





Module 14: Couche de transport

Présentation des réseaux V7.0
(ITN)



Objectifs du module

Titre du module: Couche de transport (NetACad module 14.0)

L'objectif du module: Comparer les opérations des protocoles de la La couche de transport dans la prise en charge de la communication de bout en bout.

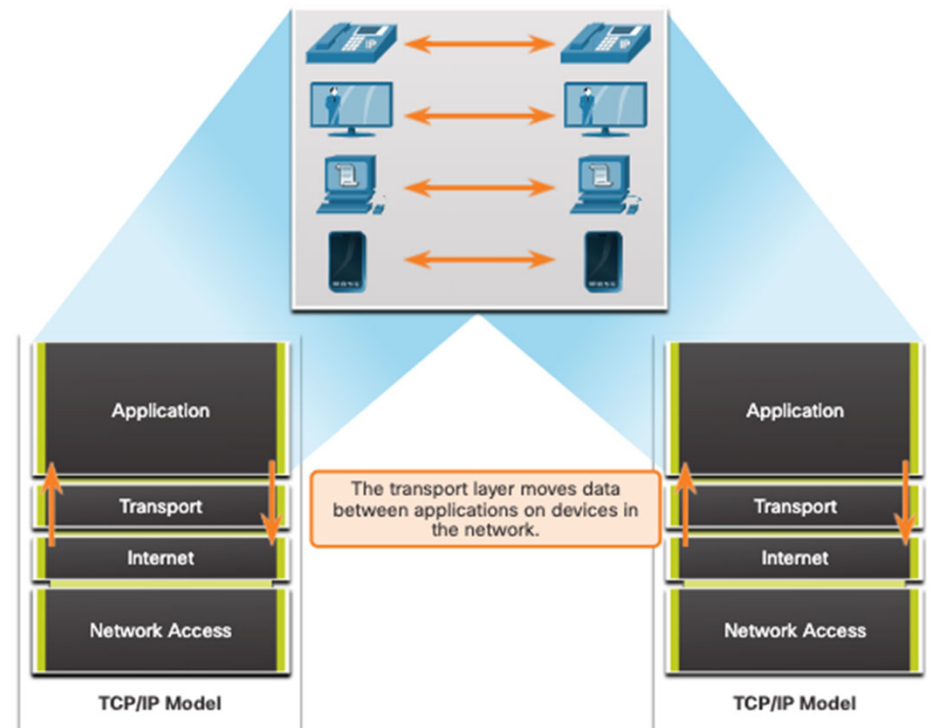
Titre du rubrique	Objectif du rubrique
Transport des données	Expliquer le rôle de la couche de transport dans la gestion du transport des données dans une communication de bout en bout.
Présentation du protocole TCP	Expliquer les caractéristiques du TCP.
Présentation du protocole UDP	Expliquer les caractéristiques de l'UDP.
Numéros de port	Expliquer comment TCP et UDP utilisent les numéros de port.
Processus de communication TCP	Expliquer comment les processus d'établissement et d'interruption de session TCP garantissent la fiabilité des communications.
Fiabilité et contrôle des flux	Expliquer comment les unités de données de protocole TCP sont transmises et comment leur réception est confirmée pour garantir l'acheminement des données.
Communication UDP	Comparer les opérations des protocoles de la La couche de transport dans la prise en charge de la communication de bout en bout.

14.1 Transport des données

Rôle de la couche transport (14.1.1)

La couche de transport est:

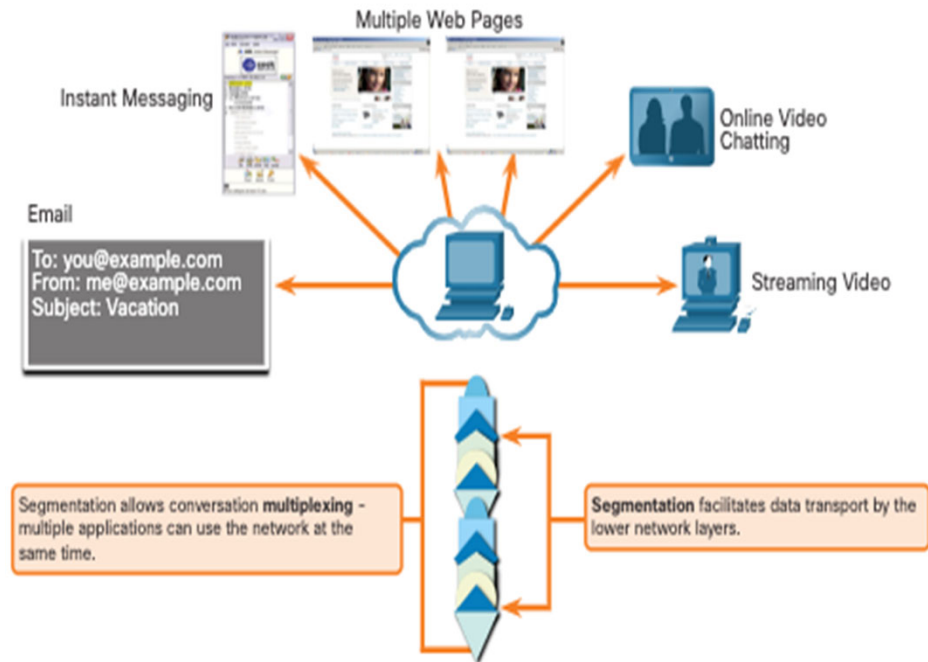
- responsable des communications logiques entre les applications exécutées sur différents hôtes.
- La liaison entre la couche d'application et les couches inférieures qui sont responsables de la transmission du réseau.



Responsabilités de la La couche de transport (14.1.2)

La couche de transport a les responsabilités suivantes:

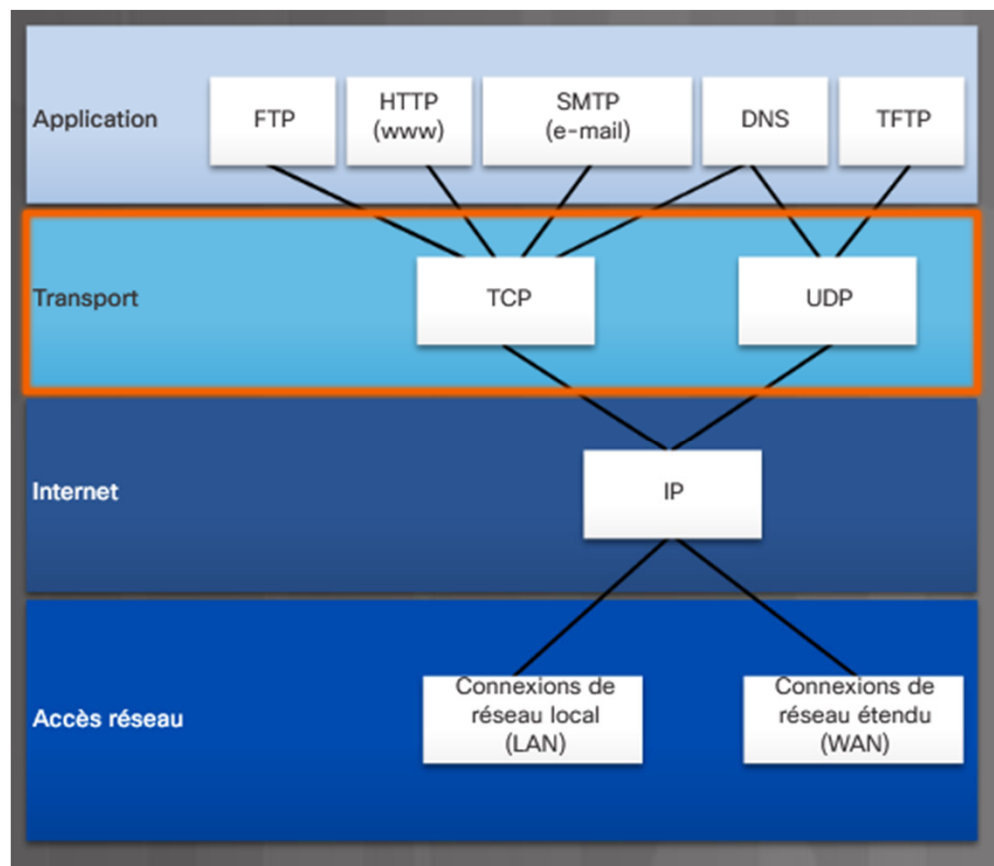
- Suivre les conversations individuelles
- Segmentation des données et reconstitution des segments
- Ajouter les informations d'en-tête
- Identifier, séparer et gérer plusieurs conversations
- Utiliser la segmentation et le multiplexage pour permettre à différentes conversations de communication d'être entrelacées sur le même réseau



Transport des données

Fiabilité de la couche transport

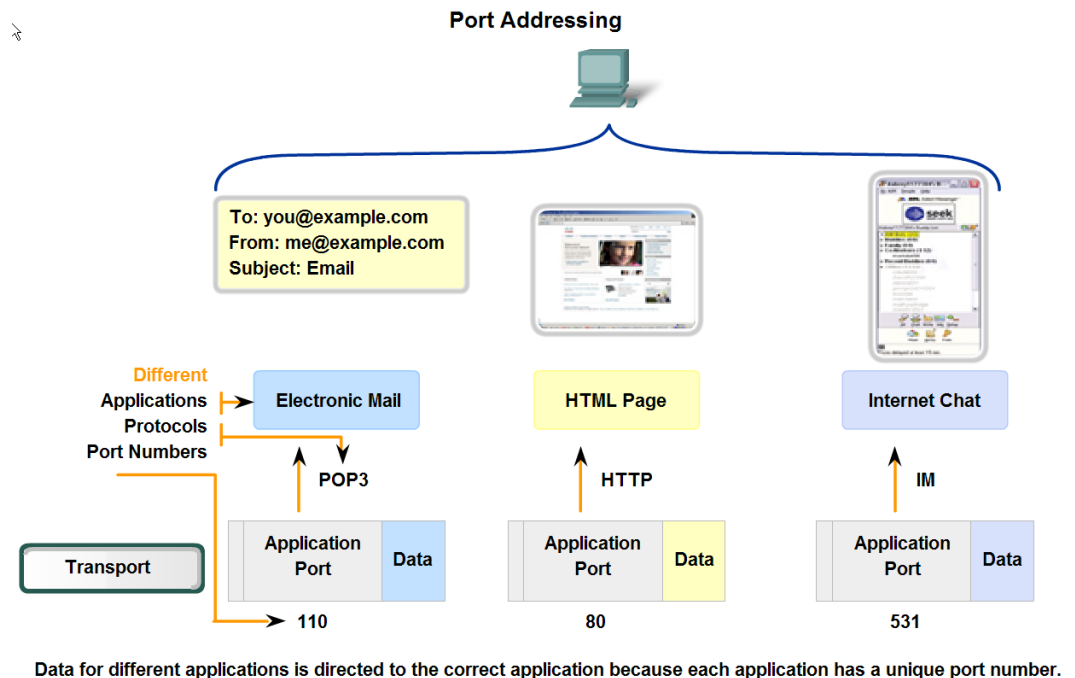
- La suite de protocoles TCP/IP propose deux protocoles de couche transport :
 - **TCP** (Transmission Control Protocol)
 - Robuste → garantie de livraison
 - (reprise de paquets perdus)
 - Considéré **comme fiable**
 - Champs supplémentaires nécessaires dans l'en-tête qui augmente la taille et engendre des retards.
 - **UDP** (User Datagram Protocol)
 - **Plus léger** → Moins de champs
 - plus rapide
 - Ne **garantit pas la fiabilité**.



