

à l'Université Pierre et Marie Curie, le 28 septembre 2004

# M2 Informatique Réseaux

## Multimédia et Qualité de Service

### Cours 1 : RTP

Timur FRIEDMAN

# A propos du module

- Cinq cours

- RTP

- multicast

- TCP Friendly

- signalisation

- Int Serv, Dif f Serv, MPLS, et RSVP

# Evaluation

## ■ Note

- 60 % écrit
- 40 % contrôle continu

## ■ Écrit

- examen (documents non autorisés)

## ■ Contrôle continu

- questions en cours : 10 %
- TDs : 30 %
- papers, normes – discussion : 30 %
- contribution au wiki : 30 %

# Pour la semaine prochaine

- préparer les exercices du TD
  - sur le site [www-rp.lip6.fr/~friedman](http://www-rp.lip6.fr/~friedman) à partir de demain
- lire le papier
  - Timer Reconsideration for Enhanced RTP Scalability
    - par Jonathan Rosenberg Henning Schulzrinne
    - Proc. Infocom 1998
- lire le RFC 3550
- visiter le Wiki
  - disponible à partir de demain
- préparer le prochain cours sur le multicast

# Survol du module

- Basé sur des connaissances réseau
  - en particulier : internet
- Revisiter les couches OSI suivantes :
  - Application
  - Transport
  - Réseau
- Les voir sous l'optique du multimédia

# Motivation

- Internet : pas conçu pour multimédia
- un réseau pour texte, données
  
- Depuis ...
- Web : images
- lecture en transit (*streaming*)  
audio et vidéo
- jeux interactifs
- voix sur IP (*VoIP*) : voix en temps réelle
- visioconférence

# Problématique

- Internet : un réseau dit de « moindre effort » (*best effort*)
  - délais
  - pertes
  - déséquencement
  - duplicatas
- Applications multimédia : besoin de garanties
  - e.g. VoIP avec trop de délai ne fonctionne pas
- Est-ce qu'un seul réseau peut tout fournir ?

# Couche application

- Le signalisation

- RTSP, SDP, H.323, SIP



# Couche transport

- Confronter le congestion dans le réseau
  - l'approche TCP-Friendly (*TCP-Friendly*)
  - DCCP/ TFRC
  - l'approche Qualité de service (*QoS*)
  - Int Serv, Dif Serv, RSVP, MPLS
  - (sur plusieurs couches, mais on en parle ici)
- 
- Fournir les fonctionnalités temps réelle
  - RTP/ RTCP

# Couche réseau

- Diffusion à grande échelle : le multicast
- IGMP (protocole de bordure)
- DVMRP et PIM (protocoles de routage)
- RMT (couche transport)
-

# Plan du cours

- Introduction
- Communication en temps réel
- RTP
- Limitation de débit RTCP
- RTP et le temps réel

# Introduction

- RTP est un protocole de transport dans l'Internet
  - Dans la couche 4, comme TCP, mais implémenté sur UDP
  - RFCs 3550 et 3551 (juillet 2003 – RFCs originaux de 1996)
- RTP est pour les applications « temps réel »
  - Par exemple la téléphonie, le visioconférence
  - TCP n'est pas adapté pour ces applications
- RTP est conçu pour la communication multipoint
  - RTP marche sur le multicast
- RTP fournit un cadre aux applications
  - Il laisse beaucoup de fonctionnalité aux applications
  - Il fournit les outils nécessaires (e.g. estampilles temporelles)

# Plan

- Introduction
- Communication en temps réel
  - Applications
  - Caractéristiques
  - Besoins
- RTP
- Limitation de débit RTCP
- RTP et le temps réel

# Communications en temps réel

- Avertissement : une certaine idée du temps réel
  - Orientée audio et vidéo interactive
  - Pas le temps réel dans le sens technique
    - Pas « systèmes durs temps réel »
- Mieux compris en fonction d'exemples...

# Applications typiques

- Voix sur IP (« VoIP »)
  - Téléphonie
- Visioconférence
  - Voix
  - Vidéo
  - Transparents
- Jeux interactives
  - Mises à jour de mouvement
  - Communication entre participants

# Applications moins typiques

- Diffusion d'actualités
  - Données (par exemple à propos de la bourse)
- Moniteur à distance
  - Collection de données
- Control télémétrique
  - Données
  - Commandes
- RTP à été conçu plutôt pour audio et vidéo
  - (Mais pas exclusivement)



# Caractéristiques

## ■ Interactivité

- Exemple : téléphonie

- Communication dans deux sens

- Exemple moins typique : vidéo sur demande

- Seulement marche/arrêt, reculer, avancer, recherche

## ■ Intolérance pour les délais

- Exemple : vidéo

- Une trame arrive  $> 500$  ms en retard

- Mieux jeter que d'utiliser

# Caractéristiques bis

- Tolérance pour les pertes
  - Exemple : voix en paquets G.711 (sans redondance)
    - Perte de 1 % de trames négligeable en termes de qualité
  - Exemple : vidéo codé en MPEG
    - Perte de trames B est permis
    - Au moins que les trames I et P arrivent
  - Exemple moins typique : données de la bourse
  
- Communications multipoint
  - Exemple : visioconférence
  - Communications bidirectionnels sont un cas particulier

