Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

2017-2018

Date : 19 avril 2018

Prénom, Nom : Yanick Zemba

Réseaux IP
Travail écrit no 3

5^6

Informations importantes :

- Le temps disponible est de 1h30. Vous pouvez aussi répondre en allemand ou en anglais.
- Le travail est individuel. La seule documentation autorisée est (1) le formulaire personnel (1 feuille A4, deux côtés, manuscrite) qui doit être rendu et (2) l'éventuel formulaire officiel, fournit avec le travail écrit, sans annotations
- Il est important de bien lire les questions jusqu'à la fin. La démarche est très importante. *Un résultat sans développement ou explication ne sera pas accepté. N'oubliez pas les unités !*

Question :	1	2	3	4	5	6	7	Total
Points :	4	6	10	14	10	6	10	60

Question 1 (4 points)

(a) (2 points) Les flux de données qui utilisent le protocole UDP ne sont soumis à aucun contrôle de flux. Peuvent-ils devenir un problème pour les applications utilisant le protocole TCP ? Justifiez votre réponse.

Le protocole TCP contrôle le flux à la destination en utilisant le système de la sliding window, prenant de réguler le flux de donnée en fonction de la vitesse de traitement du récepteur.

Cela peut ainsi devenir un problème si le récepteur devient surchargé (dû à une attaque malveillante, ou non) et que ses buffers se remplissent : il aura donc une taille de fenêtre nulle, ou très petite, bloquant la communication complètement.

c'est les mêmes buffers ?

L'idée c'était de montrer l'effet d'UDP sur TCP ("Time out" fréquent)

(b) (2 points) Est-ce que l'insertion de messages malveillants (par une personne tierce) dans une connexion est plus facile avec TCP ou avec UDP ? Justifiez votre réponse.

Il est plus difficile d'insérer des messages dans une connexion TCP. En effet, ce dernier utilise des numéros de séquence cumulatif pour numéroter les segments et le numéro de segment initial est lui-même défini sur une base aléatoire (~ lié à un compteur d'une horloge). Il faudrait donc connaître le numéro de segment actuel pour pouvoir transférer un segment bien sûr que la connexion le remarque.

UDP n'ayant pas ces mécaniques (segment et ack cumulatif), la problématique ne le concerne pas. (il n'y a aucun contrôle de données entre la source et la destination).

Question 2 (6 points)

(a) (2 points) Vous vous occupez d'un stagiaire au sein de votre entreprise. Celui-ci a configuré une interface de routeur avec l'adresse IP 194.235.12.191 et le masque 255.255.255.224.
Quelle est votre réaction? Expliquez pourquoi.

Explication binaire
224 host bits
111 00000
191
101 11111
tout à 1
→ broadcast

Son adresse IP ne fonctionnera pas, car il s'agit de l'adresse de broadcast de son sous-réseau. En effet, un masque '/27' découpe l'adresse en sous-réseaux de 32 adresses → 30 hôtes. L'adresse utilisée est dans le sous-réseau 194.235.12.160/27 la dernière adresse de ce sous-réseau est son broadcast, et c'est bien 194.235.12.191.

(b) (2 points) Quelques jours plus tard, il vous demande de configurer une interface de routeur avec l'adresse IP 194.235.23.160/29.
Quelle est votre réaction? Expliquez pourquoi.

Explication binaire
/29 host bit
11111 000
160
101 00000
tout à 0
→ adresse réseau

Cette fois, cela ne fonctionne pas car c'est l'adresse du sous-réseau qui est utilisée. De la même manière qu'au dessus, son sous-réseau s'étend de 194.235.23.160 à 194.235.23.167. Ces deux adresses "limites" ne sont pas disponibles ✓

(c) (2 points) Votre entreprise possède les blocs d'adresses IP suivantes : 194.235.0.0/20, 194.235.16.0/22, 194.235.20.0/22, 194.235.32.0/21 et 194.235.40.0/21. On vous demande de déterminer le(s) supernet(s) correspondant(s).

