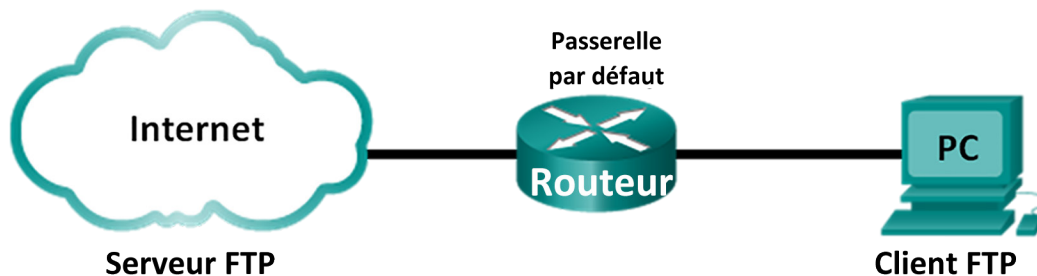


# Travaux pratiques - Utilisation de Wireshark pour l'examen de captures TCP et UDP

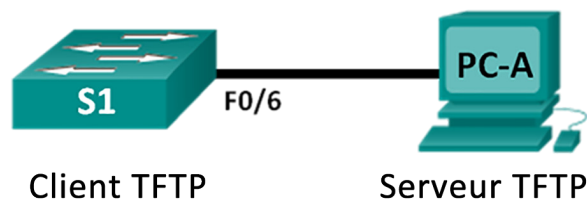
## Topologie - Partie 1 (FTP)

La première partie mettra l'accent sur une capture TCP d'une session FTP. Cette topologie est composée d'un ordinateur disposant d'un accès à Internet.



## Topologie - Partie 2 (TFTP)

La deuxième partie mettra l'accent sur une capture UDP d'une session TFTP. L'ordinateur doit disposer à la fois d'une connexion Ethernet et d'une connexion de console avec le commutateur S1.



## Table d'adressage (deuxième partie)

Appareil	Interface	Adresse IP	Masque de sous-réseau	Passerelle par défaut
S1	VLAN 1	192.168.1.1	255.255.255.0	N/A
PC-A	Carte réseau	192.168.1.3	255.255.255.0	192.168.1.1

## Objectifs

**Partie 1 :** Identifier les champs d'en-tête TCP ainsi que les opérations TCP à l'aide de la capture de session FTP de Wireshark

**Partie 2 :** Identifier les champs d'en-tête UDP ainsi que les opérations UDP à l'aide de la capture de session TFTP de Wireshark

## Contexte/scénario

Les deux protocoles de la couche transport TCP/IP sont le protocole TCP, défini dans le document RFC 761, et le protocole UDP, défini dans le document RFC 768. Les deux protocoles prennent en charge les communications du protocole de couche supérieure. Par exemple, TCP prend en charge la couche transport

pour les protocoles HTTP (HyperText Transfer Protocol) et FTP, entre autres. UDP fournit notamment la prise en charge de la couche transport pour le système de noms de domaine (DNS) et TFTP.

**Remarque** : la capacité à comprendre les éléments des en-têtes TCP et UDP ainsi que les opérations représentent une compétence cruciale pour les ingénieurs réseau.

Dans la première partie de ces travaux pratiques, vous utiliserez l'outil open source Wireshark pour capturer et analyser les champs d'en-tête de protocole TCP pour les transferts de fichiers FTP entre l'ordinateur hôte et un serveur FTP anonyme. L'utilitaire de ligne de commande Windows permet de se connecter à un serveur FTP anonyme et de télécharger un fichier. Dans la deuxième partie de ces travaux pratiques, vous utiliserez Wireshark pour capturer et analyser des champs d'en-tête UDP pour les transferts de fichiers TFTP entre l'ordinateur hôte et S1.

**Remarque** : les commutateurs utilisés sont des modèles Cisco Catalyst 2960 équipés de Cisco IOS version 15.0(2) (image lanbasek9). D'autres commutateurs et versions de Cisco IOS peuvent être utilisés. Selon le modèle et la version de Cisco IOS, les commandes disponibles et le résultat produit peuvent varier de ceux figurant dans les travaux pratiques.

**Remarque** : assurez-vous que la mémoire du commutateur a été effacée et vérifiez l'absence de configurations initiales. En cas de doute, contactez votre formateur.

**Remarque** : la première partie suppose que l'ordinateur dispose d'un accès à Internet et ne peut pas être effectuée avec Netlab. La deuxième partie est compatible avec Netlab.

### Ressources requises - Partie 1 (FTP)

1 ordinateur (Windows 7, 8 ou 10, équipé d'un accès à Internet, d'un accès aux invites de commandes et de Wireshark)

### Ressources requises - Partie 2 (TFTP)

- 1 commutateur (Cisco 2960 équipé de Cisco IOS version 15.0(2) image lanbasek9 ou similaire)
- 1 ordinateur (Windows 7, 8 ou 10, avec Wireshark et un serveur TFTP, tel que tftpd32, installé)
- Câble de console pour configurer les périphériques Cisco IOS via le port de console
- Câble Ethernet conformément à la topologie

## Partie 1 : Identifier les champs d'en-tête TCP ainsi que les opérations TCP à l'aide de la capture de session FTP de Wireshark

Dans la première partie, vous utiliserez Wireshark pour capturer une session FTP et examiner les champs d'en-tête TCP.

