

Examen 1

H2016

GIF-3001 Réseau de transmission de données

Question 1 (10 points sur 100)

(10 points)

- (a) Dessinez la pile des protocoles Internet. Indiquez **un rôle** important pour chacune de ces couches, ainsi qu'un **exemple de protocole** pour chacune des couches, s'il y a lieu.

5	Application
4	Transport
3	Réseau
2	Liaison
1	Physique

Solution :

- Application : Définit le format des messages entre les applications, et les procédures.
- Transport : Offre un service de multiplexage et démultiplexage. Peut offrir des services de transfert fiable, contrôle de congestion, etc.
- Réseau : Aiguillage des paquets, routage.
- Liaison : Mise en trame des informations
- Physique : Transport des bits sur un lien. Modulation, etc.

Question 2 (15 points sur 100)

Indiquez si chaque énoncé est **VRAI** ou **FAUX** :

(1 point)

- (a) Une connexion TCP est bidirectionnelle (*full-duplex*).

Solution : V

(1 point)

- (b) HTTP est un protocole applicatif sans état.

Solution : V

(1 point)

- (c) Un socket est toujours associé à un seul processus.

Solution : V

(1 point)

- (d) Un processus ne peut avoir qu'un seul socket.

Solution : F

(1 point)

- (e) Dans TCP, le numéro de séquence représente le numéro du segment dans le flux.

Solution : F

- (1 point) (f) Si un nœud A envoie un segment TCP avec un numéro de séquence de 38 et 4 octets de données, ce même segment contiendra alors nécessairement un numéro d'accusé de réception (ACK) de valeur de 42.

Solution : F

- (1 point) (g) Un paquet TCP valide a toujours le drapeau ACK actif;

Solution : F

- (1 point) (h) Un paquet TCP valide ne peut pas avoir les drapeaux SYN et FIN actifs;

Solution : V

- (1 point) (i) Un paquet TCP valide ne peut pas avoir les drapeaux SYN et ACK actifs;

Solution : F

- (1 point) (j) La taille de fenêtre (*Receive Window*) est échangée au début d'une connexion TCP, et varie pendant la connexion selon la mémoire disponible au récepteur;

Solution : V

- (1 point) (k) La taille maximum d'un segment TCP (*MSS*) est échangée au début d'une connexion TCP, et reste fixe pendant la connexion;

Solution : V

- (1 point) (l) Les spécifications des protocoles SMTP, DNS, HTTP et TCP sont décrites dans des documents RFC de l'IETF

Solution : V

- (1 point) (m) Si l'on capture le segment d'une session TCP à deux points différents, il est possible de déterminer combien de segments ont été transmis entre ces deux points à partir du numéro de séquence.

Solution : F

- (1 point) (n) UDP préserve les frontières entre les messages envoyés par l'application, mais pas TCP.

Solution : V

- (1 point) (o) Le corps (*message body*) d'un message de réponse HTTP n'est jamais vide.

Solution : F

Question 3 (8 points sur 100)

Dans la définition d'une machine d'état pour la transmission fiable de données, des mécanismes ont été introduits pour permettre la transmission fiable de données suite à certains événements dans un réseau. Décrivez quel(s) mécanisme(s) ont été introduit pour résoudre les problèmes suivants :

1. Paquet reçu avec erreur

Solution :

- Une somme de contrôle a été introduit pour détecter l'erreur dans le paquet.
- Un accusé de réception (ACK) et de non-réception (NAK) ont été introduits pour signaler le transmetteur que le paquet est reçu correctement ou si une retransmission est nécessaire.

2. Paquet perdu

Solution :

- Compteur de temps

3. Une retransmission de données provoquant une duplication au récepteur

Solution :

- Un numéro de séquence. (dans le paquet envoyé et dans le ACK/NAK)

4. Faible taux d'utilisation du réseau (faible efficacité)

Solution :

- Pipelining : une gamme de numéro de séquence définissent une fenêtre de transmission disponible à l'envoyeur. Ceci permet de transmettre des paquets sans attendre l'accusé de réception.

