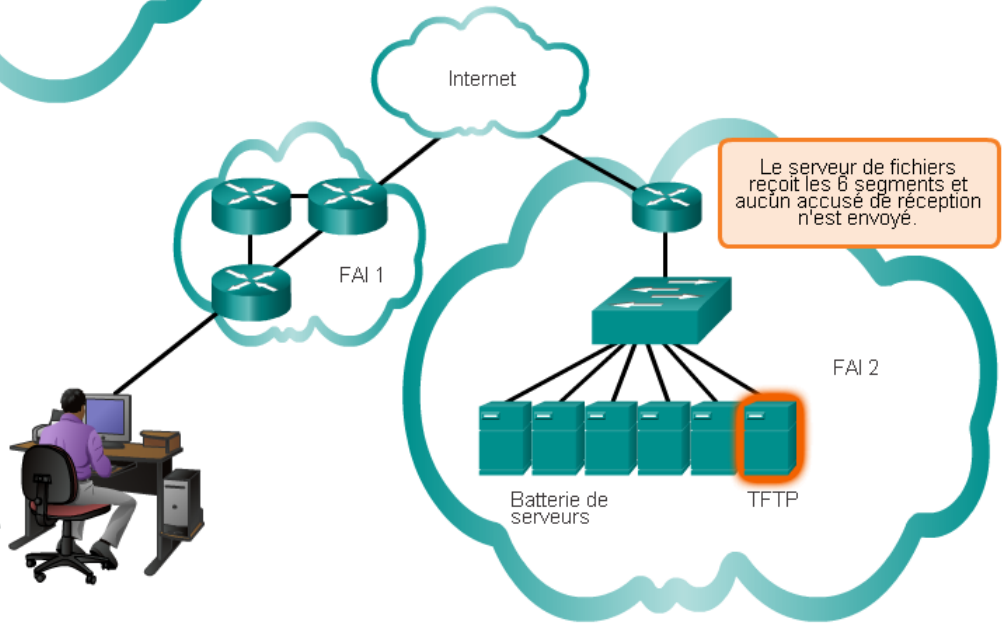


Protocole UDP



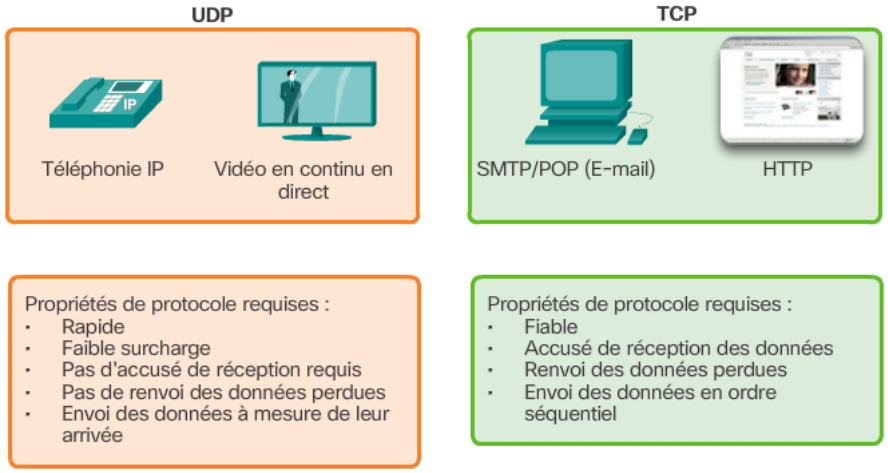
- Certaines applications ne nécessitent pas de fiabilité. La fiabilité engendre une surcharge et d'éventuels retards de transmission.
- L'ajout de cette surcharge pour garantir la fiabilité de telle ou telle application peut réduire l'utilité de l'application et même en détériorer les performances.

- Si la fiabilité n'est pas nécessaire, UDP est un meilleur choix de protocole de transport.
- Le protocole UDP fournit des fonctions de base permettant d'acheminer des segments de données entre les applications appropriées avec peu de surcharge et de vérification des données.



Protocole de couche Transport

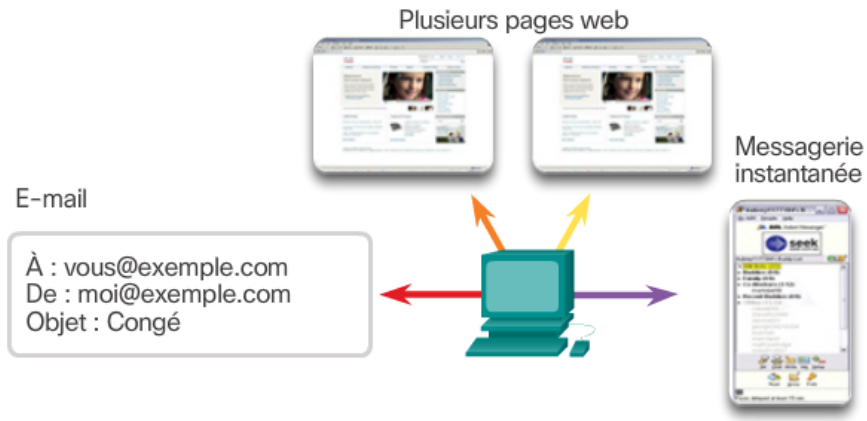
- TCP est un meilleur choix pour :
  - Les applications dont les segments doivent arriver dans un ordre bien précis pour être traités correctement
  - Les applications dans lesquelles toutes les données doivent être entièrement reçues pour être considérées comme étant utiles
- Les applications qui nécessitent TCP : bases de données, navigateurs web, clients de messagerie.
- UDP est un choix mieux adapté aux applications qui peuvent tolérer un certain niveau de perte de données durant leur transmission, mais ne peuvent accepter aucun retard.
- Applications qui utilisent UDP :
  - Streaming audio
  - Streaming vidéo
  - Voix sur IP (VoIP)



Rubrique 9.1.2 :
Présentation des protocoles TCP et UDP

Fonctions du protocole TCP

- Outre la prise en charge des fonctions de base de segmentation et de reconstitution des données, le protocole TCP offre les services suivants :
  - Établissement d'une session
  - Acheminement fiable
  - Livraison dans le même ordre
  - Contrôle de flux



L'**établissement d'une session** permet de s'assurer que l'application est prête à recevoir les données.

