Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

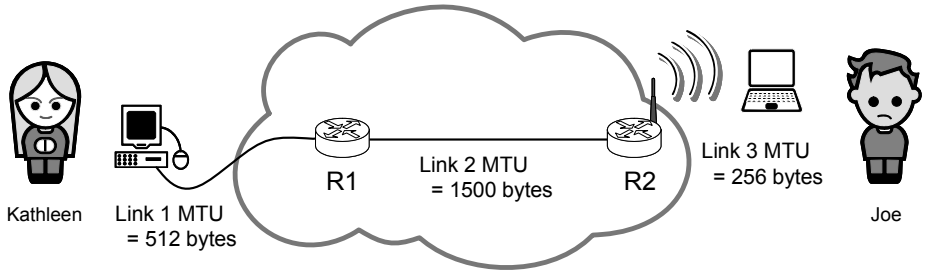
Protocoles & Prog. Réseau

Corrections TD n°3

Fragmentation et protocole temps réel

Fragmentation

1 – Questions sur la fragmentation du datagramme IP :



a. Combien de datagrammes sont générés par l'ordinateur de Kathleen dans la couche IP ?
Chaque trame sur « Link 1 » fait au plus 512 *octets ce qui permet de transporter* $512-(30+20+20) = 442$ *octets de données.*
Kathleen envoie $8 * 1024 = 8192$ *octets de données, ce qui donne après division entière par* 442 *:* $8192 = 18 * 442 + 236$
Kathleen transmet 19 *datagrammes (*18 *de* 482 *octets et un de* 276 *octets, chacun avec* 40 *octets d'entête IP&TCP).*
On ne tient pas compte des segments de connexion et de fin de la communication TCP.

b. Combien de fragments Joe reçoit dans la couche IP ? Expliquez votre calcul.
Le MTU du « Link 3 » est de taille inférieure à celui du « Link 1 », il est donc nécessaire de fragmenter les paquets provenant de la machine de Kathleen :
- *sur le « Link 3 », les trames sont de taille* 256 *octets, soit* $256-30 = 226$ *octets pour le datagramme, et* 206 *octets pour le segment TCP ;*
- *les entêtes entre « Link 1 » et « Link 3 » sont de même taille :*
  - *pour un datagramme de* 482 *octets, il est fragmenté en* $462 = 2 * 200 + 62$ *(seulement* 200 *octets pour être un multiple de 8, conformément aux exigences de taille de fragment), soient 3 datagrammes : le premier et le second font* $200+20 = 220$ *octets et le troisième fait* $62+20 = 82$ *octets ;*
  - *pour un datagramme transportant un segment de* 276 *octets, il est fragmenté en* $256 = 200+56$*, soient 2 datagrammes : un de* $200 + 20 = 220$ *octets et le second de* $56 + 20 = 76$ *octets.*
- *en tout la fragmentation va créer :* $18 * 3 + 1 * 2 = 56$ *paquets.*

c. Décrivez les en-têtes des 4 premiers datagrammes et du dernier datagramme que Joe reçoit en indiquant **uniquement** la valeur des paramètres suivants : taille du datagramme, identifiant, offset et drapeaux (on choisira la valeur 672 comme identifiant du premier datagramme émis par Kathleen).

taille	identifiant	offset	drapeaux
220	672	0	MF
220	672	200 (25)	MF
82	672	400 (50)	-
220	673	0	MF
...	...	...	...
76	690	200 (25)	-

Les 4 premiers datagrammes reçus par Joe sont les 3 premiers fragments du premier datagramme envoyé par Kathleen, plus le premier fragment du second datagramme envoyé.
L'offset correspond à la taille du contenu + l'offset du paquet précédent, par exemple pour le troisième fragment : $(25 * 8 + 25 * 8)/8 = 50$
Le dernier datagramme reçu est le second fragment résultant de la fragmentation du dernier datagramme envoyé (numéro 19 soit l'identifiant 690).

d. Que se passe-t-il si le dernier datagramme se perd sur le lien 3 ? Combien de datagrammes IP seront retransmis par l'ordinateur de Kathleen ? Combien de fragments retransmis Joe va recevoir ?
Si le dernier datagramme se perd, alors le datagramme fragmenté auquel il appartient ne peut être reconstruit et il est entièrement annulé (tous les fragments reçus sont détruits).
L'ordinateur de Kathleen va retransmettre le dernier datagramme, n°19, car il n'aura pas eu confirmation de sa réception, ce qui donnera lieu à la réception de deux datagrammes de fragmentation sur la machine de Joe.

Fragmentation et Protocole temps réel

Le protocole RTP, «Real Time Transport Protocol», RFC 1889 &3550 est généralement encapsulé dans UDP.

2 – Soit une session de téléphonie IP, *VoIP*, entre un hôte A et un hôte B :
- ⋆ chaque paquet contenant un échantillon sonore et partant de A à destination de B est numéroté ;
- ⋆ l'utilisateur sur A commence à parler au temps 0 ;
- ⋆ l'hôte A envoie un paquet toutes les $20ms$;
- ⋆ les paquets arrivent sur l'hôte B avec les temps indiqués dans le tableau ci-dessous ;
- ⋆ l'hôte B utilise un temps de retard dans son utilisation du contenu de chaque paquet $q = 210ms$.

a. Pourquoi certains paquets risquent d'être ignorés ?
Le récepteur choisit un temps de retard suffisamment grand pour annuler le jitter, puis utilise/joue chaque échantillon sonore au même rythme que le récepteur les construit et les envoi (toutes les 20ms). Les paquets reçus après le temps prévu de leur utilisation sont ignorés.

b. Complétez le tableau en indiquant le temps d'utilisation de chacun des paquets reçus (vous pouvez vous aider du graphe ci-dessous).

N°paquet	Tps arrivée r_i	Tps utilisation
1	195ms	230
2	245ms	250
3	270ms	270
4	295ms	ignoré > 290
5	300ms	310
6	310ms	330
7	340ms	350
8	380ms	ignoré > 370
9	385ms	390
10	405ms	410