# Besoins

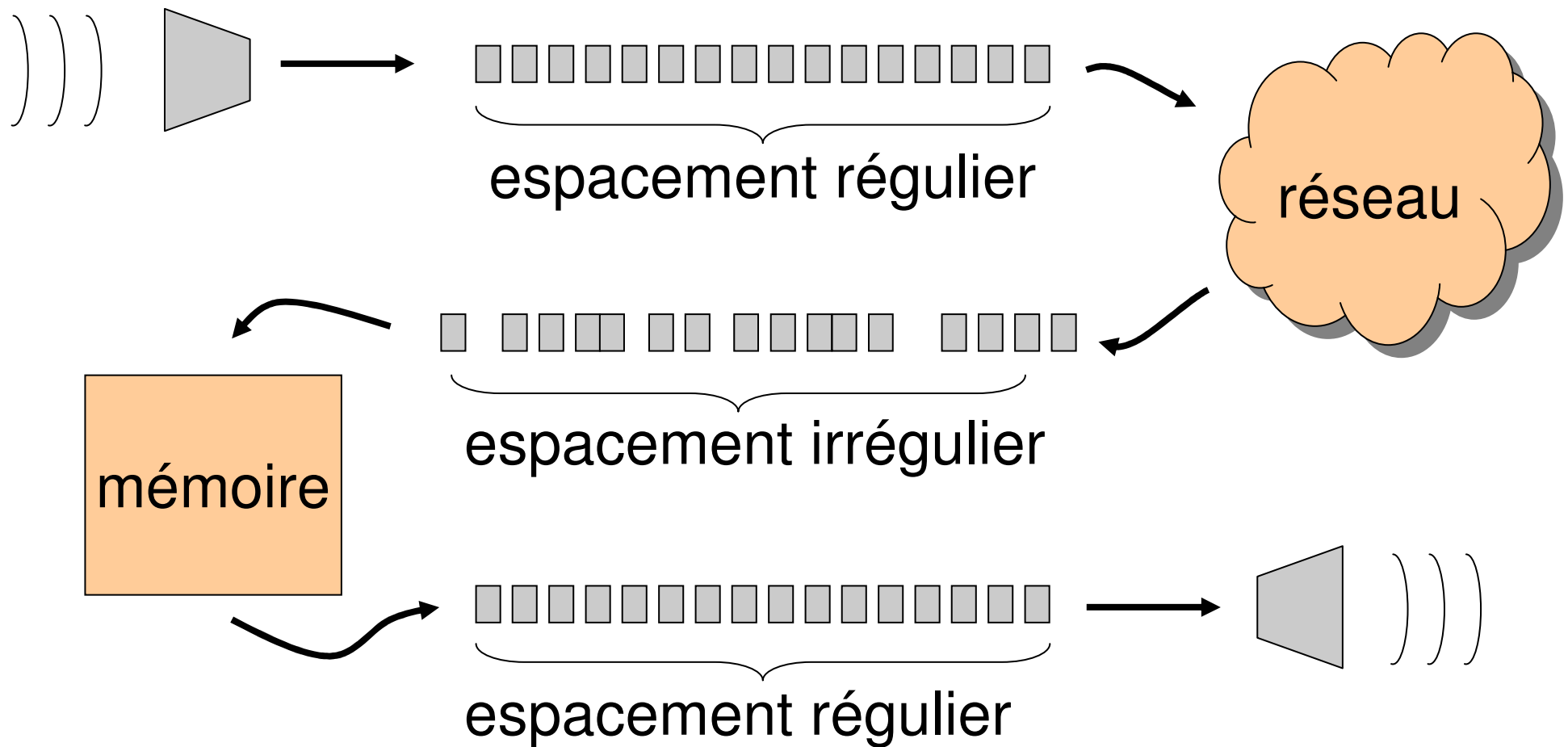
- Horodatage des données
- Synchronisation des flux
- Résistance contre les pertes, duplicatas, mauvais ordre
- Identification de participants
- Surveillance de l'état de la connexion
- Contrôle de flux
- Contrôle de congestion
- Besoins avancés :
  - Support pour le transcodage de données
  - Sécurité de données

# Horodatage des données

- l'Internet est un réseau « moindre effort »
  - Pas de garanties de délai
  - A la différence avec X.25, par exemple
    - X.25 garde les intervalles d'origine
- Contraintes temporelles pour la lecture
  - Chaque trame audio ou vidéo doit être lu à un moment précis
  - On jette les trames qui arrivent en retard
- Technique : mise en mémoire
  - Avec l'horodatage on récupère les intervalles d'origine

# Horodatage exemple

la norme G.729 : trames de voix à 62,5 Hz



# Synchronisation des flux

- Une applications peut avoir plusieurs flux
  - Audio
  - Vidéo
  - Transparents
- La lecture des flux doit être synchronisée
  - Si, par exemple, la voix n'est pas codée dans le vidéo
- Besoin d'établir l'équivalence entre :
  - L'horodatage de chaque flux
    - (Il peut être artificiellement régulier)
  - Le « temps de l'horloge murale »

# Synchronisation exemple

d'après H. Schulzrinne

estampilles temporelles audio :



460

estampilles temporelles vidéo :



1900

temps de l'horloge murale :

8:45:17.23

# Résistance contre les pertes, etc.

- L'Internet est un réseau « moindre effort », donc :
  - Pertes des paquets
  - Duplicatas des paquets
  - Paquets qui arrivent dans le mauvais ordre
- Pour faire face aux pertes : la redondance
  - L'émetteur doit connaître le taux de pertes
  - Plus il y a de pertes, plus on ajoute de la redondance
- Pour faire face aux duplicatas, mauvais ordre :
  - Numéros de séquence
- Retransmissions ?
  - Pas systématiquement (problème de délais)



# Identification des participants

- Communications multipoint : plusieurs participants
  - Identification aux autres participants
  - Démultiplexage des flux
    - Destinés vers le même adresse, même numéro de port
- Moyens d'identification
  - Nom : M Dupont
  - Adresse courriel : [m.dupont@online.fr](mailto:m.dupont@online.fr)
  - Nom logique de machine : dhcp-51.online.fr
  - Adresse IP / numéro de port : 135.227.61.57/24882
  - Autres ?

# Surveillance de l'état de la connexion

- Paramètres génériques :
  - Le taux de pertes
  - La gigue (variance des délais)
- La connaissance de ces paramètres aide :
  - Les applications adaptatives
    - Ajouter de la redondance en fonction des pertes
    - Augmenter la mise en mémoire en fonction de la gigue
  - L'administrateur du réseau
    - Reconnaissance des failles

# Contrôle de flux

- Chaque récepteur peut avoir plusieurs mémoires :
  - Pour l'audio
  - Pour le vidéo
- Dans les communications multipoints :
  - Le nombre peut être multiplié par le nombre d'émetteurs
- Comment signaler la mémoire disponible ?
  - Dans le TCP : fenêtre avertie
  - Dans le multi flux, multipoint c'est plus compliqué

