

# Brève introduction à La Qualité de Service sur Internet

Chaput Emmanuel



2020-2021



Notes :

---

---

---

---

---

---

---

---

- 1 Introduction
- 2 Spécifications des besoins
- 3 Comment partager les ressources ?
- 4 Deux architectures pour le monde IP
- 5 Références bibliographiques

Notes :

---

---

---

---

---

---

---

---

## Introduction

- 1 Introduction
  - De quoi parle-t-on ?
  - Les difficultés
  - Les outils
  - Mécanismes, protocoles, architecture
  - Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## De quoi parle-t-on ?

### 1 Introduction

- De quoi parle-t-on ?
  - Les difficultés
  - Les outils
  - Mécanismes, protocoles, architecture
  - Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## Qu'est-ce que la qualité de service ?

- Notion de service
  - Ensemble de fonctionnalités offertes aux applications
  - Fournies par le réseau au travers d'une API
  - *eg* transfert de données, ...
- Qualité du service
  - Conditions permettant une mise en œuvre satisfaisante de l'application
  - Dans des conditions définies
  - *eg* délai, jigue, débit, ...

Notes :

---

---

---

---

---

---

---

---

## Remarque préliminaire

La qualité d'un service ne se mesure pas à ses seules performances. D'autres facteurs tout aussi importants (parfois plus, selon le contexte) doivent être pris en considération

- La sécurité
- La disponibilité
- La sûreté/la fiabilité
- ...

Ce cours n'a pas la prétention d'aborder ces sujets.

Chacun d'entre eux mérite des cours et des livres complets.

### Objectif de ce cours

Nous nous intéresserons ici aux outils permettant à un réseau (qualifié de prestataire/opérateur) de garantir un certain niveau de performance à une application (qualifiée de client ou utilisateur).

Notes :

---

---

---

---

---

---

---

---

## Qualité de service : définition

Pas de définition universelle

### Une tentative de définition

"A set of quality requirements on the collective behavior of one or more object."

### Une autre

"The set of those quantitative and qualitative characteristics of a distributed multimedia system, which are necessary in order to achieve the required functionality of an application." [16]

### Une dernière pour la route

Un ensemble de paramètres permettant d'évaluer les performances d'un service offert par un système et la satisfaction qu'en tirera l'utilisateur.

Notes :

---

---

---

---

---

---

---

---

## Les difficultés

### Introduction

- De quoi parle-t-on ?
- Les difficultés**
- Les outils
- Mécanismes, protocoles, architecture
- Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## Les difficultés

Premier niveau de difficultés, la *spécification* des contraintes/besoins.

- Comment caractériser le trafic engendré par une application ?
  - Incidence sur la charge du réseau, et donc sur les moyens à mettre en œuvre
- Comment caractériser les besoins en qualité de service ?
  - Besoin de définir des métriques modélisant les besoins applicatifs
  - Paramètres pertinents et pouvant être définis sans ambiguïté
  - Traduction de ces paramètres entre les différentes entités
- Comment caractériser la qualité du service rendu par le prestataire ?
- Comment négocier la qualité de service ?
  - Notion de contrat entre un prestataire et un client
  - Service Level Agreement* (SLA)

Notes :

---

---

---

---

---

---

---

---

## Les difficultés

Deuxième niveau de difficultés, la *mise en œuvre* de la qualité de service.

- Comment assurer le respect du contrat ?
  - Mécanismes d'acceptation (refus) d'un contrat (qui ne pourra être assumé)
    - *Call/Connexion Admission Control (CAC)*
  - Mécanismes de rejet éventuel du trafic du client qui ne respecte pas sa part du contrat
    - *Policing, shaping*
- Comment mettre en œuvre la qualité de service ?
  - Implantation de mécanismes dédiés
  - Positionnement des ces mécanismes dans les différentes entités impliquées
  - Coordination de ces entités entre elles

Notes :

---

---

---

---

---

---

---

---

## Un exemple trivial : la téléphonie

- Sur le réseau téléphonique commuté
  - Appel ↔ signalisation
  - Appel accepté ou refusé
  - Circuit mis en place ↔ ressources réservées
  - Qualité (laquelle ?) garantie
- Sur un réseau paquet ( VoIP)
  - Signalisation ? Pas nativement dans IP
  - Possibilité d'accepter/refuser ? Pas nativement dans IP
  - Circuit virtuel mis en place ? Pas dans IP
  - Quelle garantie ? Jigue, pertes, ...

