Réseaux 6e édition

Andrew Tanenbaum, Nick Feamster, David Wetherall

Corrigés des exercices

Chapitre 6 : La couche Transport

ISBN : 978-2-3260-0239-5

1. L'appel LISTEN pourrait indiquer la volonté d'établir de nouvelles connexions mais sans blocage. En faisant la tentative de connexion, on pourrait envoyer à l'appelant un signal. Il exécuterait alors, par exemple, OK ou REJECT pour accepter ou refuser la connexion. Dans notre schéma d'origine, cette souplesse n'est pas permise.
2. TCP fournit un flux d'octets. Les données que renvoie **receive()** peuvent correspondre à un message complet ou partiel ou à plusieurs messages. Le protocole applicatif doit définir ses propres frontières.
3. La ligne en pointillés reliant *ÉTABLISSEMENT PASSIF EN ATTENTE* à *ÉTABLIE* n'est plus liée à l'arrivée d'un accusé de réception. La transition peut se faire immédiatement. En fait, l'état *ÉTABLISSEMENT PASSIF EN ATTENTE* peut disparaître puisqu'il n'est plus visible quelle que soit la couche.
4. Si le client envoie un paquet à SERVER_PORT et que le serveur ne soit pas à l'écoute de ce numéro de port, le paquet ne sera jamais remis au serveur.
5. a) Il faut 32 768 impulsions pour que le compteur repasse à 0 et donc 3 276,8 s. Dans le pire des cas, s'il n'y a pas de génération de données, l'émetteur entre dans la zone interdite au bout de 3 276,8 –60 = 3 216,8 s.

   b) Si les données utilisent 240 numéros par minute, le nombre de séquences effectif est 4t, t étant exprimé en secondes. Le bord gauche de la région interdite est 10(t – 3 216,8). En posant l'égalité de ces deux formules, nous trouvons un point d'intersection à t = 5 361,3 s.
6. Observez le second paquet dupliqué de la figure 6-11(b). Lorsque ce paquet arrive, ce serait une catastrophe si des accusés de réception de *y* se promenaient encore dans le réseau.
7. Un compteur sur 10 bits permet de gérer 2^{10} numéros de séquence. Un tick d'horloge toutes les 125 ms équivaut à 8 ticks par seconde. $2^{10}/8 = 2^{10}/2^3 = 2^7 = 128$ secondes avant rebouclage d'horloge. $T = 64$ sec. Le début de la région interdite après un rebouclage peut être décrit par la ligne suivante :

   $y = 8(x - 128 + 64) = 8(x - 64)$

   Les numéros de séquence utilisés peuvent être décrits par la ligne $y = 4x$.

   $4x = 8(x - 64)$

   $4x = 8x - 512$

   $x = 128$ secondes.
8. La variation de la durée aller-retour (RTT) d'un lien est en général très faible ; on peut donc assez aisément choisir une valeur de timeout optimale avant de décider de retransmettre un message. D'un autre côté, le délai RTT peut fluctuer fortement sur Internet d'abord à cause de la congestion réseau. Cette valeur doit de ce fait être mise à jour en permanence au regard des conditions réseau réelles et des retransmissions imprévues sont très probables.

9. Si les temps *AW* ou *WA* sont très courts, les événements *AC(W)* et *WC(A)* sont peu probables. L'expéditeur peut retransmettre dans l'état *S1* ; l'ordre du destinataire est sans importance.

10. Sur le chemin *R1R2*, l'allocation pour le flux *E* pourrait être de 1/2 (le flux *A* devrait alors baisser à 1/2), de même sur *R2R5* mais sur *R5R6* l'allocation pour le flux *E* ne peut pas dépasser 1/3. Donc l'allocation pour le flux *E* sera de 1/3 sur R1R2, sur R2R5 et sur R5R6. Les autres allocations ne seront pas affectées.

11. *A*, *C* et *D* ne peuvent pas obtenir plus de 1/3 de la bande passante. Il reste donc 2/3 de la bande pour *B*.

12. La fenêtre dynamique est plus simple puisqu'il n'y a qu'un seul jeu de paramètres à gérer (les bords de la fenêtre). De plus, le problème d'une fenêtre augmentant puis diminuant, avec les segments arrivant dans le mauvais ordre, n'existe pas. En revanche, le principe du crédit est plus souple et permet une gestion dynamique de la mise en tampon, indépendante des accusés de réception.

