

# **Systemes Multimedia**

## **Correction<sup>1</sup> de l'épreuve du 21 janvier 2005**

*Version 0.9 du document - Les auteurs précise que des libertes ont été prises avec le grammaraire française.*

## **1 Exercice 1 : Qualité de Service**

### **1.1 Problèmes de QoS**

Lors de la transmission sur le net, quatres phénomènes interviennent dans les pertes de paquets (cf. page 26) :

- Erreurs Binaires ( interférences )
- Dépassement des capacités des buffers de routeurs
- Reception tardive des paquets
- Expiration des TTL ( nb de passage dans des buffers )

Considérons l'émission de paquet vidéo décodé. On rappelle qu'un tel format vidéo suppose l'émission de *Group of Pictures* tels que :

- Image/Bloc I :  
images index, codées sur elles meme
- Image/Bloc P :  
images predictives, codées par rapport aux images précédentes
- Image/Bloc B :  
images bidirectionnelles, codées par rapport aux images précédentes et suivantes

Soit par ordre d'importance :  $I > P > B$

Dès lors si il y a perte d'un bloc I, l'erreur se propagera sur le reste de l'image, durant la durée de vie du GoP ( cf. gif page 49 ). Dans le cas d'une perte d'un bloc P, l'erreur se propage sur les blocs B qui en dépendent codés. Pour le cas de pertes sur un bloc B, il n'y a pas de propagation de l'erreur.

L'introduction de *checksum* permet d'une part de corriger les erreurs sur les paquets eux-memes. Pour limiter les pertes de transmission pures ( retard etc... ), des mécanismes de priorité peuvent être instaurée. Ceci inclu une gestin des files d'attentes dans les buffers/routeurs, ainsi qu'une numérotation des paquets.

---

<sup>1</sup>ou plutot "Proposition de correction de la GCTE"

## 1.2 QoS & Délais

Les principales causes de délais dans la transmission de paquet par IP sont ( cf. page 21 ) :

- Le temps de transit sur le reseau ( vitesse de propagation ) : 70 ms
- Le temps de transit dans les buffers et la gigue : 60 ms

Si l'on considère un service interactif, type *ssh*, les délais doivent être au plus court ( max 400 ms ) afin que les retours d'information ( affichage de caractère sur l'écran ) soient parfaits. Pour des applications de streaming vidéo, les délais peuvent atteindre plusieurs secondes puisque les buffers permettent de compenser la gigue qui intervient dans la transmission des paquets ( cf. page 19 ).

Le mémoire tampon en entrée de terminal de réception sert, comme évoqué au précédent paragraphe, de compenser les effets de la gigue, ainsi qu'un réordonnement des paquets qd cela est nécessaire. Ce type de buffer est appelé *buffer de synchronisation* (cf. page 25 ) et ne doivent passer 50 à 57 ms ( seuil d'interactivité ).

## 2 Exercice 2 (Transport audiovisuel)

### 2.1 Question 1

Les deux modes de streaming multicast utilisés par les services IP sont :

- multicast applicatif,
- multicast réseau (= multicast IP).

*le multicast applicatif*

Description : dans ce cas, on ouvre plusieurs sessions unicast, c'est à dire 1-to-1.

Les problèmes :

- consommation énorme des ressources du serveur,
- au niveau des clients : latence importante, QoS non assuré, performances moyennes pour les opérations de VCR,
- pour l'ISP : problèmes de congestion du trafic.

*le multicast réseau (=multicast IP)*

Description : dans ce cas, le serveur envoie un unique échantillon qui est dupliqué par les routeur ensuite. Donc ça demande moins de ressource au niveau du serveur.

Les problèmes :

- serveur : marche bien QUE si le réseau est homogène, càd dispose QUE des mêmes machines,

- ISP : impossibilité de connaître précisément le nombre de clients connectés, donc il y a des difficultés pour facturer les clients,
- problème de scalabilité, càd de dimensionnement du réseau (pb des groupes de grandes tailles, de l'éloignement géographique,...)

## 2.2 Question 2 : calculs liés à un codec audio

Les données :

- trame de 160 échantillons càd 8kHz,
- fonction d'excitation codée sur 16bits,
- 10 coefficients codés sur 8 bits,
- un gain G codé sur 8 bits.

*Calcul pour la trame de DATA util*

On code 160 échantillons avec une fonction d'excitation, 10 coefficients et un gain.

Ainsi on a :

$$16bits + (10 * 8)bits + 8bits = 104bits/160echantillons$$

Or 160 échantillons=8kHz=20ms.

Donc le débit pour la DATA util est :

$$\frac{104bits = 13octets}{20 * 10^{-3}} = 5,2kb/s$$

Donc le débit à la sortie du codec est :

$$Debit_{codec} = 5,2kb/s$$

*Calcul charge util d'un paquet RTP* La taille d'une en-tête RTP est 12 octets.