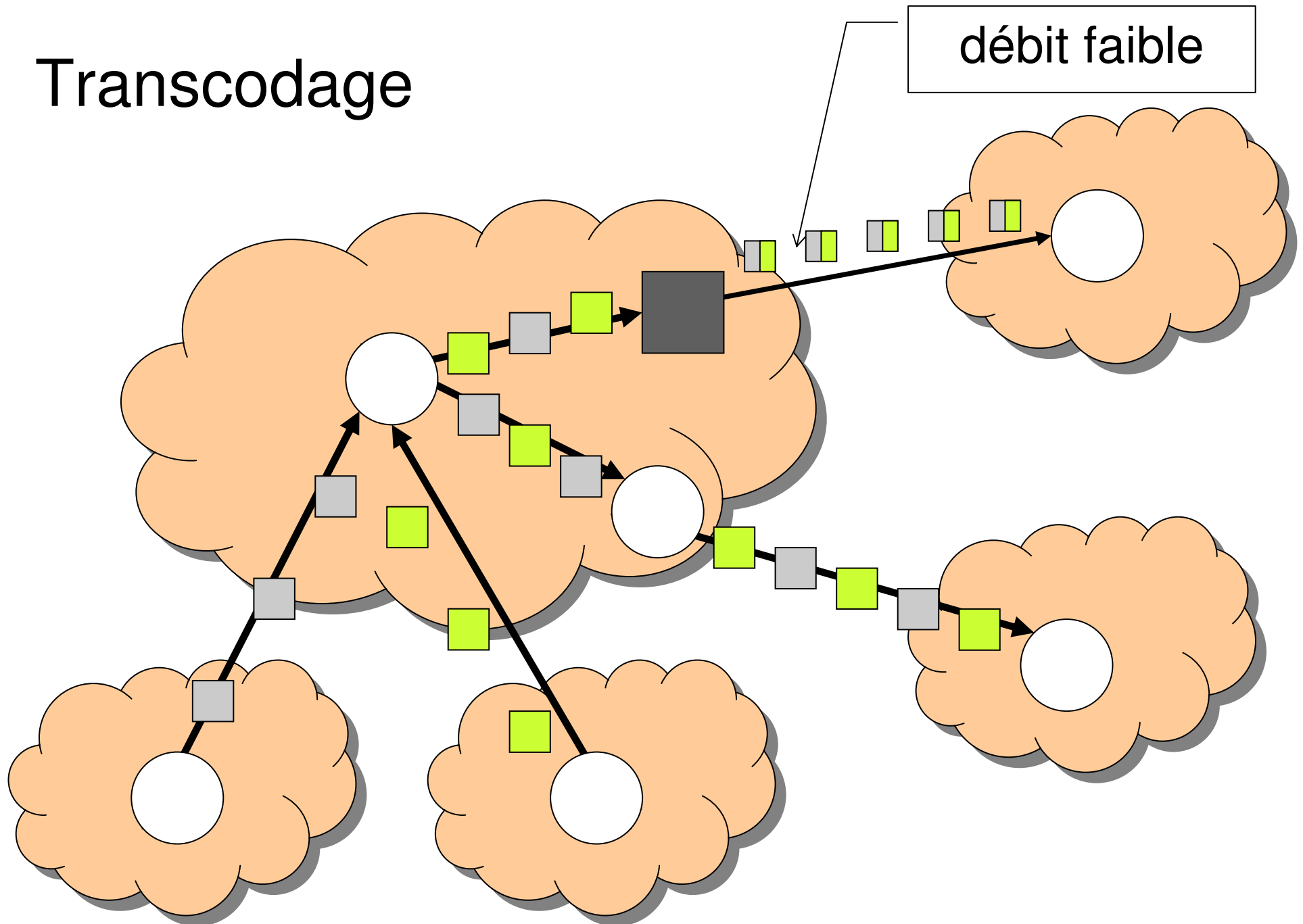
# Contrôle de congestion

- Le contrôle de congestion dans l'Internet :
  - TCP
  - Un contrôle de bout en bout
- Problèmes avec le TCP pour le temps réel :
  - TCP est monolithique :
    - Contrôle de congestion
    - Fiabilité par retransmissions
  - TCP existe exclusivement en version unicast
- Si on veut se dispenser de TCP
  - Il est conseillé d'être amicale avec TCP (« TCP-friendly »)
  - On doit connaître les délais et les taux de pertes

# Transcodage

- Des situations difficiles :
  - Un récepteur derrière un lien à débit faible
  - Un récepteur qui ne peut pas décoder un format donné
  - Un récepteur derrière un pare-feux
- On peut transcoder l'audio, le vidéo :
  - Éliminer le stéréo, diminuer la qualité
  - Rendre les images plus petites
  - Changer de formats
- On peut combiner les flux
- On peut changer les numéros de port

# Transcodage



# Sécurité

- Authentification des participants
- Autorisation des participants
- Intégrité de données
- Confidentialité de données

# Plan

- Introduction
- Communication en temps réel
- RTP
  - Pourquoi un autre protocole de transport ?
  - Séparation données/contrôle
  - Profiles différentes pour applications différentes
  - Les paquets RTP
  - Les paquets RTCP
- Limitation de débit RTCP
- RTP et le temps réel



# Pourquoi un autre protocole de transport ?

## ■ Pourquoi pas TCP ?

- TCP exige la fiabilité à 100%
- TCP favorise la fiabilité au dépens des délais
- TCP existe seulement en version unicast

## ■ Pourquoi pas UDP ?

- UDP fournit peu d'outils :
  - Les numéros de port pour le démultiplexage
  - Un checksum