### Étape 1 : Démarrez une capture Wireshark.

- a. Interrompez tout le trafic réseau superflu, par exemple, fermez le navigateur web, pour limiter le volume du trafic lors de la capture Wireshark.
- b. Lancez la capture Wireshark.

### Étape 2 : Téléchargez le fichier Readme (Lisez-moi).

- a. À partir de l'invite de commande, saisissez **ftp ftp.cdc.gov**.

- b. Connectez-vous au site FTP du Centre pour le contrôle et la prévention des maladies (Center for Disease Control and Prevention, CDC) avec l'utilisateur **anonymous** et aucun mot de passe.

```
C:\> ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
```

- c. Localisez et téléchargez le fichier Readme (Lisez-moi) en exécutant la commande **ls** afin d'afficher la liste des fichiers.

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection.
.change.dir
.message
pub
Readme
Siteinfo
up.htm
w3c
welcome.msg
226 Transfer complete.
ftp: 75 bytes received in 0.00Seconds 75000.00Kbytes/sec.
```

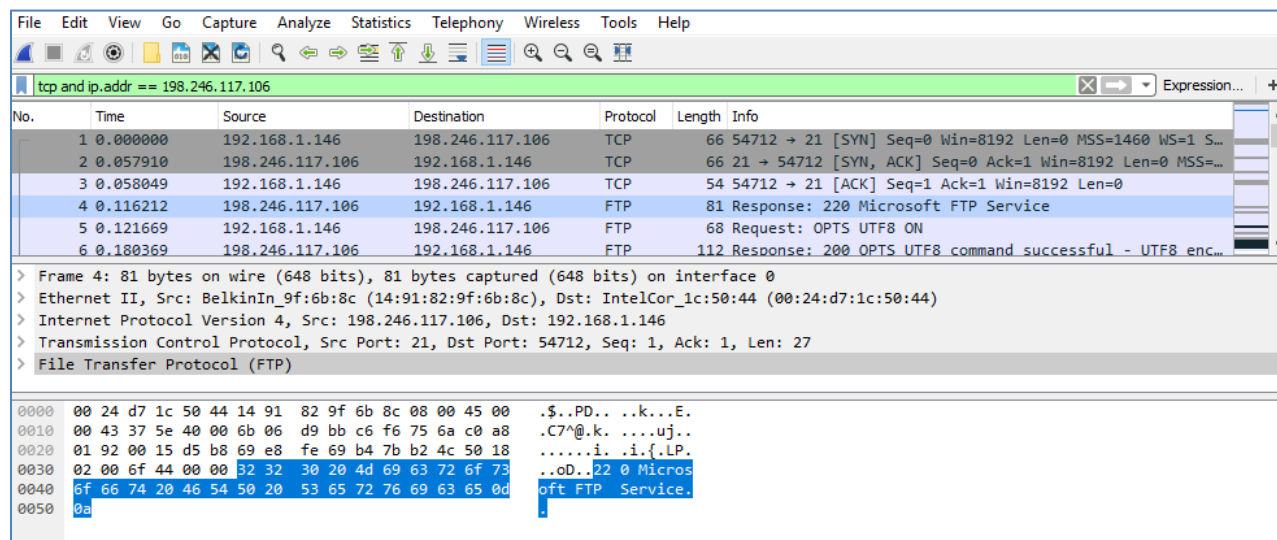
- d. Exécutez la commande **get Readme** pour télécharger le fichier. Lorsque le téléchargement est terminé, entrez la commande **quit** pour quitter.

