Università di Corsica Pasquale Paoli | Faculté Sciences et Techniques

Examen	N° de la session : **1**	Enseignant : **Paul Pina-Gherardi**
Nom du diplôme : **Licence 3 SPI Informatique**		Année : **2023-2024**
Nom de l'épreuve : **Système Exploitation et Réseaux partie Réseaux**		Durée totale de l'épreuve : **1h30**
A distribuer avec le sujet de :		
Calculatrice (FX 92 ou équivalente) autorisée : **Oui**		
Documents autorisés : spécifications TCP/IP (voir annexe)		

Le sujet est assez long, donc le barème est indicatif et susceptible d'être modifié. Jusqu'à 1 point peut être ajouté ou retiré pour la présentation de la copie. Les annexes A, B et C doivent être complétés et rendues.

Exercice 1 (5 points) Questions de cours

1. Qu'est-ce que le temps UNIX ?
2. Qu'est-ce que TCP ? Donner quelques unes de ses spécificités et comparer avec UDP.
3. Proposer et expliquer un protocole simple de synchronisation d'horloge entre deux machines. Cela peut être un protocole déjà existant.

Exercice 2 (3 points) Numération

Rappel : Un nombre IEEE-754, que l'on notera x, est égal à :

$$x = s \times 2^{e-127} \times (1 + m) \tag{1}$$

où s est le signe du nombre (-1 ou 1), e est l'exposant, et m est la valeur de la mantisse.

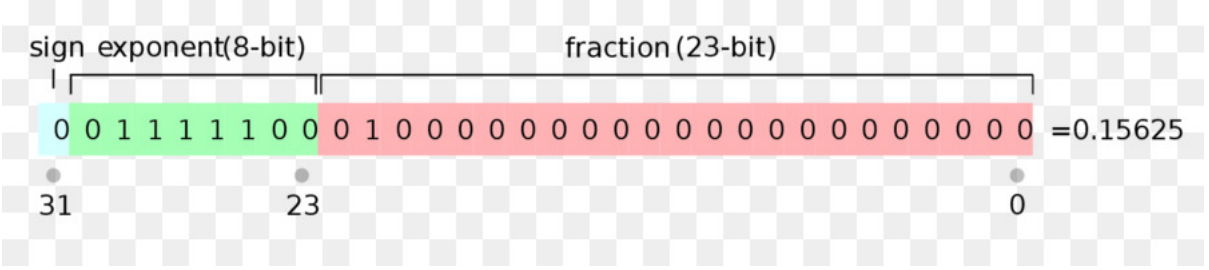

Figure 1: Schéma d'un nombre flottant à simple précision.

1. Vérifier que le nombre IEEE-754 de la figure 1 vaut bien 0,15625.
2. Convertir le nombre 174,125 en nombre flottant IEEE 754 simple précision *big endian*. Pour ce faire, on calculera d'abord ses valeurs de s, e et m, puis on écrira la série de bits.

Exercice 3 (12 points) Prédiction et réconciliation

Dans une application en ligne, où il y a un besoin de modéliser des entités et de les faire se déplacer, la difficulté augmente lorsqu'il faut partager ces informations avec les utilisateurs en temps réel, mais aussi que leur expérience soit satisfaisante, c'est-à-dire que les déplacements et animations présentés soient fluides.
On rappelle que dans une application client-serveur autoritaire, le serveur détient la vérité, et le client doit s'y plier.
Dans cet exercice, on considère que l'état du serveur est rafraîchi 10 fois par seconde. On peut alors signer par un nombre chaque rafraîchissement des états du monde : toutes les 100 ms, on incrémente la signature.
Un utilisateur contrôle une entité dans un environnement à deux dimensions x et y en envoyant des paquets à hauteur de 10 fois par seconde sur l'état de son clavier : les touches Z,Q,S,D donnent les ordres de déplacement respectivement en haut, à gauche, en bas et à droite. Un ordre de déplacement permet de se déplacer d'une unité en $100ms$: si à $t = 0ms$ en $(0; 0)$, j'envoie un ordre de me déplacer vers le haut, alors à $t = 100ms$, l'entité sera en $(0, 1)$
Pour transférer les données aux clients, le protocole de sérialisation binaire correspond à la transmission de ces données, toutes encodées en *big-endian*, dans l'ordre :

1. Le numéro de signature de l'état (entier 32 bits non-signé);
2. Le nombre d'utilisateurs (1 octet non-signé);
3. Pour chaque utilisateur :
   (a) son identifiant (1 octet non-signé);
   (b) sa coordonnée sur x (short int signé);
   (c) sa coordonnée sur y (short int signé).