Transport des données

Rôle de la couche transport

- **Identification des applications :**
- Souvent, plusieurs services sont installés sur un même serveur.
- Le numéro du port est associé à une application
- Exemples possibles:
- ftp → transfert de fichiers : ports 20/21
- http → serveur web: port 80/8080/443
- POP3/SMTP → courriels port 110/25



Présentation des protocoles TCP et UDP

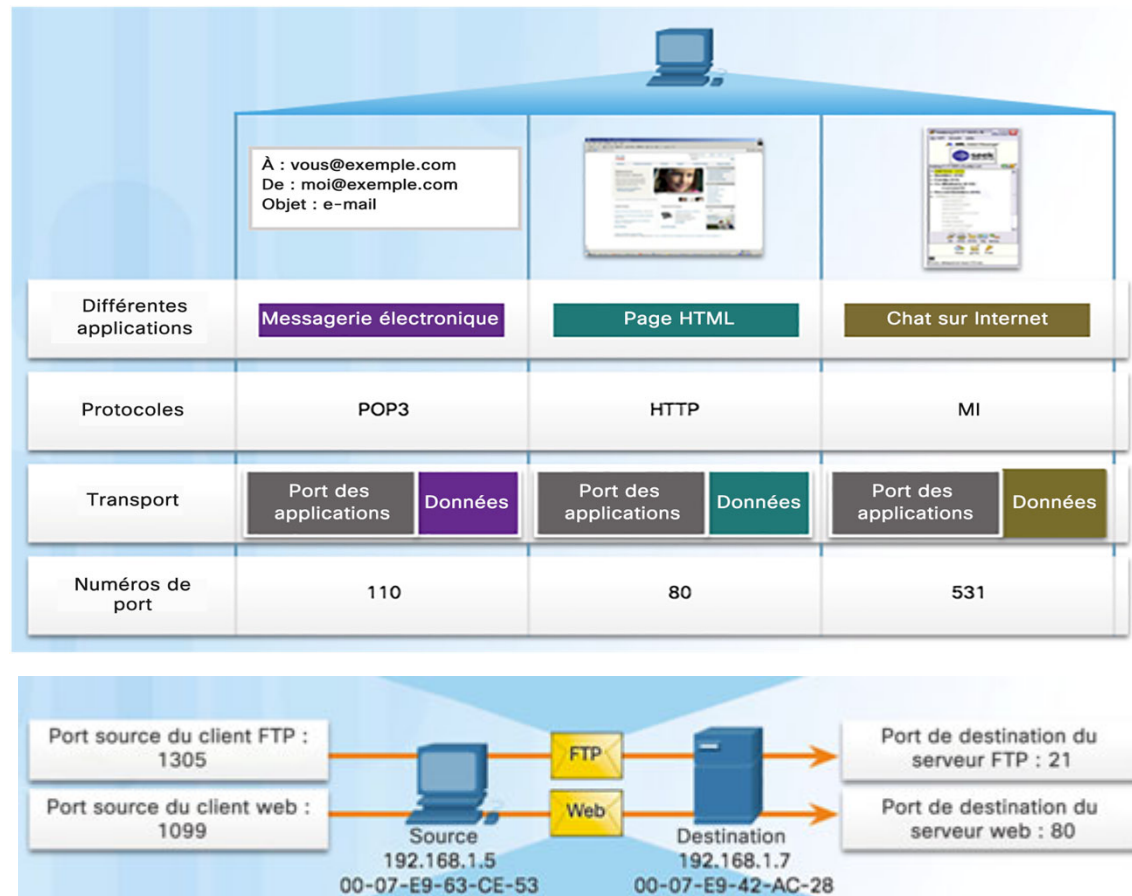
Numéros de port

■ Port source

- Port d'application d'origine **généré (random)** par l'expéditeur
- Par exemple : chaque conversation HTTP est suivie en fonction des ports sources.

■ Port de destination

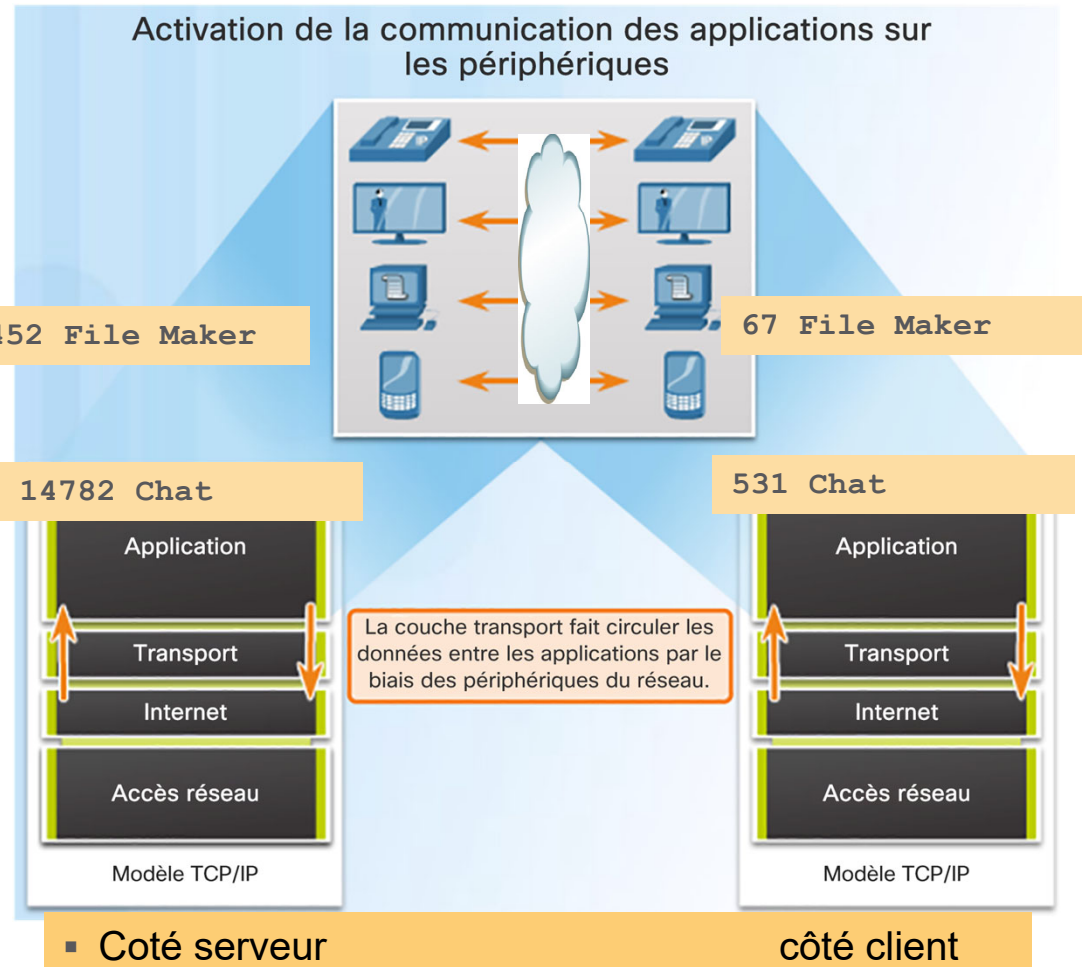
- Indique à la destination quel service est demandé
- Par exemple : les services web du port 80 sont demandés



Transport des données

Rôle de la couche transport

- Établir et maintenir une communication temporaire entre deux **applications**
- Les deux extrémités sont liées par leur «**ports**»
- Le « PDU de couche transport » contient les ports source et destination pour permettre la reconnaissance mutuelle de la liaison

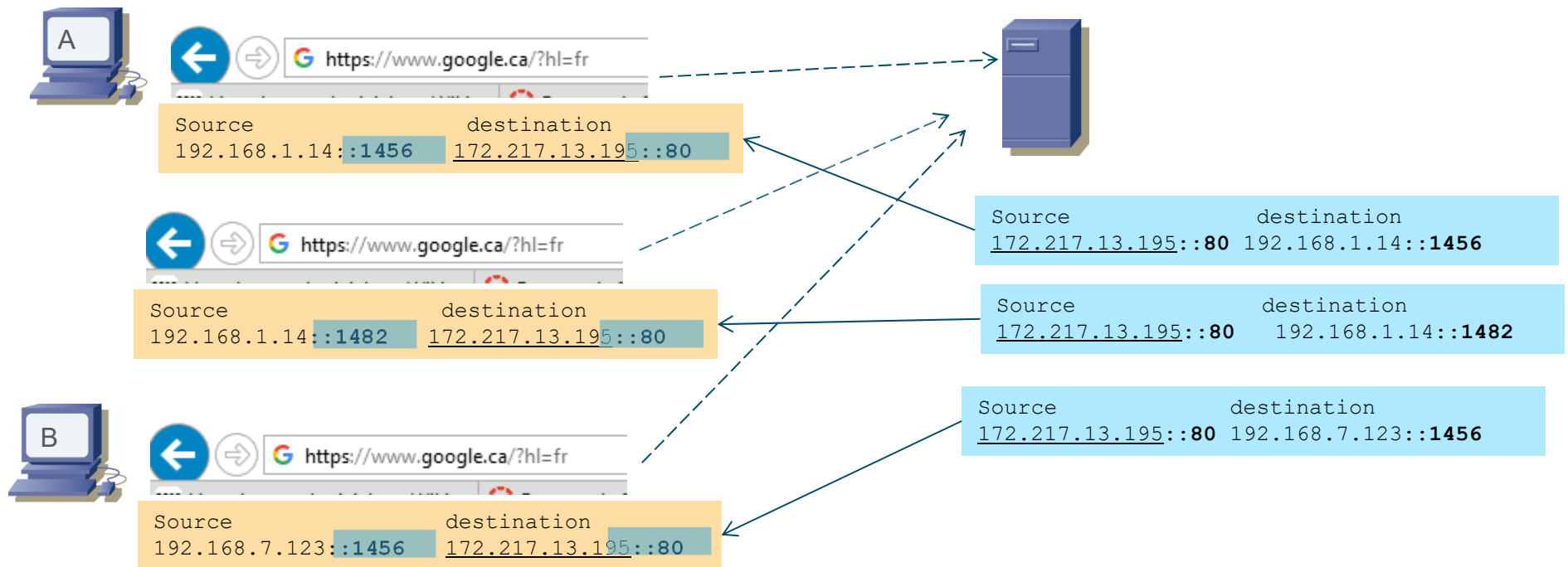


Transport des données

Multiplexage de conversations

- Demandes fréquentes

réponse « simultanée » du serveur

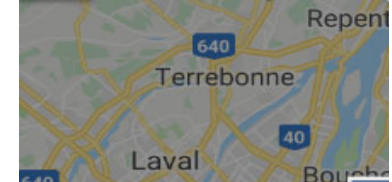
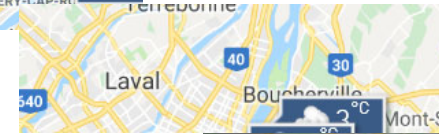
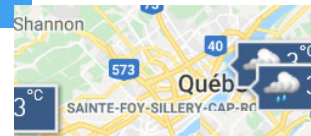


- Pour un même utilisateur, plusieurs sessions, du même protocole, peuvent fonctionner en même temps

Transport des données

Responsabilités de la couche transport

- **Segmentation** : divise les données volumineuses (exemple Page web) en **segments numérotés** pour l'assemblage à destination.
- **Suivi des conversations** : reprise de segments perdus ou brouillés.



