

le cnam

Exercices dirigés

UTC505/USRS4D -Introduction- E. Gressier-Soudan

2023-2024

Ce polycopié a été élaboré par l'équipe enseignante "Réseaux et protocoles" à partir d'exercices rédigés par MM. Florin, Gressier-Soudan qu'ils en soient ici remerciés.

ED•Encapsulation et Les 7 couches de protocoles (Auto-évaluations, pour les plus déterminés, difficiles en début de cours, faisable à la fin du cours... rien n'empêche d'essayer ?)

Texte et Correction des auto-évaluations

Auto-évaluation 1, à faire seul ou à plusieurs, l'enseignant n'intervient pas... la correction est assez détaillée.

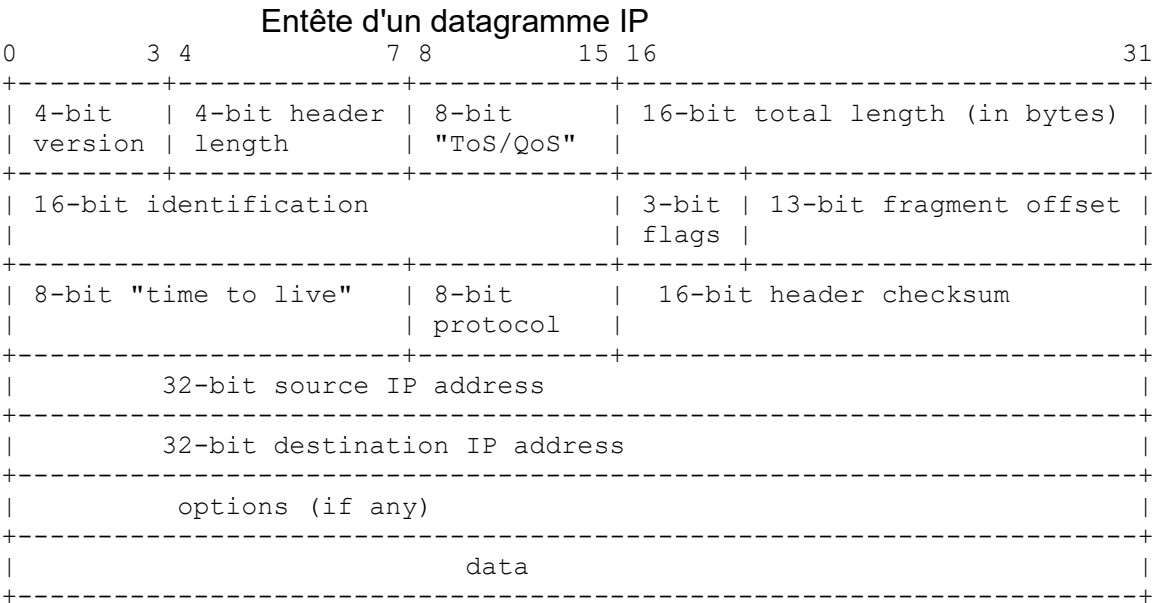
C'est un problème sur 16 points à l'origine. Une partie des questions sera plus facile après avoir fait la partie du cours sur IP et sur TCP. Mais ça reste faisable. Vous pouvez chercher les définitions qui manquent sur Internet. Le format est celui d'un examen en cours du soir.

On fournit les spécifications des différentes structures d'entête ou de message nécessaires pour analyser une trace en hexadécimal : Trame Ethernet, Entête d'un datagramme IP, entête d'un segment TCP respectivement. Ces entêtes ont déjà été données en cours et en exercices dirigés. Elles seront utiles pendant tout l'examen. Vous vous servirez de ces entêtes en fonction des questions posées.

Les spécifications des différentes structures d'entête ou de message nécessaires pour analyser une trace en hexadécimal : Trame Ethernet, Entête d'un datagramme IP, entête d'un segment TCP respectivement.

Structure d'une trame Ethernet :

Adresse MAC destination	Adresse MAC source	Type	Charge utile-Données	FCS-contrôle d'erreur
6 octets	6 octets	2 octets	46 à 1500 octets	4 octets



TCP Header																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
Offsets	Octet	0								1								2								3																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	Source port																Destination port																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
4	32	Sequence number																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
8	64	Acknowledgment number (if ACK set)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
12	96	Data offset				Reserved 0 0 0			N	C	E	U	A	P	R	S	F	Window Size																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

Question 1 : Une communication entre un programme sur un PC sous Windows, le client, et une imprimante, le serveur, a été capturée. La trame 1, telle qu'elle est capturée par l'outil Wireshark, est donnée ci-dessous au format hexadécimal.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
tcp.stream eq 1						
No.	Time	Source	Destination	Protocol	Length	Info
30	10.749644	163.173.231.161	10.173.32.97	TCP	66	50014 → 515 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
31	13.760928	163.173.231.161	10.173.32.97	TCP	66	[TCP Retransmission] 50014 → 515 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32	13.762907	10.173.32.97	163.173.231.161	TCP	66	515 → 50014 [SYN, ACK] Seq=0 Ack=1 Win=768 Len=0 MSS=1460 WS=4 SACK_PERM=1
33	13.763060	163.173.231.161	10.173.32.97	TCP	54	50014 → 515 [ACK] Seq=1 Ack=1 Win=65536 Len=0
34	13.765664	163.173.231.161	10.173.32.97	LPD	58	LPR: transfer a printer job / jobcmd: receive control file
35	13.766982	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=1 Ack=5 Win=3072 Len=0
36	13.766983	10.173.32.97	163.173.231.161	LPD	60	LPD response
37	13.767160	163.173.231.161	10.173.32.97	LPD	81	LPR: transfer a printer job / jobcmd: receive control file
38	13.778855	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=2 Ack=32 Win=3072 Len=0
39	13.778857	10.173.32.97	163.173.231.161	LPD	60	LPD response
40	13.779065	163.173.231.161	10.173.32.97	LPD	198	LPD continuation
41	13.843409	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=3 Ack=176 Win=3072 Len=0
42	13.843646	10.173.32.97	163.173.231.161	LPD	60	LPD response
43	13.843800	163.173.231.161	10.173.32.97	LPD	93	LPQ: print short form of queue status / jobcmd: receive data file
44	13.853520	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=4 Ack=215 Win=3072 Len=0
45	13.853718	10.173.32.97	163.173.231.161	LPD	60	LPD response
46	13.853976	163.173.231.161	10.173.32.97	LPD	2974	LPD continuation
47	13.883920	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=5 Ack=3135 Win=61592 Len=0
48	13.883981	163.173.231.161	10.173.32.97	LPD	8814	LPD continuation
49	13.895362	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=5 Ack=11895 Win=64512 Len=0
50	13.895423	163.173.231.161	10.173.32.97	LPD	5841	LPD continuation
51	13.897961	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=5 Ack=17682 Win=64512 Len=0
52	13.898219	163.173.231.161	10.173.32.97	TCP	54	50014 → 515 [FIN, ACK] Seq=17682 Ack=5 Win=65536 Len=0
53	13.899679	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [ACK] Seq=5 Ack=17683 Win=64512 Len=0
54	13.915481	10.173.32.97	163.173.231.161	TCP	60	515 → 50014 [FIN, ACK] Seq=5 Ack=17683 Win=64512 Len=0
55	13.915554	163.173.231.161	10.173.32.97	TCP	54	50014 → 515 [ACK] Seq=17683 Ack=6 Win=65536 Len=0

```

0000  00 08 e3 ff fc 28 30 e1 71 82 10 c5 08 00 45 00
0010  00 34 18 cf 40 00 80 06 00 00 a3 ad e7 a1 0a ad
0020  20 61 c3 5e 02 03 e8 44 72 ed 00 00 00 00 80 02
0030  fa f0 b6 83 00 00 02 04 05 b4 01 03 03 08 01 01
0040  04 02

```

Pour toutes les questions qui suivent, vous pourrez surligner/encadrer les valeurs correspondantes dans la trace en hexadécimal pourvu que ça soit facilement compréhensible pour le correcteur. N'hésitez pas à utiliser de la couleur.