Notes :

---

---

---

---

---

---

---

---

## Les outils

### 1 Introduction

- De quoi parle-t-on ?
- Les difficultés
- Les outils
  - Mécanismes, protocoles, architecture
  - Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## Quels outils ?

Sur quels équipements ?

- Les machines d'extrémité (clients, serveurs)
  - Sources du trafic
  - Clients du service  $\rightsquigarrow$  exigences
- Les éléments actifs de réseau (routeurs)
  - Mettent en œuvre le service
  - Garants de la qualité



Notes :

---

---

---

---

---

---

---

---

## Quels outils ?

Quelle échelle de temps ?

- Préventifs
  - Acceptation/refus de connexion (CAC)
  - Réserve de ressources
  - ...
- Réactifs
  - Changements de routes b
  - Adaptation des applications
  - ...
- Curatifs
  - Altération de la qualité
  - Rejet/déclassement de flux moins prioritaires
  - ...

Notes :

---

---

---

---

---

---

---

---

## Mécanismes, protocoles, architecture

### 1 Introduction

- De quoi parle-t-on ?
- Les difficultés
- Les outils
- Mécanismes, protocoles, architecture
- Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## Les différents axes de la QoS

### Des architectures

- Quelle structure ?
- Comment s'organise-t-on ?

### Des protocoles

- Quelle communication entre les entités ?
- Comment dialogue-t-on ?

### Des mécanismes

- Quelle mise en œuvre ?
- Comment fait-on ?

Notes :

---

---

---

---

---

---

---

---

## Des architectures

### Répartition des responsabilités

- Qui décide ?
- Qui implante ?

### Interfaces entre les entités

- Ajout de primitives de service supplémentaires
- Nécessité de décrire les paramètres

### Répartition des entités sur le réseau

- Où sont les entités de décision, de mise en œuvre ?

Notes :

---

---

---

---

---

---

---

---

## Des protocoles

### Plan de contrôle

- Routage
- Description de ressources
- Description de politiques (d'admission, ...)
- Contrôle de congestion

### Plan de données

- Transport
- Fiabilisation
- Estampillage
- Marquage
- Contrôle de congestion

Notes :

---

---

---

---

---

---

---

---

## Des mécanismes

- Réserve de ressources
  - Afin de pouvoir garantir un service
- Classification
  - Des paquets en fonction de spécifications de trafic
- Mesure
  - Afin de vérifier la conformité au contrat
- Conditionnement
  - Afin d'assurer la conformité au contrat
- Contrôle de congestion
  - Afin de gérer le comportement du réseau

Notes :

---

---

---

---

---

---

---

---

## Mais que fait IP ! ?



### Introduction

- De quoi parle-t-on ?
- Les difficultés
- Les outils
- Mécanismes, protocoles, architecture
- Mais que fait IP ! ?

Notes :

---

---

---

---

---

---

---

---

## L'état des lieux

- IP est non connecté
  - Pas de possibilité de négocier, accepter, refuser un appel
  - Pas d'état sur les routeurs
  - Pas de ressources associées
- Routage *hop by hop*
  - Pas de cohérence garantie dans le chemin suivi par un paquet
  - Tables de routage asynchrones
- On parle de *best effort*
  - Le réseau fait de son mieux
  - Comprendre : il ne garanti rien !
  - Il traite tous les paquets de la même façon, ...