```
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection.
226 Transfer complete.
ftp: 1428 bytes received in 0.08Seconds 18.08Kbytes/sec.
```

### Étape 3 : Arrêtez la capture Wireshark.

### Étape 4 : Affichez la fenêtre principale de Wireshark.

Wireshark a capturé de nombreux paquets pendant la session FTP sur <ftp.cdc.gov>. Pour limiter la quantité de données à analyser, tapez **tcp and ip.addr == 198.246.117.106** dans la zone **Filter: entry** et cliquez sur **Apply**. L'adresse IP 198.246.117.106 est l'adresse de <ftp.cdc.gov> à ce moment-là.



### Étape 5 : Analysez les champs TCP.

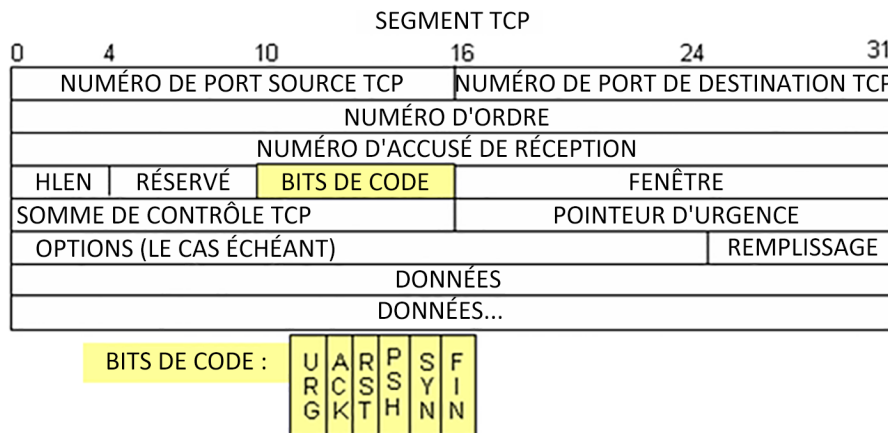
Une fois le filtre TCP appliqué, les trois premières trames dans le volet de la liste des paquets (section supérieure) affichent le protocole TCP de la couche transport, créant ainsi une session fiable. La séquence de [SYN], [SYN, ACK] et [ACK] illustre l'échange en trois étapes.

1	0.000000	192.168.1.146	198.246.117.106	TCP	66	54712 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 S...
2	0.057910	198.246.117.106	192.168.1.146	TCP	66	21 → 54712 [SYN, ACK] Seq=0 Ack=1 Win=8192 L...
3	0.058049	192.168.1.146	198.246.117.106	TCP	54	54712 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0

TCP est couramment utilisé au cours d'une session pour contrôler la transmission et l'arrivée des datagrammes ainsi que pour gérer la taille des fenêtres. Pour chaque échange de données entre le client FTP et le serveur FTP, une nouvelle session TCP est démarrée. Au terme du transfert de données, la session TCP est fermée. Une fois la session FTP terminée, TCP exécute un arrêt, puis une déconnexion.

Dans Wireshark, des informations TCP détaillées sont disponibles dans le volet de détails des paquets (section centrale). Sélectionnez le premier datagramme TCP à partir de l'ordinateur hôte et développez le datagramme TCP. Le datagramme TCP développé ressemble au volet de détails des paquets indiqué ci-dessous.

```
> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: IntelCor_1c:50:44 (00:24:d7:1c:50:44), Dst: BelkinIn_9f:6b:8c (14:91:82:9f:6b:8c)
> Internet Protocol Version 4, Src: 192.168.1.146, Dst: 198.246.117.106
▼ Transmission Control Protocol, Src Port: 54712, Dst Port: 21, Seq: 0, Len: 0
  Source Port: 54712
  Destination Port: 21
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
  1000 .... = Header Length: 32 bytes (8)
  ▼ Flags: 0x002 (SYN)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    ....0... = Congestion Window Reduced (CWR): Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....0... = Acknowledgment: Not set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    > ....0... = Syn: Set
    ....0... = Fin: Not set
    [TCP Flags: .....S.]
  Window size value: 8192
  [Calculated window size: 8192]
  Checksum: 0x13e8 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
```



L'illustration ci-dessus est un schéma de datagramme TCP. Une explication de chaque champ est fournie pour référence :

- Le **numéro de port source TCP** (TCP Source Port Number) appartient à l'hôte de session TCP qui a ouvert une connexion. Il s'agit généralement d'une valeur aléatoire supérieure à 1 023.
- Le **numéro de port de destination TCP** (TCP Destination Port Number) permet d'identifier le protocole de couche supérieure ou l'application sur le site distant. Les valeurs comprises entre 0 et 1 023 représentent les « ports réservés » et sont associées aux services et applications standard (comme décrit dans le document RFC 1700, par exemple Telnet, FTP et HTTP). La combinaison de l'adresse IP source, du port source, de l'adresse IP de destination et du port de destination identifie de façon unique la session à la fois vis-à-vis de l'émetteur et du récepteur.

**Remarque** : dans la capture Wireshark ci-dessous, le port de destination est le 21, ce qui correspond au FTP. Les serveurs FTP écoutent sur le port 21 pour les connexions clientes FTP.

- Le **numéro d'ordre** (Sequence Number) indique le numéro du dernier octet dans un segment.
- Le **numéro d'accusé de réception** (Acknowledgment Number) indique l'octet suivant prévu par le récepteur.
- Les **bits de code** ont une signification spécifique dans la gestion des sessions et dans le traitement des segments. Valeurs intéressantes :
  - ACK : accusé de réception d'un segment.
  - SYN (Synchronize, uniquement défini lorsqu'une nouvelle session TCP est négociée au cours de la connexion en trois étapes) ;
  - FIN (Finish, requête pour fermer la session TCP) ;
- La **taille de la fenêtre** (Window size) est la valeur de la fenêtre glissante. Elle détermine le nombre d'octets pouvant être envoyés avant d'attendre l'accusé de réception.
- Le **pointeur d'urgence** (Urgent pointer) n'est utilisé qu'avec un indicateur URG (Urgent), lorsque l'émetteur doit envoyer des données urgentes au récepteur.
- Les **options** ne contiennent actuellement qu'une seule valeur, définie comme étant la taille maximale d'un segment TCP (valeur facultative).

À l'aide de la capture Wireshark du démarrage de la première session TCP (bit SYNC défini sur 1), renseignez les informations concernant l'en-tête TCP.

Du PC au serveur CDC (seul le bit SYN est défini sur 1) :

Adresse IP source	
Adresse IP de destination	
Numéro du port source	
Numéro du port de destination	
Numéro d'ordre	
Numéro d'accusé de réception	
Longueur de l'en-tête	
Taille de fenêtre	

Dans la deuxième capture Wireshark filtrée, le serveur FTP CDC reconnaît la requête de l'ordinateur. Notez les valeurs des bits SYN et ACK.