- Quelle est l'adresse MAC source de la trame en hexadécimal ?
- Quelle est l'adresse MAC destination de la trame en hexadécimal ?
- Quelle est la valeur du champ type de la trame, qu'en déduisez-vous sur le type de message contenu dans la partie données de cette trame ? Si c'est un datagramme IP, est-ce de l'IP v4 ou v6 ? Justifiez votre réponse.
- Quelle est la longueur de l'adresse source de niveau 3, quelle est sa valeur en hexadécimal et en décimal ?
- Quelle est la longueur de l'adresse destination de niveau 3, quelle est sa valeur en hexadécimal et en décimal ?
- Quel est le protocole de niveau 4 transporté par le protocole de niveau 3 (IP) ?

Correction :

```

0000  00 08 e3 ff fc 28 30 e1 71 82 10 c5 08 00 45 00
0010  00 34 18 cf 40 00 80 06 00 00 a3 ad e7 a1 0a ad
0020  20 61 c3 5e 02 03 e8 44 72 ed 00 00 00 00 80 02
0030  fa f0 b6 83 00 00 02 04 05 b4 01 03 03 08 01 01
0040  04 02
  
```

- Quelle est l'adresse MAC source de la trame en hexadécimal ? 30 e1 71 82 10 c5
- Quelle est l'adresse MAC destination de la trame en hexadécimal ? 00 08 e3 ff fc 28
- Quelle est la valeur du champ type de la trame, 08 00 qu'en déduisez-vous sur le type de message contenu dans la partie données de cette trame ? Si c'est un datagramme IP, est-ce de l'IP v4 ou v6 ? C'est un datagramme IPv4. 0x0800 correspond au type IPV4, qui est confirmé dans le premier demi-octet suivant, dans l'entête IPv4 encapsulé dans la trame, où on trouve "4".
- Quelle est la longueur de l'adresse source de niveau 3, 32 bits ou 4 octets. Quelle est sa valeur en hexadécimal et en décimal ? a3 ad e7 a1. Soit 163.173.231.161 en décimal.
- Quelle est la longueur de l'adresse destination de niveau 3, même longueur que l'adresse IP source. Quelle est sa valeur en hexadécimal et en décimal ? 0a ad 20 61. Soit 10.173.32.97 en décimal.

- Quel est le protocole de niveau 4 transporté par le protocole de niveau 3 ? **06** c'est la valeur qui correspond à TCP.

Question 2 : Le système d'exploitation de la source est de type Windows. On exécute la commande `ipconfig/all`, et on obtient en partie l'affichage suivant :

Carte Ethernet eth0 :

```
Suffixe DNS propre à la connexion. . . : cnam.fr
Description. . . . . : Intel(R) Ethernet Connection I219-V
Adresse physique . . . . . : 30-E1-71-82-10-C5
Adresse IPv4. . . . . : 163.173.231.161 (préférée)
Masque de sous-réseau. . . . . : 255.255.252.0
Passerelle par défaut. . . . . : 163.173.128.2
Serveurs DNS. . . . . : 163.173.128.6
                        163.173.128.60
```

- Donner le masque du sous-réseau IP auquel appartient la source en notation compacte (/n) à partir des résultats de la commande `ipconfig` ci-dessus.
- Quelle est l'adresse de broadcast IPv4 associée à ce réseau IP ? Expliquez très brièvement comment vous la trouvez.
- Complétez les cases vides de la table de routage de la machine source ci-après en vous aidant des informations données dans la commande `ipconfig`.

Réseau IP/mask	Next-Hop	Commentaire	Inter-face	Accessi-bilité
0.0.0.0/0		Route par défaut pour	eth0	distant
127.0.0.0/8	0.0.0.0	Loopback, on ne passe par la carte NIC	lo0	direct
	0.0.0.0	Le réseau IP où je suis directement connecté	eth0	direct

Correction :

- Donner le masque du sous-réseau IP auquel appartient la source en notation compacte (/n) à partir des résultats de la commande `ipconfig` ci-dessus : /22 car le masque, c'est 22 bits à 1 en partant de la gauche.
- Quelle est l'adresse de broadcast IPv4 associée à ce réseau IP ? Pour avoir le broadcast IP, on complète la partie droite de l'adresse de réseau non couverte par les 1 du masque, par des 1, on obtient 163.173.131.255
- Complétez les cases vides de la table de routage de la machine source ci-après en vous aidant des informations données dans la commande `ipconfig`.



Réseau IP/mask	Next-Hop	Commentaire	Interface	Accessibilité
0.0.0.0/0	163.173.228.2	Route par défaut pour	eth0	distant
127.0.0.0/8	0.0.0.0	Loopback, on ne passe par la carte NIC	lo0	direct
163.173.128.0/22	0.0.0.0	Le réseau IP où je suis directement connecté	eth0	direct

Question 3 :

- Pourquoi les trames portant les numéros 30 à 33 dans la capture correspondent à une ouverture de connexion ?
- Donner l'adresse IP en décimal de l'interface qui exécute l'ouverture de la connexion et donc joue un rôle actif. Pourquoi c'est cette adresse qui correspond à un rôle actif ?
- Donner l'adresse IP de l'interface en décimal qui exécute l'acceptation de l'ouverture de connexion et donc joue un rôle passif. Pourquoi c'est cette adresse qui correspond à un rôle passif ?
- Ligne 31 on observe une retransmission pendant l'ouverture de connexion. Proposer une valeur pour le temps d'attente avant de refaire une nouvelle tentative d'ouverture de connexion. Expliquer brièvement votre raisonnement. **L'échelle de temps spécifiée dans la trace correspond à des secondes.**

Correction :

- Pourquoi les trames portant les numéros 30 à 33 dans la capture correspondent à une ouverture de connexion ?

Trame 30 : du client vers serveur : SYN mais n'est pas acquité

Trame 31 : du client vers serveur : retransmission du SYN

Trame 32 : du serveur vers client : SYN, ACK

Trame 33 : du client vers serveur ACK

On observe l'exécution du Three-way handshake caractéristique de l'ouverture de cx TCP entre un client et un serveur.