Notes :

---

---

---

---

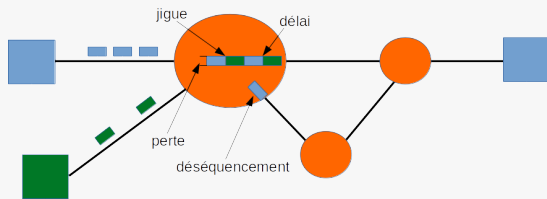
---

---

---

---

## Une situation désespérée ?



- Files d'attente FCFS, *droptail*
- Pas de connaissance des communications
- Pas de *feedback* aux applications
- Congestion subie (aucun levier pour les routeurs)

Notes :

---

---

---

---

---

---

---

---

## Des espoirs ?

- TCP fait du contrôle de congestion
  - Objectif : maintenir les files d'attente stables et vides
  - Observation du comportement du réseau
  - Moduler son trafic en conséquence
- Gestion active des files d'attente
  - Objectif : anticiper la congestion plutôt que la subir
  - Observer les files d'attente
  - Recommandé depuis 2015 [1]

~> Ce n'est pas de la qualité de service, mais cela va dans le bon sens

Notes :

---

---

---

---

---

---

---

---

## Spécifications des besoins

- ② Spécifications des besoins
  - Introduction
  - Un exemple : la modélisation du débit

Notes :

---

---

---

---

---

---

---

---



## Introduction

### 2 Spécifications des besoins

- Introduction
- Un exemple : la modélisation du débit

Notes :

---

---

---

---

---

---

---

---

## Les caractéristiques et besoins applicatifs

### Historiquement

- Téléphonie
  - Temps de traversée faible et constant
  - Débit constant
  - Erreurs acceptables
  - Utilisation sporadique
- Télévision
  - Temps de traversée quelconque mais constant
  - Diffusion à débit (constant) élevé
  - Erreurs acceptables
  - Transmission permanente
- Informatique
  - Temps de traversée "quelconque"
  - Débits variables
  - Erreurs inacceptables
  - Utilisation sporadique

Notes :

---

---

---

---

---

---

---

---

## Les caractéristiques et besoins applicatifs

Convergence des services (voix, télévision, accès Internet sur le même équipement)

- Comment traiter les flots de données ?
  - Traitement individuel
    - Réaliste à l'échelle de l'Internet ?
  - Traitement par groupe
    - Quelle efficacité pour chaque flot ?
- Comment modéliser le trafic ?
  - Certains trafics sont très difficiles à modéliser
  - Le trafic global est trop complexe
    - Métrologie
- Comment dimensionner le réseau ?
  - Plus complexe encore que les réseaux dédiés
    - Voir par exemple ATM
  - Approche plus réactive
    - Ingénierie de trafic

Notes :

---

---

---

---

---

---

---

---

## Quelles métriques ?

- Débit
  - Débit utilisable pour un trafic
  - Quelle mesure (échelle de temps) ?
  - Débit moyen, burst, ...
- Taux de perte
  - Nombre moyen max de pertes
  - Quelle distribution ?
- Gigue
  - Variation du délai
  - Important pour les applications "temps-réel"
- ...
  - Satisfaction de l'utilisateur
  - Paramètre difficile à traduire entre couches

Notes :

---

---

---

---

---

---

---

---

## Différents types de métriques

- Comment définir les métriques ?
  - Ce qui a une signification applicative n'en a pas toujours au niveau réseau (ou transport)
- Difficulté d'évaluer les métriques
  - Fenêtre glissante, moyenne mobile, ...
- Divers comportements sur un chemin
  - Métrique additive
    - $M_{r_1, r_2} = M_{r_1} + M_{r_2}$
    - *pe* délai, gigue
  - Métrique concave
    - $M_{r_1, r_2} = \min(M_{r_1}, M_{r_2})$
    - *pe* débit
  - Métrique multiplicative
    - $M_{r_1, r_2} = M_{r_1} \times M_{r_2}$
    - *pe* probabilité de non perte

Notes :