```
> Frame 2: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: BelkinIn_9f:6b:8c (14:91:82:9f:6b:8c), Dst: IntelCor_1c:50:44 (00:24:d7:1c:50:44)
> Internet Protocol Version 4, Src: 198.246.117.106, Dst: 192.168.1.146
▼ Transmission Control Protocol, Src Port: 21, Dst Port: 54712, Seq: 0, Ack: 1, Len: 0
  Source Port: 21
  Destination Port: 54712
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▼ Flags: 0x012 (SYN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    ....0... = Congestion Window Reduced (CWR): Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    > ....1... = Syn: Set
    ....0... = Fin: Not set
    [TCP Flags: .....A..S.]
  Window size value: 8192
  [Calculated window size: 8192]
  Checksum: 0xabcd [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [SEQ/ACK analysis]
```

Indiquez les informations suivantes concernant le message SYN-ACK.

Adresse IP source	
Adresse IP de destination	
Numéro du port source	
Numéro du port de destination	
Numéro d'ordre	
Numéro d'accusé de réception	
Longueur de l'en-tête	
Taille de fenêtre	

Lors de l'étape finale de la négociation visant à établir une communication, l'ordinateur envoie un accusé de réception au serveur. Notez que seul le bit ACK est défini sur 1 et que le numéro d'ordre a été incrémenté à 1.

