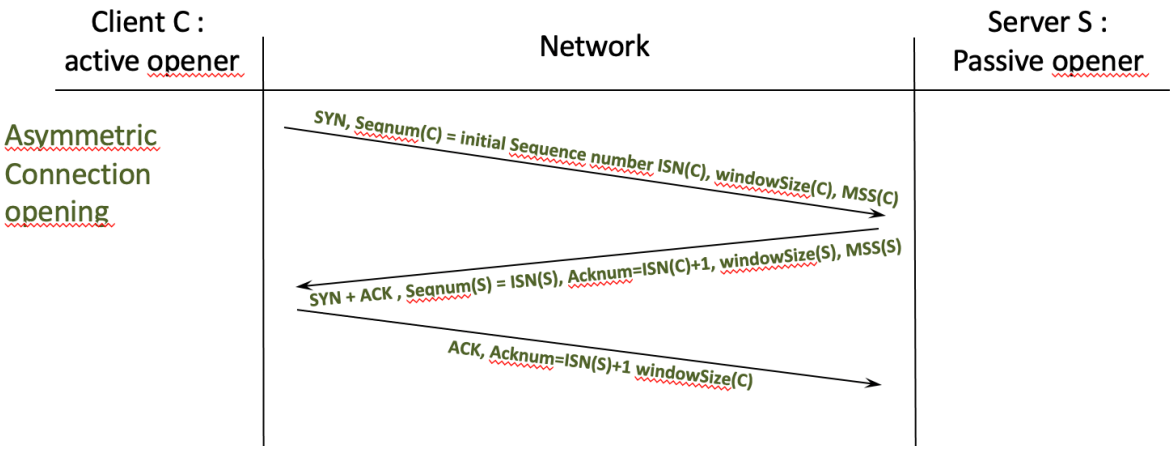
Correction TD 6

1.a. Avec 2 messages, le sous-réseau pouvant déséquencer les paquets, l'appelant de la connexion reçoit un segment de données pour une connexion qui est en attente de confirmation d'établissement. Avec 3 messages, l'appelé doit attendre de recevoir un acquittement de sa confirmation (sous la forme de données ou d'acquittement) avant de pouvoir émettre ses propres données. On évite ainsi à l'appelant de se retrouver dans un état mal défini.

1.b.



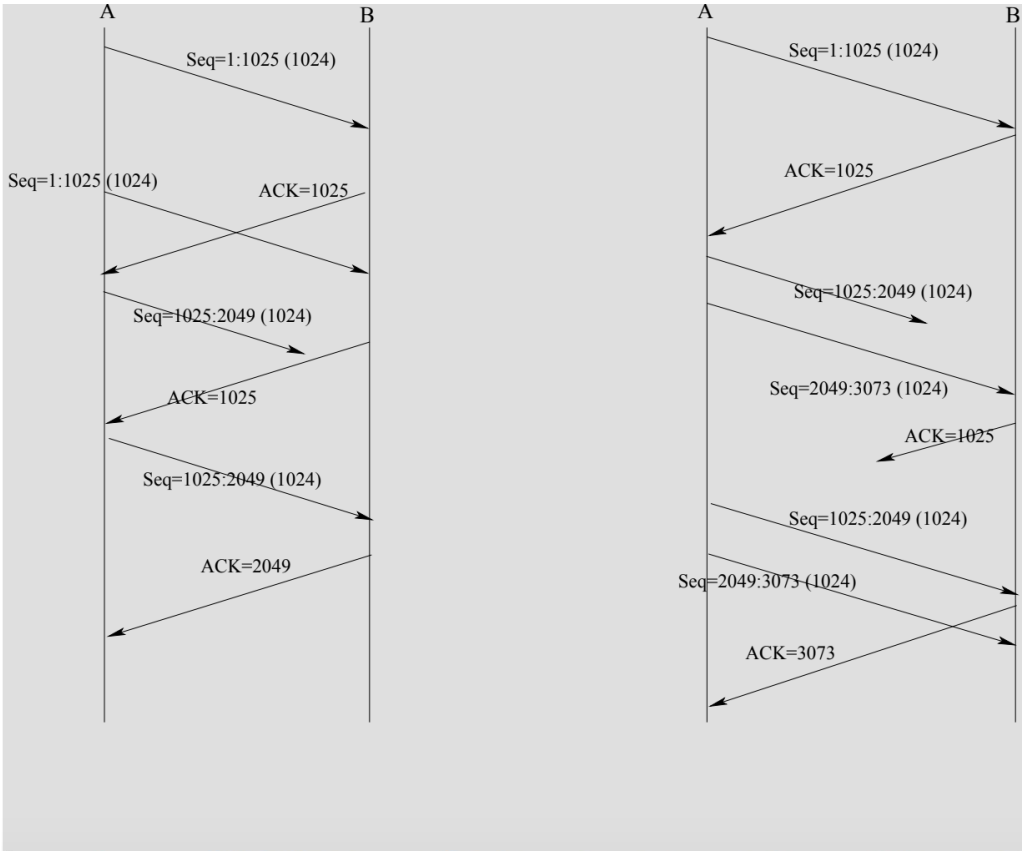
1.c. Une connexion TCP est en fait identifiée par (port S, hôte S, port D, hôte D). Il est tout à fait possible d'ouvrir une connexion entre les machines A et B, ports X et Y, de la fermer, puis quelque temps après, d'ouvrir une nouvelle connexion, toujours entre les machines A et B et les ports X et Y. On parle dans ce cas d'incarnations différentes d'une même connexion.
Si une seconde incarnation réutilise un numéro de séquence trop tôt, c'est-à-dire quand il y a encore une chance qu'un segment appartenant à une incarnation précédente interfère avec l'incarnation courante.
Le numéro de séquence initial (ISN) est choisi de façon à éviter cette situation. A l'origine, il devait être tiré aléatoirement ; en pratique, il est généralement mis à jour grâce à un compteur de 32 bits incrémenté toutes les 4s.

