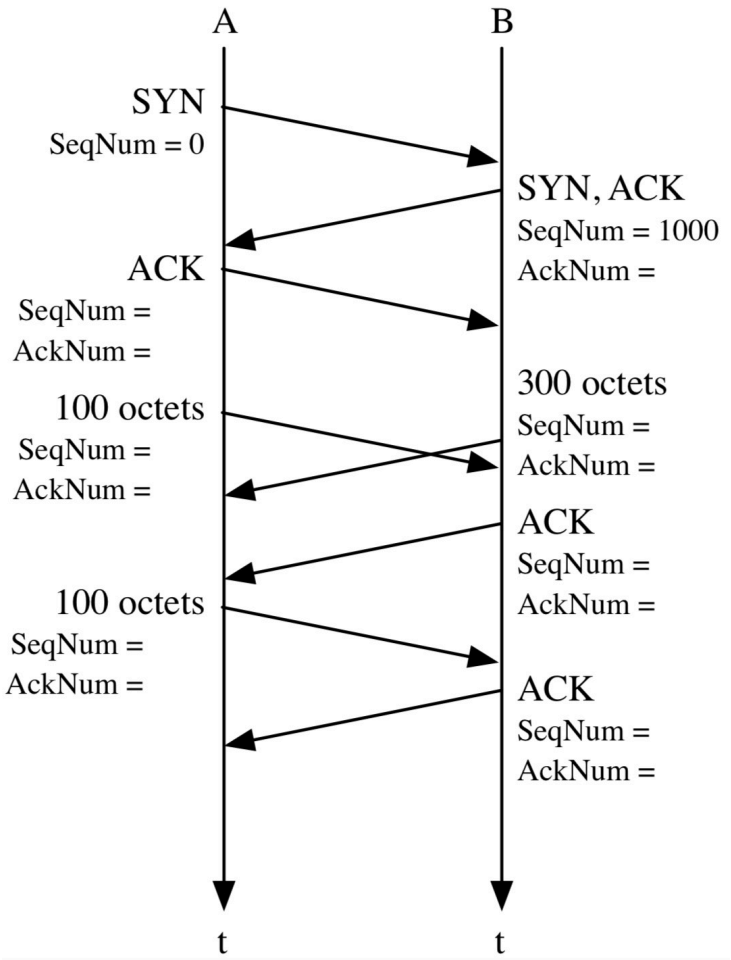
EI-2I TP1

POLYTECH SORBONNE

1. Numérotation de TCP

On considère une connexion TCP entre deux machines distantes A et B. A doit envoyer à B deux segments de 100 octets de données chacun et B doit envoyer à A un segment de 300 octets de données. A est à l'origine de l'établissement, son ISN (numéro de séquence initial) est égal à 0 et celui de B est égal à 1000.



1.1. Complétez les numéros de séquence dans le schéma.

1.2. Si le segment de 300 octets est perdu, indiquez les modifications de numéro nécessaires.

2. Analyse manuelle

L'objectif de cette section est d'étudier « manuellement » une trame observée sur un réseau local Ethernet. L'obtention d'une telle trame peut se faire à l'aide d'un analyseur de réseau multiprotocolaire (un sniffer), tel l'outil tcpdump, utilisé sur une machine connectée à ce réseau.

On considère la trame Ethernet ci-dessous, donnée sans préambule, ni CRC :

```
00 01 02 a5 fb 3a 00 01 02 a5 fc 8d 08 00 45 00
00 3c ec 26 40 00 40 06 cc cd 0a 21 b6 b6 84 e3
3c 0d 0e b5 00 50 a9 55 92 64 00 00 00 00 a0 02
3e bc a3 74 00 00 02 04 05 b4 04 02 08 0a 08 39
91 16 00 00 00 00 01 03 03 00
```

Analysez cette trame en vous aidant de l'annexe fournie à la fin du support.

2.1. Quelles informations de niveau liaison observez-vous ?

2.2. Quelles informations de niveau IP sont disponibles ?

2.3. Le paquet contient-il des options ? Justifiez.

2.4. Quelles sont la source et le destinataire du paquet ?

2.5. Quel est le protocole de transport utilisé ?

2.6. Identifiez (sans les interpréter) les différents champs contenus dans les données transportées dans le paquet IP.

3. Introduction Wireshark

L'objectif de ce TP est d'introduire un outil important dans l'etude des réseaux IP et des protocoles : l'analyseur de trames. Ce type de logiciel est capable, pour peu que l'on soit administrateur sur la machine, de capturer l'intégralité du trafic réseau émis et reçu sur une carte d'interface et de le présenter à un utilisateur pour analyse. Les logiciels habituels permettant de réaliser ce genre d'opération se nomment tcpdump ou Wireshark.
Dans ce TP, les différentes questions ne sont destinées qu'à vous familiariser avec les fonctions pratiques de ces outils. Vous êtes fortement encouragés à consulter la documentation en ligne de l'outil tout au long du TP afin d'en apprendre plus. La documentation est disponible à l'adresse : **http://www.wireshark.org/docs/**