1. **(2,5 points)** Interpréter le paquet reçu par le client ci-dessous :

```
00000000 00000000 00000000 00011010
00000011 10011101 00000100 11000000
10000000 00001111 01011001 10000000
00011110 10000000 00000010 11110110
00001000 00000000 00000000 00001000
```

2. **(1 points)** Expliquer pourquoi il est inutile d'identifier le destinataire dans le paquet de données, et indiquer l'endroit où l'identification est réalisée lors de la réalisation du paquet.
3. **(1 point)** Situation : le paquet 10 atteint le client avant le paquet 8. Que faire du paquet 8 dans cette situation?
4. **(1 point)** Avec une telle implémentation, faut-il préférer TCP ou UDP ? Réponse en **une** phrase courte.
5. **(1.5 points)** On suppose qu'il y a une latence de 80 ms entre le client et le serveur. Le client modifie l'état de son application uniquement lorsqu'il reçoit une information du serveur. Au temps $t = 100ms$, le client se situe en $(2, 4)$ et envoie "D" au serveur.

Indiquer l'état du client et du serveur à $200ms$ et à $300ms$, en sachant que le serveur a commencé à émettre à $t = 0ms$.

6. **(2 points)** La latence augmente alors à 150 ms. Le client envoie "Q" au serveur à $t = 300ms$ en étant à $(3, 4)$, puis à $t = 400ms$, il envoie "S". Compléter le schéma de la figure 3 en annexe A, et expliquer en quoi une telle architecture n'est pas envisageable pour une simulation en temps réel.

   Pour la suite, on ajoute la capacité au client de prédire son état au lieu d'attendre le serveur (voir fig. 2). Lorsque le client reçoit un paquet du serveur, il applique la logique autoritaire et déplace alors les entités aux positions désignées. Seulement, cela peut poser problème si la latence devient trop haute.

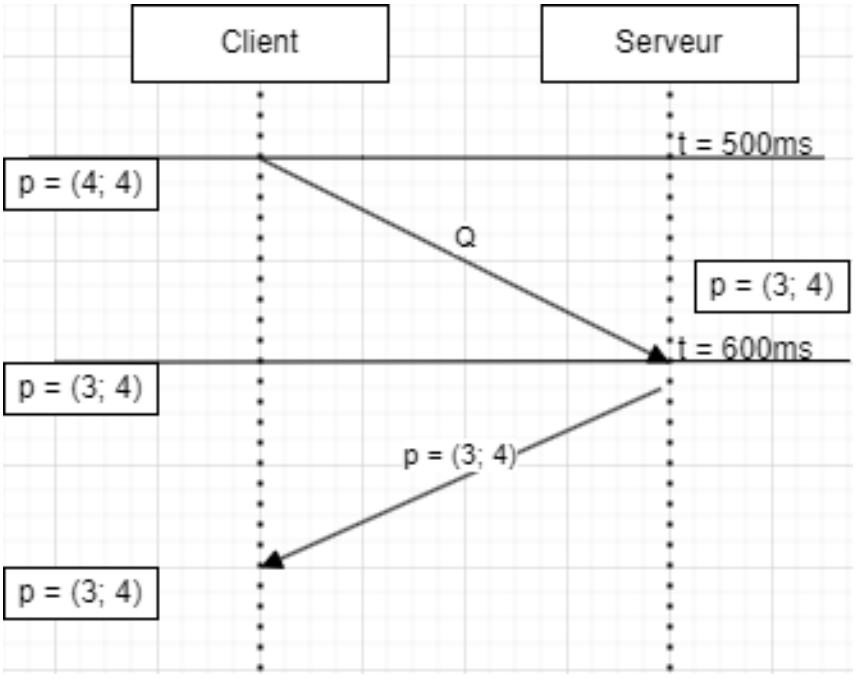

Figure 2: Schéma de la prédiction : le client voit le déplacement en direct, et lorsque le serveur envoie la position, le client est bien à la position désignée.

7. **(2 points)** La latence augmente à 125ms. Compléter l'annexe B, et expliquer la situation.
8. **(1 point)** On implémente le mécanisme de **réconciliation** suivant :
   1. Le client numérote ses paquets de commande à partir de 1;
   2. Le client prédit l'état de son entité lorsqu'il la déplace, et sauvegarde l'ensemble de ses paquets;
   3. Le client garde en mémoire le numéro du dernier paquet traité par le serveur;
   4. Lorsque le client reçoit un paquet du serveur, ce dernier spécifie le numéro du dernier paquet envoyé par le client à ce moment là. Il prend alors la position envoyée, et recalcule la position réelle à partir de ses paquets qu'il a sauvegardé.