1.d. Si le SYN est une retransmission, sa valeur de ISN est la même que dans le premier SYN. Dans le cas contraire, et si les valeurs d'ISN sont générés par une horloge, alors l'ISN du second SYN sera différent de celui du premier SYN.

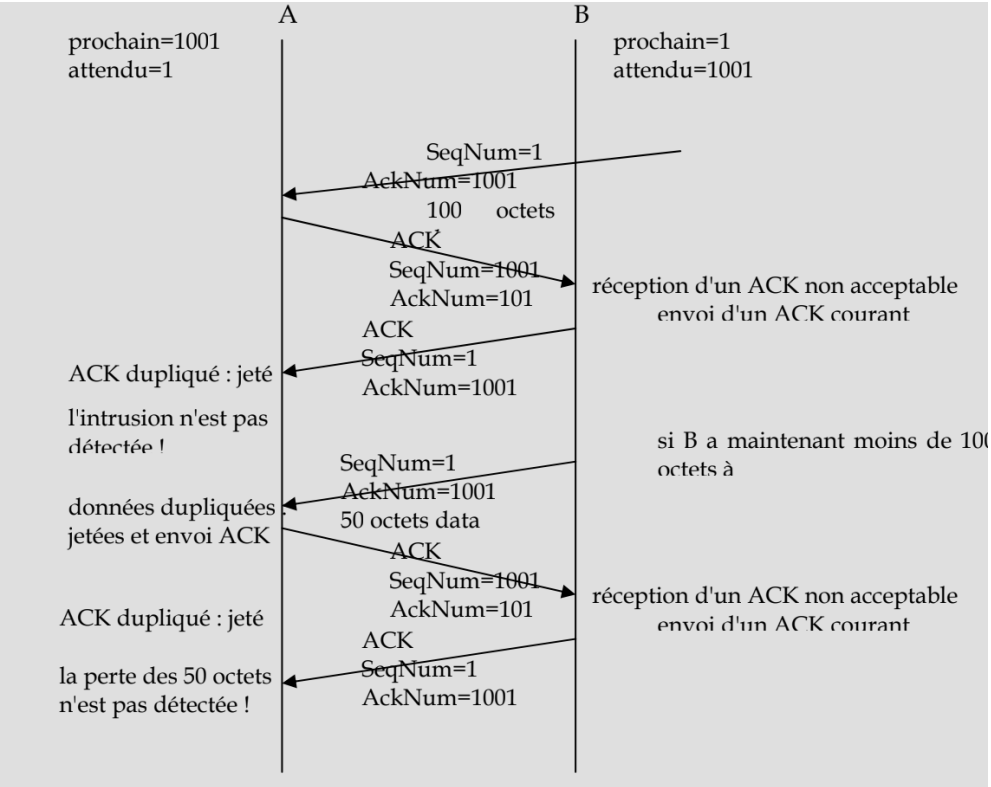
2.a. NumSeq ne démarre pas à 0 pour un transfert : le numéro initial ISN est généré, soit de façon aléatoire, soit le plus souvent sur la base d'une horloge (incrémentée toutes les 4 s).