Nous allons, lors de ce TP, étudier les différents messages des trois protocoles applicatifs vus précédemment :
HTTP, ***SMTP*** et ***FTP***. N'etant pas administrateur sur les différentes machines, vous ne pourrez pas exécuter de capture vous-même. Aussi, des fichiers pré-capturés vous sont fournis sur le dossier (https://drive.google.com/drive/folders/1Cs0f55TkBL5uUSH1F3qqXFwraX6IyDVa?usp=sharing).

Commencez donc par récupérer les quatre fichiers suivants et à les sauvegarder sur votre compte :
http.pcap
smtp-cc.pcap
smtp-bcc.pcap
ftp.pcap

4. HTTP

Lancez l'outil en tapant, dans une fenêtre de terminal ***wireshark***.
Dans le menu ***File***, utilisez la commande ***Open*** pour ouvrir le fichier **http.pcap** précédemment téléchargé.

La fenêtre principale se divise alors en trois cadres : le premier (en haut) affiche une liste de trames, le deuxième affiche les différentes parties de la trame sélectionnée (différents en-têtes et corps du message) et le dernier cadre affiche la traduction en hexadécimal et en ASCII de la trame.

Sélectionnez la première des quatre trames, puis allez dans le menu ***Analyze*** et sélectionnez la commande: ***Follow - TCP stream.***

Une nouvelle fenêtre s'affiche retraçant un échange du dialogue sans en-têtes.

4.1. A quoi cet échange correspond-t-il ?

Lors de l'examen d'un dialogue, wireshark sélectionne et n'affiche que les messages correspondant à ce dialogue. Fermez la fenêtre précédente et cliquez sur le bouton ***Clear* (X)** au dessus du premier cadre pour afficher l'intégralité des paquets. Si des trames n'ont pas été affichées dans le premier échange, elles correspondent à un second échange.

Vous pouvez examiner une partie du contenu des paquets en développant la dernière partie (hyper text transfer protocol) des trames. Pour développer cette partie, cliquez sur le triangle à gauche de la ligne visée dans le deuxième cadre.

4.2. A quoi correspond la requête envoyée lors du troisième message de l'échange ?

4.3. Allez visiter la page récupérée par la première requête HTTP et expliquez le fonctionnement du navigateur en vous basant sur l'échange, le contenu de la deuxième trame (première réponse du serveur) et sur ce que vous voyez dans votre navigateur.

Sélectionnez maintenant la commande ***Flow Graph*** du menu ***Statistics***. Dans le dialogue s'ouvrant alors, sélectionnez les options ***All packets*** ; General Flow et Standard source / destination address. Vous obtenez alors un chronogramme de la communication.

4.4. Combien de machines sont impliquées dans cette communication ? Identifiez-les et indiquez quelle machine possède quelle donnée.

4.5. Sélectionnez maintenant les options *TCP Flows*, décrivez les échanges TCP en expliquant les changement des numéro *Seq* et *Ack*.

5. SMTP

Ouvrez maintenant dans un premier temps le fichier **smtp-cc.pcap**. A l'aide des sites suivants (https://fr.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol), répondez les questions suivantes.

Sélectionnez la commande ***Flow Graph*** et examinez l'échange d'envoi d'e-mail.

5.1. Qui, du client ou du serveur, démarre le dialogue affiché ? Expliquez ce phénomène.

Sélectionnez maintenant la commande ***Follow TCP*** stream et identifiez les adresses source et destination (s) du message.

5.2. Indiquez comment est spécifié le sujet du message.

5.3. Indiquez comment un même message est envoyé à plusieurs destinataires.

Ouvrez maintenant le fichier **smtp-bcc.pcap.**

5.3. Comparez le contenu des deux échanges et identifiez la différence entre un message envoyé en copie à un second destinataire (premier exemple) et un message envoyé en copie cachée (deuxième exemple).

6. FTP

Ouvrez maintenant le fichier **ftp.pcap**. A l'aide des sites suivants (https://fr.wikipedia.org/wiki/File_Transfer_Protocol, https://fr.wikipedia.org/wiki/Liste_des_commandes_ftp et https://fr.wikipedia.org/wiki/Liste_des_codes_des_réponses_d%27un_serveur_FTP), répondez les questions suivantes.

Représentez le graphe des échanges.