- Donner l'adresse IP en décimal de l'interface qui exécute l'ouverture de la connexion et donc joue un rôle actif. C'est 163.173.231.161 qui ouvre la connexion par l'envoi d'un SYN, donc c'est lui qui joue un rôle actif.
- Donner l'adresse IP de l'interface en décimal qui accepte de l'ouverture de connexion et joue un rôle passif. C'est 163.173.231.161 qui accepte la connexion en répondant SYN,ACK. Il est en ouverture de cx passive. C'est donc le passif.
- Ligne 31 on observe une retransmission pendant l'ouverture de connexion. Proposer une valeur pour le temps d'attente avant de refaire une nouvelle tentative d'ouverture de connexion. La trame 30 est capturée par Wireshark à 10.7496, et la trame 31, la retransmission du SYN, est capturée à 13.7609. La différence est d'environ 3. Sachant que le temps spécifié dans la trace correspond à des secondes, 3 secondes séparent la demande d'ouverture de connexion SYN de sa retransmission. Le temps d'attente ou encore le délai garde pour redemander une ouverture de connexion est de 3s. C'est semble-t-il la valeur courante dans les implantations de TCP.

Question 4 : L'échange des données entre les deux applications s'effectue à travers les ports : 515 pour le serveur, et, 50014 pour le client. La capture du flot TCP est la suivante :

Time	163.173.231.161	10.173.32.97	Comment
10.749644	50014	515	Seq = 0
13.760928	50014	515	Seq = 0
13.762907	50014	515	Seq = 0 Ack = 1
13.763060	50014	515	Seq = 1 Ack = 1
13.765664	50014	515	Seq = 1 Ack = 1
13.766982	50014	515	Seq = 1 Ack = 5
13.766983	50014	515	Seq = 1 Ack = 5
13.767160	50014	515	Seq = 5 Ack = 2
13.778855	50014	515	Seq = 2 Ack = 32
13.778857	50014	515	Seq = 2 Ack = 32
13.779065	50014	515	Seq = 32 Ack = 3
13.843409	50014	515	Seq = 3 Ack = 176
13.843646	50014	515	Seq = 3 Ack = 176
13.843800	50014	515	Seq = 176 Ack = 4
13.853520	50014	515	Seq = 4 Ack = 215
13.853718	50014	515	Seq = 4 Ack = 215
13.853976	50014	515	Seq = 215 Ack = 5
13.883920	50014	515	Seq = 5 Ack = 3135
13.883981	50014	515	Seq = 3135 Ack = 5
13.895362	50014	515	Seq = 5 Ack = 11895
13.895423	50014	515	Seq = 11895 Ack = 5
13.897961	50014	515	Seq = 5 Ack = 17682
13.898219	50014	515	Seq = 17682 Ack = 5
13.899679	50014	515	Seq = 5 Ack = 17683
13.915481	50014	515	Seq = 5 Ack = 17683
13.915554	50014	515	Seq = 17683 Ack = 6

- Est-ce que la trace contient toutes les étapes d'une connexion (ouverture, transferts, fermeture) ? Pourquoi ?
- A quoi peut servir l'indicateur PSH dans les échanges entre une application et une imprimante ?
- Combien d'octets de données ont été envoyés depuis le port 50014 vers le port 515 pendant toute la connexion ? Expliquez brièvement votre réponse.
- Combien d'octets de données ont été envoyés depuis le port 515 vers le port 50014 pendant toute la connexion ? Expliquez brièvement votre réponse.

Correction :

- Est-ce que la trace contient toutes les étapes d'une connexion (ouverture, transferts, fermeture) ? Pourquoi ? Il semble bien que toutes les phases de la connexion soient présentes dans la figure ci-dessus cf encadrements ajoutés en violet.
- A quoi peut servir l'indicateur PSH dans les échanges entre une application et une imprimante ? Le PSH demande à l'automate TCP récepteur de délivrer le contenu du segment, ses données, au plus vite sans bufferiser.
- Combien d'octets de données ont été envoyés depuis le port 50014 vers le port 515 pendant toute la connexion ? Ce qui nous intéresse c'est l'envoi des données vers le serveur LPD (pilote d'impression), en gros quelle est la taille des commandes d'impression et du fichier à imprimer.
Le dernier segment d'acquiescement du serveur dans le FIN,ACK trame 54 dans la trace Wireshark, porte le numéro 17683. C'est donc le numéro du prochain octet attendu venant du client par le serveur. Le serveur acquitte donc le 17682^{ème} octet et tous ceux qui précèdent. Il a donc reçu de l'octet 0 à l'octet 17682. Toutefois, le SYN et le ACK sont des octets fantômes comptés dans cet ensemble. Le serveur a donc $17683^1 - 2$ octets de données, soit 17681 octets.
- Combien d'octets de données ont été envoyés depuis le port 515 vers le port 50014 pendant toute la connexion ? On raisonne de la même façon. Le dernier segment FIN,ACK pour le client venant du serveur porte le numéro 6. Le serveur a donc envoyé les octets 0, 1, 2, 3, 4, 5 mais 0 et 5 sont des octets fantômes. Donc LPD a envoyé 4 octets, les segments correspondants sont marqués d'une étoile dans le schéma ci-dessus.

Le découpage en phase de l'échange est indiqué dans la figure qui suit.

¹ De 0 à 17682, cela fait 17683 octets. Penser [0 ; 17682]



Time	163.173.231.161	10.173.32.97	Comment
10.749644	30 50014	→ 515	Seq = 0
13.760928	50014	→ 515	Seq = 0
13.762907	50014	← 515	Seq = 0 Ack = 1
13.763060	50014	→ 515	Seq = 1 Ack = 1
13.765664	50014	→ 515	Seq = 1 Ack = 1
13.766982	50014	← 515	Seq = 1 Ack = 5
13.766983	50014	← 515 *	Seq = 1 Ack = 5
13.767160	50014	→ 515	Seq = 5 Ack = 2
13.778855	50014	← 515	Seq = 2 Ack = 32
13.778857	50014	← 515 *	Seq = 2 Ack = 32
13.779065	50014	→ 515	Seq = 32 Ack = 3
13.843409	50014	← 515	Seq = 3 Ack = 176
13.843646	45 : ok LPD 50014	← 515 *	Seq = 3 Ack = 176
13.843800	50014	→ 515	Seq = 176 Ack = 4
13.853520	50014	← 515	Seq = 4 Ack = 215
13.853718	50014	← 515 *	Seq = 4 Ack = 215
13.853976	48 50014	→ 515	Seq = 215 Ack = 5
13.883920	50014	← 515	Seq = 5 Ack = 3135
13.883981	46 50014	→ 515	Seq = 3135 Ack = 5
13.895362	50014	← 515	Seq = 5 Ack = 11895
13.895423	50 50014	→ 515	Seq = 11895 Ack = 5
13.897961	50014	← 515	Seq = 5 Ack = 17682
13.898219	50014	→ 515	Seq = 17682 Ack = 5
13.899679	50014	← 515	Seq = 5 Ack = 17683
13.915481	50014	← 515	Seq = 5 Ack = 17683
13.915554	50014	→ 515	Seq = 17683 Ack = 6