① 194.235.0.0/20 → 194.235.0.0 à 194.235.15.255
② 194.235.16.0/22 → 194.235.16.0 à 194.235.19.255
③ 194.235.20.0/22 → 194.235.20.0 à 194.235.23.255
④ 194.235.32.0/21 → 194.235.32.0 à 194.235.39.255
⑤ 194.235.40.0/21 → 194.235.40.0 à 194.235.47.255

On a donc : 194.235.0.0/20 ✓ ①
194.235.16.0/21 ✓ ② + ③
et 194.235.32.0/20 ✓ ④ + ⑤

On ne peut faire mieux car les adresses des réseaux ~~=~~ 240 à 32.0

Question 3 (10 points)

Soit un ISP (*Internet Service Provider*) qui s'est vu attribuer un bloc de 4096 adresses réseau de classes C situés entre 123.32.0.0 et 123.47.255.0.

(a) (2 points) Quel est le préfixe le plus long (*supernet*) partagé par l'ensemble de ces adresses?

123.32.0.0 /13 ~~/13~~

8+4=13, cqfd Zamboni

(b) (3 points) Combien aurait-il fallu attribuer d'adresses de classes B à l'ISP pour être capable d'adresser le même nombre d'hôtes?

4096 adresses : (256-2) hôtes = 1'040'384 hôtes (1)
↑ classe C

1 classe B : $2^{16}-2$ hôtes = 65534 hôtes (2)

(1) / (2) = 15,8 ... il faut donc 16 adresses de classe B. ✓

(c) (3 points) Deux clients C1 et C2 de cet ISP justifient l'utilisation respective de 2000 adresses et de 1000 adresses. Quelles adresses l'ISP leurs attribue-t-il? Donner pour C1 et C2 un exemple, avec l'adresse du réseau, le masque et la plage d'adresses pour les stations.

C1 → 2000 adresses → 11 bits d'hôte → /21

plage : 123.32.0.0 (↳ adresse du réseau) à 123.32.7.255 : 123.32.0.0/21 ✓

C2 → 1000 adresses → 10 bits d'hôte → /22

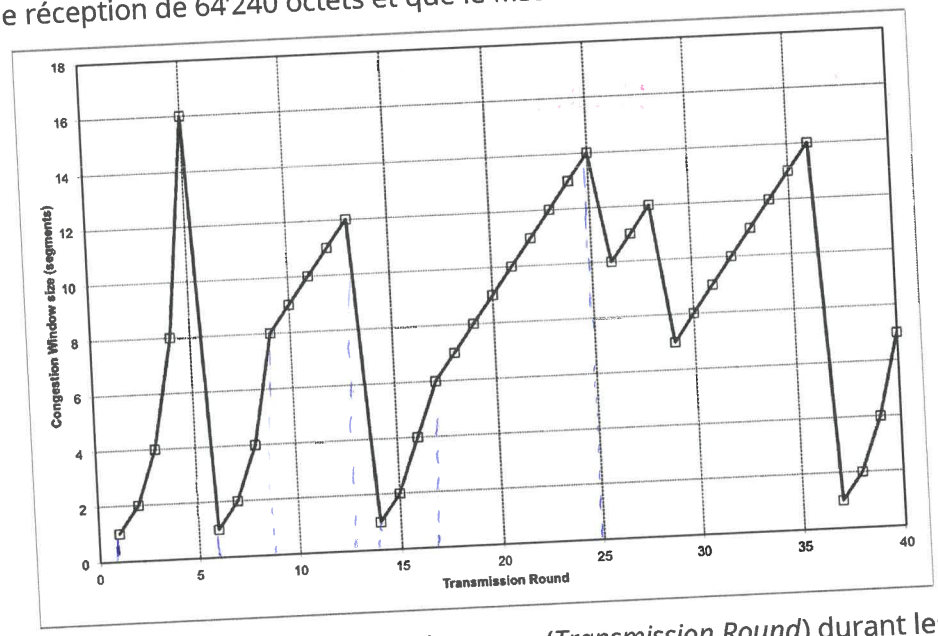
plage : 123.32.8.0 à 123.32.11.255 : 123.32.8.0/22 ✓