14.1.7 Quiz couche transport

14.2 Présentation du protocole TCP

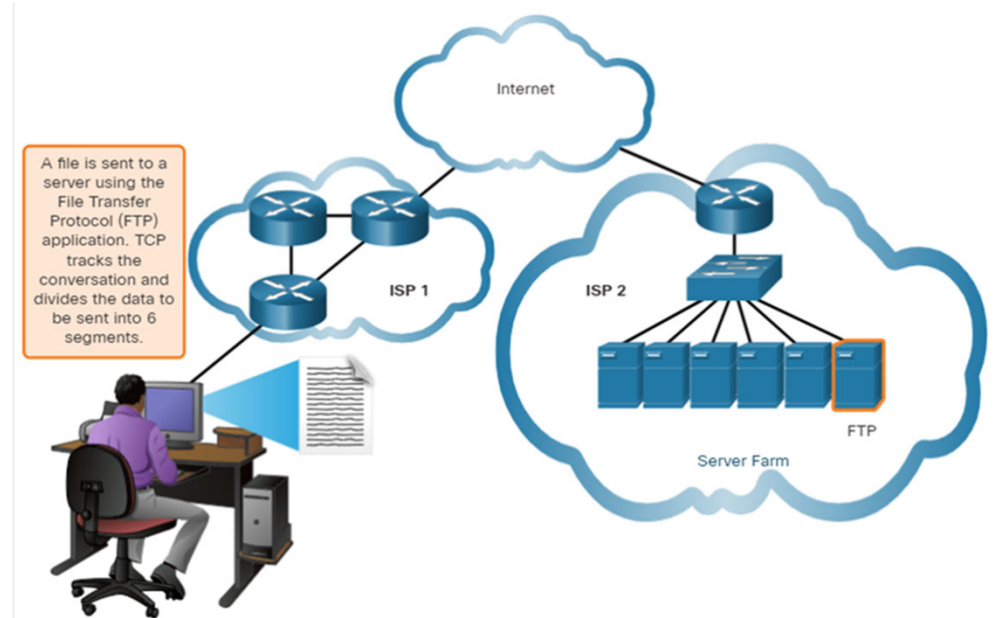
Fonctions du protocole TCP (14.2.1)

- 1) établir une session de bout en bout
 - S'assure que l'application est prête à recevoir les données
 - Négocier la quantité de trafic qui peut être acheminée à un moment donné
- 2) Acheminement fiable
 - S'assure que chaque segment envoyé par la source arrive à destination
 - **ACK: Accusé de réception** confirme la réception
- 3) Ordonnancer les segments d'une même session à l'arrivée
 - **numérotation** et **tri**.
- 4) Contrôler le flux: réagit aux contraintes des bandes passantes
 - Transmission des segments selon la fréquence négociée

Protocole TCP (Transmission Control Protocol) (14.1.4)

TCP assure la fiabilité et le contrôle du flux.
Les opérations de base de TCP:

- **Numéroter** et suivre les segments de données transmis à un hôte spécifique à partir d'une application spécifique
- **Accuser la réception** des données reçues
- **Retransmettre** toute donnée non reconnue après un certain temps
- Séquence des données qui pourraient arriver dans un ordre incorrect
- Envoyer des données à un taux efficace et acceptable par le destinataire

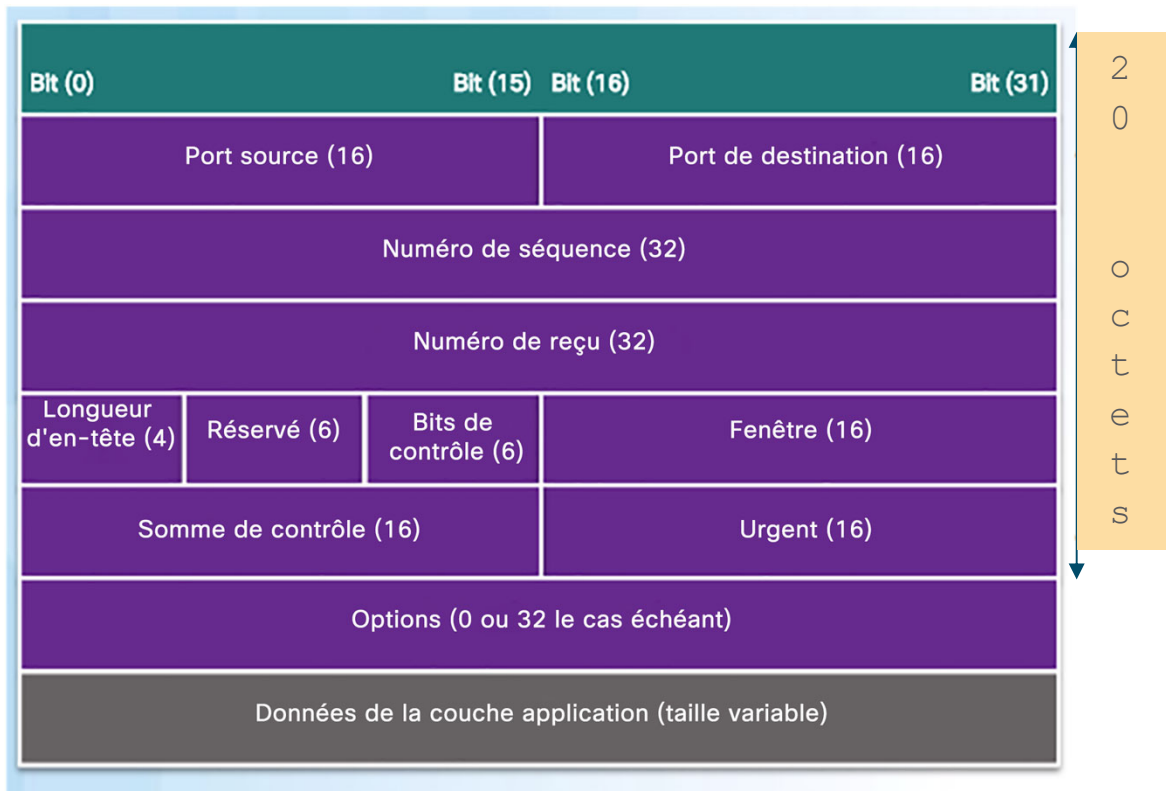


Présentation de protocole TCP

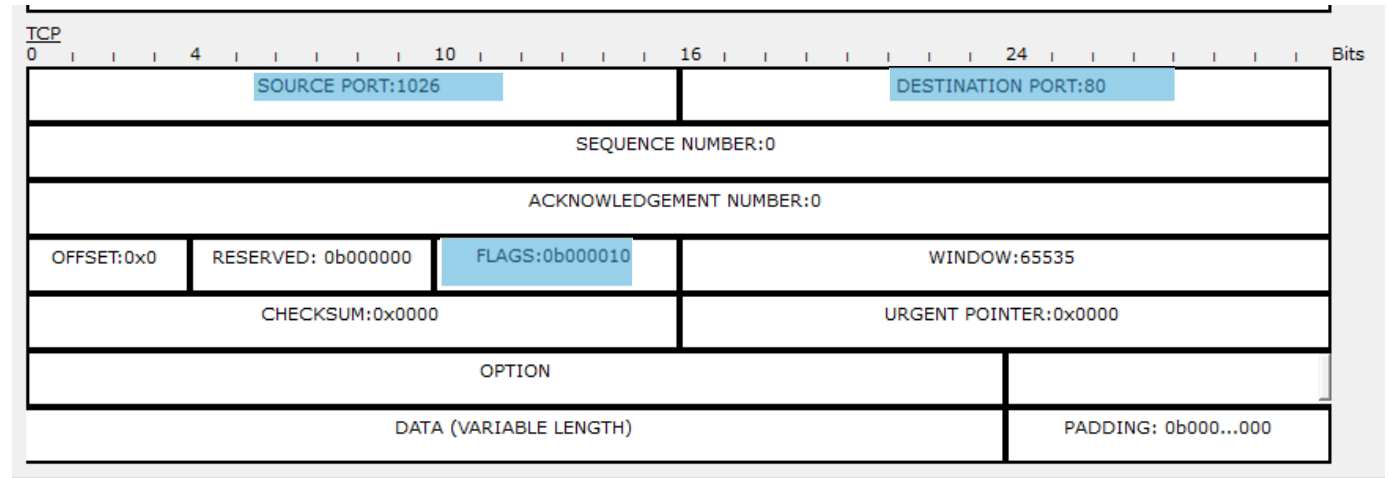
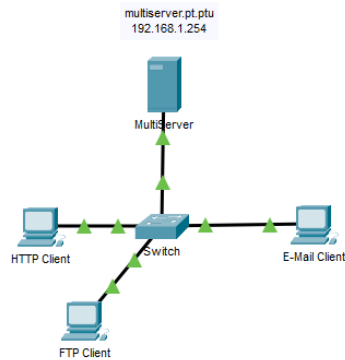
L'en-tête TCP (14.2.2)

TCP est un protocole avec état, ce qui signifie qu'il garde une trace de l'état de la session de communication.

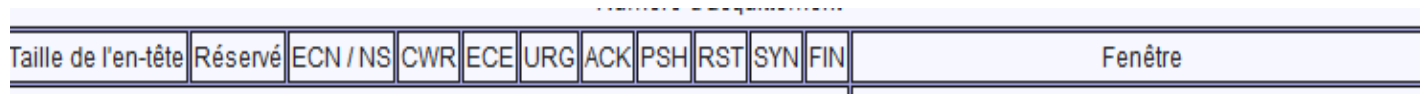
le protocole TCP enregistre les informations qu'il a envoyées et les informations qu'il a reçues.



Exemple pour l'exercice PT 14.8.1 Packet Tracer



- Flags: 6 bits de contrôle TCP pour **SYN**chronisation, **ACK**nowledgemnt, **FIN**al, etc



14.8.1 Packet Tracer - TCP and UDP Communications.pka

Présentation de protocole TCP

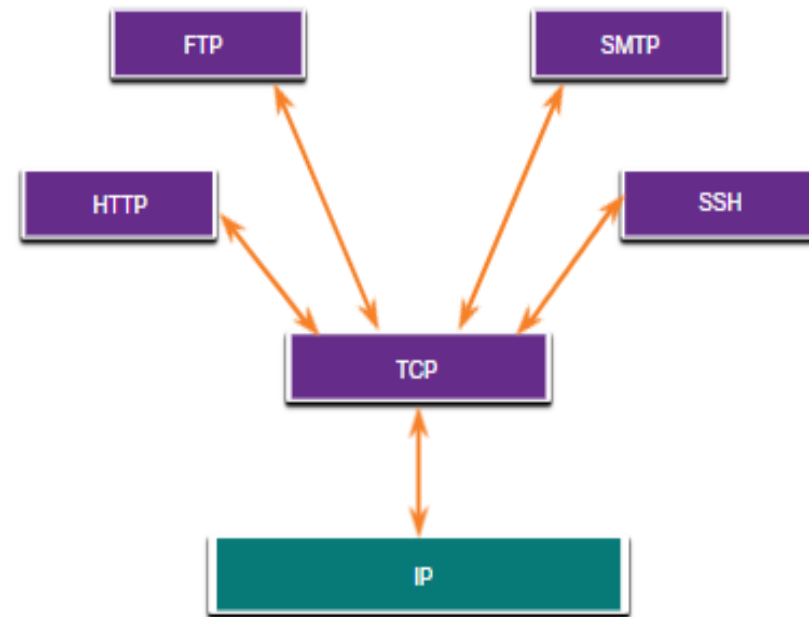
TCP Champs d'en-tête

Champ d'en-tête TCP	Description
Port source	Champ 16 bits utilisé pour identifier l'application source par le numéro de port.
Port de destination	Champ de 16 bits utilisé pour identifier l'application de destination par le numéro de port.
Numéro de séquence	Champ 32 bits utilisé à des fins de réassemblage de données.
Numéro d'accusé de réception	Champ 32 bits est utilisé pour indiquer que les données ont été reçues et l'octet suivant est prévu de la source.
Longueur d'en-tête	Champ 4 bits connu sous le nom de « offset de données » qui indique la longueur de l'en-tête du segment TCP.
Réservé	Un champ de 6 bits qui est réservé pour une utilisation future.
Bits de contrôle	Un champ de 6 bit utilisé comprennent des codes de bits qui indiquent l'objectif et la fonction du segment TCP.
Taille de fenêtre	Champ 16 bits utilisé pour indiquer le nombre d'octets qui peut être acceptés.
Somme de contrôle	Un champ de 16 bits utilisé pour la vérification des erreurs de l'en-tête du segment et des données.
Urgent	Champ 16 bits utilisé pour indiquer si les données contenues sont urgentes.