Question 4 (10 points sur 100)

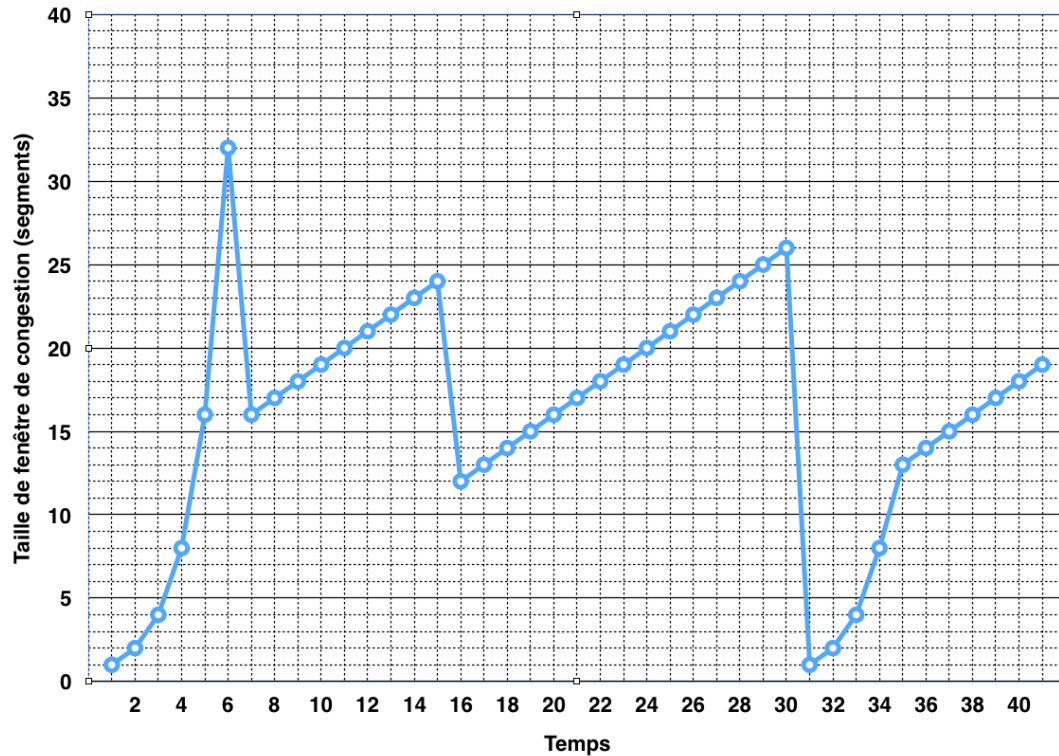
Un nœud A utilise une connexion TCP pour transmettre un énorme fichier au nœud B. Cette connexion ne subit aucune perte de paquet, et les compteurs de temps n'expirent jamais. Le lien entre les nœuds A et B offre un taux de transmission de 10 Gbps. Le délai aller-retour entre les deux nœuds est de 60 ms.

La mémoire en écriture et lecture du socket TCP sur les deux nœuds est de 4 Mo (limite défaut sur une installation Linux).

1. Qu'est-ce qui peut empêcher le nœud A de transmettre continuellement sur son socket TCP à un taux qui approche 10 Gbps ? Contrôle de flux ? Contrôle de congestion ? Autre facteurs ? **Élaborez. Votre explication doit inclure une description du comportement du contrôle de flux et contrôle de congestion dans ce scénario.**
2. Quelle devrait être la taille de fenêtre TCP pour optimiser l'utilisation du réseau lors de la transmission du nœud A vers le nœud B ?
3. Quelle mémoire faut-il ajuster ? Le nœud A ? B ? les deux ? Mémoire en écriture ou en lecture ?

Question 5 (16 points sur 100)

Le graphique suivant démontre la taille (en segments) de fenêtre de congestion TCP en fonction du temps. Le graphique est basé sur le comportement de TCP Reno, tel que discuté en classe. Répondez aux questions suivantes. **Justifiez chacune de vos réponses.**



- (3 points) (a) Identifiez dans quel(s) intervalle(s) de temps où TCP est en mode de démarrage lent (slow start).

Solution : Intervalles $[1, 6]$ et $[31, 35]$. Augmentation exponentielle de la taille de fenêtre de congestion.

- (3 points) (b) Identifiez dans quel(s) intervalle(s) de temps où TCP est en mode d'évitement de congestion (congestion avoidance).

Solution : $[7, 15]$, $[16, 30]$ et $[35, 41]$. Augmentation linéaire de la taille de fenêtre de congestion.