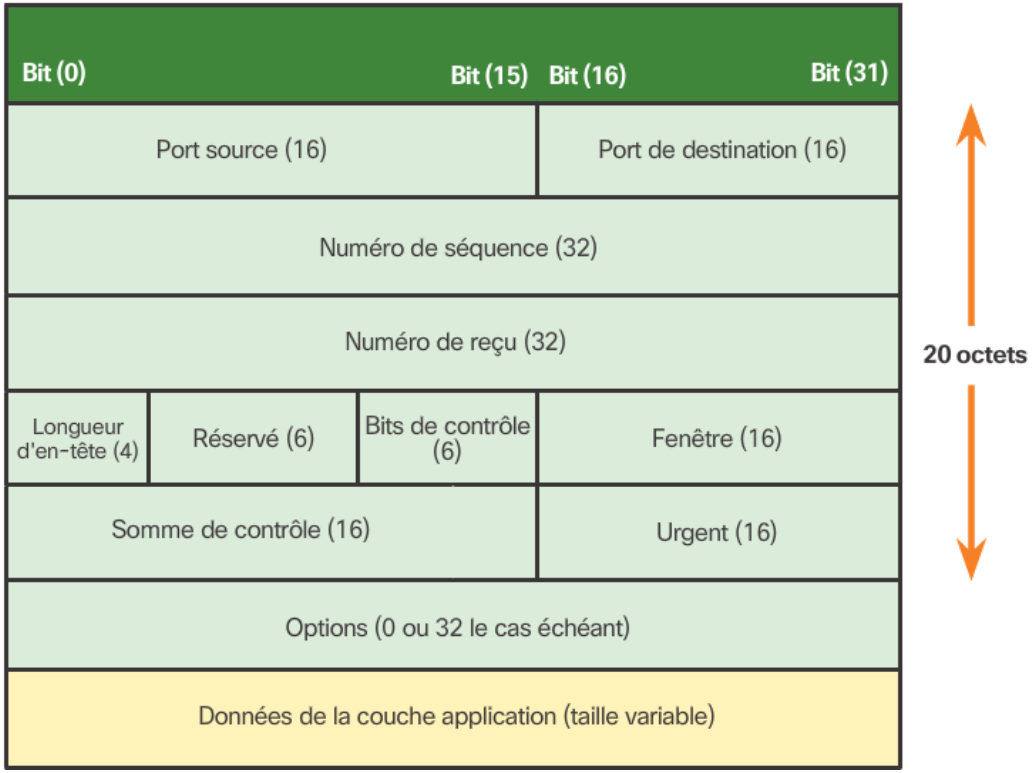
La **livraison dans un ordre défini** permet de s'assurer que les segments sont remis dans le bon ordre.

L'**acheminement fiable** signifie que les segments perdus sont renvoyés afin que les données soient reçues dans leur intégralité.

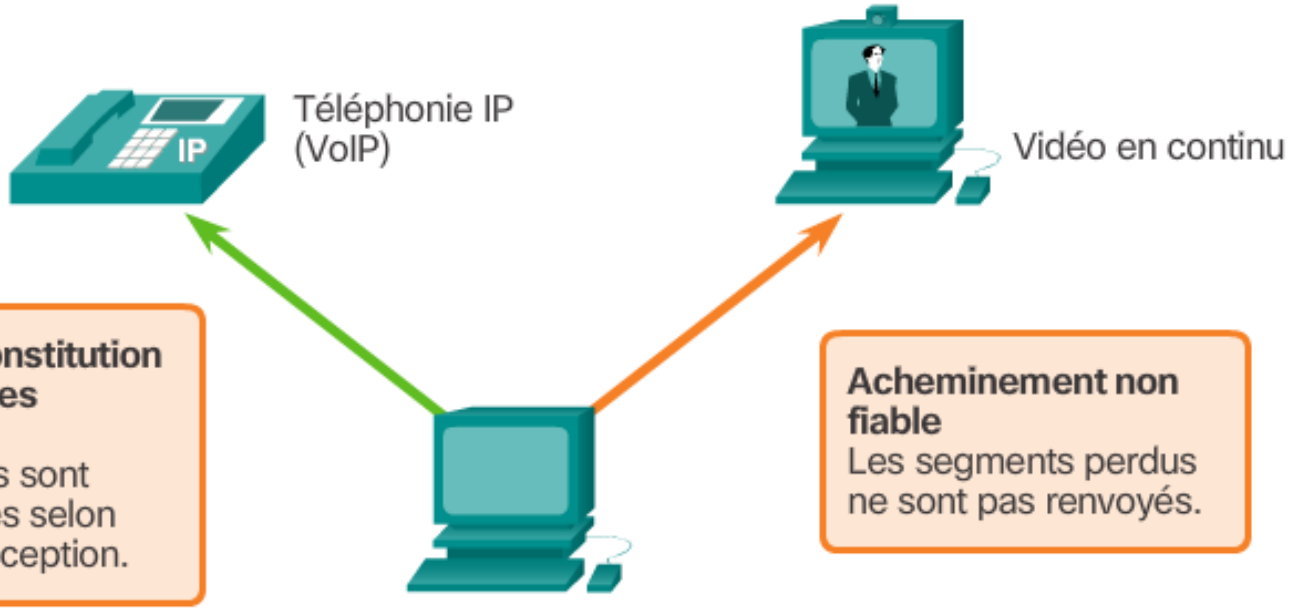
Le **contrôle de flux** permet de s'assurer que le récepteur est capable de traiter les données reçues.

En-tête TCP

- Le protocole TCP est un protocole dynamique. Il suit l'état de la session de communication en enregistrant les informations qui ont été envoyées et celles qui ont été reçues avec un accusé.
- Chaque segment du protocole TCP utilise 20 octets de surcharge dans l'en-tête pour encapsuler les données de la couche application, comme le montre l'illustration.



Fonctions du protocole UDP



Pas de reconstitution ordonnée des données
Les données sont reconstituées selon l'ordre de réception.

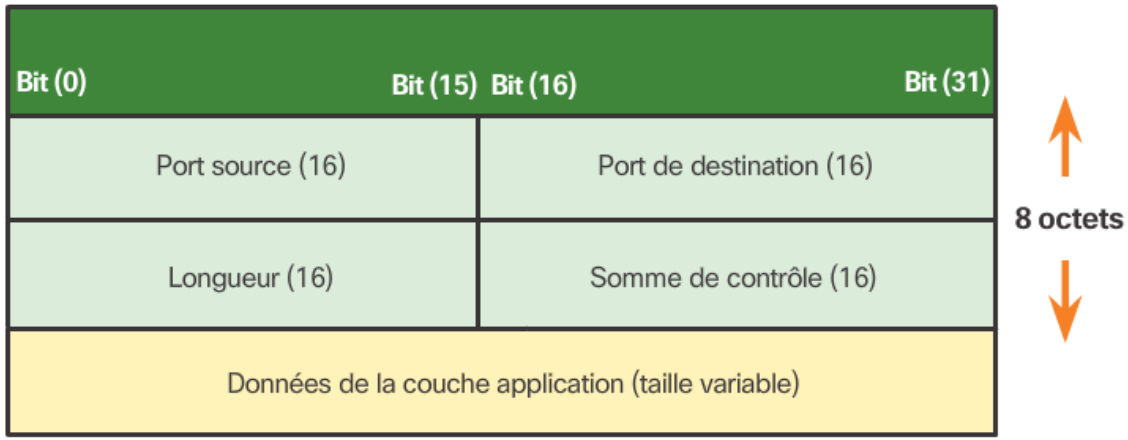
Acheminement non fiable
Les segments perdus ne sont pas renvoyés.