6.1. Quel port est sélectionné pour le transfert effectif des données ?

6.2. L'échange vous paraît-il complet ?

6.3. Quels nom d'utilisateur et mot de passe ont été utilisés pour l'échange ?

6.4. Indiquez, à votre avis, les risques liés à l'utilisation de FTP et l'une des raisons pour lesquelles l'utilisation d'outils de capture de trame est généralement réservée aux administrateurs d'une machine.

ANNEXE

TRAME ETHERNET

8 octets	6 octets	6 octets	2 octets	46-1500 octets	4 octets
Preamble	Destination address	Source address	Type	Data	CRC

! `Preamble` est un préambule qui détermine le début d'une trame (de la forme 101010...101011) ;

! `Destination address` est l'adresse physique (MAC) du destinataire de la trame ;

! `Source address` est l'adresse physique (MAC) de l'expéditeur de la trame ;

! `Type` indique le protocole de niveau supérieur encapsulé dans le champ Data de la trame. Quelques exemples :

Type	Utilisation
0800	DoD Internet (Datagramme IP)
0805	X.25 niveau 3
0806	ARP
8035	RARP
8098	Appletalk
…	…

! `Data` contient les données brutes de la trame à passer au protocole déterminé par le champ Type ;

! `CRC` est le code détecteur d'erreur (total de contrôle) de la trame permettant d'assurer son intégrité.

DATAGRAMME IP

4 bits	4 bits	8 bits	16 bits
Version	IHL	TOS	Total length
Identification		Flags	Fragment offset
TTL	Protocol	Header checksum	
Source address			
Destination address			
Options			Padding
Data			

! `Version` indique le format de l'en-tête. Ce champ sert à l'identification de la version courante du protocole. La version décrite ici (et aujourd'hui utilisée) porte le n°4 ;

! `IHL` (*IP Header Length*) est la longueur de l'en-tête IP exprimée en mots de 32 bits ;

! `TOS` (*Type Of Service*) définit le type de service à appliquer au paquet en fonction de certains paramètres comme le délai de transit, la sécurité. Il est peu utilisé et sa valeur est généralement égale à 0 ;

! `Total Length` est la longueur totale du datagramme, exprimée en octets. En pratique, il est rare qu'un datagramme IP fasse plus de 1500 octets ;

! `Identification` sert en cas de fragmentation/réassemblage du datagramme. Ce champ permet alors à l'entité réceptrice de reconnaître les fragments issus d'un même datagramme initial et qui doivent donc faire l'objet d'un réassemblage ;

! `Flags` (3 bits) est utilisé par la fragmentation. Il est composé (de gauche à droite) :

– d'un bit réservé : mis à 0 ;

– de l'indicateur DF (*Don't Fragment*): mis à 1 par l'émetteur pour interdire la fragmentation ;

– de l'indicateur MF (*More Fragment*) : mis à 1 pour signifier que le fragment courant est suivi d'un autre fragment ;

! `Fragment offset` (13 bits) donne la position relative du fragment dans le datagramme initial, le déplacement étant donné en unités de 64 bits ;

! `TTL` (*Time To Live*) donne une indication de la limite supérieure du temps de vie d'un datagramme ;

! `Protocol` indique le protocole (de niveau supérieur) utilisé pour le champ de données du datagramme. Quelques exemples :

Code (déc)	Abréviation	Nom du protocole	Référence
1	ICMP	Internet Control Message Protocol	[RFC792]
2	IGMP	Internet Group Management Protocol	[RFC1112]
6	TCP	Transmission Control Protocol	[RFC793]
8	EGP	Exterior Gateway Protocol	[RFC888]
9	IGP	any private Interior Gateway Protocol	
17	UDP	User Datagram Protocol	[RFC768]
36	XTP	XTP	
46	RSVP	Reservation Protocol	
…	…	…	

! `Header Checksum` est une zone de contrôle d'erreur portant uniquement sur l'en-tête du datagramme ;