2.b. 1 Gbit/s, cela fait exactement 125 Mo/s. Par ailleurs, les numéros de séquence rebouclent quand on envoie 2^32 octets c'est-à-dire environ 4 Go. Par conséquent, cela prend 4 Go / 125 Mo/s, soit 32 s.

2.c.

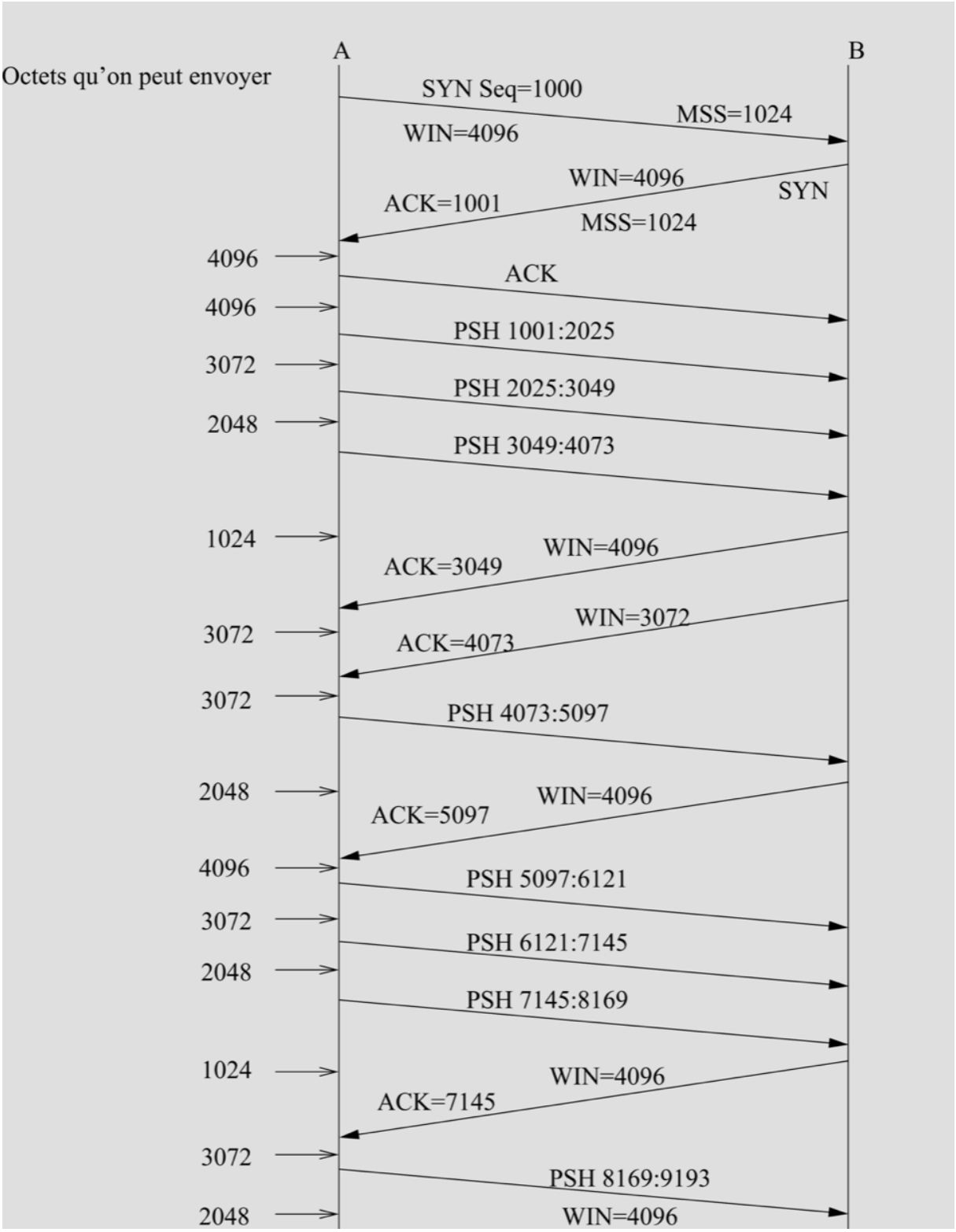


2.d



1.a la taille de la fenêtre change dynamiquement selon l'occupation du réseaux d'un côté et l'occupation de la machine liées à la file d'attente d'un autre côté.
Les acteurs s'informent mutuellement de la taille de la fenêtre en le champ « window » de l'entête TCP.

3.b



3.c. AdvWin doit être suffisamment grand pour pouvoir remplir le « pipe » (i.e. le tuyau que constitue la connexion) ; or, la contenance du tuyau correspond au produit délai * BP qui est ici de 100 ms * 100 Mbit/s = 10 Mbits = 1,25 Mo de données, ce qui demande 21 bits (221 = 2.097.152 et 220 = 1.048.576) pour le champ AdvWin.

3.d. SeqNum ne doit pas reboucler pendant MSL ; en 60s, on peut envoyer 60 x 100. 106 bits, soit 750 Mo. Or 750 Mo = 750.103.103 ≈ 750.1024.1024 = 750.210.210. Donc, il faut prendre 30 bits.

3.e.
La BP est directement déterminée par le matériel (hardware).
Le RTT est lui aussi une mesure précise, mais qui est affectée par tout changement dans la taille et la charge du réseau (le RTT varie avec la paire d'hôte considérées et l'heure de la connexion).
Le MSL est probablement la valeur la moins certaine, puisqu'elle dépend non seulement de la taille et de la complexité du réseau, mais aussi du temps mis à résoudre les boucles de routage. Le RFC 793 préconise un MSL de 120 s.