Sans connexion
Pas d'établissement de session.

Pas de contrôle de flux
L'expéditeur n'est pas informé de la disponibilité des ressources.

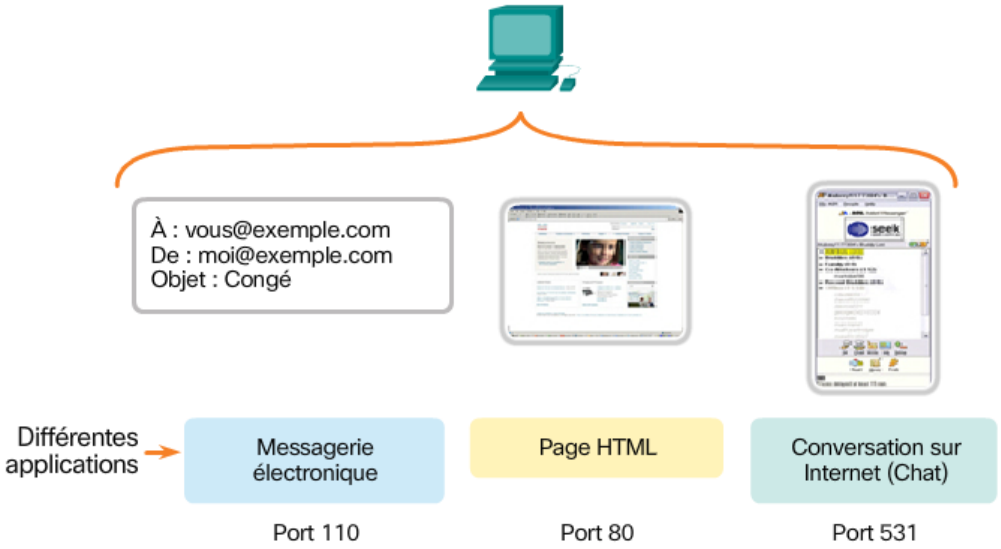
En-tête UDP

- Le protocole UDP est un protocole stateless ou « sans état ». Ni l'expéditeur ni le destinataire ne sont obligés de suivre l'état de la session de communication.
- La fiabilité doit être gérée par l'application.
- Les applications de streaming vidéo et audio doivent rapidement diffuser les données et peuvent tolérer un certain niveau de perte de données. Le protocole UDP est particulièrement adapté à ces applications.
- Les blocs de communications utilisés dans le protocole UDP sont appelés des datagrammes.
- Ces datagrammes sont envoyés « au mieux » par le protocole de la couche transport.
- Le protocole UDP a une faible surcharge de 8 octets.



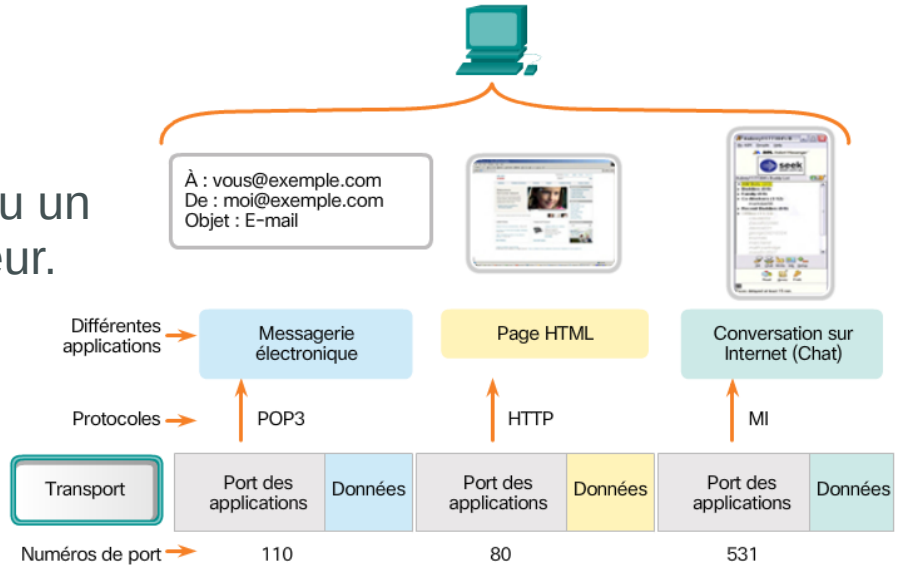
Plusieurs conversations distinctes

- La couche transport doit pouvoir séparer et gérer plusieurs communications ayant différentes exigences de transmission.
- Différentes applications envoient et reçoivent simultanément des données sur le réseau.
- Des valeurs d'en-tête uniques permettent aux protocoles UDP et TCP de gérer plusieurs conversations simultanées en identifiant ces applications.
- Ces identificateurs uniques sont les numéros de port.



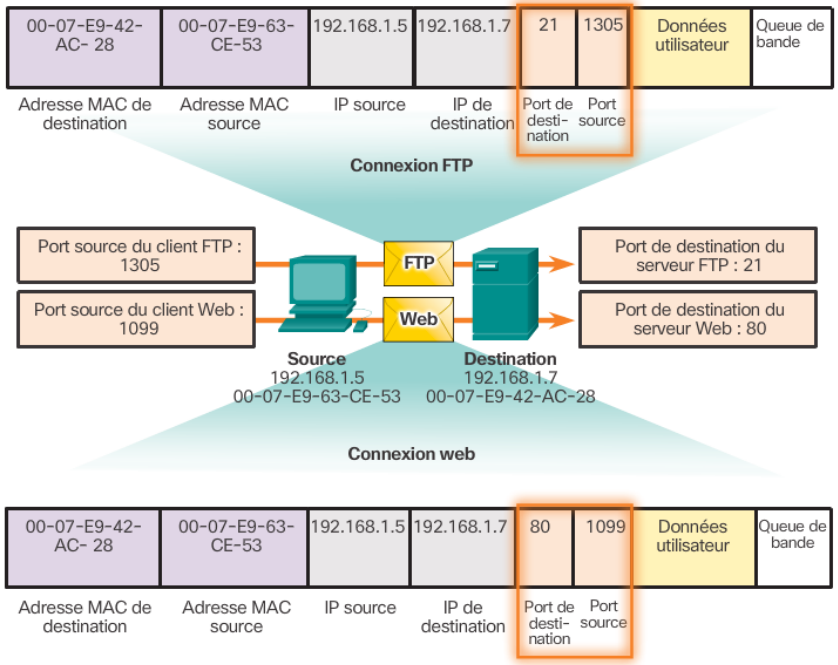
Numéros de port

- Port source
  - Le numéro du port source est choisi de manière dynamique par le périphérique émetteur pour identifier une conversation entre deux périphériques.
  - Un client HTTP envoie habituellement plusieurs requêtes HTTP à un serveur web en même temps. Chaque conversation HTTP est suivie en fonction des ports sources.
- Port de destination
  - Sert à identifier une application ou un service qui s'exécute sur le serveur.
  - Un serveur peut offrir plusieurs services à la fois, c'est-à-dire un service web sur le port 80 en même temps qu'un service FTP sur le port 21.



