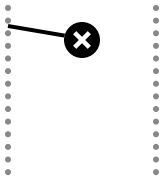
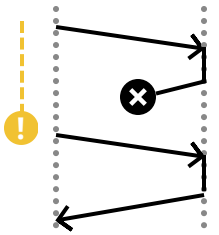
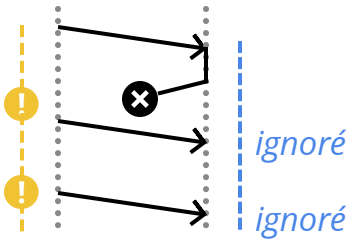
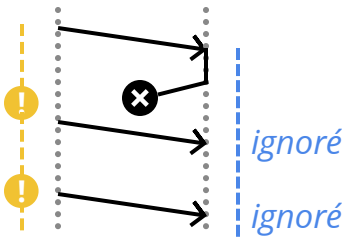
SDR

Communiquer avec des pannes – Corrigés

Olivier Lemer

Exercice

Solution 1

	Absence traitement	Duplication traitement	Absence confirmation	Duplication confirmation
R		Impossible : Le client n'envoie jamais en double.	Inapplicable : Le client ne reçoit jamais de confirmation.	Inapplicable : Le client ne reçoit jamais de confirmation.
RR	Impossible : Le client réessaie après timeout		Impossible : Le client réessaie après timeout	Possible, mais pas à cause d'une perte de message.
RR avec id	Impossible : Le client réessaie après timeout	Impossible : L'id assure l'absence de doublon coté serveur		Impossible : l'identificateur supprime les doublons
RRA	Impossible : Le client réessaie après timeout	Impossible : L'id assure l'absence de doublon coté serveur		Impossible : l'identificateur supprime les doublons

Exercice

Solution 1 *(continued)*

RR avec id et RRA ont un risque d'absence de confirmation en cas de perte de la réponse.

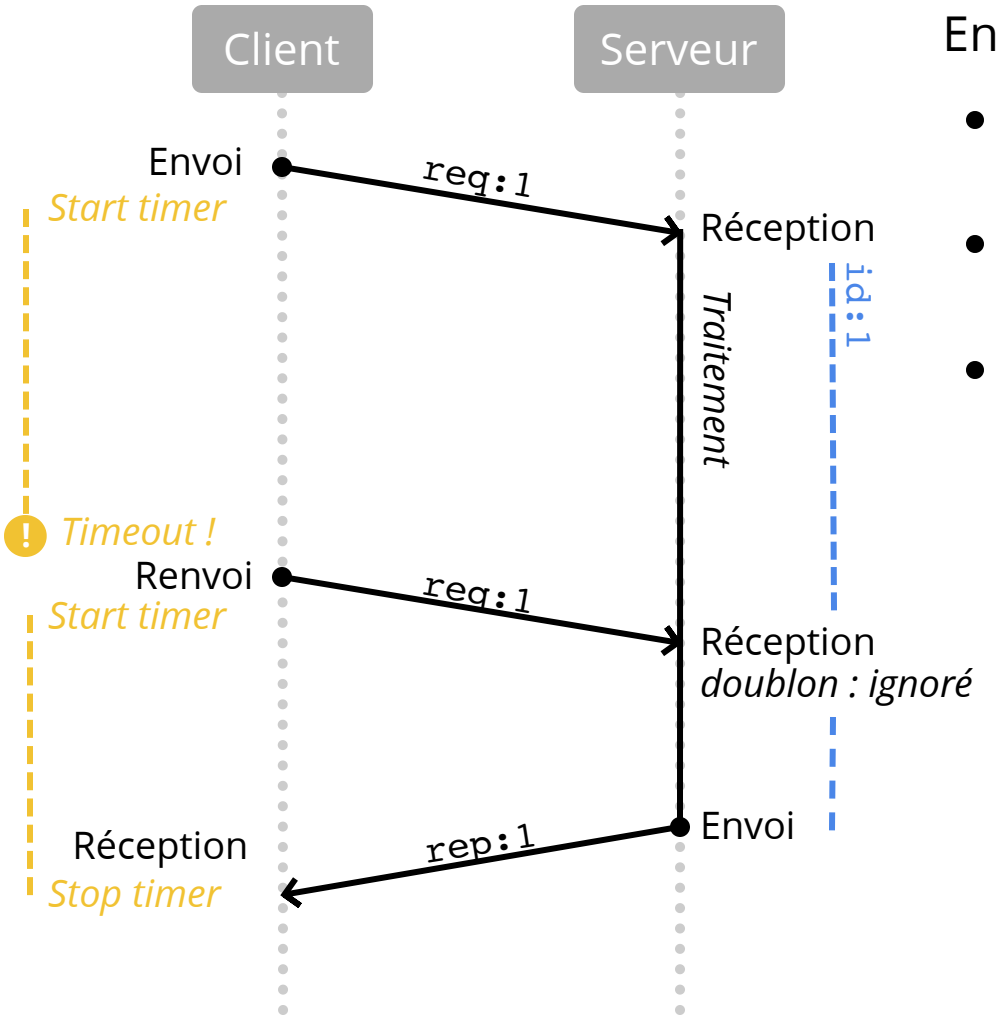
Le problème est que le serveur se reposait sur la garantie du réseau pour ignorer les doublons. Il doit maintenant renvoyer la réponse, même s'il pense être face à un doublon.