Présentation du protocole TCP

Applications utilisant le protocole TCP

Le TCP gère toutes les tâches associées à la division du flux de données en segments, à la fiabilité, au contrôle du flux de données et à la réorganisation des segments.



14.2.5 Quiz TCP

14.3 Présentation du protocole UDP

Caractéristiques du protocole UDP (14.3.1)

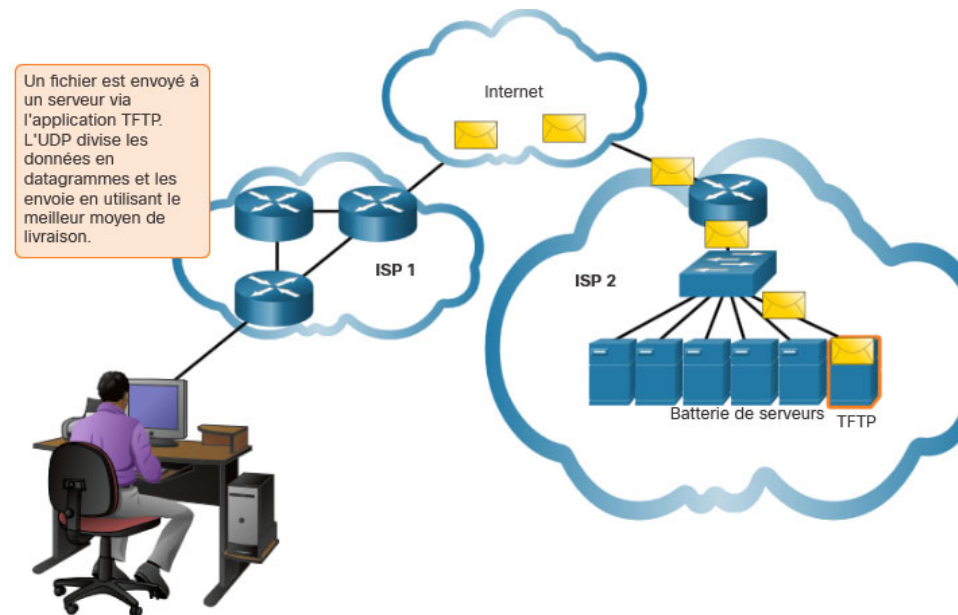
Les caractéristiques de l'UDP sont les suivantes :

- Les données sont reconstituées selon l'ordre de réception.
- Les segments qui sont perdus ne sont pas renvoyés.
- Il n'y a pas d'établissement de session.
- L'expéditeur n'est pas informé de la disponibilité des ressources.

UDP (User Datagram Protocol) (14.1.5)

UDP fournit des fonctions de base permettant d'acheminer des segments de données entre les applications appropriées tout en ne nécessitant que très peu de surcharge et de vérification des données.

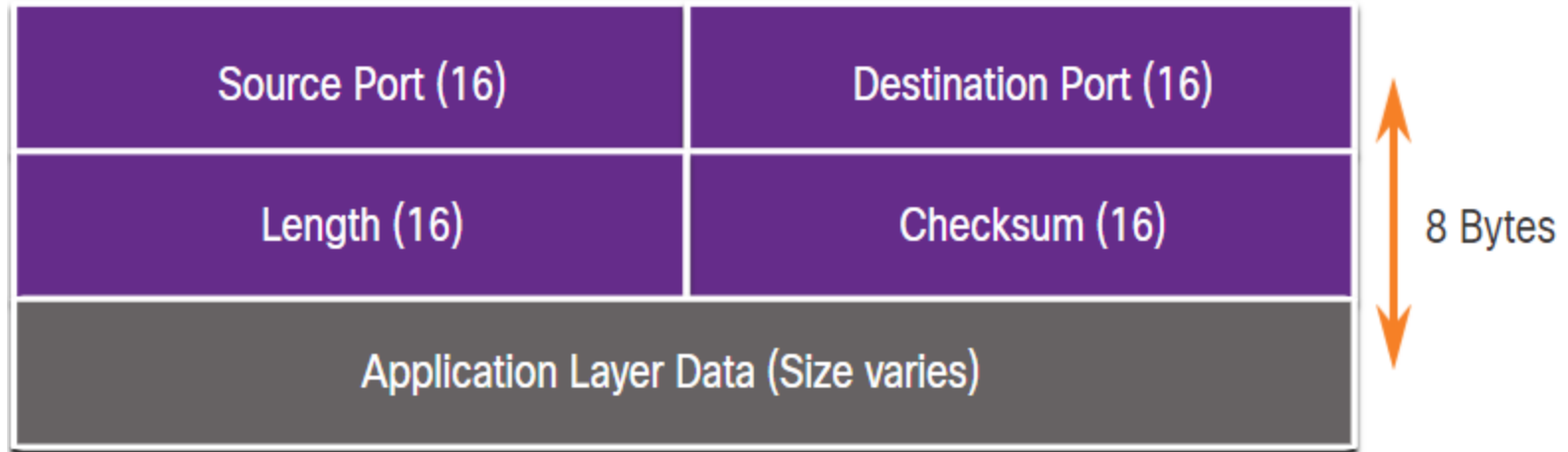
- UDP est un protocole sans connexion.
- UDP est également connu comme un protocole de livraison du meilleur effort, car il n'y a pas d'accusé de réception des données à la destination.



Présentation de protocole UDP

L'en-tête UDP (14.3.2)

L'en-tête UDP est beaucoup plus simple que l'en-tête TCP car il n'a que quatre champs et nécessite 8 octets (c'est-à-dire 64 bits).



Présentation de protocole UDP

Les Champs d'en-tête UDP

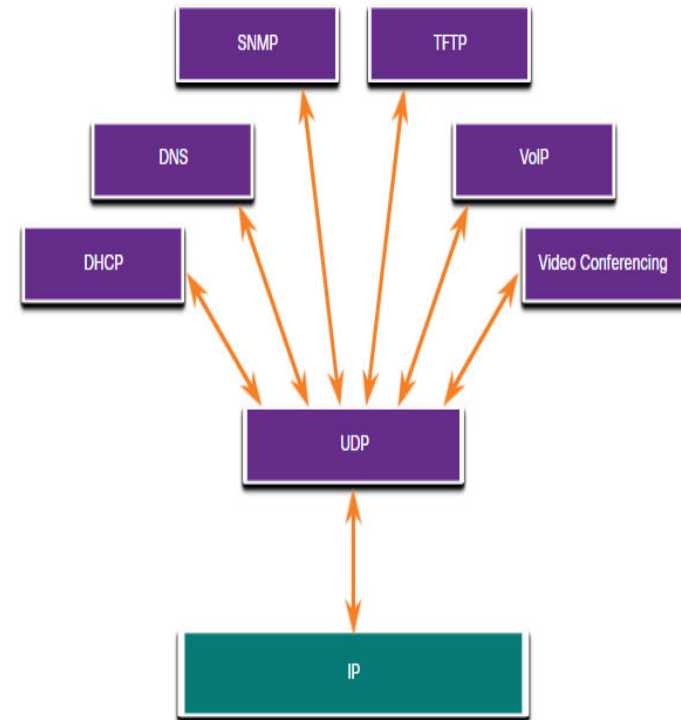
Le tableau identifie et décrit les quatre champs d'un en-tête UDP.

Champ d'en-tête UDP	Description
Port source	Champ 16 bits utilisé pour identifier l'application source par le numéro de port.
Port de destination	Champ de 16 bits utilisé pour identifier l'application de destination par le numéro de port.
Longueur	Champ 16 bits indiquant la longueur de l'en-tête de datagramme UDP.
Somme de contrôle	Champ 16 bits utilisé pour la vérification des erreurs de l'en-tête et des données du datagramme.

Présentation du protocole UDP

Applications utilisant le protocole UDP

- Les applications vidéo et multimédia en direct : Ces applications peuvent tolérer une certaine perte de données, mais ne nécessitent que peu ou pas de délai. La voix sur IP et le streaming vidéo sont de bons exemples.
- Les simples applications de requête et de réponse : ils sont des applications dont les transactions sont simples et pour lesquelles un hôte envoie une requête à laquelle il recevra ou non une réponse. Exemples incluent **DNS** et **DHCP**.
- Applications qui gèrent elles-mêmes la fiabilité- Communications unidirectionnelles où le contrôle de flux, la détection des erreurs, les accusés de réception et la récupération des erreurs ne sont pas nécessaires, ou peuvent être gérés par l'application. Exemples incluent: **SNMP** et **TFTP**.



14.4 Les Numéros de ports

Les Numéros de port

Communications multiples et séparées

Les protocoles de couches de transport TCP et UDP utilisent des numéros de port pour gérer plusieurs conversations simultanées.

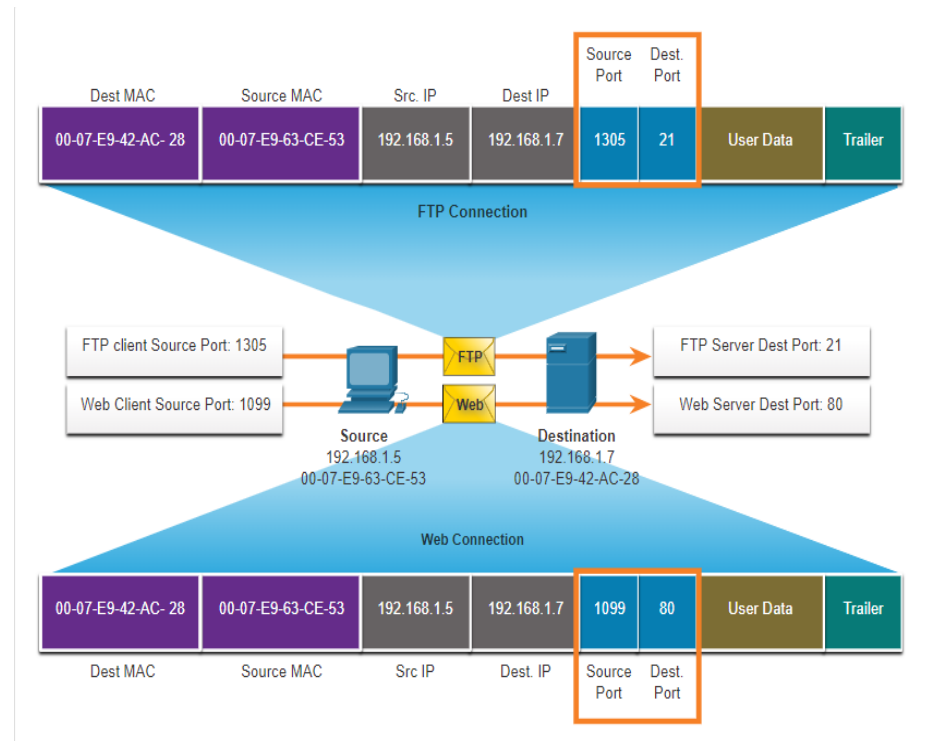
Le numéro de port source est associé à l'application d'origine sur l'hôte local tandis que le numéro de port de destination est associé à l'application de destination sur l'hôte distant.