## ■ RTP est adapté aux besoins du temps réel

## ■ RTP est léger et flexible

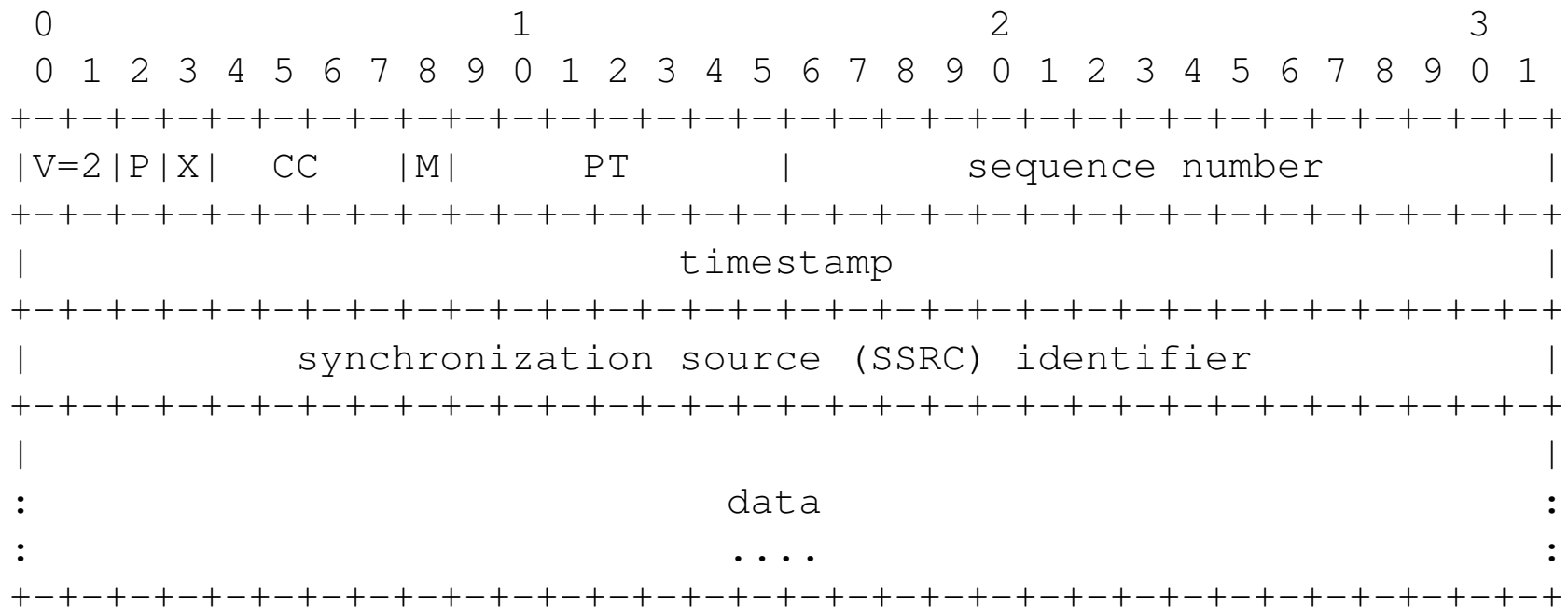
# Séparation données/contrôle

- RTP consiste en deux protocoles :
  - RTP pour l'acheminement de données
  - RTCP pour échanger les messages de contrôle
- Les différences avec TCP :
  - Chaque paquet TCP contient des champs de contrôle :
    - Acquittements, taille de la fenêtre, etc.
    - Solution adapté pour une boucle de contrôle étroite
  - RTP fonctionne en multipoint
  - RTP n'exige pas la fiabilité à 100%
- Deux numéros de ports voisins
  - Par exemple : données port 12040, contrôle port 12041

# Différentes profiles

- Une solution n'est pas adaptée à toutes les applications
  - Par exemple il existe plusieurs codecs audio
  - Chaque codec à son propre horodatage
  - Les codecs vidéo ont encore d'autres horodatages
- RTP (RFC 3550) fournit un cadre
- Les « profiles » (RFC 3551) fournissent les détails
  - Quelques profiles audio :
    - GSM, PCMA, G.722
  - Quelques profiles vidéo :
    - JPEG, H.261, MPEG 1 et MPEG 2

# Les paquets RTP



- V : version  
P : padding (at end of data)  
X : extension (after the header)  
CC : CSRC count (additional sources)  
M : marker (profile specific)  
PT : packet type

# A propos des paquets RTP

- Un format simple, principalement :
  - Identificateur de source (SSRC)
  - Type de paquet (PT)
  - Numéro de séquence
  - Estampille temporelle
  - Données
- Longueur et numéro de port dans l'en-tête UDP
- Peu de surcharge
  - Douze octets d'en-tête (par rapport à 20 pour TCP)
  - Possibilité d'ajouter des extensions

# SSRC : synchronization source identifier

- Identifiant d'un flux de paquets
  - A chaque SSRC correspond :
    - Une espace de numéros de séquence
    - Une espace temporelle
  - Une application peut avoir plusieurs SSRCs
    - Par exemple : un pour l'audio, un pour le vidéo
- Globalement unique
  - $2^{32} = 4,3 \times 10^9$  valeurs possibles
  - Choisi à l'aléatoire
  - Algorithme de détection de collisions

# Indépendance de la couche inférieure

- Le SSRC est indépendant de l'adresse machine
  - RTP fonctionne sur IPv4, IPv6, ou d'autres protocoles
- On sépare les couches réseau et transport

# PT : packet type

PT	encoding name	media type	clock rate (Hz)	channels
0	PCMU	A	8000	1
2	G726-32	A	8000	1
3	GSM	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G722	A	8000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	QCELP	A	8000	1
14	MPA	A	90000	?
15	G728	A	8000	1
16	DVI4	A	11025	1
17	DVI4	A	22050	1
18	G729	A	8000	1
25	CelB	V	90000	
26	JPEG	V	90000	
28	nv	V	90000	
31	H261	V	90000	
32	MPV	V	90000	
96-127	dynamic	?		