- (2 points) (c) Après la 15ème transmission, quel événement a provoqué la diminution de la taille de fenêtre de congestion ?

Solution : 3 ACK dupliqués, car la fenêtre de congestion est diminuée de moitié et on retombe en évitement de congestion.

- (2 points) (d) Après la 30ème transmission, quel événement a provoqué la diminution de la taille de fenêtre de congestion ?

Solution : Temporisation, car la taille fenêtre de congestion descend à 1, et le démarrage lent reprends.

- (3 points) (e) Quel est le seuil d'évitement de congestion (threshold) après la 6ème transmission ?

Solution : Seuil est à 16, car c'est à là où l'augmentation de la taille de fenêtre de congestion reprend à l'étape d'évitement de congestion.

- (3 points) (f) Quel est le seuil d'évitement de congestion (threshold) après la 30ème transmission ?

Solution : Le seuil est diminuée à 13, soit la moitié de la taille de fenêtre de congestion (26) lorsque la perte de segment est détectée (temps 30).

Question 6 (10 points sur 100)

- (2 points) (a) Un processus sur un nœud veut communiquer avec un processus en exécution sur un nœud distant. Dans le but d'établir une communication TCP ou UDP, quel information le processus a-t-il besoin pour identifier le processus distant ?

Solution : Le processus doit connaître l'adresse IP et le numéro de port TCP/UDP.

- (2 points) (b) Si vous devez créer un programme qui doit effectuer une transaction avec un client distant le plus rapidement possible. Est-ce que vous choisissez TCP ou UDP ? Pourquoi ?

Solution : UDP. On évite le temps d'établissement d'une connexion requis par TCP.

- (2 points) (c) Quelle est la différence entre le MAIL FROM: dans le protocole SMTP et le From: dans le message de courriel ?

Solution : Le MAIL FROM: est un message échangé dans le protocole SMTP et identifie l'envoyeur du message au serveur SMTP. Le From: est une entête qui fait parti du message de courriel (et non du protocole SMTP).

- (2 points) (d) À quoi sert une base de donnée «whois» ?

Solution : Contient l'information associée aux adresses IP et nom de domaine : Le registraire, le propriétaire (administratif et technique), nom des serveurs DNS, etc.

- (2 points) (e) Quelle est la différence entre un document «Draft Internet» et un document «Request for Comments (RFC)» de l'IETF ?

Solution : Un draft internet est un document de travail, alors qu'un RFC est un document stable. Un RFC peut définir un standard.

Question 7 (15 points sur 100)

- (3 points) (a) Expliquez ce que représente les records DNS suivants : A, MX et NS.

Solution :

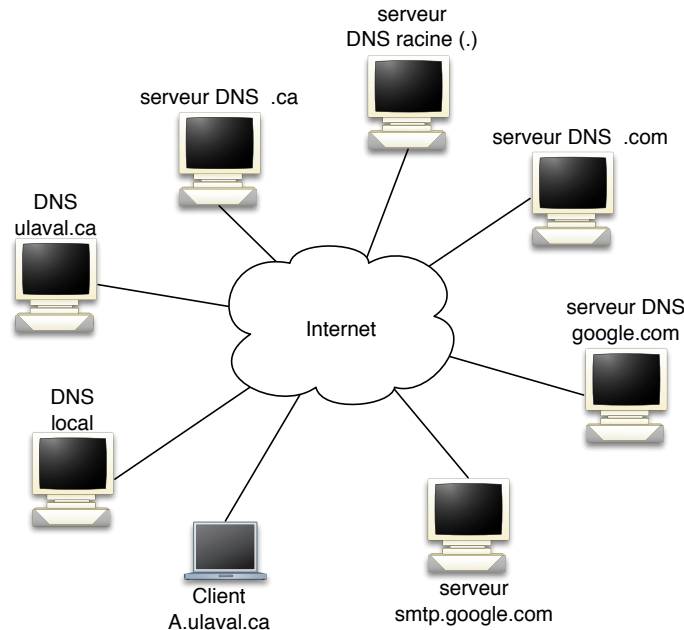
A Adresse IPv4 du nom donné

MX Nom du serveur de courrier électronique pour un nom de domaine donné

NS Nom d'un serveur DNS autoritaire pour un nom de domaine donné

- (2 points) (b) Pourquoi un *temps de vie* (TTL) est-il associé aux réponses obtenues du DNS ?