   Le client envoie "Z" à 500 ms, puis "Q" à 600ms, et on considère une latence de 125ms. Compléter l'annexe C.

Exercice 4 (2 points bonus) Réflexions supplémentaires

1. On simule des voitures roulant à plus de 100 km/h et on transmet régulièrement leur position aux utilisateurs. Proposer un mécanisme d'**extrapolation** (en anglais *dead reckoning*, navigation à l'estime) qui permet au client d'essayer de deviner où seront les voitures à chaque étape suivante du monde sans recevoir de données. Proposer une réponse simple éventuellement accompagnée d'un schéma.
2. Dans la situation de l'exercice 3, d'un état à un autre, un utilisateur ne peut se déplacer que sur 4 directions, et d'une unité à chaque fois. Les coordonnées sont représentées sur 32 bits au total. On souhaite mettre en place un *delta encoding* : plutôt que de transmettre les positions, on transmet les déplacements réalisés. Quels sont les avantages et inconvénients d'une telle méthode ?

A Serveur autoritaire

A compléter pour l'exercice 3.

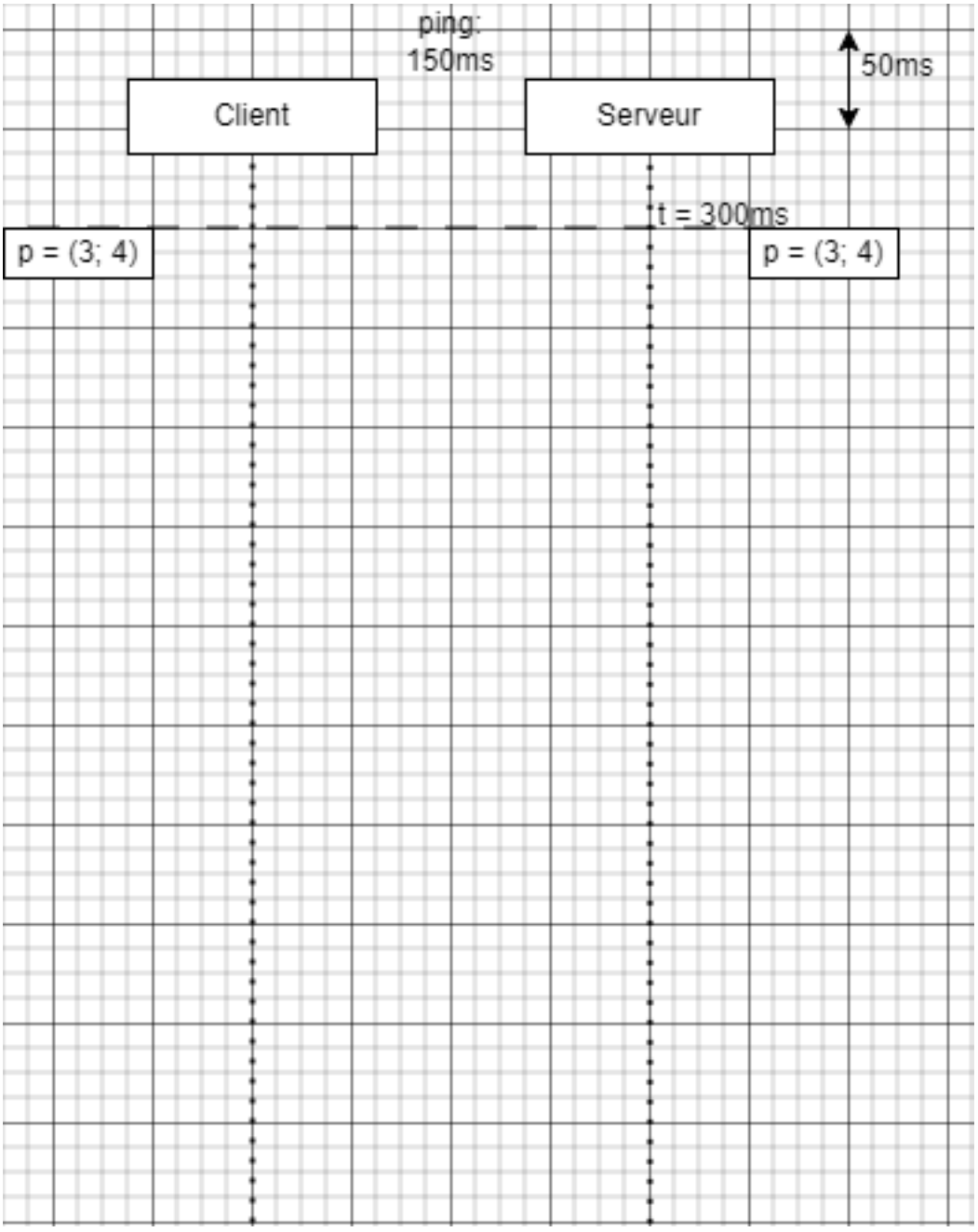

Figure 3: Déplacement sans prédiction du client, schéma à compléter. Un grand carreau pour 40ms.

B Prédiction simple

A compléter pour l'exercice 3. On ajoute l'indication pour le paquet envoyé par le serveur sur le temps auquel il a été émis. Rajouter des flèches supplémentaires si besoin.

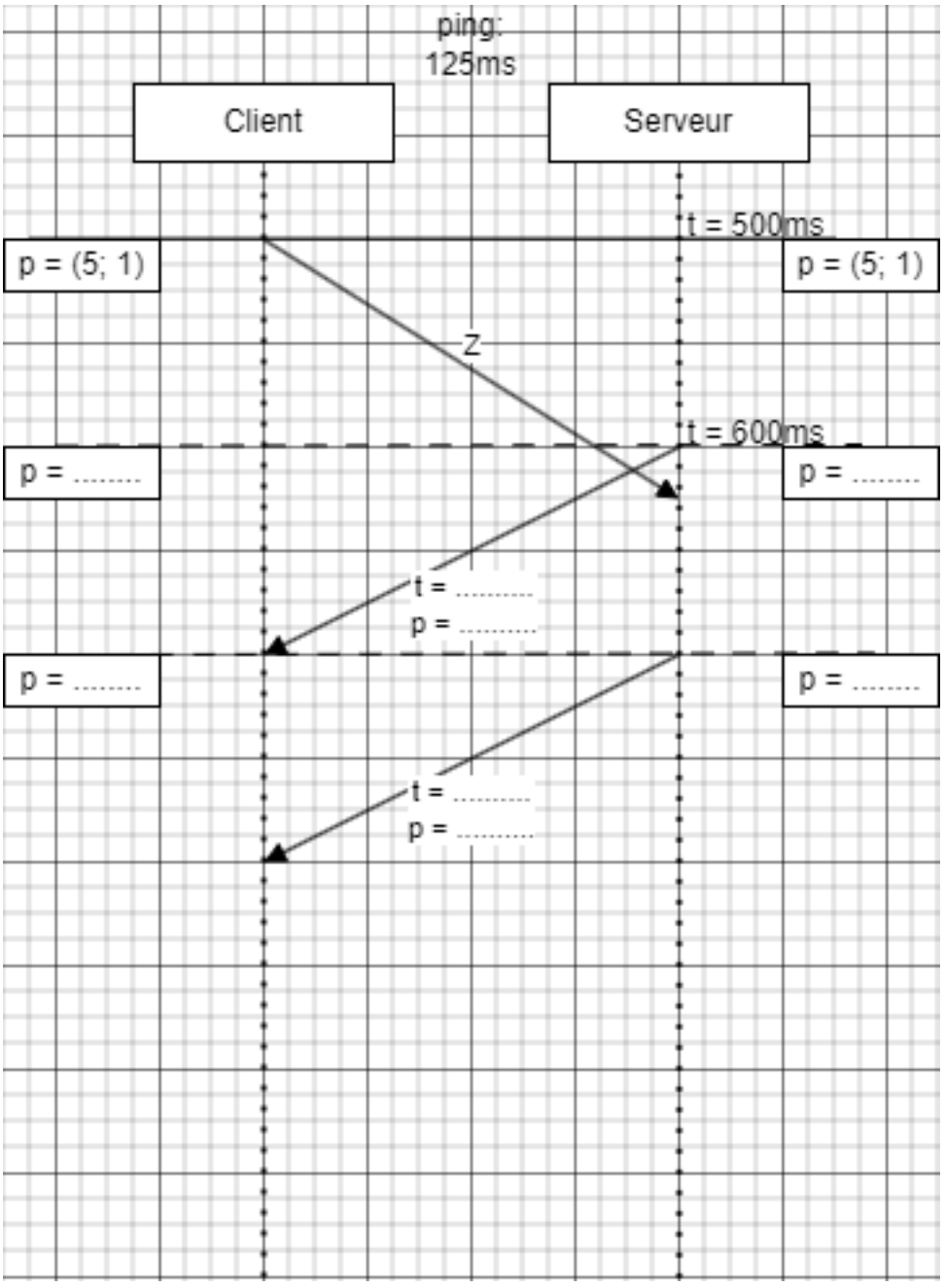

Figure 4: Déplacement avec une prédiction simple du client, schéma à compléter. Un grand carreau pour 50 ms.