```
> Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: IntelCor_1c:50:44 (00:24:d7:1c:50:44), Dst: BelkinIn_9f:6b:8c (14:91:82:9f:6b:8c)
> Internet Protocol Version 4, Src: 192.168.1.146, Dst: 198.246.117.106
▼ Transmission Control Protocol, Src Port: 54712, Dst Port: 21, Seq: 1, Ack: 1, Len: 0
  Source Port: 54712
  Destination Port: 21
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  0101 .... = Header Length: 20 bytes (5)
  ▼ Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    ....0... = Congestion Window Reduced (CWR): Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    ....0... = Syn: Not set
    ....0... = Fin: Not set
    [TCP Flags: .....A....]
  Window size value: 8192
  [Calculated window size: 8192]
  [Window size scaling factor: 1]
  Checksum: 0xec50 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > [SEQ/ACK analysis]
```

Indiquez les informations suivantes concernant le message ACK.

Adresse IP source	
Adresse IP de destination	
Numéro du port source	
Numéro du port de destination	
Numéro d'ordre	
Numéro d'accusé de réception	
Longueur de l'en-tête	
Taille de fenêtre	

Combien d'autres datagrammes TCP contenaient un bit SYN ?



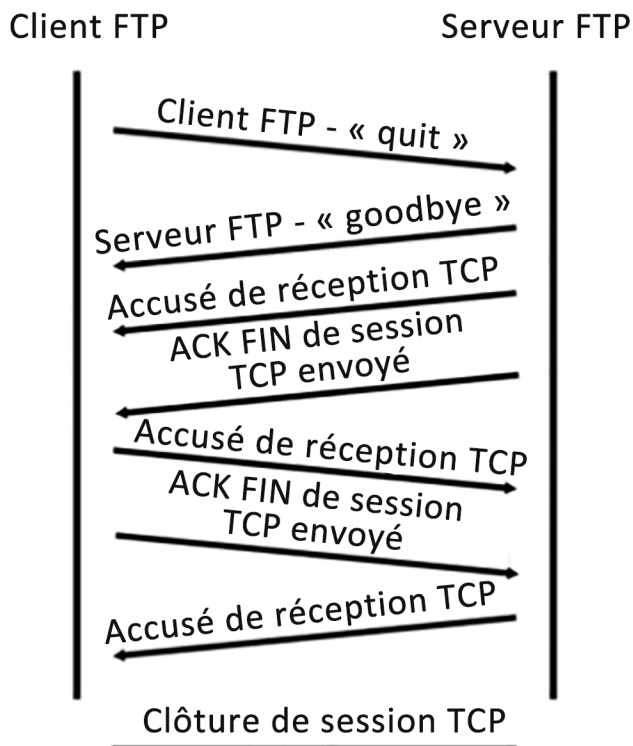
Une fois qu'une session TCP est établie, le trafic FTP peut circuler entre le PC et le serveur FTP. Le client FTP et le serveur communiquent entre eux sans tenir compte du contrôle et de la gestion de la session par TCP. Lorsque le serveur FTP envoie *Response: 220* au client FTP, la session TCP sur le client FTP envoie un accusé de réception à la session TCP sur le serveur. Cette séquence est visible dans la capture Wireshark ci-dessous.

4	0.116212	198.246.117.106	192.168.1.146	FTP	81 Response: 220 Microsoft FTP Service
5	0.121669	192.168.1.146	198.246.117.106	FTP	68 Request: OPTS UTF8 ON
6	0.180369	198.246.117.106	192.168.1.146	FTP	112 Response: 200 OPTS UTF8 command successful - UTF8 enc...

>	Frame 4: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
>	Ethernet II, Src: BelkinIn_9f:6b:8c (14:91:82:9f:6b:8c), Dst: IntelCor_1c:50:44 (00:24:d7:1c:50:44)
>	Internet Protocol Version 4, Src: 198.246.117.106, Dst: 192.168.1.146
>	Transmission Control Protocol, Src Port: 21, Dst Port: 54712, Seq: 1, Ack: 1, Len: 27
>	File Transfer Protocol (FTP)
>	220 Microsoft FTP Service\r\n
>	Response code: Service ready for new user (220)
>	Response arg: Microsoft FTP Service

Une fois la session FTP terminée, le client FTP envoie une commande pour « quitter ». Le serveur FTP accuse réception de la fin de la session FTP avec un message *Response: 221 Goodbye*. À ce stade, la session TCP du serveur FTP envoie un datagramme TCP au client FTP, et annonce ainsi la fin de la session TCP. La session TCP du client FTP accuse réception du datagramme de fin, puis envoie la fin de sa propre session TCP. Lorsque l'émetteur de la fin de la session TCP, le serveur FTP, reçoit une fin en double, un datagramme ACK est envoyé pour accuser réception de la fin et la session TCP est fermée. Cette séquence est visible dans le schéma et la capture ci-dessous.



En appliquant un filtre **ftp**, la séquence entière du trafic FTP peut être examinée dans Wireshark. Notez la séquence des événements au cours de cette session FTP. Le nom d'utilisateur **anonymous** a été utilisé pour récupérer le fichier Readme (Lisez-moi). Une fois le fichier transféré, l'utilisateur a mis fin à la session FTP.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.116212	198.246.117.106	192.168.1.146	FTP	81	Response: 220 Microsoft FTP Service
5	0.121669	192.168.1.146	198.246.117.106	FTP	68	Request: OPTS UTF8 ON
6	0.180369	198.246.117.106	192.168.1.146	FTP	112	Response: 200 OPTS UTF8 command successful - UTF8 encoding
18	34.993981	192.168.1.146	198.246.117.106	FTP	70	Request: USER anonymous
19	35.052234	198.246.117.106	192.168.1.146	FTP	126	Response: 331 Anonymous access allowed, send identity (e-mail address)
21	39.133664	192.168.1.146	198.246.117.106	FTP	61	Request: PASS
22	39.188301	198.246.117.106	192.168.1.146	FTP	75	Response: 230 User logged in.