- (10 points) (c) Décrivez le processus de résolution de nom DNS dans le scénario suivant.



- Le poste de travail A est configuré (via DHCP) pour utiliser le serveur DNS local pour le service de résolution de nom (DNS).
 - La mémoire cache du serveur DNS local est initialement vide.
 - Le client A amorce une demande de résolution d'adresse IPv6 pour smtp.google.com
- Énumérez les échanges effectués en commençant par le message DNS (question) envoyée par le poste client, suivit par questions et réponses des différents serveurs impliqués. Votre réponse doit être sous la forme d'une liste en ordre chronologique. Chaque ligne de votre liste doit contenir les informations suivante :
- Le nœud source qui transmet la requête ou la réponse ;
 - Le nœud destination qui reçoit la requête ou la réponse ;
 - La question ou la réponse envoyée, ainsi que le type de record

Source	Destination	message DNS (question ou réponse)
A		
...		

	Transmetteur	Récepteur	message
Solution :	A	DNS local	AAAA pour smtp.google.com ?
	DNS local	racine	AAAA pour smtp.google.com ?
	racine	DNS local	Demande à DNS .com (NS pour .com)
	DNS local	serveur DNS .com	AAAA pour smtp.google.com ?
	serveur DNS .com	DNS local	Demande à DNS google.com (NS google.com)
	DNS local	serveur DNS google.com	AAAA pour smtp.google.com ?
	serveur DNS google.com	DNS local	Retourne record(s) AAAA pour smtp.google.com
	DNS local	A	Retourne record(s) AAAA pour smtp.google.com

Question 8 (16 points sur 100)

- (8 points) (a) Le *socket API* offre les appels de système suivants :

<code>socket(int domain, int type, int protocol)</code>
<code>connect(int socket, const struct sockaddr *address, socklen_t address_len)</code>
<code>bind(int socket, const struct sockaddr *address, socklen_t address_len)</code>
<code>listen(int socket, int backlog)</code>
<code>accept(int socket, struct sockaddr *address, socklen_t *address_len)</code>

Identifiez lesquelles de ces fonctions sont nécessaires dans les cas suivants :

1. Un client TCP
2. Un client UDP
3. Un serveur TCP
4. Un serveur UDP

Solution :

client TCP	socket, connect
client UDP	socket
serveur TCP	socket, bind, listen, accept
serveur UDP	socket, bind

- (8 points) (b) Dans un programme, la fonction `getaddrinfo` est utilisée pour obtenir la liste des adresse(s) IP et numéro(s) de port associés à un nom de machine et nom de service. Ces informations sont retournées dans une structure de type `addrinfo`

```
int getaddrinfo(const char *hostname, const char *servname, const struct
addrinfo *hints, struct addrinfo **res);

struct addrinfo {
    int ai_flags;
    int ai_family;
    int ai_socktype;
    int ai_protocol;
    socklen_t ai_addrlen;
    struct sockaddr *ai_addr;
    char *ai_canonname;
    struct addrinfo *ai_next;
};
```

Si une application demande la résolution du nom «mail.ulaval.ca» pour le service «smtp» (`getaddrinfo("mail.ulaval.ca", "smtp", hints, res)`), quel sera le résultat dans liste chaînée `res` retournée par fonction `getaddrinfo`?

Assumez que la structure `hints` contient les valeurs suivantes :

- `hints.ai_family = PF_UNSPEC;`
- `hints.ai_socktype = SOCK_STREAM;`

Les entrées DNS sont les suivantes :

mail.ulaval.ca	A	132.203.250.26
smtp.ulaval.ca	A	132.203.250.25
mail.ulaval.ca	AAAA	2620:0:1af0::1
smtp.ulaval.ca	AAAA	2620:0:1af0::25

IMPORTANT : Il n'est pas nécessaire d'inscrire tout les éléments de la structure `addrinfo`. Votre réponse doit contenir l'information sur la famille d'adresse, le protocole, l'adresse et le numéro de port.

	résultats 1	résultats 2	...
Famille			...
Protocole			...
no. de port			...
Adresse			...

Solution :

	résultats 1	résultats 2
Famille	IPv6	IPv4
Protocole	TCP	TCP
no. de port	25	25
Adresse	2620:0:1af0::1	132.203.250.26