Examen de Protocoles Internet

Juliusz Chroboczek

13 décembre 2021

La durée de l'examen est de 2 heures. Les documents sont autorisés, le matériel électronique est interdit. Le sujet consiste de 3 pages.

Question 1. Nous sommes en 2035, et il n'y a presque plus de problèmes dûs à la fusion. L'UFR d'Informatique de l'Université de Paris a créé un serveur REST à l'intention des étudiants. L'API a la structure suivante :

- une requête `GET` à l'URL `/formation/M2-IMPAIR/` retourne la liste des intitulés des cours de l'année `M2-IMPAIR`, un par ligne ;
- une requête `GET` à l'URL `/cours/protocoles-internet/description` retourne une description du cours ayant pour intitulé `protocoles-internet` ;
- une requête `GET` à l'URL `/cours/protocoles-internet/enseignant` retourne l'adresse mail de l'enseignant responsable du cours.

On suppose que chaque requête HTTP a une taille de 100 octets, et que chaque réponse a une taille de 200 octets plus la taille des données transférées. On suppose aussi que chaque description de cours a une taille de 100 octets, et que chaque adresse mail a une taille de 15 octets. Enfin, on supposera que la connexion TCP au serveur a déjà été établie (il n'y a pas besoin de prendre en compte la latence due à l'échange `SYN-SYNACK-ACK` initial).

1. Alice veut connaître le mail de l'enseignant du cours intitulé `protocoles-internet` (elle connaît déjà l'intitulé, elle n'a pas besoin de le chercher). Donnez une estimation du nombre d'octets qu'elle doit transférer, ainsi que le nombre de RTT nécessaires.
2. Bernard est employé par des spammeurs, et il veut connaître les mails de *tous* les enseignants qui participent à l'année `M2-IMPAIR`. Indiquez comment il peut faire.
3. On suppose qu'il y a 10 cours dans l'année `M2-IMPAIR`, tous enseignés par des enseignants différents. Indiquez la quantité totale de données que Bernard va tranférer ainsi que :
   a) le nombre total de RTT nécessaires s'il effectue toutes les requêtes séquentiellement ;
   b) le nombre total de RTT nécessaires s'il parallélise les transferts autant que possible.
4. En 2036, l'administration centrale décide de remplacer le serveur REST par un service *cloud* délocalisé qui sert un simple fichier JSON contenant les données de tous les cours de l'année (description et adresse de l'enseignant). Donnez une estimation du nombre d'octets transférés et du nombre de RTT nécessaires pour Alice et Bernard. (Détaillez.)
5. Concluez en un ou deux petits paragraphes.

Question 2. Le serveur de cours décrit dans la question ci-dessus utilise HTTPS, le traffic est donc protégé par TLS.

1. Indiquez les propriétés de sécurité que garantit TLS (tel qu'il est déployé dans HTTPS).
2. Certains enseignants ne désirent pas que leur adresse mail soit publiée sur l'Internet. Est-ce que les propriétés de TLS enumérées ci-dessus suffisent pour leur garantir qu'elle ne l'est pas? Comment faire pour corriger le problème?

Question 3. Comme on est en 2035, l'Université de Paris propose enfin des cours de Français Langue Étrangère (FLE). Comme ces cours sont gratuits [1] et de haute qualité [2], mais sous-financés [3], il n'y a pas suffisamment de places pour tous les volontaires, et seuls les premiers étudiants à s'inscrire pourront y participer. Pour savoir si les inscriptions sont ouvertes, il suffit de faire une requête `GET` à l'URL `/inscriptions/fle`. On suppose que la requête fait 100 octets et que la réponse fait 200 octets.

Chloé tient absolument à s'inscrire, elle a donc écrit un programme qui fait une requête à l'URL ci-dessus à intervalles réguliers et qui la prévient dès que les inscriptions sont ouvertes.

1. Indiquez la quantité de données transférées par heure si le programme de Chloé fait une requête toutes les 10 minutes;
2. Même question si le programme fait une requête toutes les minutes.

S'étant rendus compte du problème, les ingénieurs de l'Université ont ajouté une interface asynchrone : il y a une *WebSocket* à l'URL `/inscriptions/fle/ws` qui envoie un message non-sollicité de 12 octets à tous les clients connectés dès que les inscriptions ouvrent.

3. Si un client se connecte alors que les inscriptions sont déjà ouvertes, le serveur doit-il lui envoyer une notification? Justifiez votre réponse.
4. Chloé modifie son programme pour utiliser l'interface *WebSocket*. Quelle quantité de données en octets doit-elle transférer par heure? (On suppose que les inscriptions ne sont pas encore ouvertes, et on négligera le *handshake WebSocket* initial.)
5. Malheureusement, Chloé constate que son client se fait déconnecter au bout de 60 secondes. À quoi cela peut-il être dû? (Indication : il y a plusieurs raisons possibles.)

Fort heureusement, le protocole *WebSocket* prévoit un mécanisme de *keepalive* : il existe un message *ping* que le client peut envoyer à tout moment, et le serveur y répond immédiatement par un message *pong*.

6. Chloé modifie son programme pour envoyer un message *ping* toutes les 30 secondes. Si l'on suppose que les messages *ping* et *pong* font exactement 12 octets, quelle quantité de données le programme de Chloé transfère-t-il par heure?
7. Concluez en un ou deux petits paragraphes.

Question 4. Damien implémente un jeu en ligne. Le client communique avec le serveur à l'aide d'une seule connexion *WebSocket* qui sert :

- à communiquer la position des joueurs;

1. Parce que Service Public.
2. Parce que Service Public.
3. Parce que Service Public.

- à envoyer les messages du *chat* (forum pour les discussions amicales et respectueuses entre joueurs) qui peuvent notamment contenir des images.

Damien teste son programme sur un réseau où il y a très peu de pertes de paquets. Il constate qu'il fonctionne très bien lorsque personne n'utilise le *chat*, mais que la latence augmente pour tout le monde dès lors que quelqu'un envoie une image volumineuse.

1. On rappelle que le protocole *WebSocket* est basé sur TCP. Expliquez d'où vient le problème, et indiquez la propriété de TCP qui le cause.

Damien résoud le problème en utilisant deux connexions *WebSocket* parallèles, une pour la position et l'autre pour le *chat*. Il constate que le problème est résolu sur un réseau sans pertes, mais qu'en cas de perte de paquets la latence augmente, même si personne ne se sert du *chat*.

2. Expliquez d'où vient le problème en un petit paragraphe au plus. Quelles sont les deux propriétés de TCP qui interagissent pour le causer ?

Damien résoud le problème en remplaçant une des deux connexions *WebSocket* par un flot de paquets UDP.

3. En passant à UDP, Damien a perdu certaines propriétés que TCP lui fournissait « gratuitement ». Parmi les propriétés suivantes, quelles sont celles que Damien doit réimplémenter, et quelles sont celles qu'il peut ignorer ? Justifiez chacune de vos réponses en une courte phrase ou deux.

   a) segmentation ;
   b) ordre ;
   c) fiabilité ;
   d) contrôle de flot et contrôle de congestion.

4. À la différence de la connexion *WebSocket*, qui est protégée par un protocole cryptographique, le flot UDP est envoyé en clair et sans authentification. Donnez (a) un exemple de jeu où l'absence d'authentification et un problème et (b) un exemple de jeu où l'absence de confidentialité est un problème, et donnez dans chaque cas un exemple d'attaque vaguement réaliste.