3.f. Si on responsabilise le récepteur, en lui faisant indiquer explicitement la réouverture de sa fenêtre par l'envoi spontané d'un ACK avec AdvWin > 0, le problème qui se pose est que le récepteur ne peut avoir confirmation du fait que l'émetteur a bien reçu son ACK que lorsqu'il reçoit de nouvelles données ; et si le ACK s'est perdu... On fait alors appel à un temporisateur, qui est armé à la réouverture de la fenêtre, désarmé à la réception de données ; sur expiration, un ACK avec une valeur de fenêtre éventuellement mise à jour est renvoyé.

3.g.
1) dès qu'il a MSS (Maximum Size Segment) octets de données à envoyer MSS = MTU du sous-réseau local – entête IP – entête TCP (rq : les entités TCP se communiquent leurs MSS respectifs au moment de l'établissement (champ options) et c'est la valeur la plus faible qui gagne) (rq : valeur par défaut : 536 octets (+20 en-tête)) ;
2) lorsque son application lui demande explicitement : TCP supporte une fonction push que le processus peut invoquer pour vider le buffer des octets en attente d'émission (ex : c'est utile pour l'émulation de terminaux (Telnet), où chaque caractère doit être envoyé aussitôt taper) ;
3) sur expiration d'un temporisateur : pour éviter d'attendre trop longtemps les MSS octets.

4. a. Il est nécessaire d'incrémenter le AckNum d'un FIN, de façon à ce que l'émetteur du FIN puisse déterminer si son FIN a bien été reçu, et non pas seulement les données précédentes. Pour un segment SYN, tout ACK pour des données postérieures au SYN incrémente AckNum : un tel ACK va donc acquitter implicitement le SYN (puisque des données ne peuvent pas être acquittées tant que la connexion n'est pas établie). Par conséquent, le fait d'avoir [ACK, AckNum = x+1] au lieu de [ACK, AckNum = x] relève plus d'une convention dans un souci d'homogénéité que d'une nécessité protocolaire.