Ouverture de connexion

45 : ok LPD

48

46

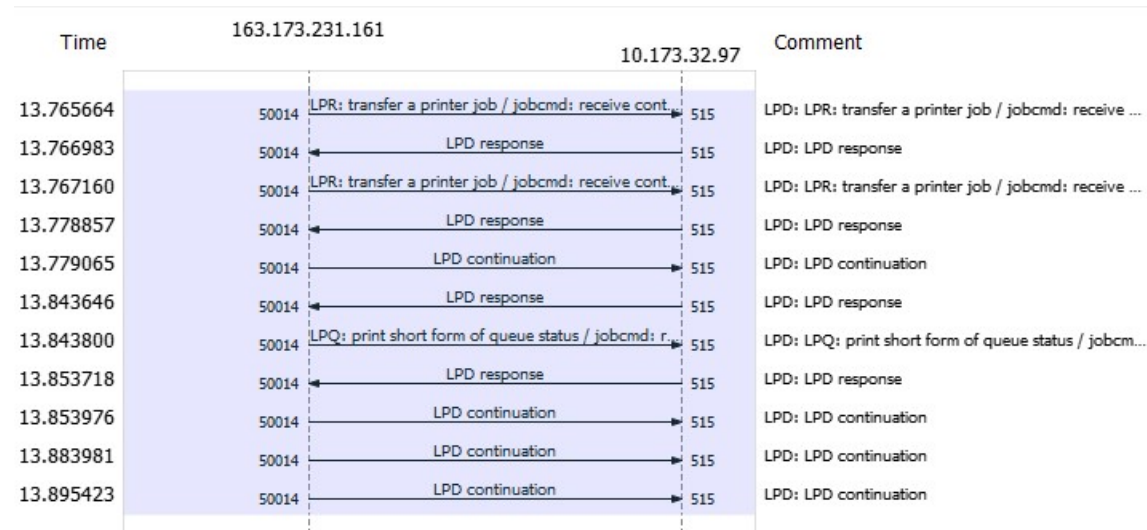
50

Fermeture de connexion

Transfert

Question 5 : Le port 515 est un port inférieur à 1023, c'est donc un port réservé. La capture Wireshark marque certaines trames avec "LPD", Line Printer Daemon, dans la colonne "protocol". LPD est un protocole applicatif associé au port 515 capturable par Wireshark.

34	13.765664	163.173.231.161	10.173.32.97	LPD	58 LPR: transfer a printer job / jobcmd: receive control file
36	13.766983	10.173.32.97	163.173.231.161	LPD	60 LPD response
37	13.767160	163.173.231.161	10.173.32.97	LPD	81 LPR: transfer a printer job / jobcmd: receive control file
39	13.778857	10.173.32.97	163.173.231.161	LPD	60 LPD response
40	13.779065	163.173.231.161	10.173.32.97	LPD	198 LPD continuation
42	13.843646	10.173.32.97	163.173.231.161	LPD	60 LPD response
43	13.843800	163.173.231.161	10.173.32.97	LPD	93 LPQ: print short form of queue status / jobcmd: receive data file
45	13.853718	10.173.32.97	163.173.231.161	LPD	60 LPD response
46	13.853976	163.173.231.161	10.173.32.97	LPD	2974 LPD continuation
48	13.883981	163.173.231.161	10.173.32.97	LPD	8814 LPD continuation
50	13.895423	163.173.231.161	10.173.32.97	LPD	5841 LPD continuation



La trace des échanges applicatifs s'obtient via la commande "follow tcp Stream" dont un sous-ensemble est extrait ci-dessous :

```
.lp
..143 cfa007DESKTOP-76E2255
.HDESKTOP-76E2255
PERIC
Jtextessaiwireshark.txt.- Bloc-notes
```



```
ldfA007DESKTOP-76E2255
Udfa007DESKTOP-76E2255
Ntextessaiwireshark.txt.- Bloc-notes
...125899906843000 dfa007DESKTOP-76E2255
..%-12345X@PJL JOB NAME="textessaiwireshark.txt.- Bloc-notes"
@PJL SET PCNAME="DESKTOP-76E2255"
@PJL SET PCNAMEW="DESKTOP-76E2255"
@PJL SET DRIVERNAM="SHARP MX-M365N PCL6\3e34"
@PJL SET JOBNAME="textessaiwireshark.txt.- Bloc-notes"
@PJL SET JOBNAMEW="textessaiwireshark.txt.- Bloc-notes"
@PJL SET SPOOLTIME="Wed, 19 Dec 2018 19:43:01 +0100"
@PJL SET FILING=OFF
@PJL SET USERNAME="ERIC"
@PJL SET USERNAMEW="ERIC"
@PJL SET NOTIFYJOBEND=OFF
@PJL SET HOLD=OFF
@PJL SET QTY=1
```

Quelques trames et leur contenu applicatif :

Trame 34 :

```
> Frame 34: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
> Ethernet II, Src: HewlettP_82:10:c5 (30:e1:71:82:10:c5), Dst: Cisco_ff:fc:28 (00:08:e3:ff:fc:28)
> Internet Protocol Version 4, Src: 163.173.231.161, Dst: 10.173.32.97
> Transmission Control Protocol, Src Port: 50014, Dst Port: 515, Seq: 1, Ack: 1, Len: 4
▼ Line Printer Daemon Protocol
  LPR: transfer a printer job / jobcmd: receive control file
  Printer/options: lp
```

0000	00 08 e3 ff fc 28 30 e1 71 82 10 c5 08 00 45 00(0- q.....E-
0010	00 2c 18 d2 40 00 80 06 00 00 a3 ad e7 a1 0a ad	.,..@...
0020	20 61 c3 5e 02 03 e8 44 72 ee 00 33 ef 31 50 18	a-^...D r--3-1P-
0030	01 00 b6 7b 00 00 02 6c 70 0a	...{...l p.

Trame 36

```
> Frame 36: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Cisco_ff:fc:28 (00:08:e3:ff:fc:28), Dst: HewlettP_82:10:c5 (30:e1:71:82:10:c5)
> Internet Protocol Version 4, Src: 10.173.32.97, Dst: 163.173.231.161
> Transmission Control Protocol, Src Port: 515, Dst Port: 50014, Seq: 1, Ack: 5, Len: 1
✓ Line Printer Daemon Protocol
  Response: Success: accepted, proceed (0)
```

```
0000 30 e1 71 82 10 c5 00 08 e3 ff fc 28 08 00 45 00 0-q.....-E-
0010 00 29 21 12 00 00 fe 06 e5 5f 0a ad 20 61 a3 ad -)!....._..a-
0020 e7 a1 02 03 c3 5e 00 33 ef 31 e8 44 72 f2 50 18 .....^3-1.Dr-P-
0030 03 00 e6 70 00 00 04 05 b4 01 01 ...p... ..
```

Trame 37

```
> Frame 37: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
> Ethernet II, Src: HewlettP_82:10:c5 (30:e1:71:82:10:c5), Dst: Cisco_ff:fc:28 (00:08:e3:ff:fc:28)
> Internet Protocol Version 4, Src: 163.173.231.161, Dst: 10.173.32.97
> Transmission Control Protocol, Src Port: 50014, Dst Port: 515, Seq: 5, Ack: 2, Len: 27
✓ Line Printer Daemon Protocol
  LPR: transfer a printer job / jobcmd: receive control file
  Printer/options: 143 cFA007DESKTOP-76E2255
```

```
0000 00 08 e3 ff fc 28 30 e1 71 82 10 c5 08 00 45 00 .....(0-q.....E-
0010 00 43 18 d3 40 00 80 06 00 00 a3 ad e7 a1 0a ad -C-@... ..
0020 20 61 c3 5e 02 03 e8 44 72 f2 00 33 ef 32 50 18 a-^...D r-3-2P-
0030 01 00 b6 92 00 00 02 31 34 33 20 63 66 41 30 30 .....1 43 cFA00
0040 37 44 45 53 4b 54 4f 50 2d 37 36 45 32 32 35 35 7DESKTOP -76E2255
0050 0a .
```

