

Travaux Dirigés N°4

Exercice N°1

Ci-contre une capture d'un entête TCP faite par *Wireshark* sous forme hexadécimale. Quel est:

00 50 05 39 e9 07 92 c0 4e e7 e7 98 70 12
20 00 fa eb 00 00 02 04 05 b4 01 01 04 02

- (a) le numéro de port Source ?
- (b) le numéro de port destination ?
- (c) la longueur d'en-tête en octet (en déduire la taille du champ Options) ?
- (d) quel est le rôle de ce segment ?
 1. segment de données
 2. demande de connexion
 3. réponse à une demande de connexion
 4. demande de fermeture de connexion

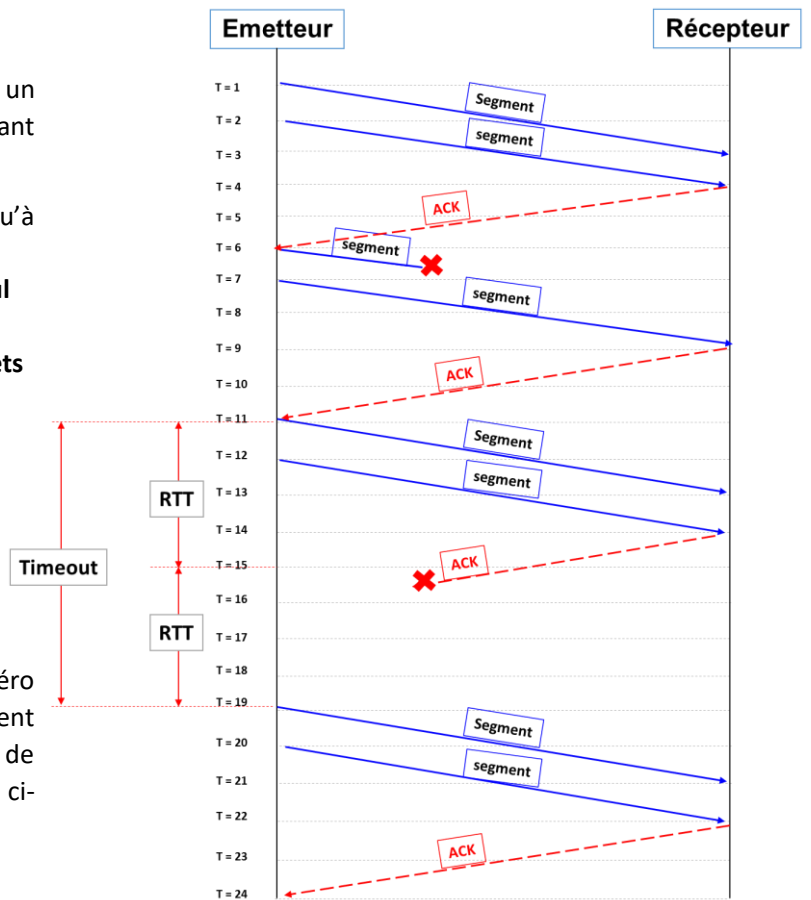
Numéro de port source (16b)						Numéro de port destination (16b)							
Numéro de séquence (32b)													
Numéro d'accusé de réception (32b)													
Longueur (4b)	0000	C	E	U	A	P	R	S	F	Taille de la fenêtre			
		W	C	R	C	S	S	Y	I				
		R	N	G	K	H	T	N					
Somme de contrôle TCP						Pointeur d'urgence							
Options													

Exercice N°2

Se référer à la figure ci-contre. Un émetteur et un récepteur communiquent via une connexion TCP ayant les caractéristiques suivantes :

- Un segment de données peut comporter jusqu'à **100 octets** de données
- Un segment d'acquittement comporte **un seul octet**
- La taille de la fenêtre glissante est de **200 octets**
- Le numéro de séquence initial (SEQ) de l'émetteur est de **110**
- Le numéro de séquence initial (SEQ) du récepteur est de **320**
- RTT = 4 unité de temps (T)
- Timeout = 2 RTT
- L'acquittement est cumulatif

Fournir le numéro de séquence (SEQ) et le numéro d'accusé de réception (N° ACK) de chaque segment échangé entre l'émetteur et le récepteur à partir de T = 1 à T = 24 (servez-vous du modèle de tableau ci-dessous)



Temps d'envoi	Direction - Segment	Numéro de séquence	Numéro d'acquittement
T = 1	Emetteur → Récepteur	110	-----
...

Exercice N° 3

Dessiner le diagramme de temps d'un algorithme à fenêtre glissante de **4 octets** de taille et des segments pouvant comporter **un seul octet** de charge, dans les deux cas (a) et (b) ci-dessous. On suppose que le récepteur envoie un **N-ACK** quand il ne reçoit pas le bon segment. Ex. il envoie N-ACK[2] quand il reçoit Segment[3] alors qu'il prévoit Segment[2]. Le récepteur envoie également un ACK cumulatif lorsqu'il reçoit tous les segments. Ex. le récepteur envoie ACK[5] dès qu'il reçoit Segment[2], Segment [3], Segment [4], et Segment [1]. Timeout = 2 RTT, le nombre de segments à envoyer est 8.

- (a) Segment [2] est perdu, la retransmission aura lieu après un timeout
- (b) Segment [2] est perdu, la retransmission aura lieu dès la réception du premier NACK.