Paires d'interfaces de connexion

- La combinaison de l'adresse IP source et du numéro de port source, ou de l'adresse IP de destination et du numéro de port de destination, est appelée une interface de connexion.
- L'interface de connexion sert à identifier le serveur et le service demandés par le client.
- Deux interfaces de connexion forment une paire d'interfaces de connexion (192.168.1.5:1099, 192.168.1.7:80).
- Les interfaces de connexion permettent à plusieurs processus actifs sur un client et aux multiples connexions à un processus serveur de se distinguer les uns des autres.
- Le numéro du port source fait office d'adresse de retour pour l'application envoyant la requête.
- Le rôle de la couche transport est de suivre les interfaces de connexion actives.



Groupes de numéros de port

L'Internet Assigned Numbers Authority (IANA) est l'organisme de normalisation responsable de l'attribution de diverses normes d'adressage, notamment des numéros de port.

Numéros de port

Plage de numéros de port	Groupe de ports
De 0 à 1023	Ports réservés
De 1024 à 49151	Ports inscrits
De 49152 à 65535	Ports dynamiques et/ou privés

Numéros de port réservés

Numéro de port	Protocole	Application	Acronyme
20	TCP	Protocole FTP (File Transfer Protocol) (données)	FTP
21	TCP	Protocole FTP (File Transfer Protocol) (contrôle)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Protocole SMTP (Simple Mail Transfer Protocol)	SMTP
53	UDP, TCP	Domain Name Service (service de noms de domaines)	DNS
67	UDP	Protocole DHCP (Dynamic Host Configuration Protocol) (serveur)	DHCP
68	UDP	Protocole DHCP (Dynamic Host Configuration Protocol) (client)	DHCP
69	UDP	Protocole TFTP (Trivial File Transfer Protocol)	TFTP
80	TCP	Protocole HTTP (Hypertext Transfer Protocol)	HTTP
110	TCP	Protocole POP (Post Office Protocol) version 3	POP3
143	TCP	Protocole IMAP (Internet Message Access Protocol)	IMAP
161	UDP	Protocole SNMP (Simple Network Management Protocol)	SNMP
443	TCP	Protocole HTTPS (Hypertext Transfer Protocol Secure)	HTTPS

Commande netstat

- Les connexions TCP inexpliquées peuvent présenter de sérieuses menaces pour la sécurité.
- La commande netstat est un utilitaire de réseau important qui peut servir à vérifier les connexions actives sur un hôte.
- Utilisez la commande **netstat** pour répertorier les protocoles utilisés, l'adresse et les numéros de port locaux, l'adresse et les numéros de port distants, ainsi que l'état de la connexion.
- Par défaut, la commande netstat tente de mapper des adresses IP avec des noms de domaine et les numéros de port avec des applications connues.
- L'option **-n** sert à afficher les adresses IP et les numéros de port dans leur format numérique.

```
C:\> netstat

Active Connections

Proto   Local Address    Foreign Address          State
TCP     kenpc:3126       192.168.0.2:netbios-ssn  ESTABLISHED
TCP     kenpc:3158       207.138.126.152:http     ESTABLISHED
TCP     kenpc:3159       207.138.126.169:http     ESTABLISHED
TCP     kenpc:3160       207.138.126.169:http     ESTABLISHED
TCP     kenpc:3161       sc.msn.com:http          ESTABLISHED
TCP     kenpc:3166       www.cisco.com:http       ESTABLISHED

C:\>
```

Section 9.2 : TCP et UDP

À la fin de cette section, vous saurez :

- Expliquer comment les processus d'établissement et d'interruption de session TCP garantissent la fiabilité des communications
- Expliquer comment les unités de données de protocole TCP sont transmises et comment leur réception est confirmée pour garantir l'acheminement des données
- Décrire les processus client UDP permettant d'établir la communication avec un serveur.
- Comparer UDP et TCP

Rubrique 9.2.1 : Processus de communication TCP

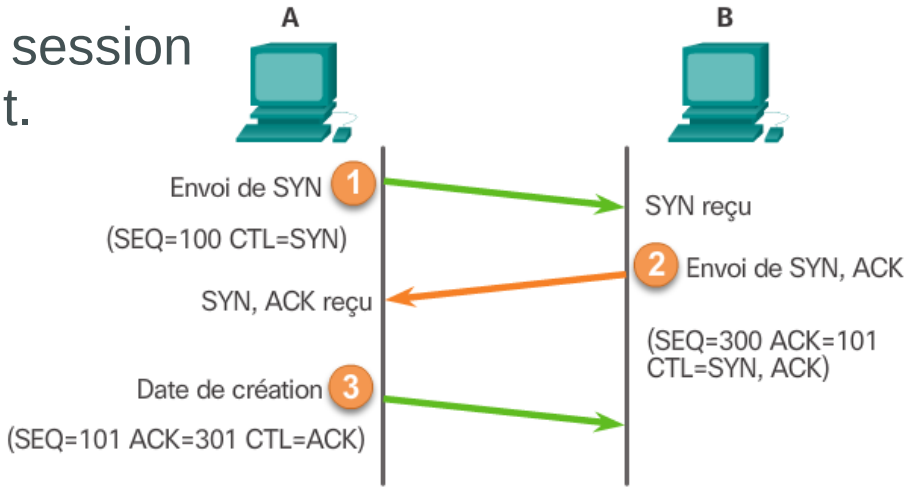
Processus serveur TCP

- Chaque processus d'application actif sur le serveur utilise un numéro de port.
- Deux services ne peuvent pas être affectés au même numéro de port d'un serveur au sein des mêmes services de la couche transport.
- Quand une application de serveur active est attribuée à un port spécifique, on considère que ce port est ouvert.
- Chaque demande de client envoyée à un port ouvert est acceptée et traitée par l'application de serveur liée à ce port.
- De nombreux ports peuvent être ouverts simultanément sur un serveur, chacun étant destiné à une application de serveur active.