Le débit avec l'en-tête RTP devient alors :

$$\frac{104bits + 12octets * 8}{20 * 10^{-3}} = 10kb/s$$

Donc la charge util d'un paquet RTP est :

$$\frac{12 * 8}{20 * 10^{-3}} = 4,8kb/s$$

*Calcul de la charge util d'un paquet UDP* de même la charge utile d'un paquet UDP (8 octets) est :

$$\frac{8 * 8}{20 * 10^{-3}} = 3,2kb/s$$

*Calcul de la charge util d'un paquet IP* de même la charge utile d'un paquet IP (20 octets) est :

$$\frac{20 * 8}{20 * 10^{-3}} = 8kb/s$$

Le flux total sur le réseau est alors :

$$5,2 + 4,8 + 3,2 + 8 = 21,2kb/s$$

Donc :  $flux_{total} = 21,2kb/s$

Les pourcentage obtenus sont alors :

– RTP :  $\frac{4,8}{flux_{total}} = 22,6\%$ ,

– UDP :  $\frac{3,2}{flux_{total}} = 15,1\%$ ,

– IP :  $\frac{8}{flux_{total}} = 37,7\%$ ,

*Les problèmes de l'overhead [CECI N'EST PAS SUR]*

Plus les paquets sont courts, plus l'overhead est élevé. Or pour diminuer les délai il faut justement des paquets cours. Par conséquent pour transmettre plus d'info, il faut avoir un débit élevé.

## 2.3 Question 3

### 2.3.1 pourquoi UDP et TCP ça marche pas ?

TCP ne marche pas car :

- TCP ne supporte pas le multicast IP,
- l'audio et la vidéo ont déjà des contrôles naturels d'erreurs dans le système audio-visuel humain, donc les contrôles d'erreurs liés à TCP sont superflus,
- fiabilité par acquittement,
- réémission de paquets manquant,
- contrôle de flux par fenêtre de congestion,
- UNIQUE INTERET : numero de la sequence pour reorganiser les paquets.

UDP ne marche pas car :

- UDP n'as pas en natif de gestion du temps réel.
- UDP peut consommer 100% de la bande passante,
- UDP n'a pas le BONUS du TCP (cf "unique interet")

### 2.3.2 grandes fonctionsde RTP

Description de RTP :

- protocol non disponible au niveau de l'OS,
- dispose de la gestion du temps réel ;
- doit bien gérer le multicast IP [A CONFIRMER],
- systeme d'horodatage pour restaurer la base des temps (=timestamp),
- numérotation des séquences,
- détection de la perte des paquets et information de la source dans des délais compatibles avec le service.

### 2.3.3 Fonctions de RTCP

RTCP envoie une boucle de “feedback” (=retour) sur la qualité de la transmission. C’est basé sur la transmission périodique de paquets de contrôle. Sinon 4 grandes fonctionnalités :

- fournir des informations sur la qualité de la session,
- garder une trace de tous les participants à une session,
- contrôler le débit auquel les participants à une session RTP transmettent leurs paquets RTCP,
- transmettre des informations de contrôle sur la session.

## 2.4 Question 4 : streaming RTP exploitation d’un graphique

### 2.4.1 Commentaire de la figure

UDP utilise beaucoup plus de bande passante que TCP car :

- pas de garanti de transport,
- pas de contrôle d’erreur,
- pas de temps réel.

### 2.4.2 partage équitable entre TCP et UDP

La solution à ce problème est l’introduction du TCP Friendly. C’est un TCP sans la gestion des erreurs et implémenté au dessus de UDP. Donc il s’adapte bien à la congestion des réseaux.

### 2.4.3 Mécanisme d’adaptation entre le seveur et le client

On utilise l’équation TFRC :

$$R_{TCP} = \frac{S}{RTT \sqrt{\frac{2*B*P}{3}} + t_{RTO} * 3 \sqrt{\frac{3*B*P}{8}} p(1 + 32p^2)}$$

avec :

- S : taille d’un paquet,
- RTT : temps d’un aller-retour,
- P : taux de perte de paquets,
- B : nombre de paquets acquitté par un seul ACK,
- $t_{RTO}$  : time out de transmission.

Grâce à la numérotation des paquets, le client mesure le taux de perte de paquets. Ensuite il envoie ces taux de pertes au serveur qui via l’équation TFRC calcule le débit de la session. Ensuite le serveur ajuste le débit en fonction de la valeur calculée avec la formule TFRC.

### 3 End of the corrigé

Ce document a été réalisé par la GCTE promo 2006 (Geek Clubbing Team Elec). Pensez à mettre radio FG le soir dès 22h. En bonus, quelques photos de la promo 2006...déchéance toizza ...