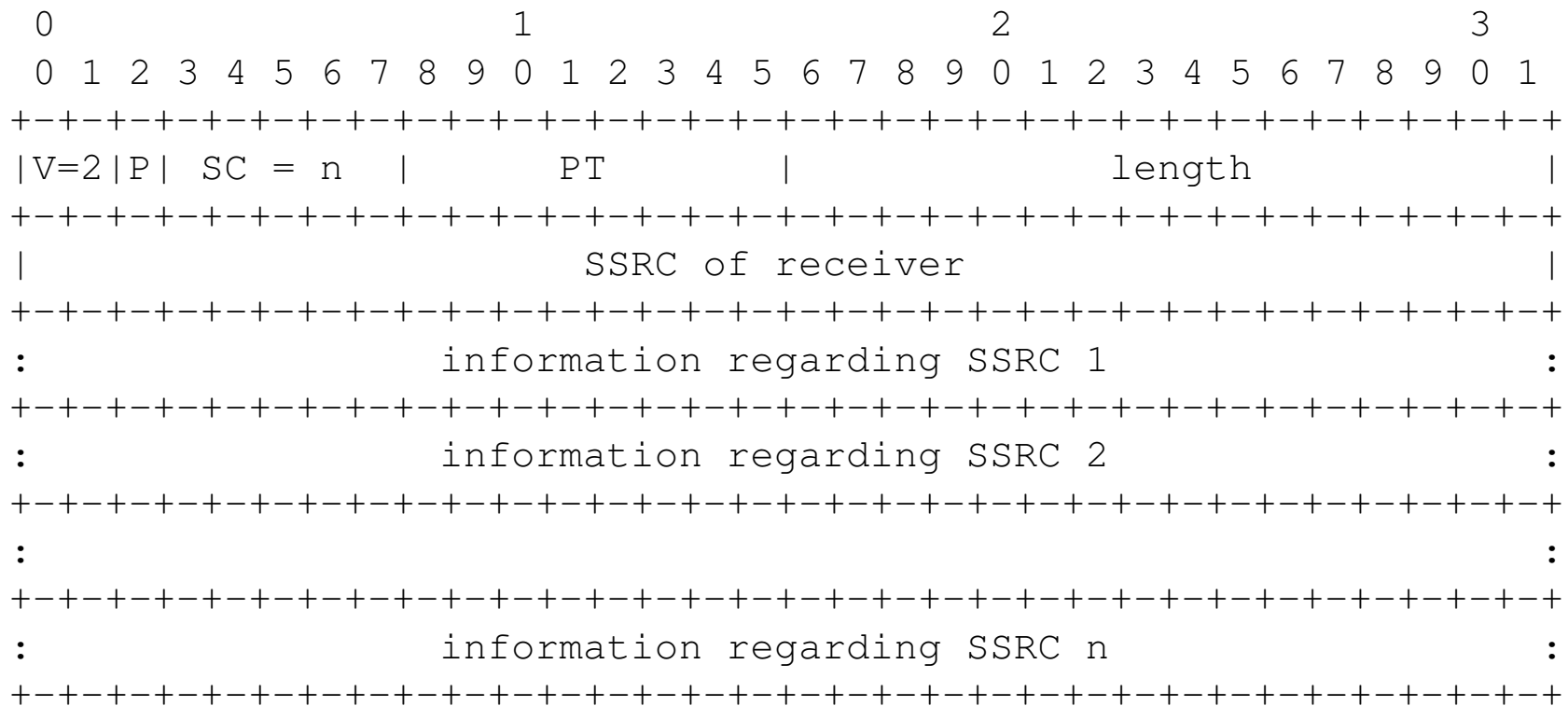
# Numéro de séquence

- Permet de reconstruire l'ordre de paquets
  - Mais aucun mécanisme de retransmission
  - Simplement un support pour l'application
- Deux octets :  $2^{16} = 65\ 536$  numéros possibles
  - Une espace de numéros par SSRC
  - Numéro initial choisi à l'aléatoire
    - Facilite la confidentialité par l'encryption
  - Augmente par 1 même si l'horloge n'avance pas

# Estampille temporelle

- Essentielle pour la lecture des paquets
  - Sert aussi à calculer la gigue
- Les unités sont dépendants de l'application
- Quatre octets :  $2^{32} = 4,3 \times 10^9$  valeurs possibles
  - Une espace de valeurs par SSRC
  - Valeur initiale choisie à l'aléatoire
    - Facilite la confidentialité par l'encryption
  - Augmente dans une manière régulière
    - N'augmente pas s'il s'agit d'une même trame

# Les paquets RTCP



- V : version
- P : padding (at end of data)
- SC : source count
- PT : packet type

# A propos des paquets RTCP

- Un cadre pour des rapports :
  - Identificateur de récepteur (SSRC)
  - Type de paquet (PT)
  - Longueur de paquet
    - Il peut y avoir plusieurs paquets par paquet UDP
  - Nombre de sources
    - Un rapport par source
- Plusieurs genres de rapport possibles
  - Receiver Report (RR), Sender Report (SR), autres