Établissement d'une connexion TCP

Une connexion TCP est établie en trois étapes :

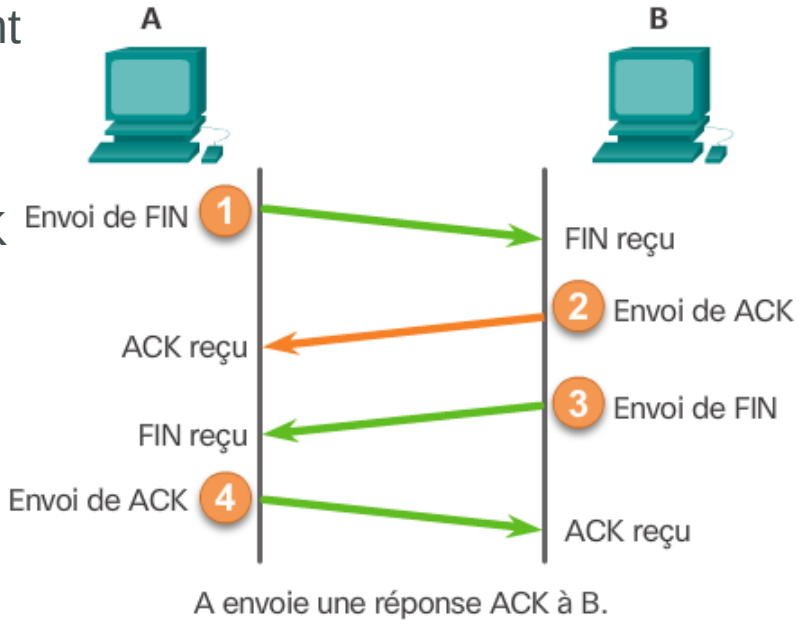
1. Le client demande l'établissement d'une session de communication client-serveur avec le serveur.
2. Le serveur accuse réception de la session de communication client-serveur et demande l'établissement d'une session de communication serveur-client.
3. Le client accuse réception de la session de communication serveur-client.



Fermeture d'une session TCP

L'indicateur FIN TCP permet de mettre fin à une connexion TCP.

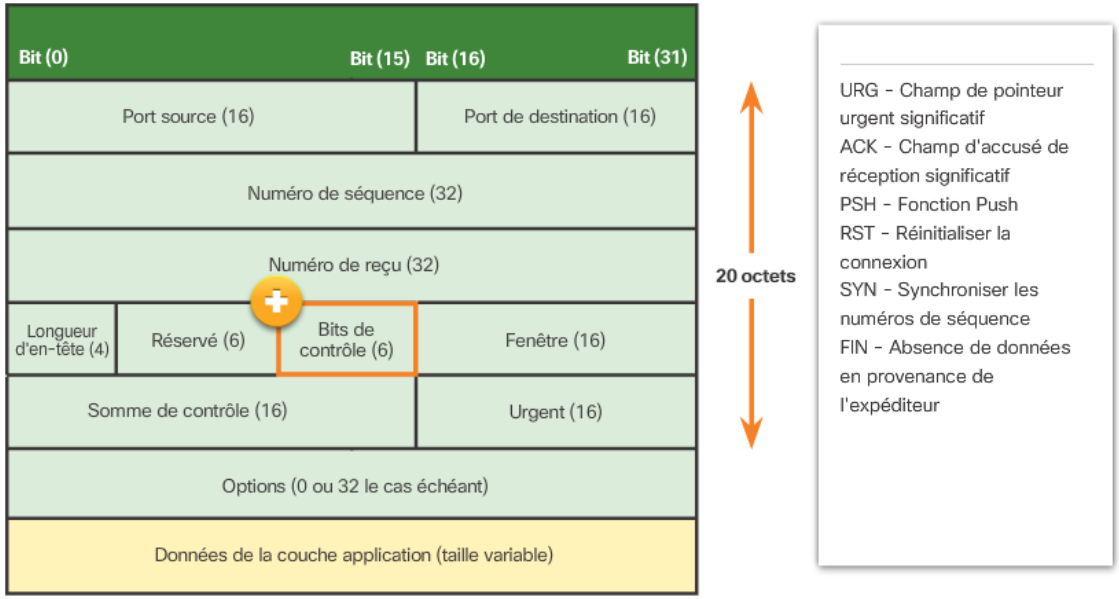
1. Quand le client n'a plus de données à envoyer dans le flux, il envoie un segment dont l'indicateur FIN est défini.
2. Le serveur envoie un segment ACK pour informer de la bonne réception du segment FIN afin de fermer la session du client au serveur.
3. Le serveur envoie un segment FIN au client pour mettre fin à la session du serveur au client.
4. Le client répond à l'aide d'un segment ACK pour accuser réception du segment FIN envoyé par le serveur.
5. Quand la réception de tous les segments a été confirmée, la session est fermée.



Analyse de la connexion TCP en trois étapes

La connexion en trois étapes :

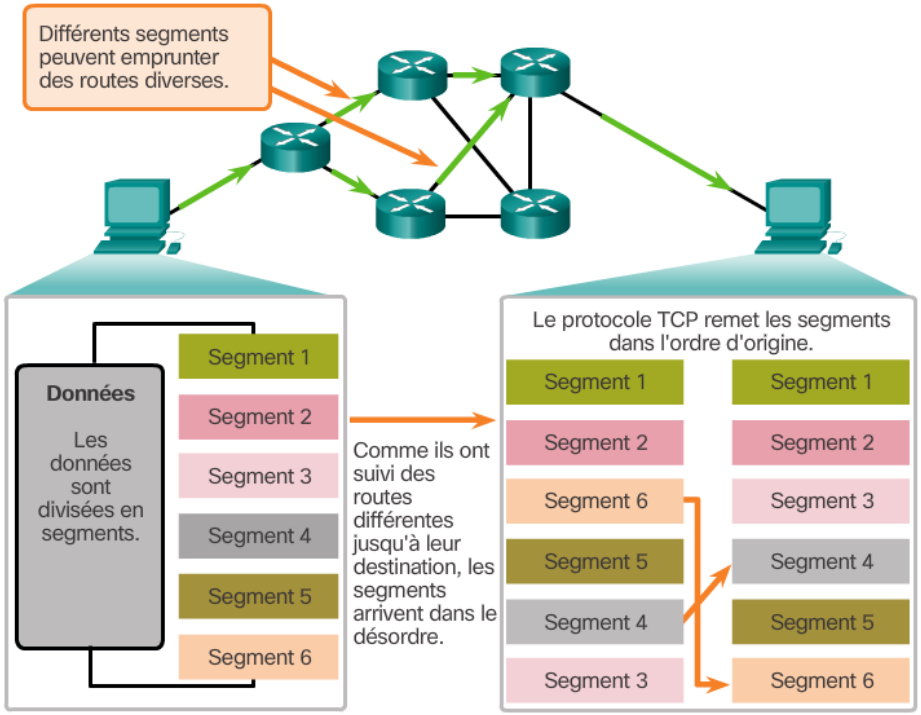
- Vérifie que le périphérique de destination est bien présent sur le réseau
- S'assure que le périphérique de destination a un service actif et qu'il accepte les requêtes sur le numéro de port de destination que le client qui démarre la session a l'intention d'utiliser
- Informe le périphérique de destination que le client source souhaite établir une session de communication sur ce numéro de port