---

---

---

---

---

---

---

---

## Un exemple : la modélisation du débit

### 2 Spécifications des besoins

#### • Introduction

- Un exemple : la modélisation du débit
  - Le problème
  - Un exemple de mise en œuvre : le token bucket
  - Un autre exemple de mise en œuvre : le GCRA
  - Bursts instantanés ?
  - Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

---

---

---

## Le problème

### 2 Spécifications des besoins

- Un exemple : la modélisation du débit
  - Le problème
    - Un exemple de mise en œuvre : le token bucket
    - Un autre exemple de mise en œuvre : le GCR
    - Bursts instantanés ?
    - Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

---

---

---

## Le problème

- Qu'est-ce qu'un débit ?
  - Un volume d'information par unité de temps
- Comment le définir dans le cadre d'un contrat de QoS ?
  - Quelle échelle de temps ?
  - Quelle variation acceptable ?

### Un exemple : 64 Kbit/s

- Pour un service téléphonique
  - Un échantillon de 8 bits toutes les  $125 \mu s$
  - Pendant toute la durée de la communication et "à chaque instant"
- Pour un service de transfert de fichier
  - Un fichier de 1 Mo transféré en 128 secondes
  - Durée moyenne et rythme éventuellement variable

Notes :

---

---

---

---

---

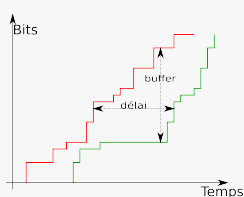
---

---

---

## Définition microscopique

- Observons ce qui se passe sur un équipement (un routeur) à l'échelle temporelle d'un paquet
  - Chaque instant d'arrivée
  - Chaque instant de départ
- La "pente" de chaque courbe est un débit
  - Localement "infini" (égale au débit du support)
- Notion de *courbe de trafic*
  - $a(t)$  est le nombre de bits cumulés à l'instant  $t$



Notes :

---

---

---

---

---

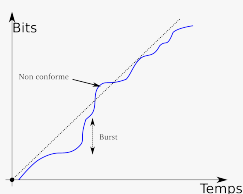
---

---

---

## A une échelle "mésoscopique"

- Vérifier qu'un trafic respecte un débit
  - la courbe de trafic  $a(t)$  est-elle sous la droite (débit constant)  $y = r.t$ ?
- Simple à implanter
- Efficace en moyenne sur du long terme
- Et à court terme ?
  - N'empêche pas les salves (*burst*)



Notes :

---

---

---

---

---

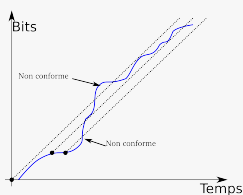
---

---

---

## Translation

- Idée
  - Repartir de  $(0, 0)$  à chaque instant significatif
  - La demi-droite du débit "glisse" le long de la courbe
- On oublie le passé
  - $\forall t_0 \geq 0, \forall t \geq t_0, a(t) < a(t_0) + r.(t - t_0)$
- Notion d'*enveloppe de trafic*



Notes :

---

---

---

---

---

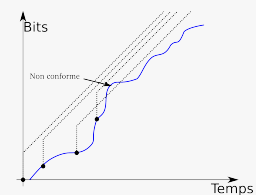
---

---

---

## Introduction des bursts

- Trop contraignant (bursts impossibles)
  - Ne permet pas de "rattraper le retard"
  - Le débit ne peut que décroître
- On autorise des bursts d'amplitude  $b$ 
  - $\forall t_0 \geq 0, \forall t \geq t_0, a(t) < a(t_0) + b + r.(t - t_0)$
- Comment implanter ?