Trame 40

```
> Frame 40: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits) on interface 0
> Ethernet II, Src: HewlettP_82:10:c5 (30:e1:71:82:10:c5), Dst: Cisco_ff:fc:28 (00:08:e3:ff:fc:28)
> Internet Protocol Version 4, Src: 163.173.231.161, Dst: 10.173.32.97
> Transmission Control Protocol, Src Port: 50014, Dst Port: 515, Seq: 32, Ack: 3, Len: 144
> Line Printer Daemon Protocol
```

0010	00 b8 18 d4 40 00 80 06 00 00 a3 ad e7 a1 0a ad@....
0020	20 61 c3 5e 02 03 e8 44 73 0d 00 33 ef 33 50 18	a.^...D s..3.3P.
0030	01 00 b7 07 00 00 48 44 45 53 4b 54 4f 50 2d 37HD ESKTOP-7
0040	36 45 32 32 35 35 0a 50 45 52 49 43 0a 4a 74 65	6E2255.P ERIC.Jte
0050	78 74 65 73 73 61 69 77 69 72 65 73 68 61 72 6b	xtessaiw ireshark
0060	2e 74 78 74 a0 2d 20 42 6c 6f 63 2d 6e 6f 74 65	.txt-- B loc-note
0070	73 0a 6c 64 66 41 30 30 37 44 45 53 4b 54 4f 50	s.ldfA00 7DESKTOP
0080	2d 37 36 45 32 32 35 35 0a 55 64 66 41 30 30 37	-76E2255 .UdfA007
0090	44 45 53 4b 54 4f 50 2d 37 36 45 32 32 35 35 0a	DESKTOP- 76E2255.
00a0	4e 74 65 78 74 65 73 73 61 69 77 69 72 65 73 68	Ntextess aiwiresh
00b0	61 72 6b 2e 74 78 74 a0 2d 20 42 6c 6f 63 2d 6e	ark.txt. - Bloc-n
00c0	6f 74 65 73 0a 00	otes--

Trame 43

- > Frame 43: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface 0
- > Ethernet II, Src: HewlettP_82:10:c5 (30:e1:71:82:10:c5), Dst: Cisco_ff:fc:28 (00:08:e3:ff:fc:28)
- > Internet Protocol Version 4, Src: 163.173.231.161, Dst: 10.173.32.97
- > Transmission Control Protocol, Src Port: 50014, Dst Port: 515, Seq: 176, Ack: 4, Len: 39
- ✓ Line Printer Daemon Protocol

LPQ: print short form of queue status / jobcmd: receive data file

Printer/options: 125899906843000 dfa007DESKTOP-76E2255

0000	00 08 e3 ff fc 28 30 e1 71 82 10 c5 08 00 45 00(0- q.....E-
0010	00 4f 18 d5 40 00 80 06 00 00 a3 ad e7 a1 0a ad	-0-..@...
0020	20 61 c3 5e 02 03 e8 44 73 9d 00 33 ef 34 50 18	a-^...D s..3-4P-
0030	01 00 b6 9e 00 00 03 31 32 35 38 39 39 39 30 361 25899906
0040	38 34 33 30 30 20 64 66 41 30 30 37 44 45 53	843000 d fA007DES
0050	4b 54 4f 50 2d 37 36 45 32 32 35 35 0a	KTOP-76E 2255-

Trame 46

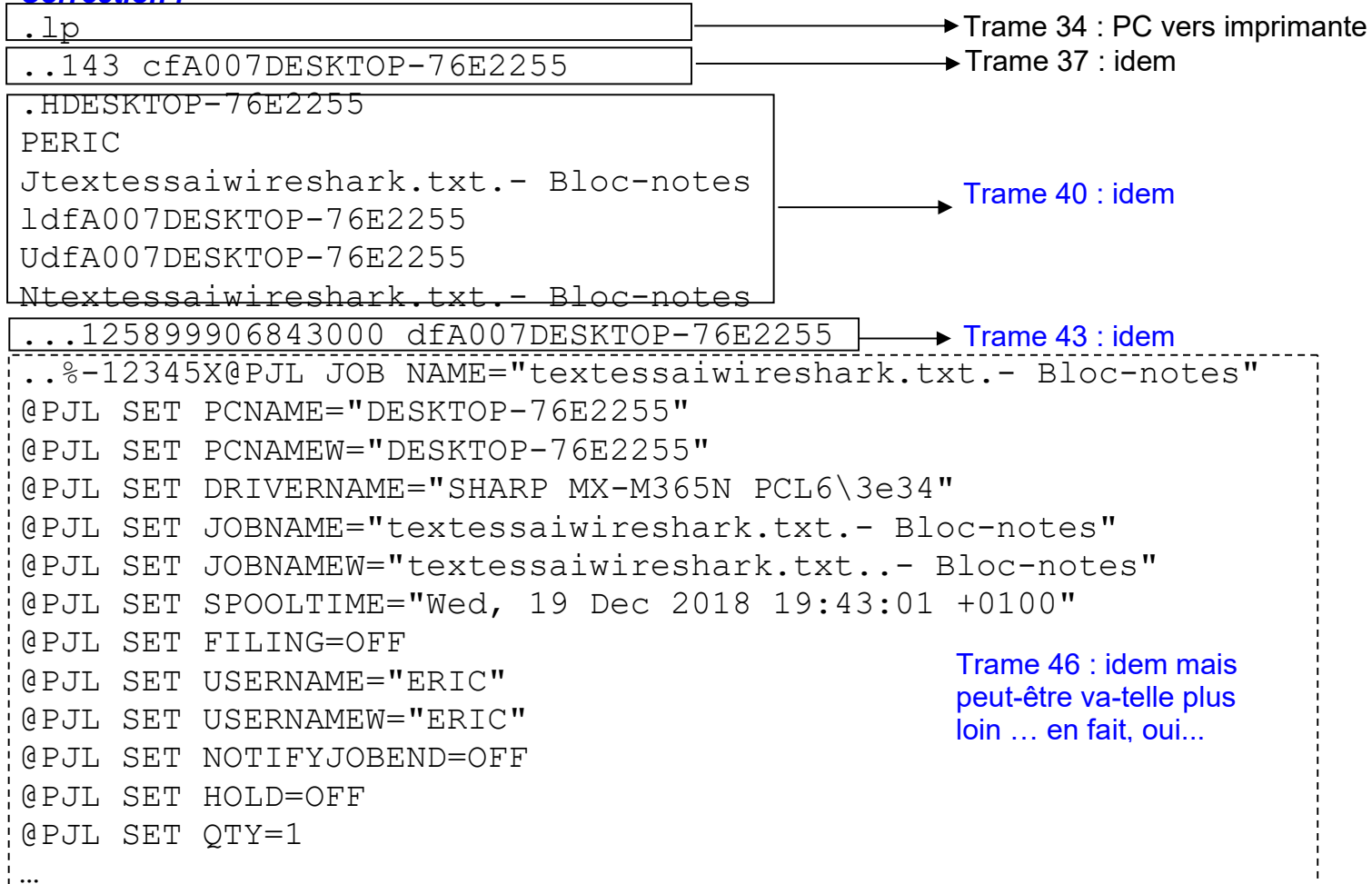
- > Frame 46: 2974 bytes on wire (23792 bits), 2974 bytes captured (23792 bits) on interface 0
- > Ethernet II, Src: HewlettP_82:10:c5 (30:e1:71:82:10:c5), Dst: Cisco_ff:fc:28 (00:08:e3:ff:fc:28)
- > Internet Protocol Version 4, Src: 163.173.231.161, Dst: 10.173.32.97
- > Transmission Control Protocol, Src Port: 50014, Dst Port: 515, Seq: 215, Ack: 5, Len: 2920
- ✓ Line Printer Daemon Protocol
 - > Data (2920 bytes)