(d) (2 points) Quel(s) est(sont) le(s) préfixe(s) CIDR (ou supernet) qui comprend (comprennent) les adresses entre 123.32.12.0 et 123.32.31.0? ← adresse des réseaux

de 123.32.12.0 à 123.32.15.255 → 123.32.12.0/22
de 123.32.16.0 à 123.32.31.255 → 123.32.16.0/20 ✓

Question 4 (14 points)

L'algorithme de contrôle de congestion appelé TCP Reno à ajouté les mécanismes de *Fast Recovery* et *Fast Retransmit*. On considère le graphique suivant qui représente la taille de la fenêtre de congestion (*cwnd*, en multiple de segments) en fonction de la transmission de segments (*Transmission Round*). De plus, on considère que la station réceptrice à une fenêtre de réception de 64'240 octets et que le MSS vaut 1460 octets.



(a) (5 points) Identifiez les intervalles de temps (*Transmission Round*) durant lesquels les algorithmes *slow start (SS)* ou *congestion avoidance (CA)* (prévention des congestions) sont actifs.

Justifiez votre réponse.

seuil ? ✓

Intervalle [début, fin]	Phase (SS, CA)	sshthresh [segments]	Evénement / causes
[1, 5]	SS	44	Lancement de la connexion.
[6, 9]	SS	8 (16/2)	- Time out en '5' - seuil /2
[9, 13]	CA	8	- seuil atteint
[14-17]	SS	6 (12/2)	- Time out en '13' - seuil /2
[17-25]	CA	6	- seuil atteint
[26, 28]	CA	7 (14/2)	- 3 fois le même Ack reçu → fast retransmit → on recommence à 7+3
[29, 36]	CA	7	fast recovery → on recommence à 7
[37, 40]	SS	7 (14/2)	- Time out en 36

(b) (2 points) Au 25ème *Transmission Round*, la perte du segment est-elle détectée par l'expiration d'un temporisateur ou par la réception de 3 duplicata ACK? Justifiez votre réponse.

Par la réception de 3 duplicatas. On passe en effet en fast retransmit avec une fenêtre de congestion à 10 MSS (seuil + 3 MSS) puis en fast recovery.

Si l'expiration du temporisateur avait fait tomber la fenêtre de congestion à 1 MSS.

(c) (3 points) Au 40ème *Transmission Round*, combien de bytes auront été transféré?

Au 40e round, la fenêtre de congestion est de 7 MSS. En comptant sans en-tête IP et TCP de 40 octets, on transfère
7 · 1500 = 10.500 bytes = 10.5 KB

Sur toute la connexion, où la fenêtre de réception a été capable de "suivre", on transfère 316 segments, → 316 · 1500 = 474 kB

calcul?

(d) (4 points) En situation stable, a plein régime, quel sera le débit TCP effectif pour un RTT de 12ms?

A plein régime, W vaut 64'240 octets.

Le débit TCP est donné par $D = \frac{W}{RTT} = \frac{64'240 \cdot 8}{12 \cdot 10^{-3}} = 42.8$ Mb/s

Question 5 (10 points)

Vous observer la trame suivante avec un *analyseur de protocole* :

```
Ethernet II,
     Destination: 00:04:23:a5:b9:57
     Source: 00:1b:63:ae:45:1b
     Type: IP (0x0800)
Internet Protocol,
     Version: 4
     Header length: 20 bytes
     Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
         0000 00.. = Differentiated Services Codepoint: Default (0x00)
         .... ..0. = ECN-Capable Transport (ECT): 0
         .... ...0 = ECN-CE: 0
     Total Length: 64
     Identification: 0x1eb1 (7857)
     Flags: 0x04 (Don't Fragment)
         0... = Reserved bit: Not set
         .1.. = Don't fragment: Set
         ..0. = More fragments: Not set
     Fragment offset: 0
     Time to live: 64
     Protocol: TCP (0x06)
     Header checksum: 0x9ca0
     Source: 160.98.31.130
     Destination: 160.98.31.32
Transmission Control Protocol
     Source port: 55001 (55001)
     Destination port: ssh (22)
     Sequence number: 1352386613
     Header length: 44 bytes
     Flags: 0x02 (SYN)
         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
         .... ..1. = Syn: Set
         .... ...0 = Fin: Not set
     Window size: 65535
     Checksum: 0xa1e2
     Options: (24 bytes)
         Maximum segment size: 1460 bytes
         NOP
         Window scale: 3 (multiply by 8)
         NOP
         NOP
         SACK permitted
```