Numéros de port

Paires d'interfaces de connexion (14.4.2)

- Les ports sources et de destination sont placés à l'intérieur du segment.
- Les segments sont ensuite encapsulés dans un paquet IP.
- 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 interface de connexion.
- Les interfaces de connexion permettent à plusieurs processus exécutés sur un client de se différencier les uns des autres, et aux multiples connexions à un processus serveur de se distinguer les unes des autres.



Groupes de numéros de port (14.4.3)

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

- Ports réservés (numéros 0 à 1023) : ces numéros sont réservés à des services et des applications: exemple HTTP, FTP, . . .
- Ports enregistrés (numéros 1024 à 49151) : ces numéros de port sont affectés par l'IANA à une entité demandeuse (exemple Microsoft, Cisco,..) pour des applications spécifiques.
- Ports privés ou dynamiques (numéros 49152 à 65535) : client génère un port aléatoirement

Les Numéros de port

Groupes de numéros de port

Groupe de ports	Gamme de numéros	Description
Ports réservés	de 0 à 1023	<ul style="list-style-type: none">• Ces numéros de port sont réservés aux services et aux applications courants ou populaires tels que les navigateurs web, les clients de messagerie électronique et les clients d'accès à distance.• Les ports bien connus définis pour les applications serveur courantes permettent aux clients d'identifier facilement le service associé requis.
Ports inscrits	de 1024 à 49151	<ul style="list-style-type: none">• ces numéros de port sont affectés par l'IANA à une entité demandeuse pour être utilisés avec des processus ou des applications spécifiques.• Ces processus sont essentiellement des applications particulières qu'un utilisateur a choisi d'installer plutôt que des applications courantes qui recevraient un numéro de port réservé.• Par exemple, Cisco a enregistré le port 1812 pour son processus d'authentification du serveur RADIUS.
Ports privés et/ou dynamiques	de 49152 à 65535	<ul style="list-style-type: none">• Ces ports sont également connus sous le nom de <i>ports éphémères</i>.• Le système d'exploitation du client attribue généralement des numéros de port dynamique lorsqu'une connexion à un service est lancée.• Le port dynamique est ensuite utilisé pour identifier l'application cliente pendant la communication.

Les Numéros de port


Groupes de numéros de port (Suite)

Numéros de ports reconnus

Numéro de port	Protocole	Application
20	TCP	FTP (File Transfer Protocol) - Données
21	TCP	File Transfer Protocol (FTP) - Contrôle
22	TCP	SSH (Secure Shell)
23	TCP	Telnet
25	TCP	Protocole SMTP
53	UDP, TCP	Service de noms de domaine (Domain Name Service, DNS)
67	UDP	Serveur DHCP (Dynamic Host Configuration Protocol)
68	UDP	Client DHCP (Dynamic Host Configuration Protocol)
69	UDP	Protocole TFTP (Trivial File Transfer Protocol)
80	TCP	Protocole HTTP (Hypertext Transfer Protocol)
110	TCP	Protocole POP3 (Post Office Protocol version 3)
143	TCP	IMAP (Internet Message Access Protocol)
161	UDP	Protocole SNMP (Simple Network Management Protocol)
443	TCP	protocole HTTPS (Hypertext Transfer Protocol Secure)

Présentation des protocoles TCP et UDP

Groupes de numéros de port (14.4.3)



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

Numéros de port reconnus

Numéros de port

La commande netstat (14.4.5)

Les connexions TCP inexpliquées peuvent poser un risque de sécurité majeur. Netstat est un outil important pour vérifier les connexions.

```
C:\> netstat
Active Connections
Proto Local Address           Foreign Address State
TCP 192.168.1.124:3126    192.168.0.2:netbios-ssn ESTABLISHED
TCP 192.168.1.124:3158    207.138.126.152:http ESTABLISHED
TCP 192.168.1.124:3159    207.138.126.169:http ESTABLISHED
TCP 192.168.1.124:3160    207.138.126.169:http ESTABLISHED
TCP 192.168.1.124:3161    sc.msn.com:http ESTABLISHED
TCP 192.168.1.124:3166    www.cisco.com:http ESTABLISHED
```

Aide sur la commande

```
C:\> netstat /?
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y]
[intervalle]
```

14.5 Processus de communication TCP

Protocoles avec connexion



Sans connexion	
Activité courante	Protocoles
Lettre à la poste	Protocole IP
Envoi de texto	

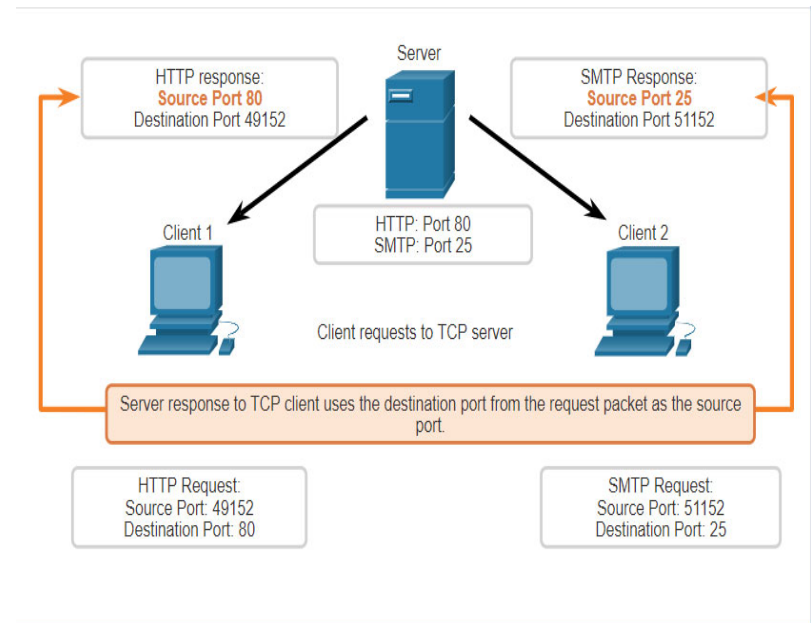
Avec connexion	
Activité courante	Protocoles
Appel téléphonique	Requêtes web

Processus de communication TCP

Processus de serveur TCP (14.5.1)

Chaque processus de demande s'exécutant sur un serveur est configuré pour utiliser 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.
- Une application de serveur active affectée à un port spécifique est considérée comme étant ouverte, ce qui signifie que la couche transport accepte et traite les segments adressés à ce port.
- Toute demande entrante d'un client qui est adressée à l'interface de connexion correcte est acceptée et les données sont transmises à l'application de serveur.



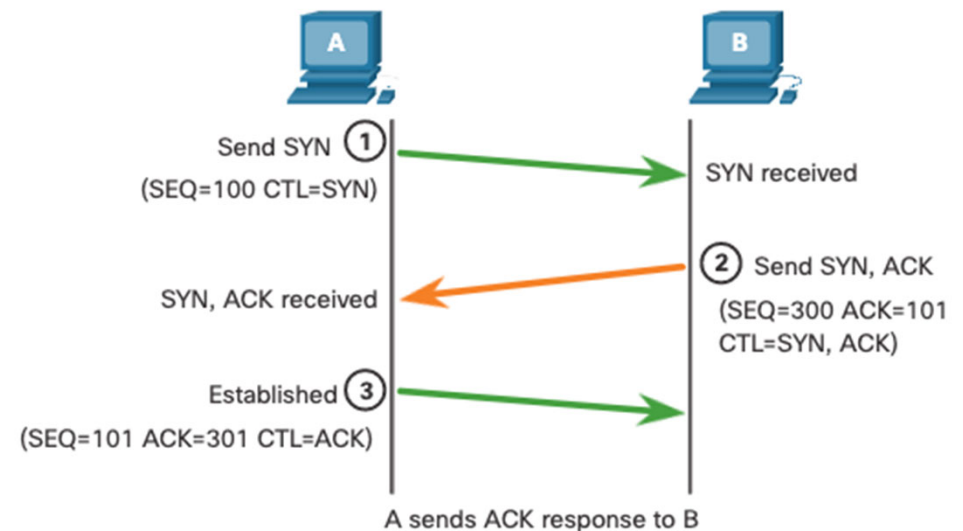
Processus de communication TCP

Établissement d'une connexion TCP

Étape 1: Le client demande l'établissement d'une session de communication client-serveur avec le serveur.

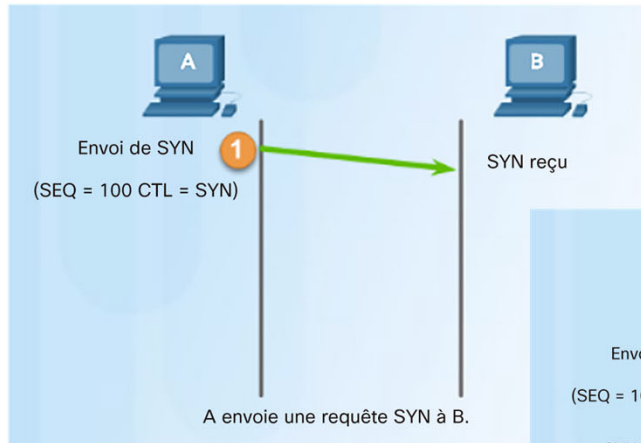
Étape 2: Le serveur accuse la réception de la session de communication client-serveur et demande l'établissement d'une session de communication serveur-client.

Étape 3: Le client accuse réception de la session de communication serveur-client.

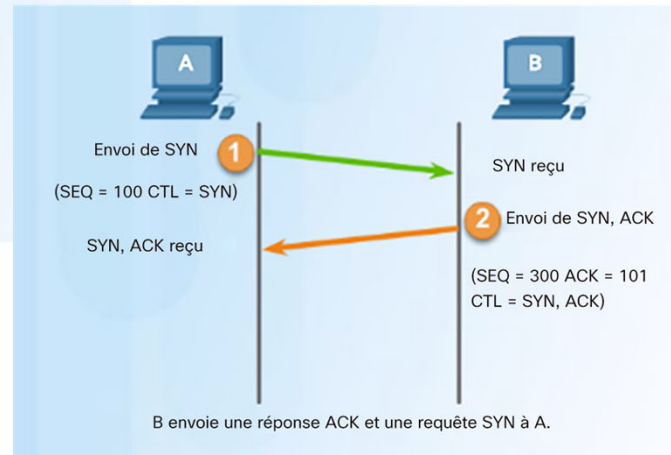


Processus de communication TCP

Connexion d'une session de bout en bout TCP (14.5.2)

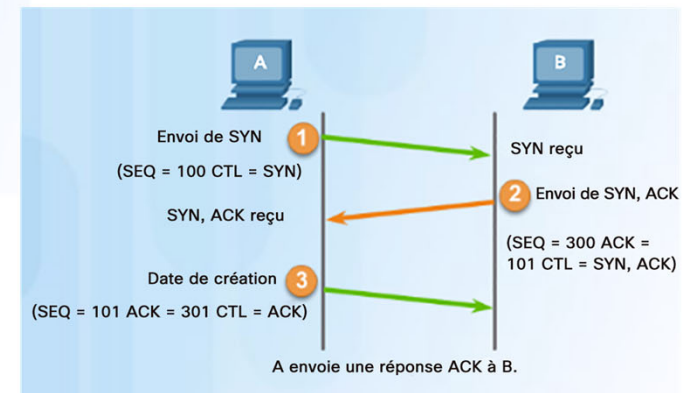


- Étape 1 : le client demande l'établissement d'une session avec le serveur.