C Prédiction avec réconciliation

A compléter pour l'exercice 3. Ajouter aux paquets envoyés par le client un numéro d'input, et aux paquets du serveur le dernier input validé. Rajouter des flèches supplémentaires si besoin.

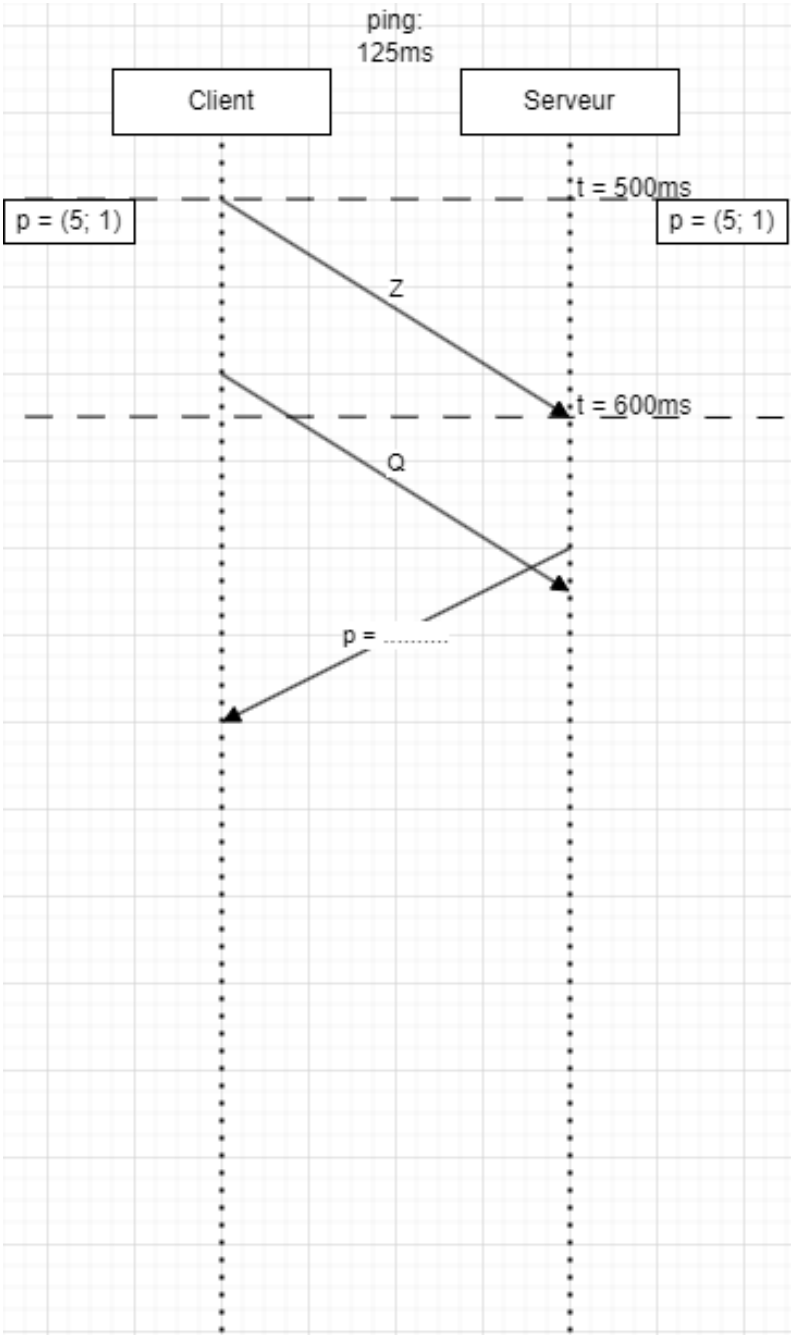

Figure 5: Déplacement avec prédiction et réconciliation, schéma à compléter. Un grand carreau pour 50 ms.

D En-têtes TCP et UDP

(Pas à rendre)

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	