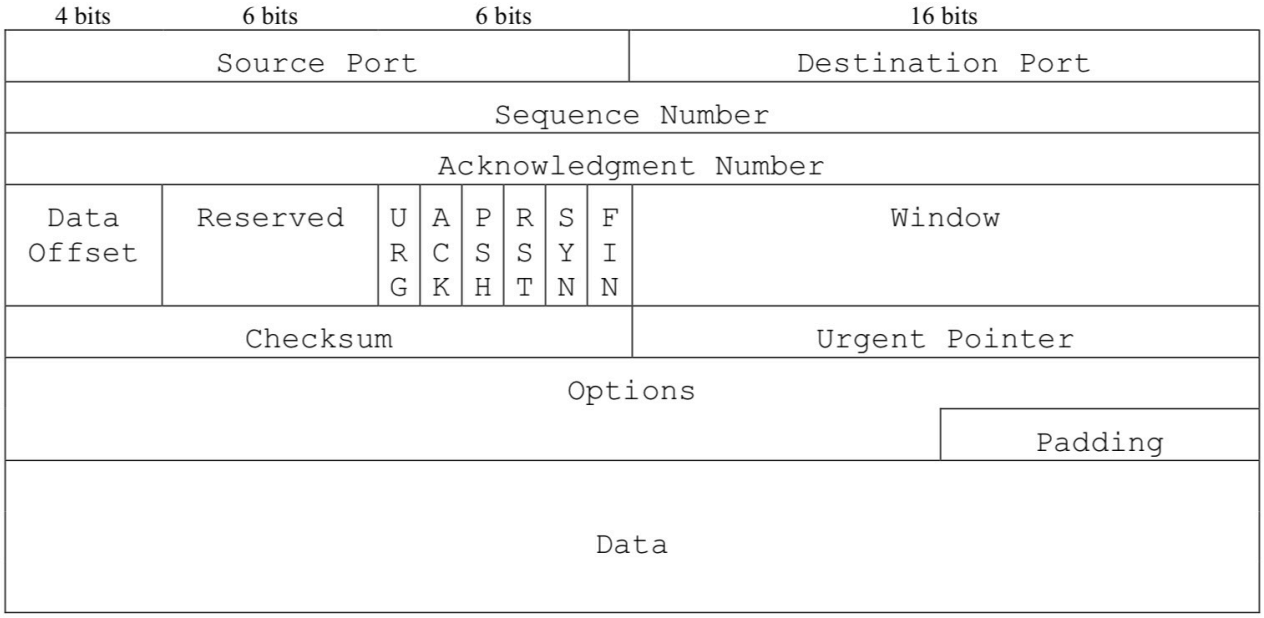
! `Source Address` est l'adresse IP de la source du datagramme ;

! `Destination Address` est l'adresse IP de destination du datagramme ;

! `Options` est de longueur variable. Il sert à des fonctions de contrôle utiles dans certaines situations. Il est constitué d'une succession d'options élémentaires, également de longueurs variables. Les options sont codées sur le principe TLV (Type, Longueur, Valeur). La longueur indique la taille complète de l'option en octets.

! `Padding` est de longueur variable : il permet d'aligner l'en-tête sur 32 bits.

SEGMENT TCP



! `Source Port` et `Destination Port` identifient les extrémités locales de la connexion. Les numéros jusqu'à 1024 correspondent à des ports réservés (Exemple : le port 21 correspond à FTP, le port 80 à HTTP) ;

! `Sequence Number` est le numéro de séquence du premier octet de données du segment TCP ; si le drapeau SYN est à 1, ce numéro est l'ISN (Initial Sequence Number) et le premier octet de données sera numéroté ISN+1 ;

! `Acknowledgment Number` est le numéro d'acquittement ; si le drapeau ACK est à 1, ce numéro contient la valeur du prochain numéro de séquence que l'émetteur est prêt à recevoir ;

! `Data Offset` est la longueur de l'en-tête TCP exprimée en mots de 32 bits ; elle indique donc où les données commencent ;

! `Reserved` n'est pas utilisé et doit être mis à zéro ;

! `URG` drapeau positionné à 1 si le pointeur d'urgence est en cours d'utilisation ;

! `ACK` drapeau positionné à 1 pour indiquer la validité du numéro d'acquittement ;

! PSH drapeau positionné à 1 pour indiquer au destinataire de remettre les données à l'application concernée dès leur arrivée ;

! RST drapeau positionné à 1 pour réinitialiser une connexion devenue incohérente, pour rejeter un segment altéré ou pour rejeter une tentative d'ouverture de connexion ;

! SYN drapeau positionné à 1 lors d'une phase d'établissement de connexion ;

! FIN drapeau positionné à 1 lors d'une phase de libération de connexion ;

! Window fenêtre d'anticipation de taille variable ; la valeur de ce champ indique au récepteur combien il peut émettre d'octets après l'octet acquitté ;

! Cheksum champs de contrôle portant sur tout le segment augmenté d'un pseudo en-tête, constitué principalement des adresses IP source et destination ;

! Urgent Pointer pointeur indiquant l'emplacement des données urgentes ; utilisé uniquement si le drapeau URG est positionné à 1 ;

! Options champs de longueur variable offrant des possibilités non offertes dans l'en-tête de base. Exemple : détermination du MSS ou Maximum Segment Size, la taille maximum d'un segment TCP (en fait la taille maximum du champs de données d'un segment TCP, car la MSS n'inclue pas l'en-tête) ;

! Padding champs permettant de cadrer l'en-tête TCP sur des mots de 32 bits.