0030	01 00 b6 63 00 00 1b 25 2d 31 32 33 34 35 58 40	...c-..-% -12345X@
0040	50 4a 4c 20 4a 4f 42 20 4e 41 4d 45 3d 22 74 65	PJL JOB NAME="te
0050	78 74 65 73 73 61 69 77 69 72 65 73 68 61 72 6b	xtessaiw ireshark
0060	2e 74 78 74 a0 2d 20 42 6c 6f 63 2d 6e 6f 74 65	.txt-- B loc-note
0070	73 22 0d 0a 40 50 4a 4c 20 53 45 54 20 50 43 4e	s"..@PJL SET PCN
0080	41 4d 45 3d 22 44 45 53 4b 54 4f 50 2d 37 36 45	AME="DES KTOP-76E
0090	32 32 35 35 22 0d 0a 40 50 4a 4c 20 53 45 54 20	2255"..@ PJL SET
00a0	50 43 4e 41 4d 45 57 3d 22 44 45 53 4b 54 4f 50	PCNAMEW= "DESKTOP
00b0	2d 37 36 45 32 32 35 35 22 0d 0a 40 50 4a 4c 20	-76E2255 "..@PJL
00c0	53 45 54 20 44 52 49 56 45 52 4e 41 4d 45 3d 22	SET DRIV ERNAME="
00d0	53 48 41 52 50 20 4d 58 2d 4d 33 36 35 4e 20 50	SHARP MX -M365N P

Utiliser les trames ci-dessus pour continuer à délimiter les échanges dans le schéma ci-après :

.lp	→ Trame 34 : PC vers imprimante
..143 cfa007DESKTOP-76E2255	→ Trame 37 : idem

.HDESKTOP-76E2255
PERIC
Jtextessaiwireshark.txt.- Bloc-notes
ldfA007DESKTOP-76E2255
UdfA007DESKTOP-76E2255
Ntextessaiwireshark.txt.- Bloc-notes
...125899906843000 dfa007DESKTOP-76E2255
..%-12345X@PJL JOB NAME="textessaiwireshark.txt.- Bloc-notes"
@PJL SET PCNAME="DESKTOP-76E2255"
@PJL SET PCNAMEW="DESKTOP-76E2255"
@PJL SET DRIVERNAME="SHARP MX-M365N PCL6\3e34"
@PJL SET JOBNAME="textessaiwireshark.txt.- Bloc-notes"
@PJL SET JOBNAMEW="textessaiwireshark.txt.- Bloc-notes"
@PJL SET SPOOLTIME="Wed, 19 Dec 2018 19:43:01 +0100"
@PJL SET FILING=OFF
@PJL SET USERNAME="ERIC"
@PJL SET USERNAMEW="ERIC"
@PJL SET NOTIFYJOBEND=OFF
@PJL SET HOLD=OFF
@PJL SET QTY=1
...

Correction :

En observant la trace Wireshark complète et de façon précise sur les échanges du protocole LPD, on s'aperçoit qu'après les commandes de configuration, un premier morceau du contenu du fichier à imprimer est présent. C'est à la fois normal, le segment fait 2920 octets de données. Le suivant, le 48, fait 8760 octets de données, et le 50 fait 5787 de données et vont toujours du client vers le serveur d'impression ce qui est dans la logique applicative.

Question 6 : A travers la trace, on a pu observer que 10.173.32.97 communiquait avec 163.173.231.161. Le cours apprend que la première adresse est une adresse privée non routable hors de son domaine de routage, et, que la seconde est une adresse publique. 2 hypothèses à étudier pour tenter d'expliquer cette observation.

- **Hypothèse 1 :** *Un routeur spécifique dédié au réseau d'imprimantes et qui exécute le protocole NAT, Network Address Translation, permet de faire communiquer l'imprimante avec le client et réciproquement.*
- **Hypothèse 2 :** *Les réseaux 10/8 et 163.173.228.0/22 cohabitent sur le même réseau physique. Un routeur du réseau Cnam route le réseau 10/8 comme tout autre sous-réseau du Cnam. Le DNS local connaît les 2 réseaux et c'est comme ça que l'imprimante peut être trouvée par le client dont l'utilisateur ne connaît que le nom de serveur.*

Laquelle des deux hypothèses vous semble la plus réaliste, expliquer rapidement pourquoi.

Correction :

- **Hypothèse 1 :** *Un routeur spécifique dédié au réseau d'imprimantes et qui exécute le protocole NAT, Network Address Translation, permet de faire communiquer l'imprimante avec le client et réciproquement.*

Cette hypothèse est crédible. On pourrait imaginer les imprimantes sur un LAN virtuel spécifique, et le routeur qui route entre les différents VLAN² du réseau du Cnam et fait du NAT (protocole Network Address Translation). Cela permet d'avoir le même réseau physique mais son partitionnement isole le réseau des imprimantes des autres réseaux. Mais qui dit NAT dit à l'interface extérieure une adresse publique. Le client utiliserait cette adresse publique pour imprimer. Ça ne colle pas avec la trace Wireshark qui est fait sur le client. Ce n'est pas dit dans l'énoncé mais on s'en doute, il est plus compliqué d'aller sniffer du trafic près d'une imprimante qui est dans un local technique géré spécifiquement.

Clairement, le client connaît l'adresse IP du serveur d'impression explicitement, même si c'est une adresse privée, et lui envoie directement ses datagrammes... Il faut trouver une autre explication.

- **Hypothèse 2 :** *Les réseaux 10/8 et 163.173.228.0/22 cohabitent sur le même réseau physique. Un routeur du réseau Cnam route le réseau 10/8 comme tout autre sous-réseau du Cnam. Le DNS local connaît les 2 réseaux et c'est comme ça que l'imprimante peut être trouvée par le client dont l'utilisateur ne connaît que le nom de serveur.*

L'hypothèse du client sur un VLAN et du serveur sur un VLAN d'impression est toujours possible dans cette hypothèse puisqu'on fait l'hypothèse d'une cohabitation. Le DNS local peut très bien connaître l'adresse IP privée du serveur d'impression ciblé par le client et résoudre la correspondance quand le client veut lancer une impression. Le routage fait le reste pour acheminer le fichier à imprimer vers l'imprimante visée.