Rubrique 9.2.2 :
Fiabilité et contrôle des flux

Fiabilité du protocole TCP – Livraison ordonnée

- Les segments TCP utilisent des numéros d'ordre pour identifier et accuser réception de chaque segment, suivre l'ordre des segments et indiquer le mode de reconstitution et de réorganisation des segments reçus.
- Un numéro d'ordre initial (ISN) est attribué aléatoirement durant la configuration de la session TCP. L'ISN est ensuite incrémenté du nombre d'octets transmis.
- Le processus TCP récepteur met en mémoire tampon les données de segments jusqu'à ce qu'elles soient toutes reçues et réorganisées.
- Les segments reçus dans le mauvais ordre sont conservés et traités ultérieurement.
- Les données sont remises à la couche application à condition qu'elles aient été entièrement reçues et réorganisées.



Fiabilité du protocole TCP – Numéros d'ordre et accusés de réception

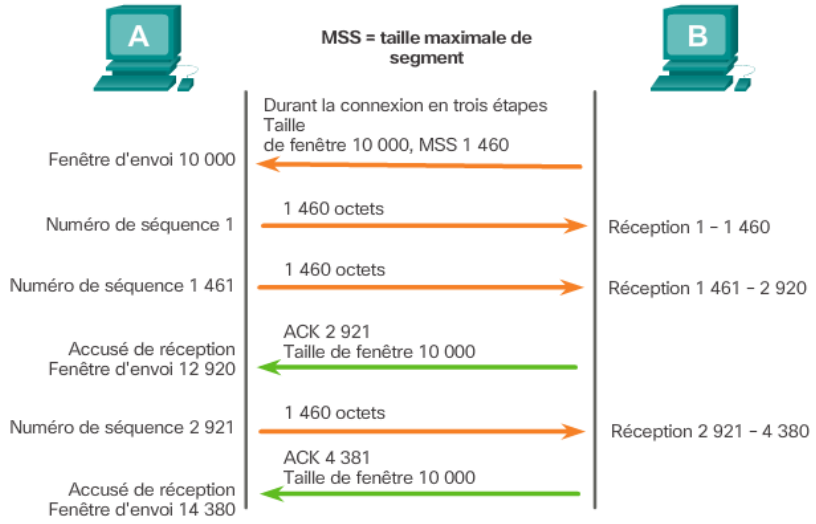
- Le protocole TCP est conçu de sorte à confirmer que chaque segment est bien parvenu à sa destination.
- La configuration des sessions TCP garantit que la destination est non seulement accessible, mais également prête à recevoir des données.
- Le processus TCP sur l'hôte de destination accuse réception des données reçues en provenance de l'application source.
- Le protocole TCP permet la retransmission des segments manqués.
- Il garantit que tous les segments sont correctement réorganisés dès leur réception.
- La fermeture d'une session TCP permet aux parties de progressivement mettre fin à la session TCP une fois qu'il n'y a plus de données à transférer (indicateur FIN).
- Si nécessaire, un terminal TCP peut soudainement mettre fin à une session (indicateur RST).
- La vidéo de la page 9.2.2.2 porte sur les numéros d'ordre et accusés de réception TCP.

Fiabilité du protocole TCP – Perte de données et retransmission

- Le protocole TCP fournit des méthodes de gestion des pertes de segments.
- Citons notamment un mécanisme de retransmission des segments pour les données sans accusé de réception.
- La vidéo de la page 9.2.2.3 porte sur la retransmission TCP.

Contrôle de flux TCP – Taille de fenêtre et accusés de réception

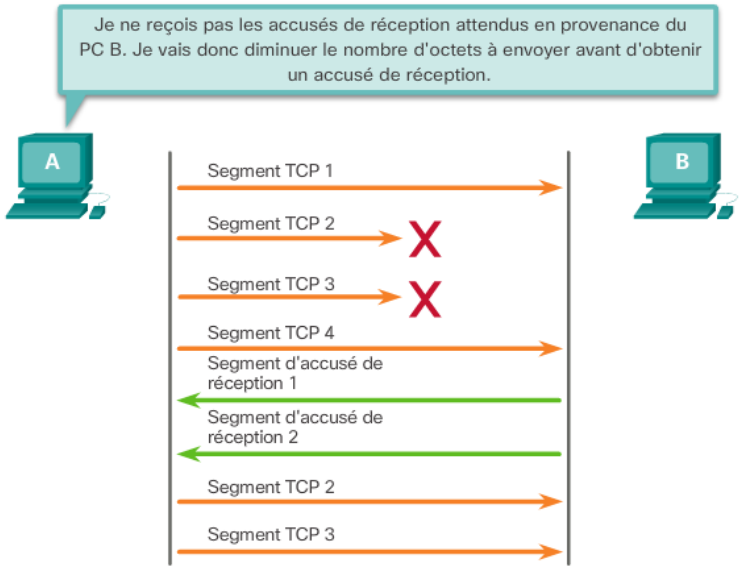
- Le protocole TCP offre des mécanismes de contrôle des flux.
- Le contrôle des flux permet aux terminaux TCP de recevoir et de traiter les données de manière fiable.
- TCP gère le contrôle de flux en ajustant la vitesse des flux de données entre la source et la destination pour une session donnée.
- Cette fonction de contrôle de flux dépend d'un champ d'en-tête TCP de 16 bits appelé Window size (taille de fenêtre). Il s'agit du nombre d'octets que le périphérique de destination d'une session TCP peut accepter et traiter en une seule fois.
- La source et la destination TCP conviennent d'une taille de fenêtre initiale lors de l'établissement de la session TCP.
- Si nécessaire, les terminaux TCP peuvent modifier la taille de fenêtre durant une session.



La **taille de fenêtre** détermine le nombre d'octets envoyés avant de pouvoir recevoir un accusé de réception.
Le numéro d'**accusé de réception** est le numéro du prochain octet attendu.

Contrôle de flux TCP – Prévention des encombrements

- L'encombrement du réseau engendre habituellement la mise au rebut de paquets.
- Les segments TCP non remis déclenchent une retransmission. Cette retransmission de segments TCP peut même aggraver le niveau d'encombrement.
- La source peut évaluer le niveau d'encombrement du réseau en examinant la vitesse à laquelle les segments TCP sont envoyés, sans être reçus avec un accusé.
- La source peut réduire le nombre d'octets qu'elle envoie avant de recevoir un accusé dès qu'un encombrement du réseau est détecté.
- Elle réduit le nombre d'octets non réceptionnés avec un accusé qu'elle envoie, et non pas la taille de fenêtre qui est déterminée par la destination.
- En général, la destination ne sait pas que le réseau est encombré et ne propose donc pas de changer la taille de fenêtre.