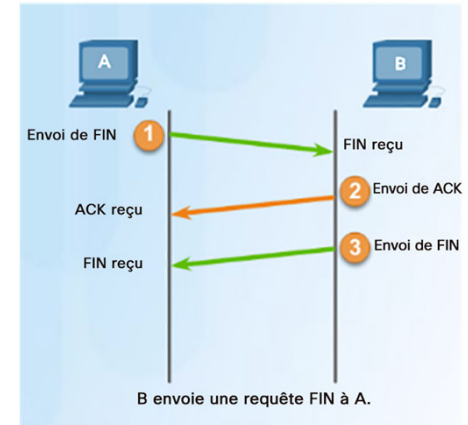
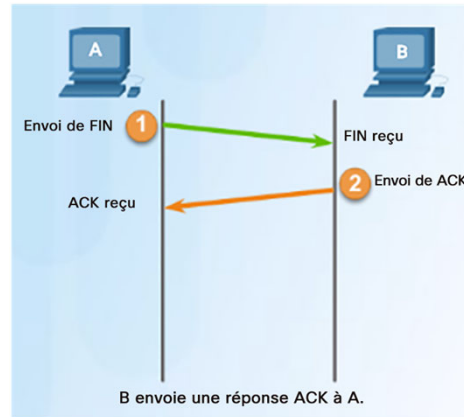
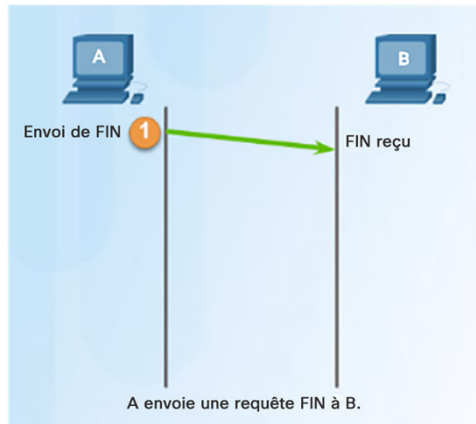
- Étape 2 : le serveur accuse réception et demande l'établissement d'une session avec le client.

- Étape 3 : le client accuse réception de la session de communication avec le serveur.



Processus de communication TCP

Fermeture d'une session TCP (14.5.3)



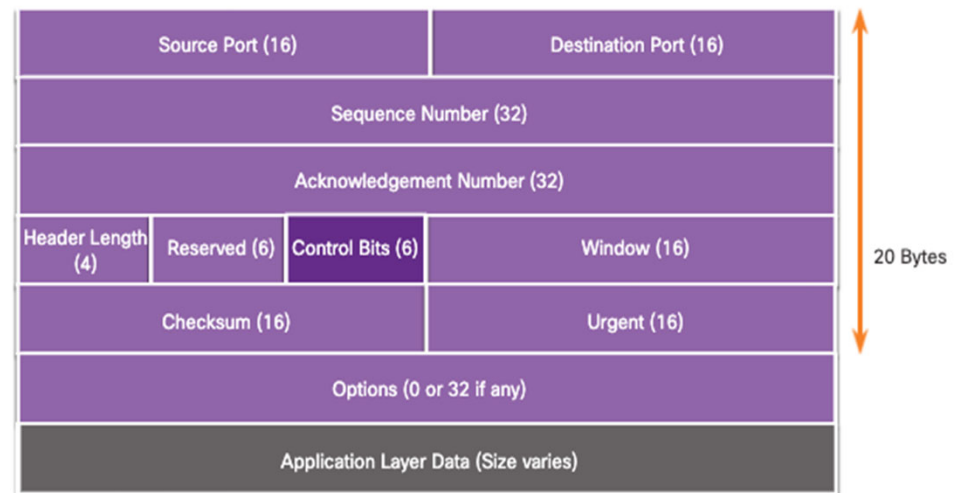
■ Vous fermez votre navigateur web

- La liaison doit être détachée.
- Une trame contenant un indicateur de contrôle FIN (Finish) est inscrit dans l'en-tête de segment.
- Le serveur répond par 2 trames successives contenant :
 - segment ACK et d'un segment FIN.
- Votre PC envoie une trame avec un segment ACK

Analyse de la connexion TCP en trois étapes (Suite) (14.5.4)

Les six indicateurs de bits de contrôle sont les suivants:

- **URG** - Champ de pointeur urgent significatif (Urgent pointer field significant)
- **ACK** - Indicateur d'accusé de réception utilisé dans l'établissement de la connexion et la fin de la session
- **PSH** - Fonction push (Push function)
- **RST** - Réinitialisation de la connexion en cas d'erreur ou de dépassement de délai
- **SYN** - Synchroniser les numéros de séquence utilisés dans l'établissement de connexion
- **FIN** - Plus de données de l'expéditeur et utilisées dans la fin de session



Processus de Communication TCP

Démonstration vidéo – Connexion TCP en trois étapes

Cette vidéo présentera les points suivants:

- La connexion TCP en trois étapes
- Terminaison d'une conversation TCP.

14.5.6 – Vidéo: Vérifiez votre compréhension - Processus de communication TCP

14.5.5 Quiz Connexion TCP en trois étapes

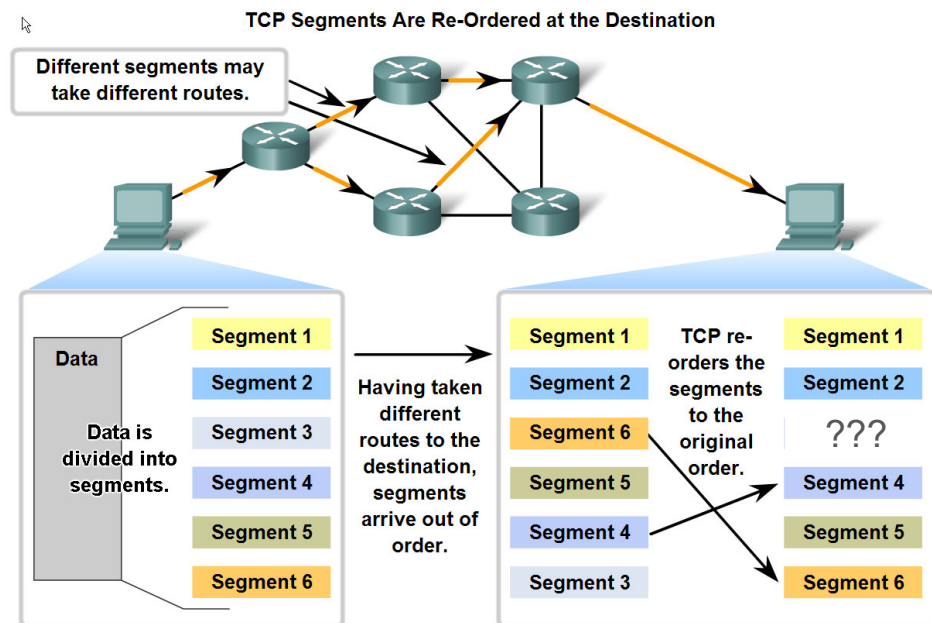
14.5.6 - Vérifiez votre compréhension - Processus de communication TCP

14.6 Fiabilité et contrôle de flux

Fiabilité et contrôle de flux

Fiabilité du protocole TCP – Livraison ordonnée (14.6.1)

- chaque paquet contient un **numéros d'ordre** des segments.
- numéro d'ordre initial est produit par le premier segment. Les autres segments auront un numéro incrémental
- (exemple 101 , 102, 103 etc
- Le récepteur ordonnance les segments, vérifie les segments manqués.



Fiabilité et contrôle de flux

Démonstration vidéo -Fiabilité du protocole TCP - Numéros d'ordre et accusés de réception

Cette vidéo indique un exemple simplifié de fonctionnement du protocole TCP.

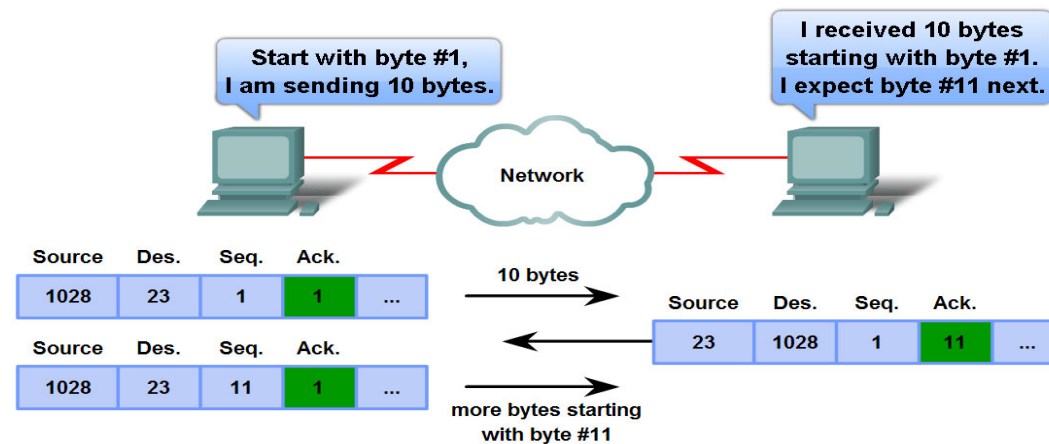
14.6.2 - Vidéo - Fiabilité du protocole TCP - Numéros d'ordre et accusés de réception

Contrôle de flux TCP

- Le récepteur confirme l'arrivée d'un segment valide.
- Il demande un segment suivant, ce peut être le suivant immédiat ou un plus en avant

