

16 juin 2016

## 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.**

Barème indicatif : Exercice 1 : 6, Exercice 2 : 5 Exercice 3 : 4, Exercice 4 : 5.

### Exercice 1

1. On considère l'adresse 192.20.16.133/29. Combien de bits sont utilisés pour identifier la partie réseau ? Combien de bits sont utilisés pour identifier la partie hôte ? Justifiez vos réponses en convertissant en binaire.
2. On considère un réseau d'adresse 120.102.0.0/16. On veut découper ce réseau en 6 sous-réseaux de même taille :
  - (a) Combien faut-il de bits supplémentaires pour définir 6 sous-réseaux ?
  - (b) Après avoir expliqué le principe d'un masque de réseau, donnez le masque réseau nécessaire pour créer les 6 sous-réseaux ?
  - (c) Combien faut-il de bits supplémentaires pour définir 6 sous-réseaux ?
  - (d) Donnez l'adresse réseau de chacun de ces 6 sous-réseaux en expliquant comment vous les obtenez ?
3. On considère un réseau d'adresse 172.15.16.40/21.
  - (a) Combien de bits sont nécessaires sur la partie hôte de l'adresse attribuée pour avoir au moins 20 machines ?
  - (b) Quel est le nombre maximum d'adresses utilisables dans chaque sous-réseau ?
  - (c) Quel est le nombre maximum de sous-réseaux définis ? Donnez les adresses de tous les sous-réseaux définis.

### Exercice 2

Dans cet exercice, vous allez étudier le comportement d'une application de transfert de documents. Par exemple, cela pourrait être *W3* avec le protocole de transfert de documents hypertexte, *Http*. On considère dans un premier temps que le document transféré est représenté par une suite de caractères.

Pour recevoir un document, une application de type *Client* demande l'établissement d'un circuit virtuel avec un *Serveur* par le biais du protocole *TCP*. Une fois la connexion établie, le client envoie la référence du document (selon le protocole d'application convenu) et le serveur lui renvoie ce document. On suppose enfin que le serveur ferme la connexion à la fin de l'expédition. Le client se contente de lire la suite de caractères jusqu'à la réception de l'information de fermeture.

1. Est-ce que le serveur peut expédier le document en une seule instruction d'expédition ? est-ce que le client peut recevoir tout le document en une seule lecture ? est-ce que le client peut déduire le nombre de paquets *tcp* qui ont été acheminés ?
2. Rappeler très rapidement comment est reçue l'information de fermeture chez le client, à savoir, sous l'aspect programmation, comment est-ce que le client récupère cette information ; ajouter une explication brève.
3. Montrer que malgré la fiabilité de *tcp* le client n'est pas sûr d'avoir reçu la totalité du document.
4. Proposer une solution pour qu'il puisse le savoir.



5. Ajouter une solution permettant de récupérer par la suite la partie manquante seule.
6. On suppose maintenant qu'un document est composé de plusieurs parties différentes. Chaque partie peut être vue comme un document en soi (une image par exemple). Proposer une solution permettant une reprise «au mieux» en cas de réception partielle.

### Exercice 3

Sur un hôte *A* une application lancée régulièrement utilise un fichier en mode lecture/écriture. Les données (enregistrements du fichier) sont des entiers et des structures plus complexes, composées de caractères et d'entiers (lire à ce sujet la remarque finale). Ce fichier est stocké sur un hôte *B*, qui sert comme serveur d'espace disque.

Lorsque l'application est lancée sur *A*, elle commence par le transfert du fichier de *B* sur *A*. Juste avant la fin de l'application, après la fermeture du fichier il est retransféré avec ses modifications, sur *B*. Ainsi, tout le travail sur le fichier se fait sur *A*, sans que le réseau soit impliqué, hormis les deux transferts au début et à la fin.

Les deux hôtes *A* et *B* ont des architectures différentes, en particulier, des codages internes différents des entiers.

1. Lors du transfert du fichier (de *B* vers *A* ou de *A* vers *B*) est-il nécessaire de transcoder tout le fichier de la forme locale à la forme réseau au départ, puis de la forme réseau à la forme locale à l'arrivée ?
2. Après quelques temps de service, l'hôte *A* devient obsolète. On décide de le remplacer par une nouvelle machine *C*. Cette nouvelle machine, utilise un codage différent de celui de *A*. Les applications vont migrer progressivement de *A* vers *C*. L'application concernée fonctionnera dans les mêmes conditions : le fichier sera toujours sauvegardé sur *B*. Quelles transformations faut-il effectuer sur les données afin que l'application continue à marcher sur *C* ? Proposer deux solutions différentes (voir les principes ci-dessous) afin que l'application sur *C* puisse utiliser le fichier sans ennui.
  - (a) une dans laquelle on envisagera une période durant laquelle les machines *A* et *C* cohabitent sur le réseau (*B* étant là de toute façon),
  - (b) une autre dans laquelle il n'y a pas de cohabitation.

**Remarque :** Vous pouvez traiter ce problème tout d'abord en supposant que les données sont uniquement des entiers, puis l'étendre au cas de données structurées plus complexes.

### Exercice 4

On considère une application sur un hôte qui lance une recherche DNS avec un nom inexistant.

1. Rappelez la définition d'une recherche DNS et son fonctionnement.
2. Décrire ce qui se passe en envisageant plusieurs erreurs possibles lorsque le domaine de premier niveau est inexistant (par exemple, l'adresse se termine avec `.truc`) ;
3. Décrire ce qui se passe en envisageant plusieurs erreurs possibles lorsque le domaine de premier niveau existe, le premier sous-domaine aussi, et l'erreur est ensuite dans le sous-domaine en 2<sup>ème</sup> position ;
4. Décrire ce qui se passe en envisageant plusieurs erreurs possibles lorsque seul le nom d'hôte contient une erreur ;
5. Profiter de cette analyse pour généraliser.