# Les rapports RR : Receiver Report

```

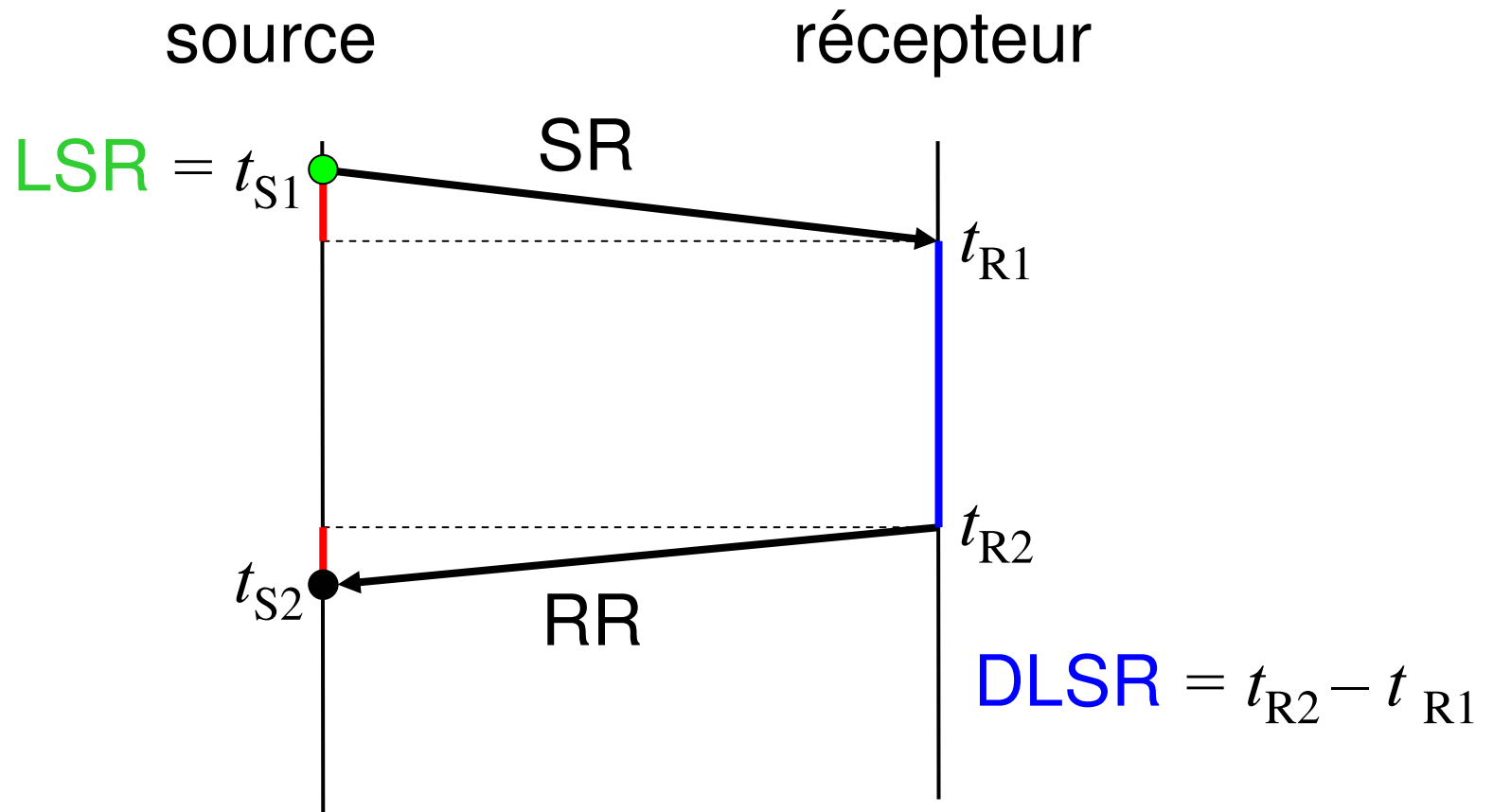
0                1                2                3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                SSRC of source                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| fraction lost |      cumulative number of packets lost      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|           extended highest sequence number received           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                interarrival jitter                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                last SR (LSR)                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                delay since last SR (DLSR)                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

PT : 201

# A propos des rapports RR

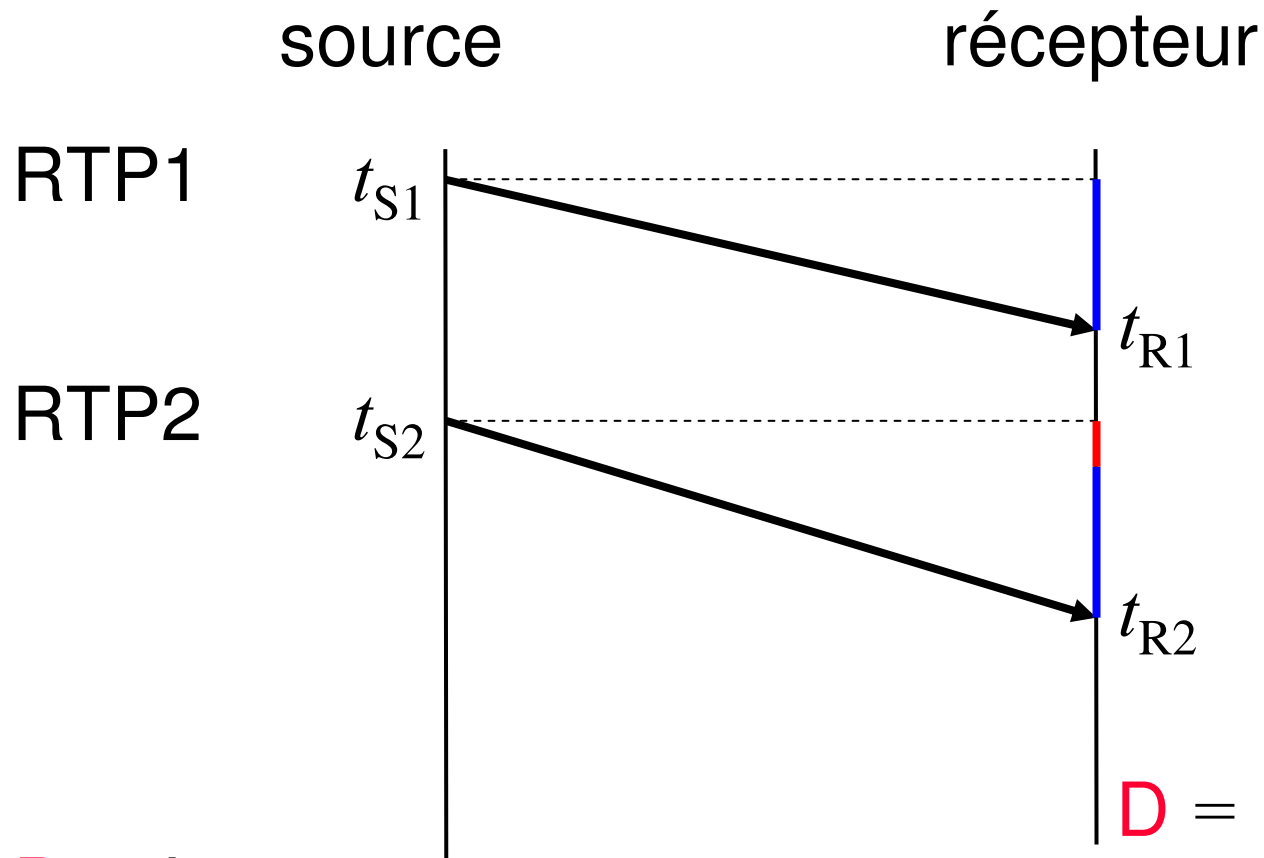
- Information sur chaque source (SSRC) :
  - Numéro de séquence
  - Pertes
    - Depuis le dernier rapport (en pourcentage)
    - Depuis le début (en nombre brut)
  - Information concernant le RTT
    - RTT = « round trip time »
    - Temps aller-retour depuis la source
    - Ne demande pas de réponse immédiat
  - Gigue

# Le calcul du RTT



$$RTT = (t_{S2} - LSR) - DLSR$$

# Le calcul de la gigue



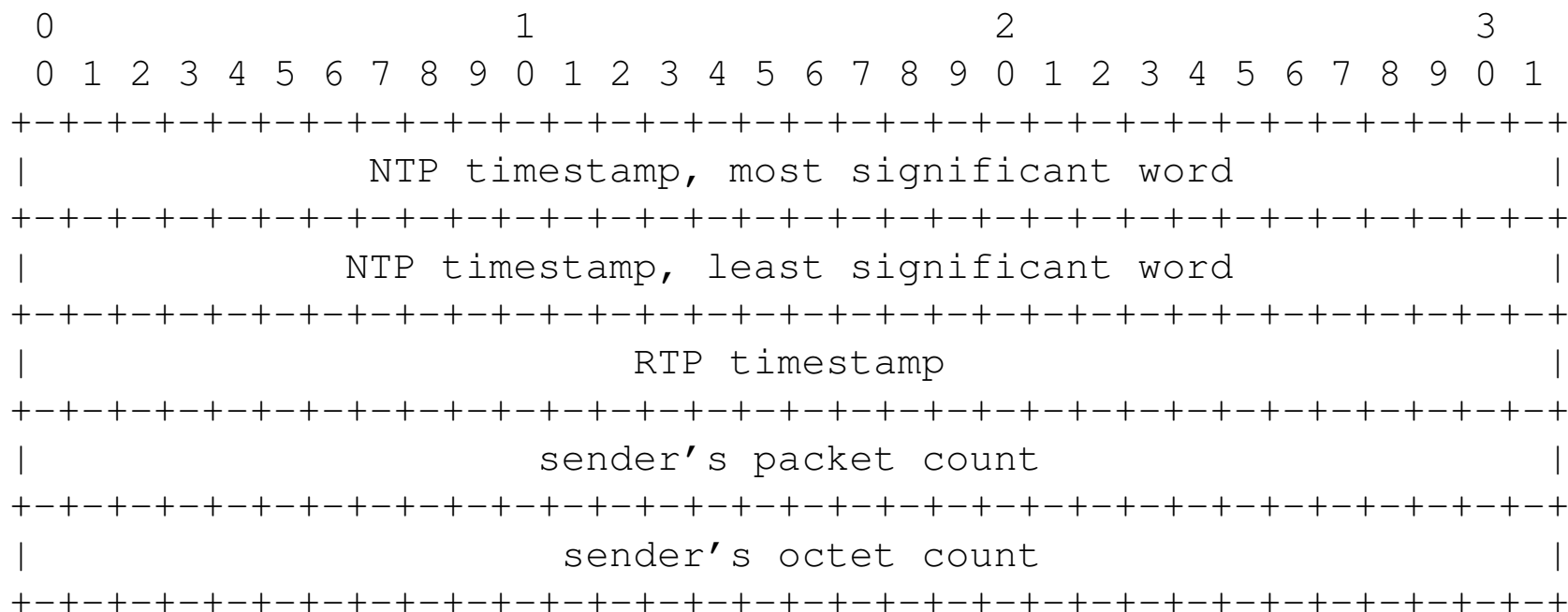
$$J' = J + (D - J)/16$$

gigue : déviation moyen de  $D$

$$D = | (t_{R2} - t_{S2}) - (t_{R1} - t_{S1}) |$$



# Les rapports SR : Sender Report



PT : 200

Cet information va entre l'en-tête et les rapports RR

# A propos des rapports SR

## ■ Estampilles temporelles NTP

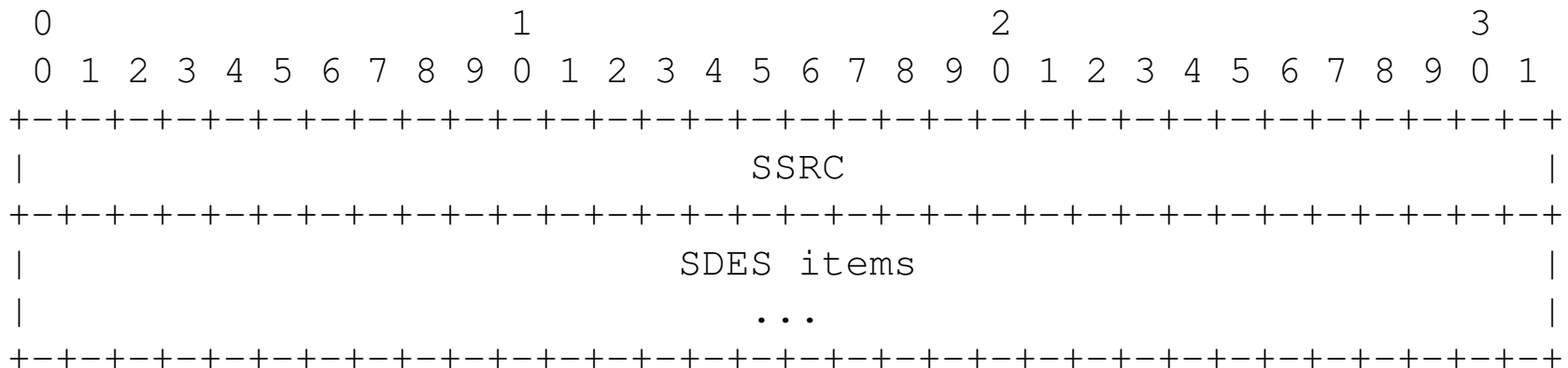
- Temps de l'horloge murale
- NTP = « Network Time Protocol »
- Secondes depuis 0h UTC le 1 janvier 1900
  - 32 premiers bits indiquent le nombre de secondes
  - 32 derniers bits indiquent la portion d'une seconde

## ■ Estampille temporelle RTP

- Le temps équivalent en unités de l'application