Dans ce cas, la capture Wireshark sur le client, peut tout à fait voir les adresses 10/8.

L'hypothèse 2 semble la plus probable...

Auto-évaluation 2, à faire seul, l'enseignant n'intervient pas... Ce sujet n'est pas facile.

L'objet de cet exercice est d'étudier la différence entre les entêtes IPv4 et IPv6, et l'importance de chaque champ dans une entête (Header en anglais), en particulier ici, le premier demi-octet qui indique la version d'IP qui correspond au datagramme analysé, et qui détermine la suite de l'analyse de l'entête.

No.	Time	Source	Destination	Protocol	Length
1	0.000000	2.1.1.2	2.1.1.1	IPv4	
<					
> Frame 1: 1010 bytes on wire (8080 bits), 1010 bytes captured (8080 bits)					
> Ethernet II, Src: PcsCompu_fc:6a:c9 (08:00:27:fc:6a:c9), Dst: PcsCompu_e2					
✓ Internet Protocol Version 4, Src: 2.1.1.2, Dst: 2.1.1.1					
0100 = Version: 4					
.... 0101 = Header Length: 20 bytes (5)					
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)					
Total Length: 996					
Identification: 0xb5d0 (46544)					
> Flags: 0x2000, More fragments					
Fragment offset: 0					
Time to live: 64					
Protocol: ICMP (1)					
Header checksum: 0x9b44 [validation disabled]					
[Header checksum status: Unverified]					
Source: 2.1.1.2					
Destination: 2.1.1.1					
[Reassembled IPv4 in frame: 2]					
> Data (976 bytes)					
0000	08 00 27 e2 9f a6 08 00	27 fc 6a c9 08 00	45 00	..'-... 'j...E..	
0010	03 e4 b5 d0 20 00 40 01	9b 44 02 01 01 02	02 01@..D.....	
0020	01 01 08 00 4d 71 13 c2	00 01 14 2b d2 59 00 00		...Mq...+..Y..	
0030	00 00 3d 2a 08 00 00 00	00 00 10 11 12 13 14 15		..=*.....	
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25	! "\$ %	

datagramme IPv4

No.	Time	Source	Destination	Protocol	Length
1	0.000000	fc00:2:0:2::1	fc00:2:0:1::1	TCP	
> Frame 1: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)					
✓ Ethernet II, Src: 86:93:23:d3:37:8e (86:93:23:d3:37:8e), Dst: 22:1a:95:d6					
> Destination: 22:1a:95:d6:7a:23 (22:1a:95:d6:7a:23)					
> Source: 86:93:23:d3:37:8e (86:93:23:d3:37:8e)					
Type: IPv6 (0x86dd)					
✓ Internet Protocol Version 6, Src: fc00:2:0:2::1, Dst: fc00:2:0:1::1					
0110 = Version: 6					
> 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)					
.... 1101 0110 1000 0100 1010 = Flow Label: 0xd684a					
Payload Length: 40					
Next Header: TCP (6)					
Hop Limit: 64					
Source: fc00:2:0:2::1					
Destination: fc00:2:0:1::1					
> Transmission Control Protocol, Src Port: 43424, Dst Port: 8080, Seq: 0, Len: 40					
0000	22 1a 95 d6 7a 23 86 93	23 d3 37 8e 86 dd 60 0d		"...z#...#-7...`.	
0010	68 4a 00 28 06 40 fc 00	00 02 00 00 00 02 00 00		hJ.(-@...	
0020	00 00 00 00 00 01 fc 00	00 02 00 00 00 01 00 00		
0030	00 00 00 00 00 01 a9 a0	1f 90 02 1b 63 8c 00 00	c...	
0040	00 00 a0 02 67 5c 8e b9	00 00 02 04 0b 7c 04 02	g\...	
0050	08 0a 80 1d a5 22 00 00	00 00 01 03 03 07	"	

datagramme IPv6

² Un VLAN, Virtual LAN, est un réseau logique qui s'appuie sur un réseau physique partagé de type LAN. Le partitionnement correspond à des besoins fonctionnels ou organisationnels en général. Le découpage en VLAN isole les réseaux logiques les uns des autres.

Dans le datagramme IPv6, décrit dans la partie verbeuse de Wireshark (centre de la page de trace), on voit apparaître une adresse source IPV6. Elle apparaît dans le format dit compact. Donner cette adresse au format compact, et donner la valeur correspondante qu'on trouve dans la trace en hexadécimal. En déduire une règle de compactage des adresses IPv6 pour les écrire.

On peut regarder la correction après chaque question.

Correction :

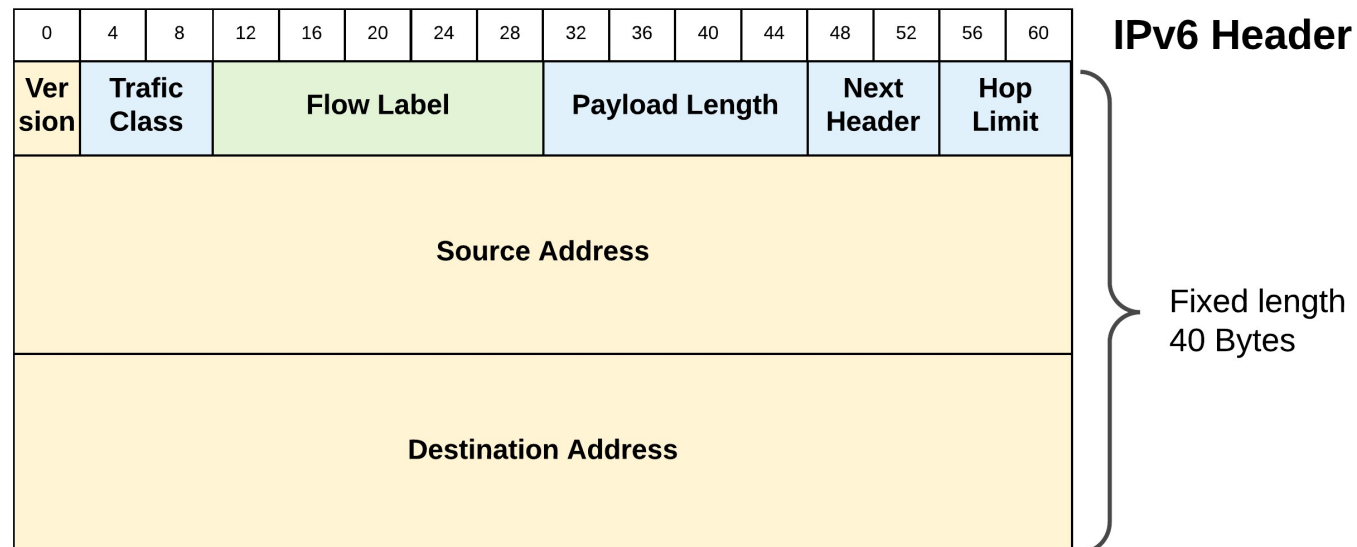
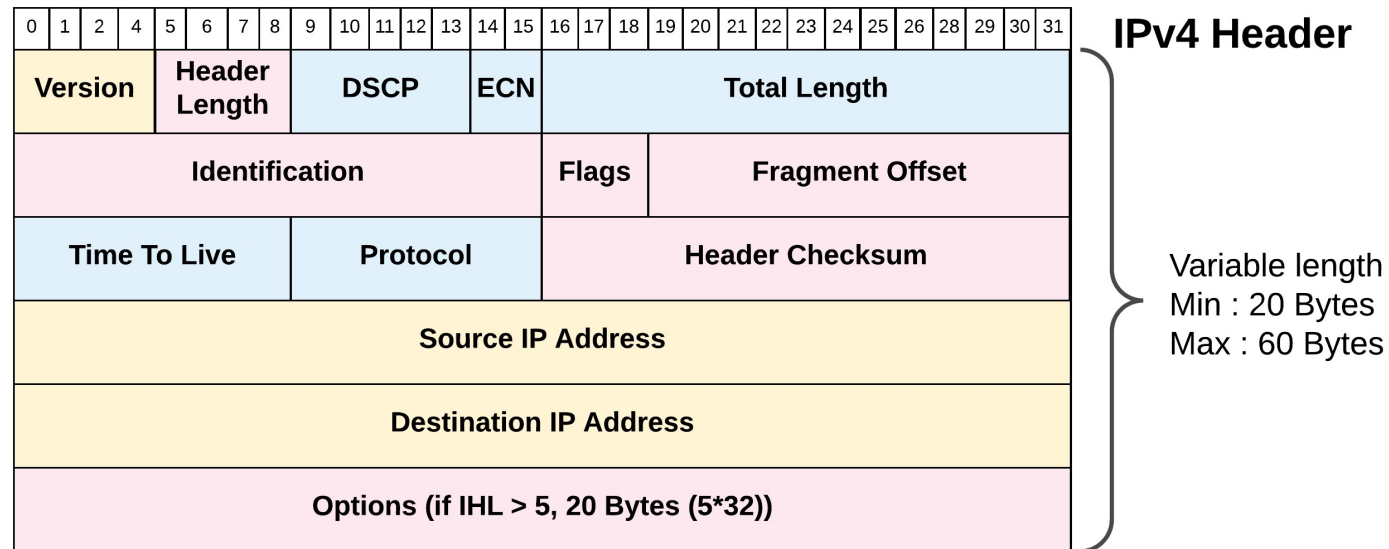
`fc00:2:0:1::1` apparaît comme adresse IP source du datagramme IPv6. Dans la partie hexadécimale de la trace, on observe une valeur différente longue de 16 octets : `fc 00 00 02 00 00 00 02 00 00 00 00 00 00 00 01`