Afin d'éviter le cout de recalculer la réponse, celle-ci peut être stockée en cache avec l'id du message. Ceci sera couteux en mémoire pour RR avec id, mais résolu avec le système d'ack de RRA.

Noter aussi le problème de la perte du ACK dans RRA : l'id (et la réponse) ne seront jamais supprimés chez le serveur. La conséquence n'est cependant pas un échec fatal du serveur ou du protocole.

Exercice

Solution 2

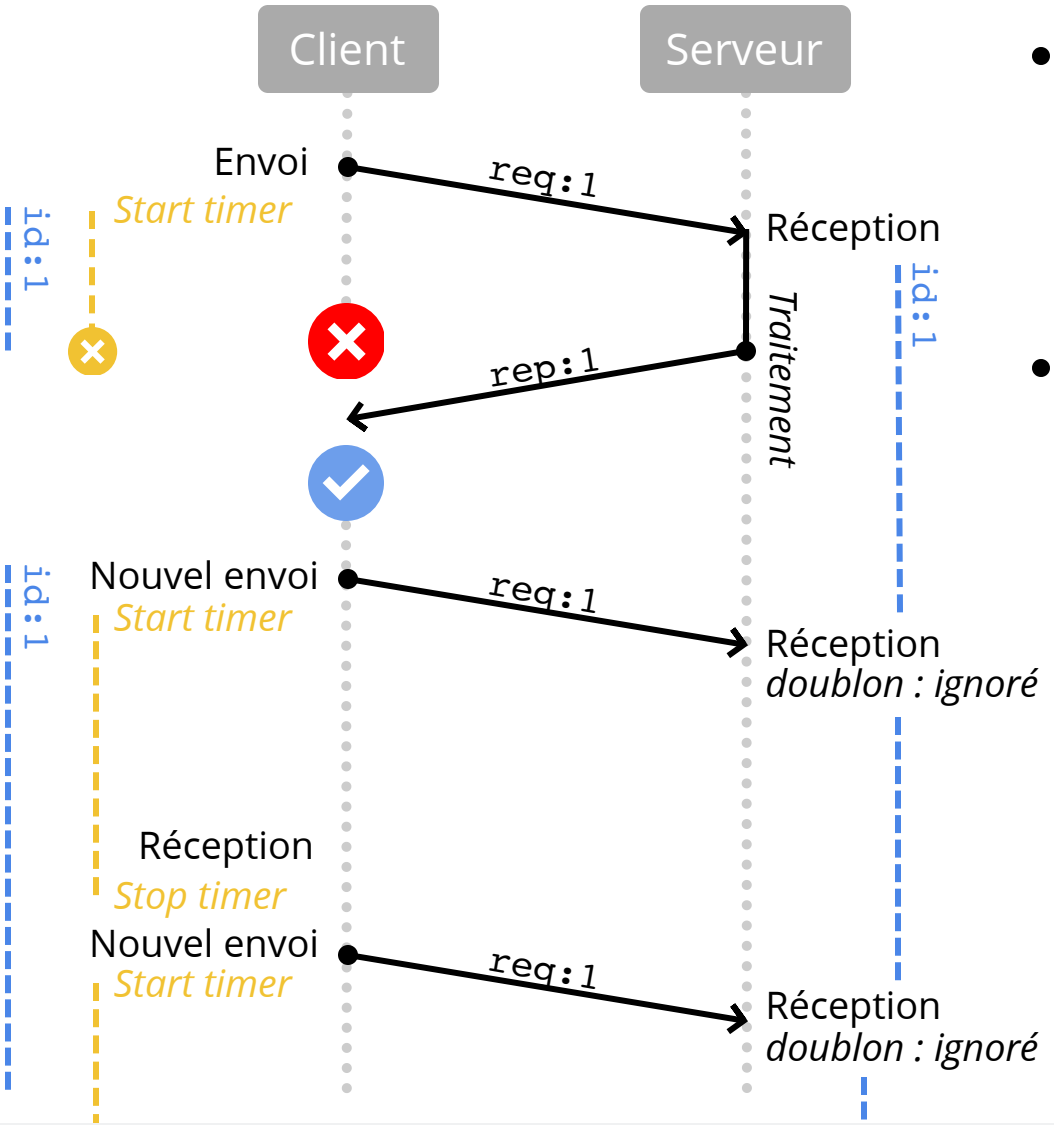


En cas de traitement long :

- Le client timeout et renvoie la requête.
- Le serveur utilise l'id pour détecter le doublon et l'ignore.
- La réponse du serveur est perçue comme réponse à la seconde requête du client et stoppe le timer.