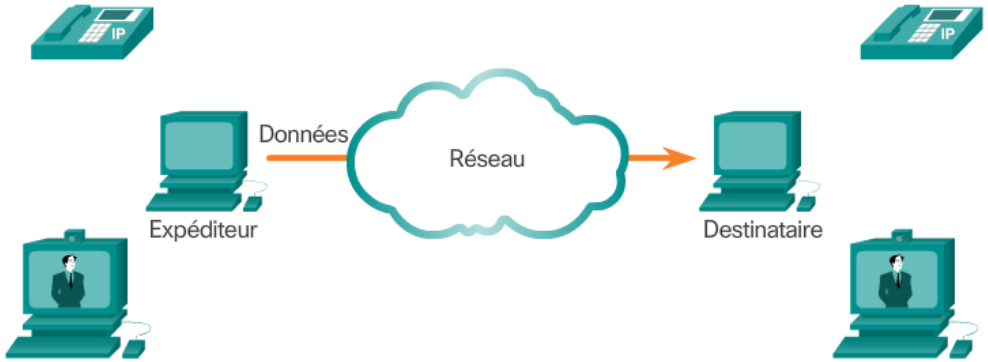


Les numéros d'accusé de réception se réfèrent à l'octet suivant attendu et non à un segment.
Les numéros de segment utilisés sont simplifiés pour les besoins de l'illustration.

Rubrique 9.2.3 : Communication UDP

Faible surcharge et fiabilité du protocole UDP

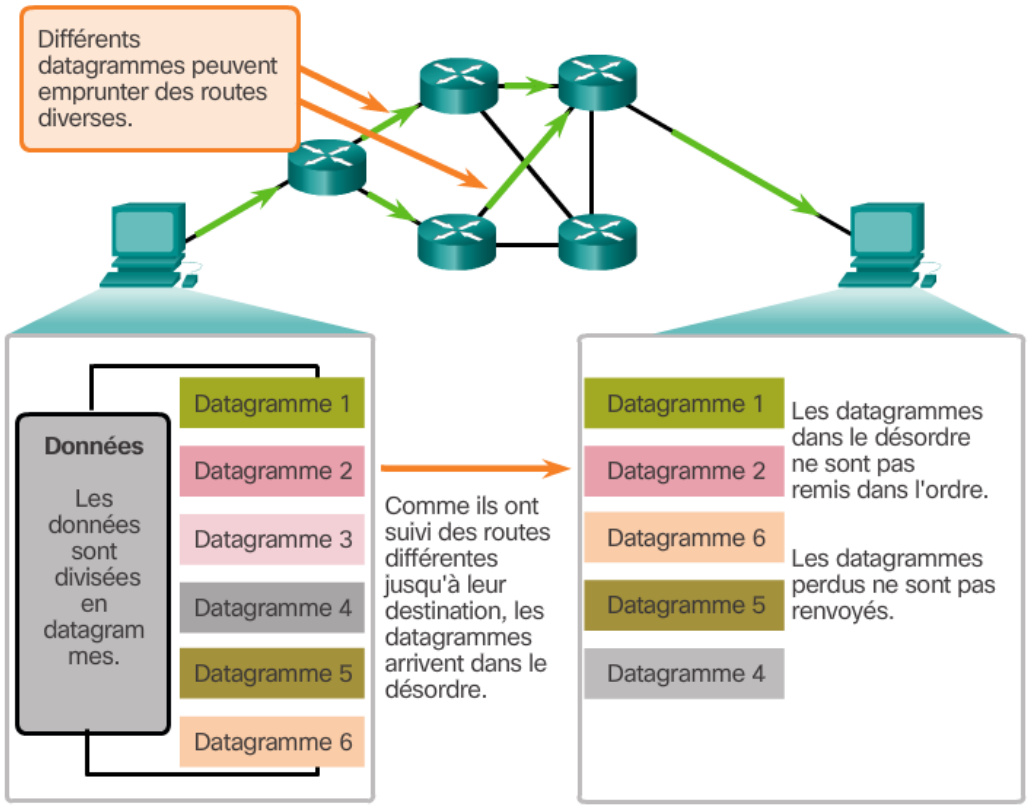
- Le protocole UDP est un protocole simple.
- Il offre des fonctions de base de la couche transport.
- Le protocole UDP subit beaucoup moins de surcharge que le protocole TCP.
- Il est orienté connexion et n'offre pas de mécanismes avancés de retransmission, de séquençage et de contrôle de flux.
- Les applications qui exécutent le protocole UDP peuvent toujours utiliser la fiabilité, mais celle-ci doit être mise en œuvre dans la couche application.
- Toutefois, UDP n'est pas un protocole inférieur.
  Il est conçu pour être plus simple et plus rapide que le protocole TCP au détriment de la fiabilité.



Le protocole UDP n'établit pas de connexion avant d'envoyer des données.

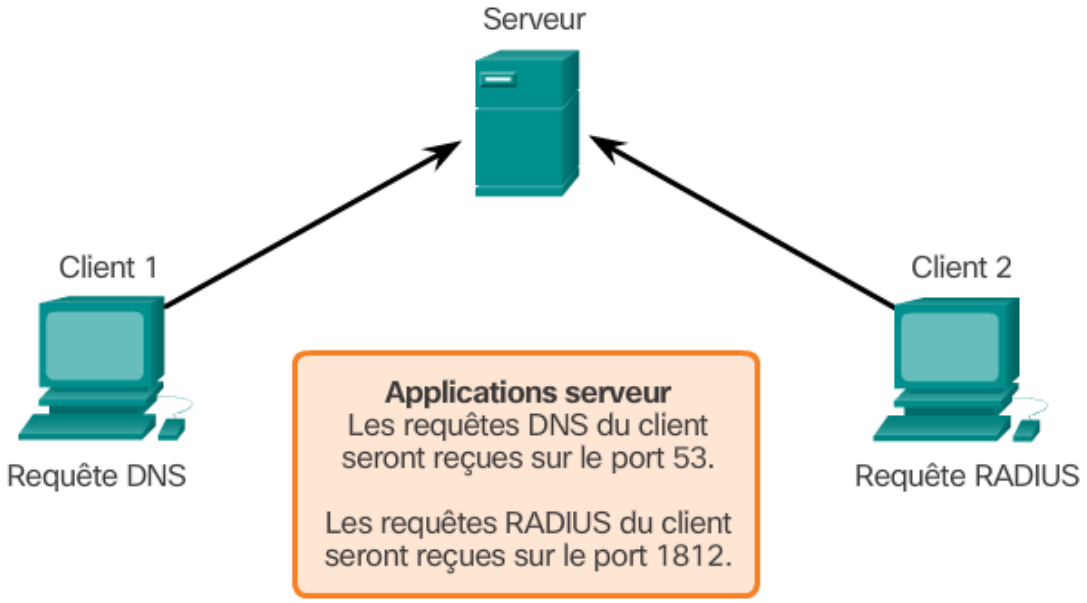
Réassemblage de datagrammes UDP

- Le protocole UDP n'effectue pas de suivi des numéros d'ordre comme le fait le protocole TCP.
- Il n'a en effet aucun moyen de réordonnancer les datagrammes pour leur faire retrouver leur ordre de transmission d'origine.
- Le protocole UDP reconstitue les données dans l'ordre dans lequel elles ont été reçues.
- L'application doit identifier le bon ordre des données, si nécessaire.