26	43.325986	192.168.1.146	198.246.117.106	FTP	82	Request: PORT 192,168,1,146,213,185
29	43.381803	198.246.117.106	192.168.1.146	FTP	84	Response: 200 PORT command successful.
30	43.390255	192.168.1.146	198.246.117.106	FTP	60	Request: NLST
35	43.447231	198.246.117.106	192.168.1.146	FTP	108	Response: 125 Data connection already open; Transfer starting
36	43.448271	198.246.117.106	192.168.1.146	FTP	78	Response: 226 Transfer complete.
40	55.104521	192.168.1.146	198.246.117.106	FTP	82	Request: PORT 192,168,1,146,213,186
43	55.171392	198.246.117.106	192.168.1.146	FTP	84	Response: 200 PORT command successful.
44	55.182471	192.168.1.146	198.246.117.106	FTP	67	Request: RETR Readme
49	55.247925	198.246.117.106	192.168.1.146	FTP	108	Response: 125 Data connection already open; Transfer starting
53	55.294530	198.246.117.106	192.168.1.146	FTP	78	Response: 226 Transfer complete.
56	61.170643	192.168.1.146	198.246.117.106	FTP	60	Request: QUIT
58	61.723390	198.246.117.106	192.168.1.146	FTP	68	Response: 221 Goodbye.

Appliquez le filtre TCP à nouveau dans Wireshark pour examiner la fin de la session TCP. Quatre paquets sont transmis à la fin de la session TCP. Comme la connexion TCP est en mode duplex intégral, chaque sens doit se terminer indépendamment. Examinez les adresses source et de destination.

Dans cet exemple, le serveur FTP n'a plus de données à envoyer dans le flux. Il envoie un segment avec l'indicateur FIN défini dans la trame 59. L'ordinateur envoie un segment ACK pour accuser réception de l'indicateur FIN afin de fermer la session du serveur vers le client dans la trame 60.

Dans la trame 61, le PC envoie un paquet FIN au serveur FTP pour mettre fin à la session TCP. Le serveur FTP répond par un paquet ACK pour accuser réception du paquet FIN envoyé par le PC dans la trame 65. À présent, la session TCP est interrompue entre le serveur FTP et le PC.

No.	Time	Source	Destination	Protocol	Length	Info
57	61.457683	192.168.1.146	198.246.117.106	TCP	60	[TCP Retransmission] 54712 → 21 [PSH, ACK] Seq=113 Ack=410 Win=0 Len=0
58	61.723390	198.246.117.106	192.168.1.146	FTP	68	Response: 221 Goodbye.
59	61.723391	198.246.117.106	192.168.1.146	TCP	54	21 → 54712 [FIN, ACK] Seq=409 Ack=119 Win=130816 Len=0
60	61.723507	192.168.1.146	198.246.117.106	TCP	54	54712 → 21 [ACK] Seq=119 Ack=410 Win=7784 Len=0
61	61.729268	192.168.1.146	198.246.117.106	TCP	54	54712 → 21 [FIN, ACK] Seq=119 Ack=410 Win=7784 Len=0
62	61.752612	198.246.117.106	192.168.1.146	TCP	68	[TCP Out-Of-Order] 21 → 54712 [FIN, PSH, ACK] Seq=395... Len=0
63	61.752678	192.168.1.146	198.246.117.106	TCP	66	[TCP Dup ACK 60#1] 54712 → 21 [ACK] Seq=120 Ack=410 Win=0 Len=0
64	62.028356	198.246.117.106	192.168.1.146	TCP	66	[TCP Dup ACK 58#1] 21 → 54712 [ACK] Seq=410 Ack=119 Win=0 Len=0
65	62.028357	198.246.117.106	192.168.1.146	TCP	54	21 → 54712 [ACK] Seq=410 Ack=120 Win=130816 Len=0

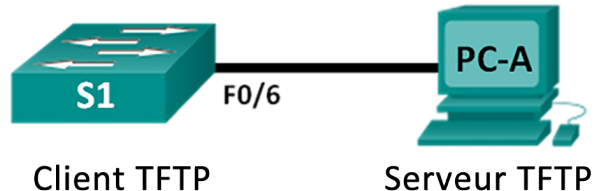
  

>	Frame 59: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
>	Ethernet II, Src: BelkinIn_9f:6b:8c (14:91:82:9f:6b:8c), Dst: IntelCor_1c:50:44 (00:24:d7:1c:50:44)
>	Internet Protocol Version 4, Src: 198.246.117.106, Dst: 192.168.1.146
>	Transmission Control Protocol, Src Port: 21, Dst Port: 54712, Seq: 409, Ack: 119, Len: 0

## Partie 2 : Identifier les champs d'en-tête UDP ainsi que les opérations UDP à l'aide de la capture de session TFTP de Wireshark

Dans la deuxième partie, vous utiliserez Wireshark pour capturer une session TFTP et examiner les champs d'en-tête UDP.

**Étape 1 : Installez cette topologie physique et préparez la capture TFTP.**



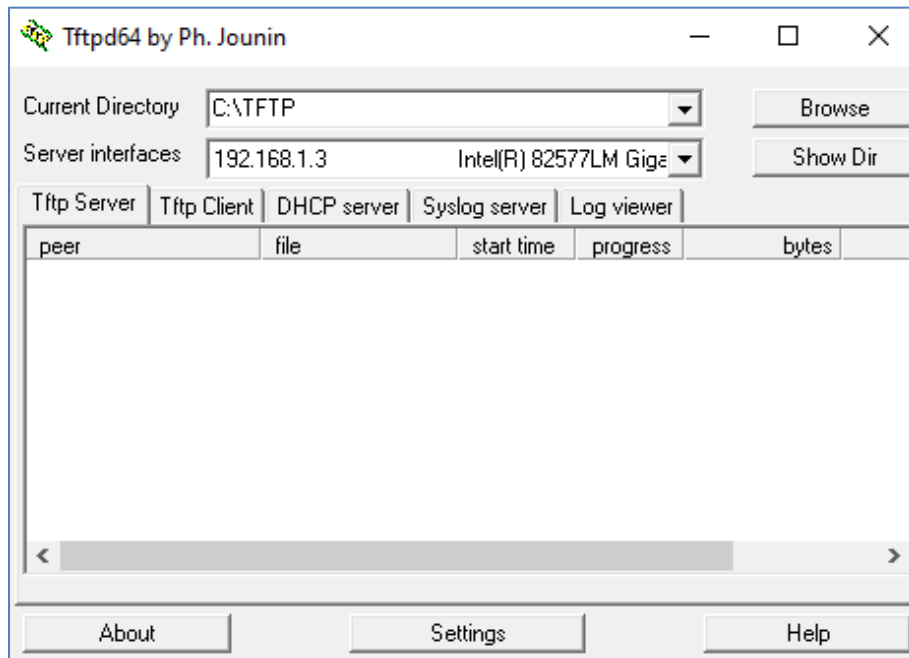
- Établissez une connexion de console et une connexion Ethernet entre PC-A et S1.
- Configurez manuellement l'adresse IP sur le PC à la valeur 192.168.1.3. Il n'est pas obligatoire de définir la passerelle par défaut.
- Configurez le commutateur. Attribuez l'adresse IP 192.168.1.1 à VLAN 1. Vérifiez la connectivité avec le PC en envoyant une requête ping à 192.168.1.3. Le cas échéant, procédez à un dépannage.