## ■ Nombre de paquets, d'octets depuis le début

# Les rapports SDES : Source Description



PT : 202

Information descriptif à propos de la source

# Exemples des rapports SDES

- CNAME : doe@sleepy.megacorp.com
  - Nom constant à travers des SSRCs
- NAME : John Doe, Bit Recycler, Megacorp
- EMAIL : John.Doe@megacorp.com
- PHONE : +1 908 555 1212
- LOC : Murray Hill, New Jersey
- TOOL : videotool 1.2
- NOTE : « ligne occupée »
- PRIV : usage privé

# Autres rapports RTCP

## ■ BYE

- Pour terminer une session RTP

## ■ APP

- Spécifique à l'application

## ■ XR (RFC 3611)

- Rapports détaillés de pertes et de délais
- Métriques VoIP

## ■ Proposé :

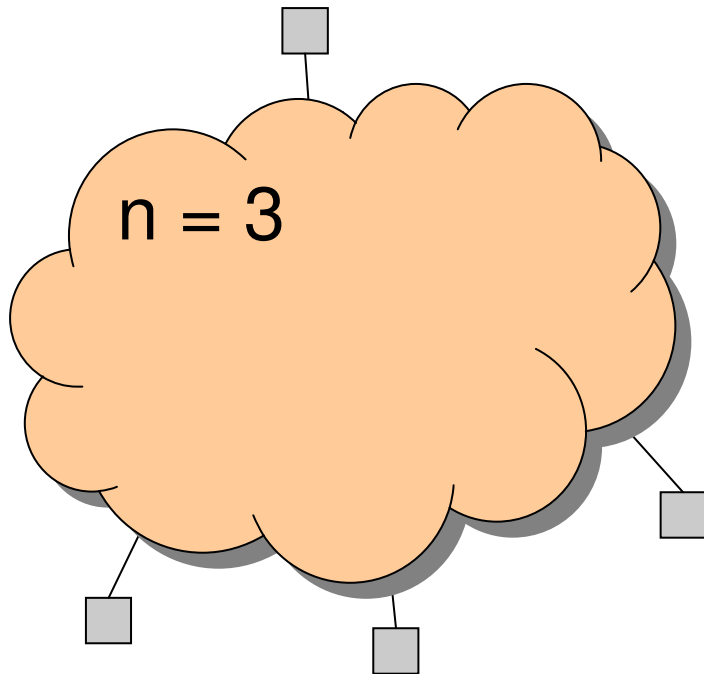
- Acquittements pour RTP en unicast

# Plan

- Introduction
- Communication en temps réel
- RTP
- Limitation de débit RTCP
- RTP et le temps réel

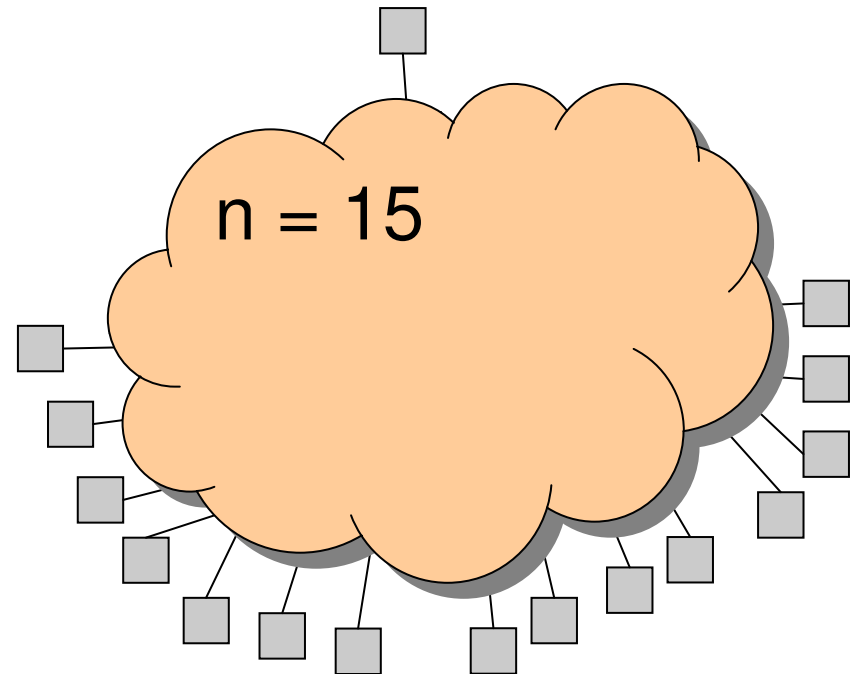
# Problème de résistance au facteur d'échelle

Débit RTP = 10 Kbps



Débit RTCP = 0,5 Kbps

Débit RTP = 10 Kbps



Débit RTCP = 2,5 Kbps ? **X**  
Débit RTCP = 0,5 Kbps

# Un algorithme distribué

## ■ Soit :

- $d$  = débit RTP (connu par tout le monde)

- $d'$  = débit RTCP =  $0,05 d$

- $d''$  = débit RTCP des récepteurs =  $0,75 d'$

- $n$  = nombre de récepteurs (estimé)

- $T$  = taille moyenne des paquets RTCP (estimée)

- $f$  = fréquence cible d'émission =  $d'' / nT$

## ■ Délai entre émission de paquets RTCP :

- Choisi à l'aléatoire entre  $0,5/f$  et  $1,5/f$



# Estimation du nombre de récepteurs

- Chaque participant compte les participants
  - Arrivé d'un SR : compte une source
  - Arrivé d'un RR : compte un récepteur
- S'il s'agit d'un nouveau participant
  - Il n'a pas encore eu le temps de compter
  - Il attend une intervalle minimale

# Plan

- Introduction
- Communication en temps réel
- RTP
- Limitation de débit RTCP
- RTP et le temps réel

# Rappel des besoins

- Horodatage des données
- Synchronisation des flux
- Résistance contre les pertes, duplicatas, mauvais ordre
- Identification de participants