Processus de serveur UDP

- Les applications de serveur UDP se voient également attribuer des numéros de port réservés ou enregistrés.
- Les applications et services UDP qui s'exécutent sur un serveur acceptent les requêtes de client UDP.
- Les requêtes reçues sur un port particulier sont transmises à l'application correspondante en fonction des numéros de port.



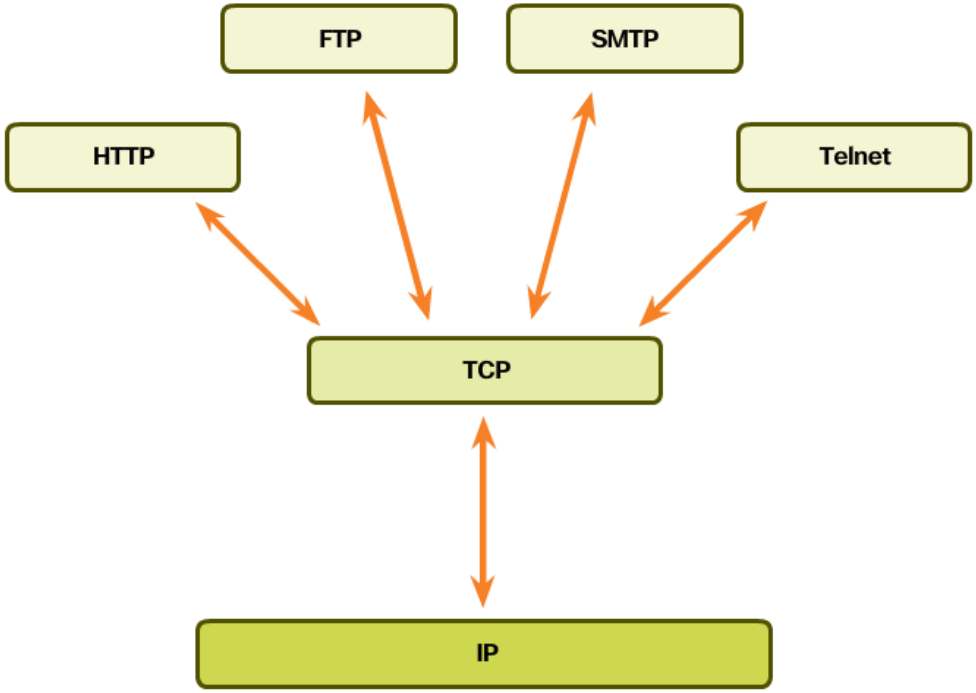
Processus de serveur UDP

- La communication UDP de client à serveur est également lancée par une application cliente.
- Le processus du client UPD sélectionne dynamiquement un numéro de port et l'utilise comme port source.
- Le port de destination est généralement le numéro de port réservé ou inscrit affecté au processus serveur.
- La même paire source-destination de ports est reprise dans l'en-tête de tous les datagrammes utilisés dans la transaction.
- Les données renvoyées du serveur vers le client utilisent des numéros de port source et de destination inversés dans l'en-tête du datagramme.

Rubrique 9.2.4 : TCP ou UDP

Applications utilisant le protocole TCP

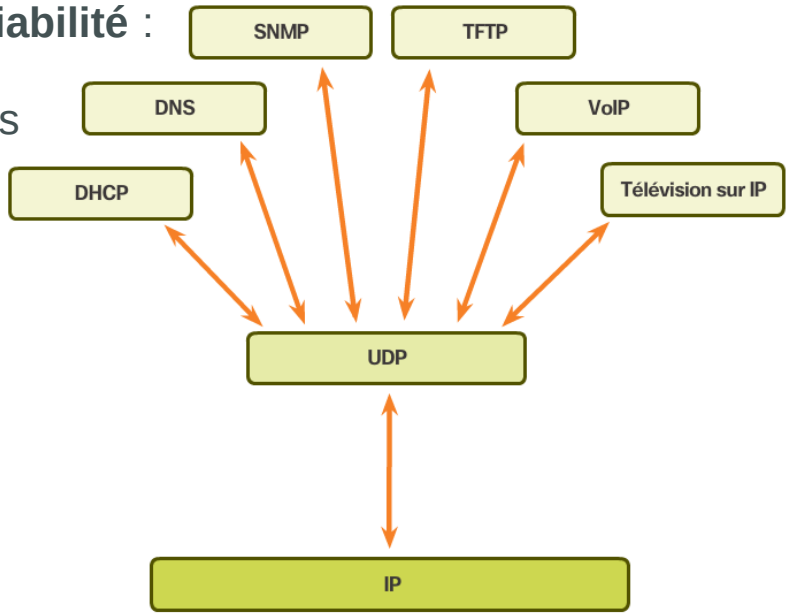
- Le protocole TCP gère toutes les tâches relatives à la couche transport.
- Cela dispense les applications d'avoir à les gérer elles-mêmes.
- Les applications peuvent simplement envoyer le flux de données à la couche transport et utiliser les services du protocole TCP.



Applications utilisant le protocole UDP

Il existe trois types d'application plus adaptés au protocole UDP :

- **Applications multimédia et de streaming vidéo** : applications pouvant tolérer certaines pertes de données, mais peu ou pas de retard. La voix sur IP et le streaming vidéo en sont de bons exemples.
- **Simples applications de requête et de réponse** : applications dont les transactions sont simples et pour lesquelles un hôte envoie une requête à laquelle il recevra ou non une réponse. Exemples : DNS et DHCP.
- **Applications qui gèrent elles-mêmes la fiabilité** : communications unilatérales où le contrôle de flux, la détection des erreurs, les accusés de réception et la reprise sur erreur ne sont pas nécessaires ou peuvent être gérés par l'application.
  Exemples : SNMP et TFTP.



Section 9.3 : Résumé

Objectifs du chapitre :

- Expliquer comment les protocoles et services de la couche transport prennent en charge les communications sur les réseaux de données
- Comparer les opérations des protocoles de la couche transport dans la prise en charge de la communication de bout en bout

Rubrique 9.3.1 : Conclusion

Merci.

Cisco Networking Academy

Mind Wide Open