```
Switch> enable
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.1 255.255.255.0
S1(config-if)# no shut
*Mar 1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar 1 00:37:50.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed state to up
S1(config-if)# end
S1# ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
d. Enregistrez la configuration en cours dans la mémoire NVRAM.
S1# copy run start
```

**Étape 2 : Préparez le serveur TFTP sur le PC.**

- S'il n'existe pas encore, créez un dossier sur le disque C: de l'ordinateur appelé **TFTP**. Les fichiers du commutateur seront copiés à cet emplacement.
- Démarrez **tftpd32** ou **Tftpd64** sur l'ordinateur.
- Cliquez sur **Parcourir** et remplacez le répertoire en cours par **C:\TFTP**.

Le serveur TFTP doit être similaire à celui-ci :



Notez que **Répertoire actif** indique l'interface du serveur TFTP (PC-A) avec l'adresse IP **192.168.1.3**.

- d. Testez la possibilité de copier un fichier en utilisant TFTP à partir du commutateur vers le PC. Le cas échéant, procédez à un dépannage.

```
S1# copy start tftp
```

```
Address or name of remote host []? 192.168.1.3
```

```
Destination filename [s1-config]?
```

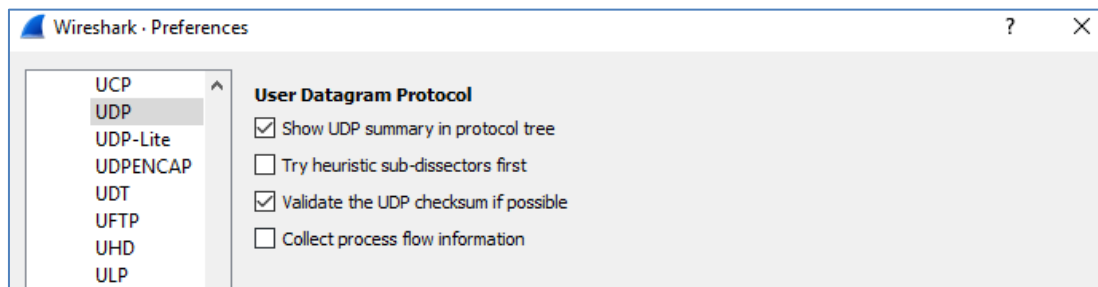
```
!!
```