Exercice

Solution 3

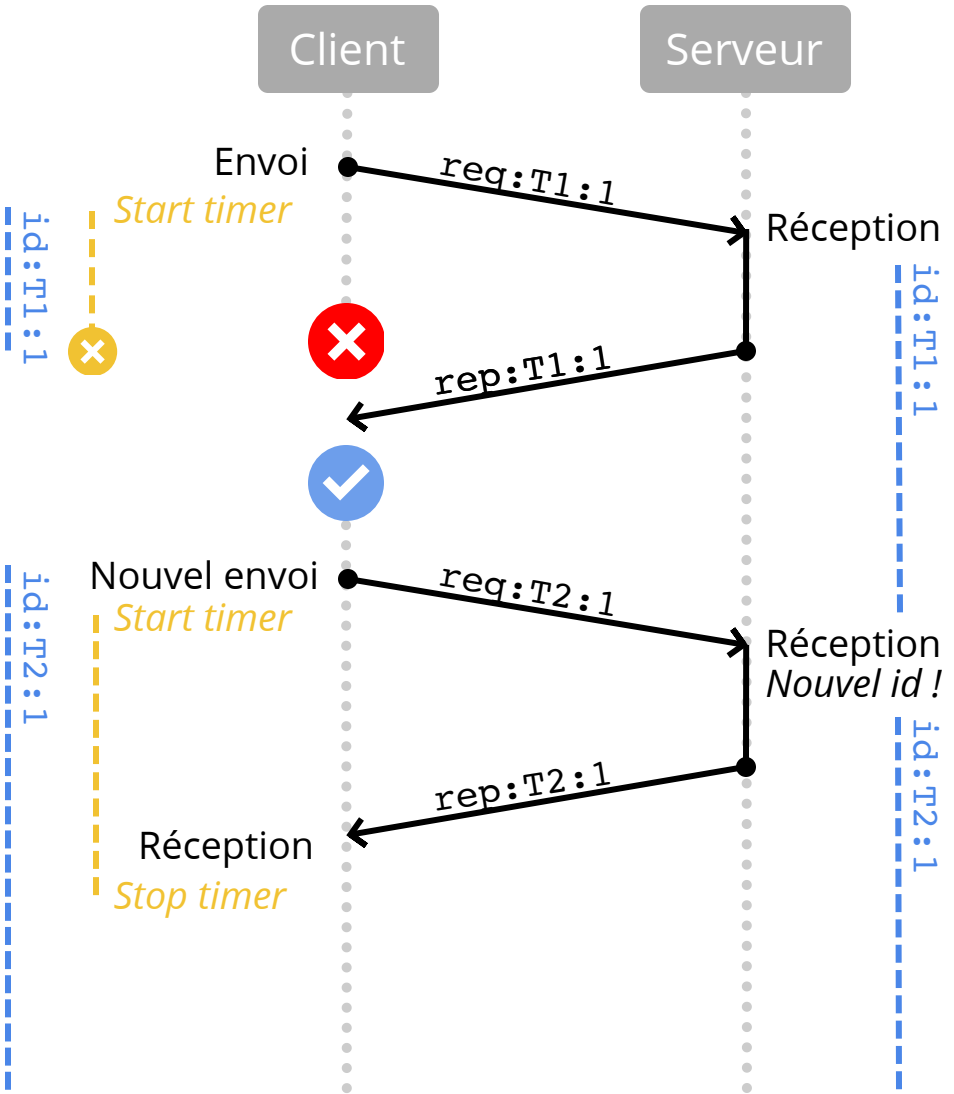


En cas de panne du client :

- Lorsqu'il revient, s'il veut envoyer un nouveau message (potentiellement indépendant du précédent), celui-ci sera ignoré par le serveur.
- Le client réessaiera infiniment, entrant dans une boucle infinie.

Exercice

Solution 3 *(continued)*



Solution

Le client doit envoyer un identifiant unique de l'éxécution en cours.

Ainsi, les ids seront différents lors de la seconde éxécution, et le serveur ne croira pas à un doublon.

Une manière de garantir cette unicité des identifiants peut être d'y accoller l'heure du début de son éxécution.

Exercice

Solution

Précision : perdue.

Le processus est signalé en panne alors qu'il répondra de nouveau. Le détecteur l'a même retiré de `surveillés`, donc il ne se corrigera jamais.

Complétude : conservée.

Elle ne parle que des pannes permanentes, et celles-là sont toujours détectées.

→ C'est exactement ce que le détecteur **parfait un jour** vient corriger, en annulant ses suspicions.