Examen de HLIN303

durée : deux heures

Feuille A4 recto-verso manuscrite autorisée

8 janvier 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 : 6 Exercice 3 : 8.

29 bits reseau
3 bits hote

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 120.162.0.0/16. On veut découper ce réseau en 8 sous-réseaux de même taille :
   (a) Combien faut il de bits supplémentaires pour définir 8 sous-réseaux ?
   (b) Après avoir expliqué le principe d'un masque de réseau, donnez le masque réseau nécessaire pour créer les 8 sous-réseaux ? Donnez l'adresse réseau de ces 8 sous-réseaux en expliquant comment vous les obtenez ?
3. On considère un réseau d'adresse 201.15.16.0/24. On veut définir un masque réseau qui permet de placer 20 machines/hôtes dans chaque sous-réseau.
   (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-réseaux ?
   (c) Quel est le nombre maximum de sous-réseaux définis ? Donnez les adresse de tous les sous-réseaux définis.

Exercice 2

Dans cet exercice, on s'intéresse à l'évolution d'un message partant d'une application sur un hôte H_1 jusqu'à son arrivée sur l'hôte destinataire H_2. La notion de message s'applique à l'entité échangée vue par les applications, par exemple un fichier s'il s'agit d'applications de transfert de fichiers. Le protocole utilisé est `TCP` et les ports attribués par les systèmes respectifs sont 2048 sur H_1 et 4096 sur H_2. H_1 et H_2 sont sur le même réseau local et ont une seule connexion au réseau chacune. Leurs numéros IP respectifs sont `192.3.6.8` (Ip_1) et `192.3.6.16` (ip_2). Le réseau local est un réseau *ethernet* et les adresses physiques sont `8:4:CF:20:36:AB` (eth_1) et `8:20:FE:10:20:48` (eth_2).

1. Après avoir rappelé le nombre et nommé les couches dans le modèle OSI, décrire et expliquer le cheminement d'un message de l'hôte H_1 à l'hôte H_2 à travers chacune des couches. Vous schématiserez de plus le cheminement.
2. On veut maintenant étudier les interfaces entre couches ainsi que les découpages et réassemblages. Supposons que la taille du *message* soit supérieure à la taille maximale d'un paquet `IP` : un **gros** fichier, une image, ...Vu de l'application, il n'y a qu'un et un seul message à transférer, et de fait, une écriture suffit.
   Détailler les transformations subies par le message au fur et à mesure de son évolution, jusqu' à son arrivée à destination. Préciser les interfaces entre couches. Détailler le rôle de chaque couche si cela n'a pas été fait précédemment.

3. On suppose maintenant que H_1 et H_2 sont sur deux réseaux distincts. Prendre pour adresse IP de M_2 `195.16.32.64` et ignorer la référence précédente.
   Décrire l'évolution du paquet lorsqu'un seul routeur relie les deux réseaux.
4. Si plus d'un routeur intervient, expliquer ce qui se passe dans chaque routeur.

Exercice 3

Dans cet exercice, on étudie la notion de fragmentation dans IPV4 et les raisons de son abandon dans IPV6. Dans IPV4 un routeur peut être amené à fragmenter un datagramme, c'est-à-dire redécouper un datagramme entrant d de longueur l en plusieurs datagrammes $d_1, d_2, \ldots, d_n$ appelés *fragments*. La raison de ce redécoupage provient du fait que les liaisons associées à la suite du routage du datagramme ne supportent pas la longueur l. Les fragments d_i vont ensuite circuler individuellement, conformément au protocole IP. Bien évidemment, un fragment peut lui-même être redécoupé.

Chaque fragment comporte la même entête que le datagramme initial, sauf pour les champs suivants :
— longueur de ce fragment qui remplace la longueur du datagramme initial,
— la position de ce fragment (premier octet de ce fragment) dans le datagramme initial est indiquée dans l'entête,
— un autre champ dans l'entête (*flags*) permet d'identifier le dernier fragment du datagramme initial.

Noter que le datagramme initial est identifié (champ *identification*) et que cette identification est donc reportée sur tous les fragments. Noter aussi que les fragments ne sont pas numérotés, mais repérés par le couple $(position, longueur)$.

1. Où doit être fait le réassemblage des fragments (quel hôte, quelle couche) ? Penser à justifier. Montrer que l'on peut reconstituer complètement le datagramme initial lorsque tous les fragments sont présents, quel que soit l'ordre d'arrivée (ne pas dépasser une dizaine de lignes).
2. Si un fragment est perdu, que proposez-vous comme solution (rejet partiel, complet, tentative de récupération, autre) ?
3. Supposons que le protocole de transport utilisé soit udp. Si les fragments déjà arrivés à destination sont dans l'ordre, $d_1, d_2, \ldots, d_i$, sans que le dernier fragment ne soit arrivé, peut-on délivrer ces fragments sans attendre la suite ?
4. Même question si le protocole est `tcp`.
5. La somme de contrôle du datagramme est calculée dans IPV4 sur l'entête du datagramme. Il est donc nécessaire de le calculer séparément pour chacun des fragments. Il n'empêche que chaque routeur doit de toute façon recalculer la somme de contrôle pour chaque datagramme, qu'il soit fragmenté ou non. Pourquoi ?

IPV6 abandonne cette notion de fragmentation. Or il doit fonctionner sur les mêmes réseaux physiques qu'IPV4. Donc si un datagramme IPV6 est de taille supérieure au maximum admissible, il va être rejeté.

6. Acceptons que ce soit globalement une bonne solution ; étudier l'incidence de cette décision (rejet des datagrammes trop longs) sur les protocoles de transport udp et `tcp` (quelques lignes pour chaque cas).
7. Déduire qu'il peut y avoir famine (destinations impossibles à atteindre à partir d'un routeur) et proposer une solution permettant d'éviter autant que possible cette situation.