13. Dans les mécanismes AIAD et MIMD, le comportement des utilisateurs oscille autour de la ligne d'efficacité et ne converge pas. Le mécanisme MIAD converge, lui, comme AIMD. On constate toutefois qu'aucun de ces mécanismes n'est suffisamment stable. La réduction additive en AIAD et MIAD est relativement douce alors que l'incrémentation multiplicative en MIMD et MIAD est plutôt brusque.

14. Oui. En utilisant les racines carrées, les grandes valeurs sont plus fortement réduites que les petites.

15. UDP sur l'hôte A va demander 800 kbit/s, ce qui ne laisse que 200 kbit/s pour TCP sur l'hôte B.

16. Non. Les paquets IP contiennent des adresses IP qui spécifient une machine de destination. À l'arrivée d'un tel paquet, comment le gestionnaire de réseau peut-il savoir à quel processus il doit le remettre ? Les paquets UDP contiennent un numéro de port de destination. Cette information est essentielle à la bonne remise du paquet.

17. Il est possible qu'un client n'obtienne pas le bon fichier. Supposons que le client *A* envoie une requête demandant le fichier *f1* puis tombe en panne. Un autre client *B* utilise alors le même protocole pour demander le fichier *f2*. Supposons que le client *B*, qui tourne sur la même machine que *A* (même adresse IP), relie son socket UDP au même numéro de port que celui qu'utilisait *A* avant de tomber en panne. De plus, supposons que la requête de *B* se perde. Lorsque la réponse du serveur (à *A*) arrive, c'est le client *B* qui la reçoit et qui suppose que c'est la réponse à sa propre demande.

18. Envoyer 1 000 bits sur une liaison à 1 Gbit/s demande 1 µs. La vitesse de la lumière dans une fibre optique étant de 200 km/ms, il faut 0,5 ms pour que la requête arrive et à nouveau 0,5 ms pour avoir la réponse. En tout, il aura fallu 1 ms pour transmettre les 1 000 bits. C'est équivalent à 1 Mbit/s soit une efficacité de 1 ‰.

19. À 1 Gbit/s, le temps de réponse dépend de la vitesse de la lumière. Au mieux, il est de 1 ms. À 1 Mbit/s, il faut environ 1 ms pour émettre les 1 024 bits, 0,5 ms pour que le dernier atteigne le serveur et encore 0,5 ms pour que la réponse atteigne le client. Le meilleur temps de réponse possible d'un RPC est donc de 2 ms. Conclusion : en augmentant le débit d'un facteur 1 000, on améliore le temps de réponse d'un facteur 2. Cela ne vaut probablement pas la peine, sauf si le coût de la liaison à 1 Gbit/s devient ridiculement bas.

20. On peut donner trois raisons. D'abord, les ID de processus dépendent des systèmes d'exploitation. Utiliser ces ID, ce serait donc se rendre dépendant des systèmes d'exploitation. Deuxième raison, un processus peut établir plusieurs canaux de communication. Un unique ID de processus comme identificateur de destination ne permettrait pas de faire la distinction entre ces canaux. Troisième raison : il est facile d'avoir des processus écoutant sur des ports réservés, il serait impossible d'avoir des ID de processus réservés.

21. Les flux audio et vidéo sont globalement insensibles aux pertes de paquets mais très sensibles aux fluctuations du délai de transmission. TCP garantit la livraison des données en forçant la retransmission des paquets perdus, mais cette retransmission génère de nouveaux délais. Avec ces flux, toute donnée arrivant trop tard n'a plus d'intérêt. De plus, TCP ne supporte pas le mode de multi-diffusion multicast. Si le flux audio ou vidéo n'est pas en temps réel et que le réseau perd beaucoup de paquets, il est préférable d'opter pour RTP sur TCP. Il est possible d'obtenir un débit suffisant en TCP sur un réseau avec beaucoup de pertes, mais les pertes non compensées en UDP vont entraîner une énorme chute de qualité audio ou vidéo.

22. Dans le cas de *N*1, comme le délai est uniforme et au maximum de 10 s, un buffer permettant de stocker un peu plus de 10 secondes de données à destination de *D* est suffisant. Cela suppose bien évidemment que la gigue soit faible ou inexistante. Dans le cas de *N*2, un buffer plus petit, permettant de stocker 2 à 3 secondes de données, peut suffire. Toutefois, quelques trames (celles qui présentent des délais importants) seront écartées.

23. Le segment par défaut fait 536 octets. TCP ajoute 20 octets ainsi qu'IP, ce qui conduit à une valeur de 576 octets.

24. Même si l'on suppose que tous les datagrammes arrivent intacts, il est possible que ces derniers arrivent dans le désordre. C'est pourquoi TCP peut avoir à remettre en ordre les différentes parties d'un message.