(a) (1 point) Comment est-ce que l'analyseur de protocole a reconnu qu'il s'agissait du protocole de couche 4 TCP ?

le Champ protocole vaut 0x06 → TCP ✓

(b) (1 point) Est-ce que cette trame fait partie de l'établissement d'une connexion TCP ? Pourquoi ? Si oui, qui est à l'origine de ce segment (le client ou le serveur) ?

Oui, car le flag SYN est à 1.
C'est le client qui est à l'origine de ce segment (la destination est un well-known port). Il s'agit du premier segment de la connexion 3-way handshake. ✓

(c) (1 point) Quel est le numéro du port *well-known port number* utilisé par la connexion TCP? De quelle application s'agit-il?

22 → ssh ✓

(d) (1 point) Pourquoi est-ce que les ports de source et de destination TCP ne sont pas les mêmes?

Cela permet d'avoir plusieurs connexions distinctes entre la source et la destination. D'autant que "la source" représente souvent plusieurs stations cachées derrière un NAT/PAT. ✓

(e) (2 points) Quelle est la taille de la fenêtre de réception annoncée?

Elle est de "window size" : 2 "window scale"
soit 65535 bytes $\cdot 2^3 = 524'280$ bytes ✓

(f) (2 points) Quelles sont les options supportées (avec leur valeur) par la source de ce segment TCP? Décrivez l'utilité de chacune de celles-ci.

- MSS : 1460 bytes : Permet de définir la taille maximale des segments supportés par le réseau sans compter l'en-tête IP et TCP. ✓
- Window scale : 3 : Permet d'augmenter la taille de la fenêtre d'un facteur 2 (WS) pour permettre un meilleur débit. La temp window d'origine ne permet plus d'être "réaliste" avec les débits actuels. ✓
- SACK : permitted : Autorise l'acquittement sélectif, ce qui permet d'acquitter des segments "perdus" sans faire de time-out ✓ (fast retransmit → fast recovery)

(g) (2 points) Complétez l'entête TCP du segment qui va être émis en réponse au segment ci-dessus.

```
1  Transmission Control Protocol, Len: 0
2      Source port           : 22
3      Destination port      : 55001
4      Sequence number       : 12345
5      Acknowledgement number: 135 23866 14
6      Header length         : 40 bytes
7      Flags:
8          0... .... = Congestion Window Reduced (CWR): Not set
9          .0.. .... = ECN-Echo: Not set
10         ..0. .... = Urgent: Not set
11         ...1 .... = Acknowledgment
12         .... 0... = Push
13         .... .0.. = Reset
14         .... ..1. = Syn
15         .... ...0 = Fin
16     Window size     : 5792
```

on ne peut pas savoir ✓ ✓

Question 6 (6 points)

On considère une station A effectuant un transfert de fichier de taille f vers une station B au travers d'une session TCP. Le MSS est de 1460 bytes et le débit du lien est de 100Mb/s.

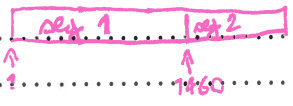
(a) (4 points) Sachant que le champ *Numéro de séquence* dans l'entête TCP est de 4 bytes, on vous demande de calculer la taille maximum f de telle sote que le nombre de numéro de séquence à disposition ne soit pas épuisés. Remarque : On part du principe que 1 kByte = 1000 Bytes.

On a 2^{32} numéro de séquence ✓

On peut donc envoyer $2^{32} \cdot 1460$ bytes de données au maximum

soit : 6,2 TB

$2^{32} = 4,3$ GB et c'était tout



1 numéro de séquence = 1 Byte

(b) (2 points) Combien de temps faudra-t-il pour trasnférer ce fichier de taille f ? On considère une émission en continue et en régime stationnaire, avec aucune perte dans le réseau. Les entêtes de la couche Liaison de donnée (Ethernet, 26 Bytes), réseau et transport sont à tenir compte dans votre calcul.