- Surveillance de l'état de la connexion
- Contrôle de flux
- Contrôle de congestion
- Besoins avancés :
  - Support pour le transcodage de données
  - Sécurité de données

# Horodatage

- Estampille temporelle RTP
  - Spécifique à l'application
  - Pour la lecture des données

# Synchronisation des flux

- Estampille temporelle NTP
  - Temps de l'horloge murale
  - Coordination entre les estampilles d'applications

# Résistance contre les pertes, etc.

- Récolte d'informations :
  - Numéro de séquence RTP
  - Taux de pertes
  - Nombre de paquets perdus
- Pas de mécanisme intégré
  - A l'application de réagir
  - Chaque application à ses propres besoins
    - Exemple : redondance audio
    - Exemple : protection de trames I en MPEG

# Identification de participants

## ■ CNAME

- Exemple : doe@sleepy.megacorp.com

- Unique et constant à travers les flux

## ■ SSRC

- Numéro unique par participant par flux

## ■ Informations SDES

- Informations supplémentaires

# Surveillance de l'état de la connexion

## ■ Les RR

- Pertes depuis le dernier rapport
- Pertes depuis le début
- Information concernant le RTT
- Gigue

## ■ D'autres rapports

- Les XR : information détaillé sur pertes, délais, métriques VoIP



# Contrôle de flux

## ■ Données RTP

- Pas de mécanismes de contrôle de flux

## ■ Rapports RTCP

- Mécanisme de limitation du débit des rapports
- Typiquement à 5% du débit des données RTP

# Contrôle de congestion

- Pas de mécanisme dans l'RTP
- RTCP fourni des informations
  - Taux de pertes, par exemple
  - Ils peuvent être utilisés par une application

# Support pour le transcodage de données

- RTP permet de mélanger les flux
  - Plusieurs SSRCs attachés à un flux mélangé
- Les détails dépendent de l'application

# Sécurité de données

- RTP est compatible avec la sécurité
  - Numéro de séquence initialisé à une valeur aléatoire
  - Pareil pour l'estampille temporel
- L'encryption n'est pas encore dans le norme
  - Un sujet de travail actuel