

15 décembre 2014

Indications

Les exercices et problème suivants sont indépendants les uns des autres. Vous pouvez les traiter dans l'ordre qui vous semble le plus efficace. Il est indispensable de présenter toutes les parties d'un même exercice consécutivement. **Toute réponse doit être justifiée.**

Exercice 1

Dans un réseau local, supposons qu'une communication TCP soit mise en place entre deux applications sur deux hôtes distincts.

La connexion est établie ; le routage est direct (sans intervention de routeur) ; les échanges sont bidirectionnels.

1. Montrer qu'il peut y avoir perte de paquets dans la couche TCP (i.e. tout se passe correctement avant l'arrivée à TCP).

Réponse (1 point) : tampon TCP plein côté réception ; dans ce cas, il y aura réémission des paquets.

On suppose maintenant que le routage met en œuvre un routeur entre les deux hôtes.

2. Donner deux raisons de pertes de paquets autres que la précédente ?

Réponse (2 points) :

- une erreur de routage suite à une modification (mise à jour) du routage ; en effet, si la communication a démarré, il n'y avait pas d'erreur, au moins pendant un moment ;
- tampon du routeur plein, soit côté réception, soit côté émission, soit les deux... ;
- on est obligés d'accepter, mais en dernier, une panne très temporaire d'un des deux réseaux, pile lorsque les paquets circulent (moitié des points seulement si c'est la seule réponse).

3. Que se passe-t-il alors dans la couche TCP des deux hôtes ?

Réponse (.5 point) il y a tentatives de réémission et éventuellement abandon si plusieurs échecs.

4. Est-il possible que ces événements provoquent la coupure de la communication entre les deux applications ?

Réponse (.5 point) oui, on vient de le voir ci-dessus, ou alors panne sans rétablissement.

Exercice 2

On veut empêcher un même client de faire plusieurs demandes de connexion successives à un même serveur. Par exemple, éviter qu'un moteur de recherche surcharge un serveur par des requêtes en rafale, empêchant toute autre demande de connexion à ce serveur.

1. Quelles sont les informations permettant d'identifier un client ? Analyser dans votre réponse toutes les configurations d'un serveur, quel que soit son type, concurrent ou itératif, utilisant UDP ou TCP.

Réponse : le total de l'exercice fait 4 au lieu de 3 dans le barème.

Réponse (1 point) : Que le serveur soit itératif ou concurrent, qu'il soit en Udp ou Tcp, il reçoit l'adresse de la BR du client (le triplet (num. hôte, num. port, proto)). La seule chose qui change est qu'en mode non connecté c'est sur un `recvfrom()` qu'on le reçoit alors qu'en mode connecté c'est sur un `accept()`.

2. Proposer une solution permettant au serveur de refuser plus de n demandes de connexion par minute. On pourra admettre l'existence d'une fonction de mesure du temps (choisissez une mesure qui vous convient, admettez son existence). Décrire ce que fait le serveur et ce qui se passe chez le client.

Réponse (2 points) : **Principe** : le serveur enregistre dans un tableau chaque adresse de client, l'heure de la demande s'il s'est passé plus d'une minute depuis la demande précédente du même. Lors de toute nouvelle demande, il suffit de regarder si le demandeur existe dans le tableau. Si oui, comparer l'heure courante à celle enregistrée. S'il s'est passé plus d'une minute, enregistrer la nouvelle heure et réinitialiser le nombre de demandes à 1, sinon comparer avec le nb. max accepté.

Algo facile à écrire. Noter qu'un client peut avoir plus de n connexions actives, si elles sont distantes de plus d'une minute.

Le client reçoit un retour 0 (zéro) sur la réception dans le cas tcp, lorsque n est dépassé. Dans le cas udp, tout dépend du type de lecture. Si bloquante, il ne recevra rien. Sinon, il recevra -1, **pour cause de non réception**, sans savoir que le serveur refuse.

3. Est-ce que votre solution empêche le client de saturer la file d'attente dans le cas d'un serveur en mode connecté ? dans le cas d'un serveur en mode sans connexion ?

Aucun moyen d'empêcher à ce niveau (transport ou application) un client de saturer la file d'attente. On ne peut que constater qu'il sature. Hors-barème : pour bloquer une telle saturation, il faudra agir au niveau du routage (pare-feu).

Exercice 3

On veut partager une adresse de réseau de type «classe C» en plusieurs sous-réseaux avec les caractéristiques suivantes :

- deux sous-réseaux (SR_1, SR_2) de capacité maximale 30 hôtes,
- trois sous-réseaux (SR_3, SR_4, SR_5) de capacité maximale 62 hôtes.
- l'ensemble de ces sous-réseaux seront reliés par un routeur qui réalise aussi la communication avec le monde extérieur.

Prendre comme exemple l'adresse de réseau 200.201.202.

1. Comment faut-il réaliser l'adressage ? Décrire **toutes les caractéristiques** des sous-réseaux.

Réponse (2 points) **Principe** faire comme si on avait quatre sous-réseaux de 62 hôtes (2 bits d'adresse de sous-réseau) et diviser l'un des quatre à nouveau en deux, qui auront donc 3 bits d'adresse de sous-réseau. Dans le tableau ci-dessous on commence avec ceux de 62 hôtes. Les bits des adresses de sous-réseaux sont : 01, 10, 11 pour ceux de 62 hôtes puis 000, 001 pour les 30 hôtes. On peut bien sûr choisir d'autres combinaisons, à condition de respecter les longueurs de ces adresses, ainsi que le partage de la même base (ici 00) pour les petits réseaux.

Res.	adr. réseau	adr. tous	masque	hôtes
SR_3	200.201.202.64	200.201.202.127	255.255.255.192	200.201.202.65 à 200.201.202.126
SR_4	200.201.202.128	200.201.202.191	255.255.255.192	200.201.202.129 à 200.201.202.190
SR_5	200.201.202.192	200.201.202.255	255.255.255.192	200.201.202.193 à 200.201.202.254
SR_1	200.201.202.0	200.201.202.31	255.255.255.224	200.201.202.1 à 200.201.202.30
SR_2	200.201.202.32	200.201.202.63	255.255.255.224	200.201.202.33 à 200.201.202.62

2. Donner la table de routage du routeur.

Réponse (1 point) On désigne par $iface_{srX}$ l'interface (carte réseau) du routeur pour le réseau X. Dans la partie *contact*, on peut accepter comme réponse l'adresse du routeur dans **chacun** des sous-réseaux, mais c'est inutile en toute rigueur. On peut accepter qu'il n'y ait pas de réponse masque pour la ligne *défaut*.

destination	contact	masque	interface
200.201.202.0	direct	255.255.255.224	$iface_{sr1}$
200.201.202.32	direct	255.255.255.224	$iface_{sr2}$
200.201.202.64	direct	255.255.255.192	$iface_{sr3}$
200.201.202.128	direct	255.255.255.192	$iface_{sr4}$
200.201.202.192	direct	255.255.255.192	$iface_{sr5}$
défaut	routeur suivant	0.0.0.0	$iface_{ext}$

3. Une erreur de masque est faite sur SR_2 . Elle provoque l'impossibilité de joindre les hôtes de SR_1 , alors que tous les autres hôtes restent joignables. Quelle erreur a été faite ?

Réponse (1 point) : si le masque est 255.255.255.192, alors on a l'erreur indiquée, car tout se passe comme si les hôtes de SR_1 étaient sur le même sous-réseau, ici SR_2 .