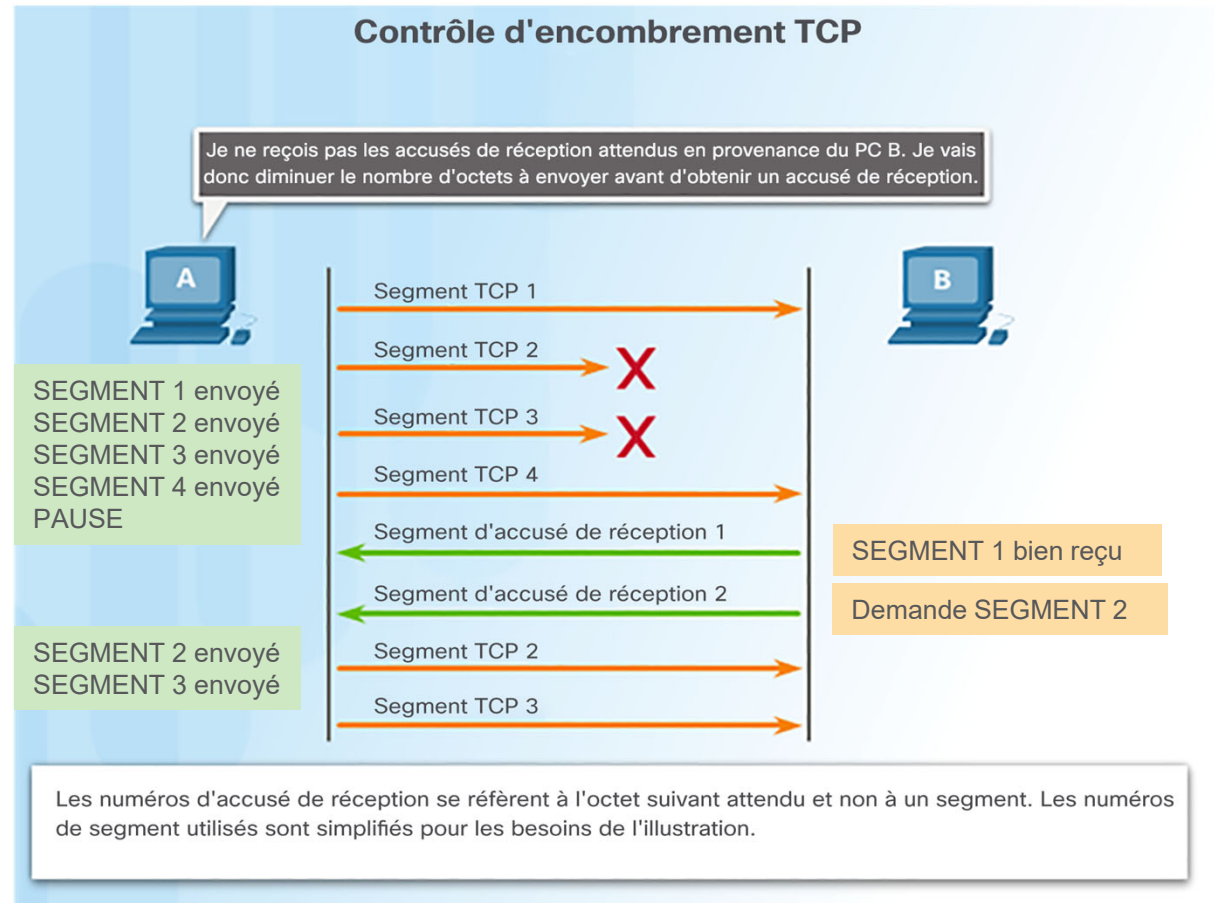
Acknowledgement of TCP Segments

Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...
-------------	------------------	-----------------	-------------------------	-----



Contrôle de flux TCP – Éviter l'encombrement (14.6.3)

- Par exemple : diminuer le nombre d'octets à envoyer avant la réception d'un accusé de réception.

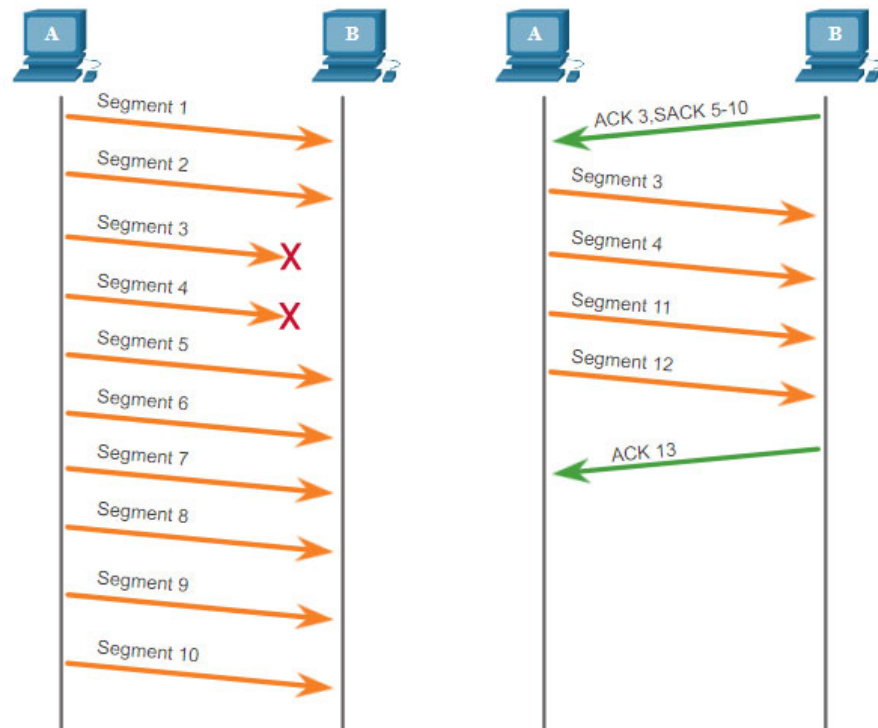


Fiabilité et contrôle du flux

Fiabilité TCP — Perte et retransmission de données (Suite)

reconnaissance sélective (SACK),

Permet de réduire le nombre d'ACK en déclarant un ensemble de segments demandés



Fiabilité et contrôle de flux

Démonstration vidéo – Perte de données et retransmission

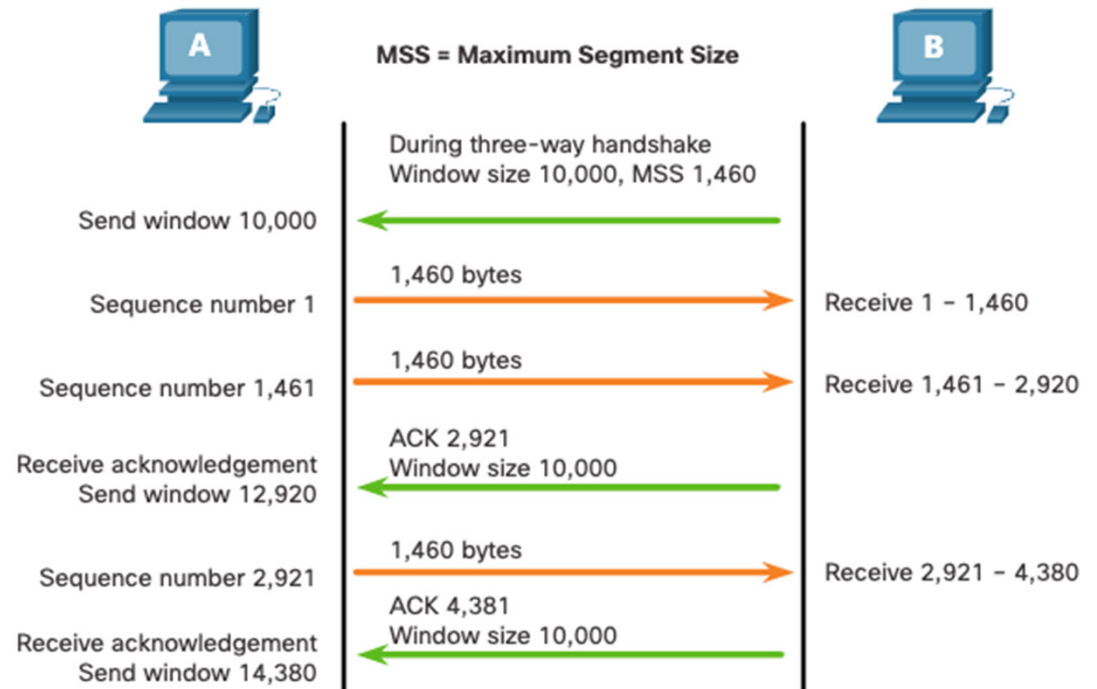
Cette vidéo indique le processus de renvoi des segments qui ne sont pas reçus initialement par la destination.

14.6.4 - Vidéo – Fiabilité du TCP – Perte de données et retransmission

Contrôle de flux TCP – Taille de fenêtre et accusés de réception (14.6.5)

mécanismes de contrôle des flux
comme suit:

- « fenêtre »: nombre total d'octets possibles.
- MSS: taille de chaque segment
- Exemple $10\,000 / 1460 = 6$ segments de suite avant acquittement (ACK)

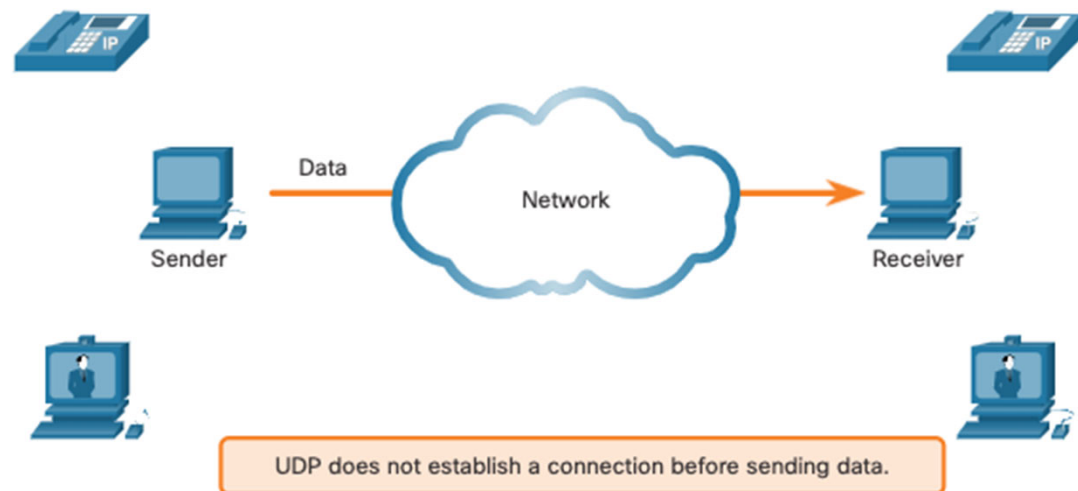


14.7 Communication du protocole UDP

Communication du protocole UDP

Faible surcharge et fiabilité du protocole UDP (14.7.1)

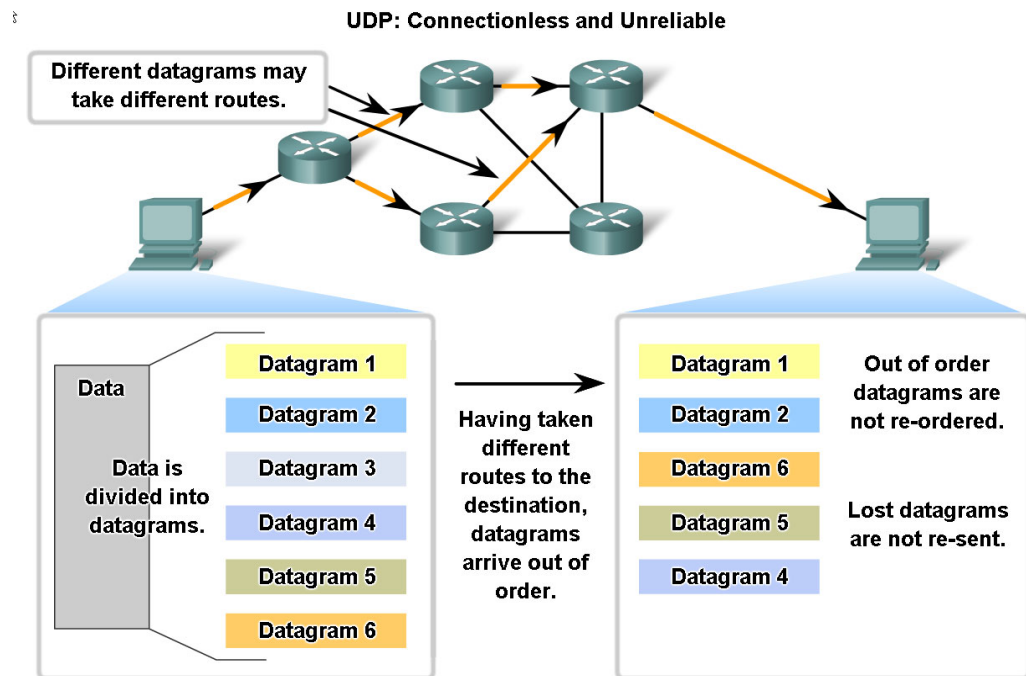
Le protocole UDP n'établit pas de connexion. Le protocole UDP fournit un transport de données à faible surcharge. Son en-tête de datagrammes est petit. UDP n'offre pas de gestion du trafic réseau. UDP ne reprend pas un segment perdu