25. Tout dépend de la taille de paquet. En considérant la taille maximale en Ethernet de 1 500 octets, avec 40 octets d'en-tête IP/UDP/RTP, il reste 1 460 octets d'espace pour les échantillons. Chaque échantillon occupant 4 octet, on peut en placer 365 par paquet. Avec 44 100 échantillons par seconde et 365 par paquet, l'envoi d'une minute d'audio correspond à 44100 / 365 soit 120 paquets. Pour une taille de paquet inférieur, leur nombre sera évidemment plus grand.

26. Non. Une connexion n'est identifiée que par ses sockets. C'est pourquoi la connexion (1, *p*) – (2, *q*) est la seule possible entre ces deux ports.

27. Le bit *ACK* permet de savoir si l'on utilise le champ de 32 bits. S'il est absent, il faudrait toujours employer ce champ, le cas échéant en acquittant un octet déjà acquitté. Ce bit n'est donc pas essentiel pour le trafic de données normal mais il joue un rôle crucial dans l'établissement de la connexion. En effet, il est utilisé dans les deuxième et troisième messages du protocole en trois étapes.

28. TCP attribue un numéro de séquence à chaque octet des données utiles. Des segments plus grands réduisent le nombre d'octets consacrés aux en-têtes, mais pas le nombre d'octets à numéroter.

29. La première méthode consiste à démarrer avec un `listen`. Si l'on reçoit un *SYN*, le protocole passe dans l'état *SYN REÇU*. Seconde méthode : un processus essaie de réaliser une ouverture active et envoie un *SYN*. Si l'autre côté est également ouvert et que le *SYN* soit bien reçu, on passe également dans l'état *SYN REÇU*.

30. Même si l'utilisateur tape son texte à vitesse constante, les caractères vont apparaître en rafales. Il va taper plusieurs caractères sans que rien ne s'affiche puis l'affichage va rattraper la saisie, ce que certains trouvent désagréable.

31. Oui. L'option TCP NODELAY peut produire ce comportement. Une autre cause éventuelle est une congestion réseau qui augmente la gigue.

32. Les premières rafales contiennent 2 Ko, 4 Ko, 8 Ko et 16 Ko. La suivante fait 24 Ko et survient après 40 ms.

33. La transmission suivante fait la taille maximale d'un segment, puis de 2, de 4 et de 8. Après 4 succès, la taille de la fenêtre sera de 8 Ko.

34. Un démarrage lent atteint les 64 ko au tour 6. L'augmentation additive est alors utilisée deux fois, amenant la fenêtre à 66 lors du tour 8.

35. Les estimations successives sont 29,6, 29,84 et enfin 29,256.

36. On peut envoyer une fenêtre toutes les 20 ms. Cela conduit à 50 fenêtres par seconde et donc à un débit maximal de 3,3 Mo/s, soit 26,4 Mbit/s. L'efficacité de la liaison est donc de 26,4/1 000, soit environ 2,6 %.

37. L'objectif est d'envoyer 232 octets en 120 secondes, soit 35 791 394 octets de charge utile par seconde. Cela correspond à 23 860 trames de 1 500 octets par seconde. La surcharge TCP est de 20 octets, celle d'IP de 20 octets également. Celle d'Ethernet étant de 26 octets, il faut émettre 1 566 octets pour transporter 1 500 octets de charge utile. Si l'on a à envoyer 23 860 trames de 1 566 octets chaque seconde, il faut une liaison de 299 Mbit/s. Avec une liaison de débit supérieur, on prend le risque d'avoir, en même temps, deux segments TCP qui portent le même numéro.

38. IP est un protocole de niveau réseau tandis que TCP est un protocole de transport de bout en bout. Tous les changements de spécifications du protocole IP doivent être pris en compte par tous les routeurs de l'Internet. Quant à TCP, il fonctionne très bien pourvu que les deux extrémités disposent d'une même version du protocole. Il est ainsi possible que des versions différentes de TCP fonctionnent en même temps sur des hôtes différents. Ce qui n'est pas le cas du protocole IP.

39. Un émetteur ne peut envoyer plus de 255 TPDU, c'est-à-dire 255 × 128 × 8 bits en 30 secondes. Le débit de données plafonne donc à 8,704 kbit/s.