On doit transférer ~~2^{32}~~ segments de $\frac{4,3\,GB}{1460\,B}$

1460 + 20 + 20 + 26 = 1526 bytes à 100 Mb/s ✓

$\frac{2^{32} \cdot 1526 \cdot 8}{100 \cdot 10^6} = 5243300 \rightsquigarrow 6$ jours

Question 7 (10 points)

Pour les différentes questions ci-dessous, veuillez sélectionner la ou les bonnes réponses, selon les indications. Sans commentaires particuliers, il y a une seule réponse possible.

(a) (1 point) La route indiquée par traceroute peut ne pas être réelle car il y a éventuellement plusieurs chemins sur Internet.

- ✓ A. Vrai
- B. Faux

(b) (1 point) Combien de stations peuvent être adressées par le bloc d'adresse `160.98.30.64/27`

- A. 32
- B. 64
- ✓ C. 30
- D. 62

(c) (1 point) Lorsqu'un fragment d'une communication IP a été perdu, que se passe-t-il au niveau du destinataire?

- A. Le destinataire demande la retransmission du fragment perdu, grâce au numéro d'indentification du paquet IP.
- B. La source détecte la perte du fragment et effectue une retransmission
- ✓ C. Le destinataire supprime tous les fragments reçus. La couche supérieure (p.ex. TCP) doit gérer la retransmission.
- D. La source envoi plusieurs copies des fragments pour palier aux erreurs de transmission

(d) (1 point) Le PAT (*Port Address Translation*) traduit les addresses privées en se basant sur :

- A. port de source
- B. port de destination
- ✗ C. les deux
- D. aucun des deux

(e) (1 point) Quelle commande permet à un serveur FTP avec l'adresse *inside local* `10.5.9.100` d'être accessible depuis l'interface *ethernet0* (qui est directement connectée à Internet)?

- A. `ip nat inside source static tcp interface ethernet0 21 10.5.9.100 21`
- ✓ B. `ip nat inside source static tcp 10.5.9.100 21 interface ethernet0 21`
- C. `ip nat inside destination static tcp interface ethernet0 21 10.5.9.100 21`
- D. `ip nat inside destination static tcp 10.5.9.100 21 interface ethernet0 21`

(f) (1 point) Un message ICMP "*time exceeded*" est généré lorsque :

- A. Le RTT entre deux stations est proche de zéro
- B. Les fragments d'un message arrive hors délai
- ✗ C. Le calcul du checksum IP prend trop de temps
- D. Aucune des réponses ci-dessus → [redacted] = 0 TTL

(g) (1 point) Quelle est la notation CIDR du masque 255.128.0.0

A. /7

B. /8

C. /9

D. /10

(h) (1 point) Le syndrome de la fenêtre stupide (*Silly Windows Syndrome*) est évité grâce à deux mécanismes :

A. L'envoi de nouveaux segments sont réglés selon l'algorithme de Fast Recovery

B. La destination modifie (ouvre) sa fenêtre de réception une fois que son buffer est vide

C. La destination modifie (ouvre) sa fenêtre de réception d'au moins un MSS

D. L'envoi de nouveaux segments sont réglés selon l'algorithme de Nagle

E. Lors de l'apparition de ce syndrome, l'algorithme de *Slow Start* est activé

(i) (1 point) Dans quel(s) cas l'émetteur d'une session TCP va retransmettre le même segment ?

A. Si le récepteur n'envoie pas un acquittement durant un intervalle de temps

B. Si le récepteur envoie 3 acquittements dupliqués

C. Dans les deux cas mentionnés ci-dessus

(j) (1 point) Si une station reçoit un segment TCP avec SEQ=1234, ACK=5678 et si celui-ci à une longueur de 1000 octets, quel sera sa réponse ?

A. SEQ=6678, ACK=1234

B. SEQ=2234, ACK=5678

C. SEQ=1234, ACK=2234

D. SEQ=5678, ACK=2234