```
1083 bytes copied in 0,84 secs
```

Si vous voyez que le fichier a été copié, vous êtes prêt à passer à l'étape suivante. Si le fichier n'a pas été copié, procédez à un dépannage. Si vous obtenez l'erreur **%Error opening tftp (Permission denied)**, vérifiez d'abord que votre pare-feu ne bloque pas le protocole TFTP et que vous effectuez la copie vers un emplacement pour lequel votre nom d'utilisateur dispose des autorisations appropriées, comme l'ordinateur de bureau.

### Étape 3 : Capturer une session TFTP dans Wireshark

- a. Ouvrez Wireshark. À partir du menu **Edit** (Edition), choisissez **Preferences** (Préférences) et cliquez sur le signe (+) pour développer **Protocols** (Protocoles). Faites défiler vers le bas, puis sélectionnez **UDP**. Activez la case à cocher **Validate the UDP Checksum if Possible** et cliquez sur **OK**.

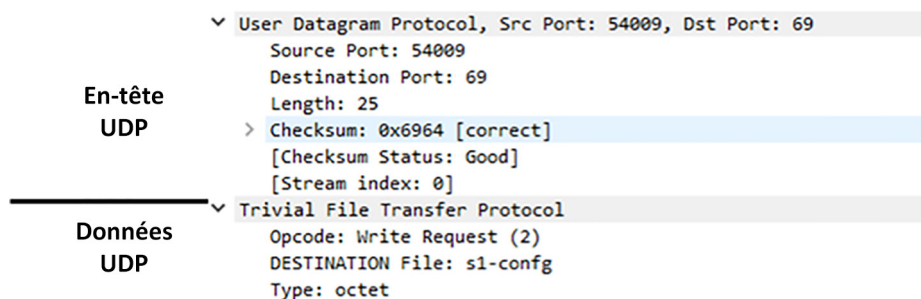


- b. Démarrez une capture Wireshark.
- c. Exécutez la commande **copy start tftp** sur le commutateur.
- d. Arrêtez la capture Wireshark.

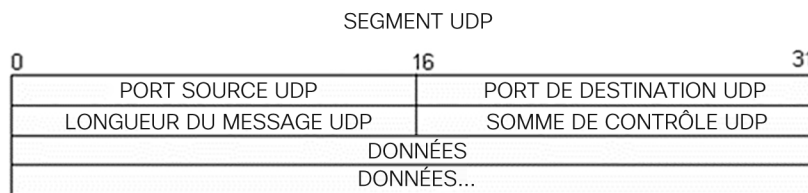
No.	Time	Source	Destination	Protocol	Length	Info
10	17.006137	192.168.1.1	192.168.1.3	TFTP	64	Write Request, File: s1-config, Transfer type
11	17.008212	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 0
12	17.012084	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 1
13	17.012376	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 1
14	17.014029	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 2
15	17.014133	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 2
16	17.017114	192.168.1.1	192.168.1.3	TFTP	105	Data Packet, Block: 3 (last)
17	17.017219	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 3

- e. Définissez le filtre sur **tftp**. Les informations affichées doivent être similaires à celles figurant ci-dessus :  
Ce transfert TFTP permet d'analyser les opérations UDP de la couche transport.

Des informations UDP détaillées sont disponibles dans le volet de détails des paquets Wireshark. Sélectionnez le premier datagramme UDP à partir de l'ordinateur hôte, et déplacez le pointeur de la souris vers le volet de détails des paquets. Il peut s'avérer nécessaire de modifier le volet de détails des paquets et de développer l'enregistrement UDP en cliquant sur la zone de développement du protocole. Le datagramme UDP développé doit être semblable au schéma ci-dessous.



La figure ci-dessous représente un schéma de datagramme UDP. Les informations d'en-tête sont peu nombreuses par rapport au datagramme TCP. De même que pour le protocole TCP, chaque datagramme UDP est identifié par les ports source et de destination UDP.



À l'aide de la capture Wireshark du premier datagramme UDP, renseignez les informations concernant l'en-tête UDP. La somme de contrôle est une valeur hexadécimale (base 16), identifiée par le code 0x précédent :

Adresse IP source	
Adresse IP de destination	
Numéro du port source	
Numéro du port de destination	
Longueur du message UDP	
Somme de contrôle UDP	

De quelle manière UDP vérifie-t-il l'intégrité du datagramme ?

Examinez la première trame renvoyée par le serveur tftpd. Complétez les informations sur l'en-tête UDP :

Adresse IP source	
Adresse IP de destination	
Numéro du port source	
Numéro du port de destination	
Longueur du message UDP	
Somme de contrôle UDP	

```
▼ User Datagram Protocol, Src Port: 65001, Dst Port: 54009
  Source Port: 65001
  Destination Port: 54009
  Length: 12
  > Checksum: 0x8372 incorrect, should be 0xab99 (maybe caused by "UDP checksum offload?")
    [Checksum Status: Bad]
    [Stream index: 1]
▼ Trivial File Transfer Protocol
  Opcode: Acknowledgement (4)
  [DESTINATION File: s1-config]
  Block: 0
```

Remarque : le datagramme UDP de retour possède un port de source UDP différent. Toutefois, ce dernier sert au transfert TFTP restant. Étant donné que la connexion n'est pas fiable, seul le port source d'origine utilisé pour commencer la session TFTP sert à gérer le transfert TFTP.

Notez également que la somme de contrôle UDP est incorrecte. Ceci provient très probablement du déchargement de la somme de contrôle UDP. Pour plus d'informations sur la raison de cet événement, effectuez une recherche sur « UDP checksum offload » (Déchargement de la somme de contrôle).

### Remarques générales

Ces travaux pratiques ont permis aux participants d'analyser les opérations des protocoles TCP et UDP à partir de sessions FTP et TFTP capturées. En quoi le protocole TCP gère-t-il la communication différemment du protocole UDP ?

---

---

---

---

### Challenge

Étant donné que les protocoles FTP et TFTP ne sont pas sécurisés, toutes les données transférées sont envoyées en texte clair. Ceci comprend les ID d'utilisateurs, les mots de passe ou le contenu des fichiers en clair. L'analyse de la session FTP de couche supérieure permet d'identifier rapidement l'ID d'utilisateur, le mot de passe ainsi que les mots de passe pour le fichier de configuration. L'analyse des données TFTP de couche supérieure est un peu plus complexe. Toutefois, le champ de données peut être examiné et les informations d'ID d'utilisateur et de mot de passe pour la configuration peuvent être extraites.

### Nettoyage

Sauf indication contraire de votre formateur :

- 1) Supprimez les fichiers qui ont été copiés sur votre ordinateur.
- 2) Supprimez les configurations sur S1.
- 3) Supprimez l'adresse IP manuelle de l'ordinateur et restaurez la connectivité Internet.