Grossièrement on pourrait résumer la règle ainsi :

- L'adresse est découpée en blocs de 2 octets séparés par ":"
- Si des zéros sont à droite d'un nombre dans un bloc on doit les laisser
- Si des zéros sont à gauche d'un nombre dans un bloc on peut les supprimer
- Si on a une suite de zéros on peut les compacter et les remplacer par `::`, on apprend aussi qu'il ne peut y avoir deux fois `::` dans une adresse IPv6

On observe bien que suivant qu'on est en IPv4 ou en IPv6 la structure de l'entête diffère tant par les champs et leur longueur que leur rôle. Certains ont complètement disparu.

Dans la figure qui suit, compléter les différents champs à partir des deux traces Wireshark, vous pouvez utiliser les valeurs décimales.



Correction :

Remplissage pour l'entête IPv4 :

No.	Time	Source	Destination	Protocol	Length
1	0.000000	2.1.1.2	2.1.1.1	IPv4	
<					
> Frame 1: 1010 bytes on wire (8080 bits), 1010 bytes captured (8080 bits)					
> Ethernet II, Src: PcsCompu_fc:6a:c9 (08:00:27:fc:6a:c9), Dst: PcsCompu_e2					
v Internet Protocol Version 4, Src: 2.1.1.2, Dst: 2.1.1.1					
0100 = Version: 4					
.... 0101 = Header Length: 20 bytes (5)					
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)					
Total Length: 996					
Identification: 0xb5d0 (46544)					
> Flags: 0x2000, More fragments					
Fragment offset: 0					
Time to live: 64					
Protocol: ICMP (1)					
Header checksum: 0x9b44 [validation disabled]					
[Header checksum status: Unverified]					
Source: 2.1.1.2					
Destination: 2.1.1.1					
[Reassembled IPv4 in frame: 2]					
> Data (976 bytes)					
0000	08 00 27 e2 9f a6 08 00	27 fc 6a c9 08 00 45 00	..'. '.j...E.		
0010	03 e4 b5 d0 20 00 40 01	9b 44 02 01 01 02 02 01@. .D.....		
0020	01 01 08 00 4d 71 13 c2	00 01 14 2b d2 59 00 00Mq... ..+..Y..		
0030	00 00 3d 2a 08 00 00 00	00 00 10 11 12 13 14 15	...*=.....		
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 !"#%\$		

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4				5				CS0 00								996															
b5d0															10			0													
64					1										9b44																
2.1.1.2																															
2.1.1.1																															
Vide, car pas d'option, longueur entête 20 octets																															

Remplissage pour l'entête IPv6

No.	Time	Source	Destination	Protocol	Length
1	0.000000	fc00:2:0:2::1	fc00:2:0:1::1	TCP	60
> Frame 1: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)					
Ethernet II, Src: 86:93:23:d3:37:8e (86:93:23:d3:37:8e), Dst: 22:1a:95:d6:7a:23 (22:1a:95:d6:7a:23)					
Destination: 22:1a:95:d6:7a:23 (22:1a:95:d6:7a:23)					
Source: 86:93:23:d3:37:8e (86:93:23:d3:37:8e)					
Type: IPv6 (0x86dd)					
Internet Protocol Version 6, Src: fc00:2:0:2::1, Dst: fc00:2:0:1::1					
0110 = Version: 6					
> 0000 0000 = Traffic Class: 0x00 (DSCP: CS0)					
..... 1101 0110 1000 0100 1010 = Flow Label: 0xd684a					
Payload Length: 40					
Next Header: TCP (6)					
Hop Limit: 64					
Source: fc00:2:0:2::1					
Destination: fc00:2:0:1::1					
> Transmission Control Protocol, Src Port: 43424, Dst Port: 8080, Seq: 0, Len: 0					
0000	22 1a 95 d6 7a 23 86 93 23 d3 37 8e 86 dd 60 0d	"...z#...#7...~..			
0010	68 4a 00 28 06 40 fc 00 00 02 00 00 00 02 00 00	hJ-(-@-.....			
0020	00 00 00 00 00 01 fc 00 00 02 00 00 00 01 00 00			
0030	00 00 00 00 00 01 a9 a0 1f 90 02 1b 63 8c 00 00c...			
0040	00 00 a0 02 67 5c 8e b9 00 00 02 04 0b 7c 04 02	...g\... ...			
0050	08 0a 80 1d a5 22 00 00 00 00 01 03 03 07".....			

0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
6	00	d684a						40				6	64		
fc00:2:0:2::1															
fc00:2:0:1::1															



Merci pour votre attention et votre persévérance !!!!!

³ Troupe d'élite dans BoomBeach de SuperCell, bombardier lanceur de pastèques explosives, https://boombeach.fandom.com/wiki/Melon_Bombardier (29/08/2021)