Communication de protocole UDP

Réassemblage de datagrammes UDP (14.7.2)

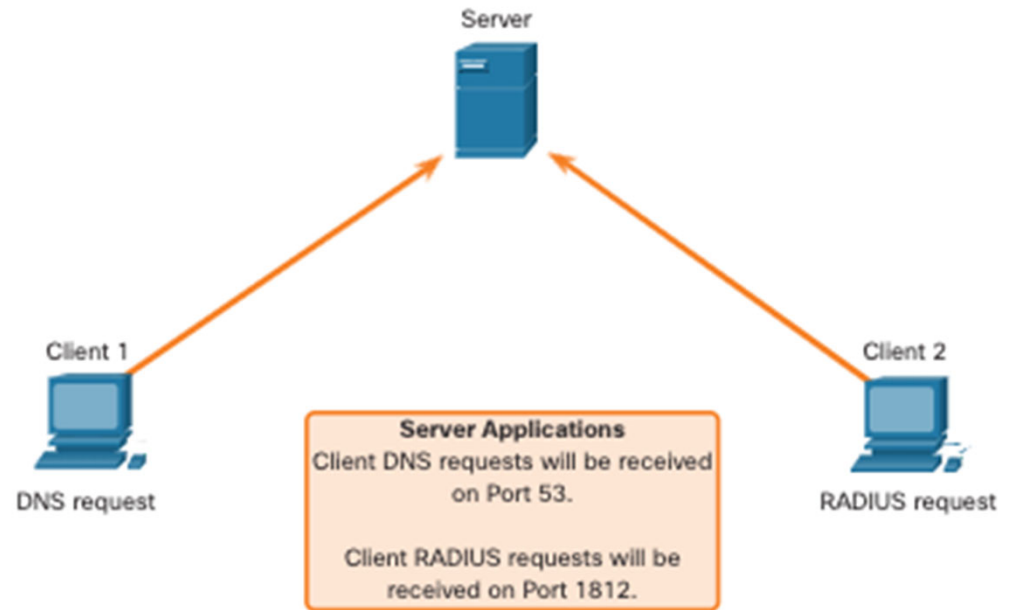
- **pas de suivi** des numéros d'ordre.
- DONC: aucun moyen de réordonner les datagrammes
- DONC: aucun moyen pour retrouver des segments perdus.
- réassembler les données dans l'ordre d'arrivée



Processus et requêtes des serveurs UDP

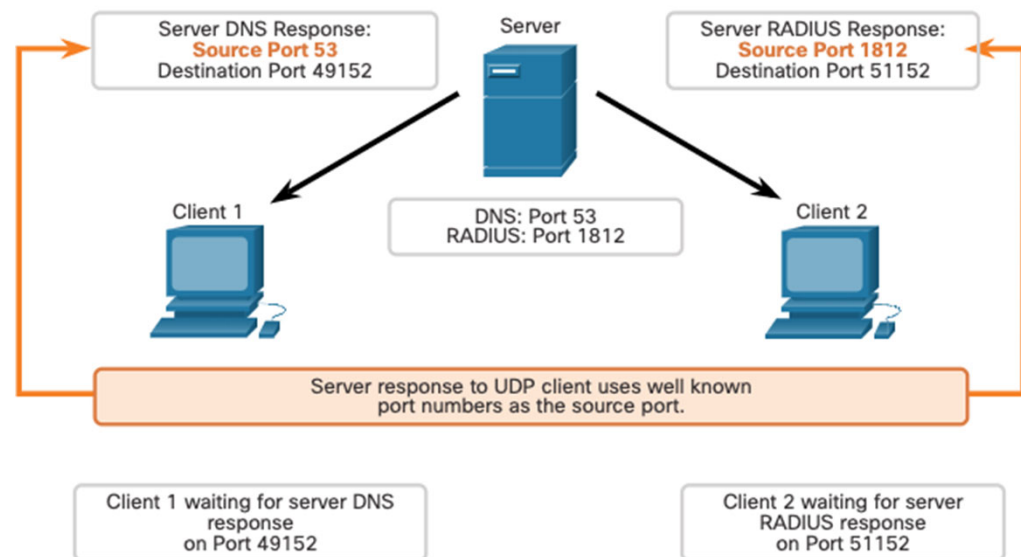
Les applications serveur basées sur l'UDP se voient attribuer des numéros de port connus ou enregistrés.

Le protocole UDP reçoit un datagramme destiné à l'un de ces ports, il transmet les données applicatives à l'application appropriée d'après son numéro de port.



Communication du protocole UDP Processus des clients UDP (14.7.5)

- Le processus client UDP sélectionne dynamiquement un numéro de port dans une plage de numéros de ports et il l'utilise en tant que port source pour la conversation.
- Le port de destination est généralement le numéro de port réservé affecté au processus serveur.
- Une fois qu'un client a choisi le port source et le port de destination, la même paire de ports est utilisée dans l'en-tête de tous les datagrammes employés dans la transaction.



14.8 Module pratique et questionnaire

Module 14: Activities

What activities are associated with this module?

Page #	Activity Type	Activity Name	Optional?
14.1.7	Check Your Understanding	Transportation of Data	Recommended
14.2.5	Check Your Understanding	TCP Overview	Recommended
14.3.5	Check Your Understanding	UDP Overview	Recommended
14.4.5	Check Your Understanding	Port Numbers	Recommended
14.5.5	Video	TCP 3- Handshake	Recommended
14.5.6 BON	Check Your Understanding	TCP Communication Process	Recommended
14.6.2 BON	Video	TCP Reliability- Sequence Numbers and Acknowledgments	Recommended
14.6.4	Video	TCP Reliability – Reliability and Flow control	Recommended
14.6.8 bon	Check Your Understanding	Reliability and Flow Control	Recommended
14.7.5bon	Check Your Understanding	UDP Communication	Recommended
14.8.1	Packet Tracer	Packet Tracer - TCP and UDP Communications	Recommended

Module pratique et questionnaire

Packet Tracer - Communications des protocoles TCP et UDP

Dans le cadre de ce Packet Tracer, vous ferez ce qui suit :

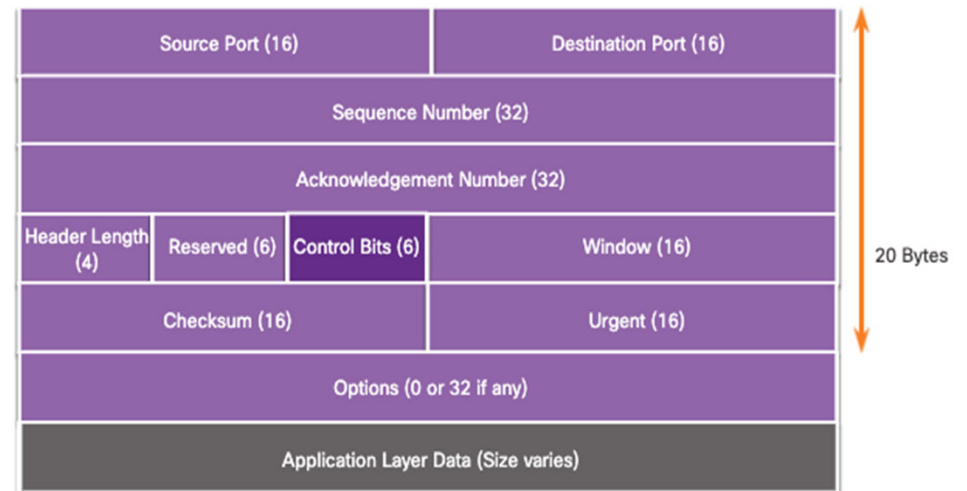
- Générer du trafic sur le réseau en mode Simulation.
- Examiner les fonctionnalités des protocoles TCP et UDP.

Qu'est-ce que j'ai appris dans ce module?

- 2 Protocoles: TCP et UDP
- TCP
 - Établir une liaison de bout en bout
 - Découper un message en segments, numérotés pour les replacer dans le bon ordre à l'arrivée
 - Ajuster le flux de transmission pour éviter les pertes de segments
- UDP
 - Léger : ne gère pas l'ordre d'arrivée et les pertes de segments
 - Effort au « mieux », selon l'achalandage et la qualité du réseau
- Numéro de port
 - « Réservés »: un par application
 - « éphémères »: déterminé au hasard ou par un administrateur de réseau, servant à identifier de façon unique une liaison entre un poste et un serveur

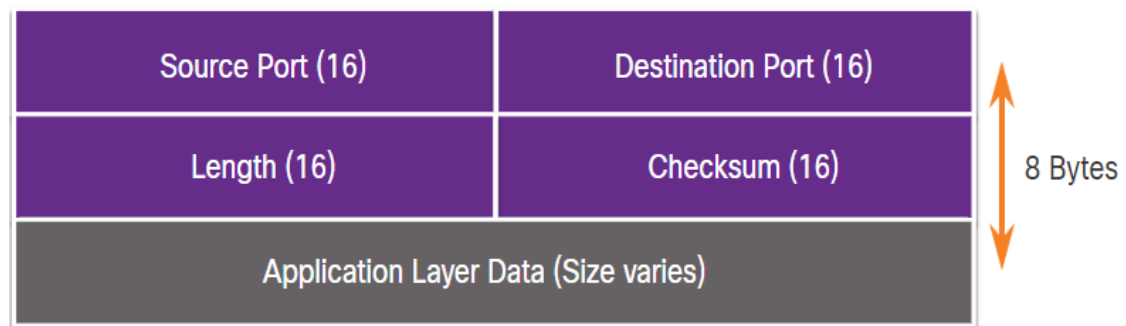
Qu'est-ce que j'ai appris dans ce module (Suite)?

- Un segment TCP contient
 - Un numéro de segment: remettre en ordre le message complet
 - Un numéro d'acquittement: confirme la liste des segments reçus et leur qualité
 - Un ensemble de bits de contrôle
 - Priorité, acquittement de message, congestion, fin de transmission
 - Port source et port destination:
 - Assure la liaison



Qu'est-ce que j'ai appris dans ce module (Suite)?

- Un segment UDP contient
 - Port source et port destination
 - Valeur CRC: assure du segment



New Terms and Commands

- Conversation Multiplexing
- Segments
- Datagrams
- Connection-Oriented Protocol
- Connectionless Protocol
- Stateless Protocol
- Flow Control
- Same-Order Delivery
- Socket Pairs
- **netstat**

- Three-Way Handshake
- SYN
- ACK
- FIN
- URG
- PSH
- RST
- Initial Sequence Number (ISN)
- Selective Acknowledgement (SACK)
- Sliding Window
- Maximum Segment Size (MSS)
- Maximum Transmission Unit (MTU)
- Congestion Avoidance