40. Il faut au moins 128 secondes pour utiliser 2^{32} numéros de séquence. $2^{32}/_{2^7} = 2^{25} = 32$ Mo/s ou 256 Mbit/s.

41. Le débit de sortie sans congestion est de 69,33 kbit/s ; sur l'ensemble de la connexion, le débit moyen est de 43,64 kbit/s ; le taux de perte moyen correspond au nombre de segments perdus divisé par le nombre de segments émis. Le tampon ralentissant (de sténose) est rempli lors des tours 3, 4 et 7. C'est lors du tour 4 qu'il contient le plus de paquets.

42. Calculons la moyenne : (270 000 × 0 + 730 000 × 1 ms)/1 000 000. Cela prend 730 µs en moyenne.

43. Il faut 4 × 10 instructions pour copier 8 octets. 40 instructions prennent 40 ns. Donc, chaque octet nécessite 5 ns de temps unité centrale. Le système est ainsi capable de traiter 200 Mo/s, soit 1 600 Mbit/s. Il peut gérer une liaison de 1 Gbit/s s'il n'y a pas d'autre goulet d'étranglement.

44. La taille de l'espace de séquence est de 2^{64} octets, ce qui correspond à environ 2×10^{19} octets. Un émetteur à 75 Tbit/s parcourt cet espace à la vitesse de $9{,}375 \times 10^{12}$ numéros de séquence par seconde. Il lui faut donc 2 millions de secondes pour reboucler. Puisqu'il y a 86 400 secondes par jour, cela représente 3 semaines, même à 75 Tbit/s. Bref, avec un champ de 64 bits, nous sommes tranquilles pour un moment.

45. a) Un paquet de 128 octets contenant 16 mots de 64 bits, le traitement des données requiert 160 instructions. En y ajoutant l'en-tête, il faut 210 instructions par paquet. Un milliard d'instructions/s équivaut à 1 000 000 000 / 210 = 4 761 904 paquets/s. Le débit utile s'élève à 609 523 712 octets/s.

    b) Un paquet de 1024 octet est traité avec 1280 + 50 = 1330 instructions, ce qui donne un débit maximal de 751 879 paquet/s. Le débit utile monte dans ce cas à 769 924 812 octets/s.

46. La vitesse de la lumière dans une fibre optique ou une paire de cuivre est d'environ 200 km/ms. Pour une liaison de 20 km, le délai est de 100 µs en aller simple et de 200 µs en aller-retour. Un paquet de 1 Ko contient 8 192 bits. Si le temps de l'émission de 8 192 bits et de l'acquittement est de 200 µs, le délai de transmission et le temps de propagation sont égaux. Si B est le temps d'un bit, on doit avoir $8\,192B = 2 \times 10^{-4}$ secondes. Le débit de données, $1/B$ est donc d'environ 40 Mbit/s.

47. Les réponses sont : 1) 18,75 Ko, 2) 125 Ko, 3) 562,5 Ko, 4) 1,937 Mo. Avoir une fenêtre de 16 bits, cela signifie qu'un émetteur ne peut envoyer plus de 64 Ko avant de devoir attendre un acquittement. On ne peut donc transmettre de façon continue avec TCP, en plein régime, si l'on utilise une technologie Ethernet, T3 ou STS-3.

48. Le délai d'aller-retour est d'environ 540 ms. Donc, avec un canal à 50 Mbit/s, le produit largeur de bande/délai est de 27 Mo, soit 3 375 000 octets. Avec des paquets de 1 500 octets, il faut 2 250 paquets pour remplir le conduit. La taille de la fenêtre doit ainsi être au moins de 2 250 paquets.

49. Certaines des limites des tests de débit du côté client concernent le logiciel et le matériel du client, le réseau sans fil, l'infrastructure de test (notamment les serveurs des tests), le chemin réseau de bout en bout entre client et serveur de test ainsi que la manière de concevoir le test.

50. Les champs suivants ne seraient plus alignés sur une frontière de mot, ce qui risquerait d'augmenter le temps de traitement des paquets.

51. Les questions 51 à 53 sont des travaux pratiques à faire éventuellement évaluer par un enseignant ou un expert.

Note à propos de la réponse à l'exercice 6.41

La dernière question n'a pas été répondue, car elle n'était pas disponible au moment de la mise en ligne de ces corrigés. Lorsque nous obtiendrons cette information de la part des auteurs, nous ajouterons un fichier en conséquence. Nous invitons les lecteurs qui aimeraient lire cette réponse à revenir de temps à autre sur la page du livre pour voir si ce complément est apparu sous forme d'un fichier texte.