Notes :

---

---

---

---

---

---

---

---

## Un exemple de mise en œuvre : le token bucket

### 2 Spécifications des besoins

- Un exemple : la modélisation du débit
  - Le problème
  - Un exemple de mise en œuvre : le token bucket
  - Un autre exemple de mise en œuvre : le GCR
  - Bursts instantanés ?
  - Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

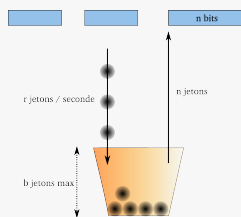
---

---

---

## Un exemple de mise en œuvre : le token bucket

- Principe du seau à jetons
  - Un seau de capacité  $b$  tokens
  - est rempli au rythme permanent de  $r$  token par seconde
- Un paquet de  $n$  bits est conforme si
  - Le seau contient au moins  $n$  tokens
  - Les  $n$  tokens sont alors supprimés du seau
- Sinon, selon la stratégie et les ressources, le paquet est
  - Détruit
  - Mis en attente
  - Marqué



Notes :

---

---

---

---

---

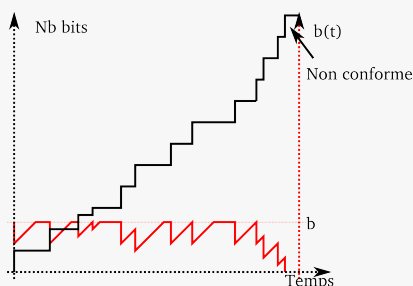
---

---

---

## Un exemple de mise en œuvre : le token bucket

- Exemple d'application du *token bucket* à une courbe de trafic
- À chaque paquet, on décroît  $b(t)$  de la taille du paquet
- $b(t)$  croît linéairement avec  $t$  (coefficient  $r$ )
- $\forall t, b(t) \leq b$



Notes :

---

---

---

---

---

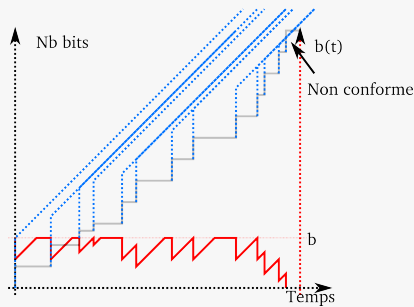
---

---

---

## Un exemple de mise en œuvre : le token bucket

- Superposons en chaque temps  $t_0$  une courbe de la forme  $a(t_0) + b(t) + c.(t - t_0)$
- Attention, ici, on utilise  $b(t)$  pour un effet mémoire
- Ainsi le *token bucket* déclare le dernier paquet non conforme



Notes :

---

---

---

---

---

---

---

---

---

---

## Un autre exemple de mise en œuvre : le GCRA

### 2 Spécifications des besoins

- Un exemple : la modélisation du débit
  - Le problème
  - Un exemple de mise en œuvre : le token bucket
  - Un autre exemple de mise en œuvre : le GCRA
  - Bursts instantanés ?
  - Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

---

---

---

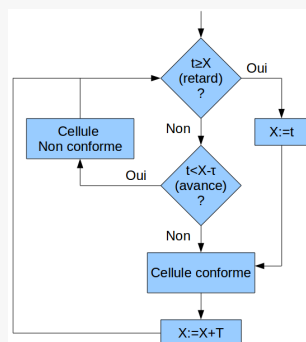
---

---

## L'algorithme GCRA

Utilisé dans ATM

- Algorithme  $GCRA(T, \tau)$
- $T$  est la durée entre deux cellules ATM
- $\tau$  est une tolérance sur  $T$
- Algorithme appliqué à chaque cellule
  - $t$  date d'arrivée réelle de la cellule
  - $X$  est la date prévue
  - Initialement  $X = t_0$  (date d'arrivée de la première cellule)



Notes :

---

---

---

---

---

---

---

---

---

---

## Bursts instantanés ?

### 2 Spécifications des besoins

- Un exemple : la modélisation du débit
  - Le problème
  - Un exemple de mise en œuvre : le token bucket
  - Un autre exemple de mise en œuvre : le GCRA
- **Bursts instantanés ?**
  - Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

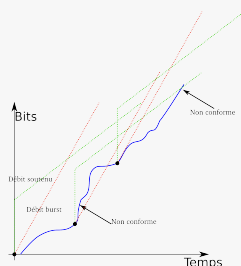
---

---

---

## Bursts instantanés ?

- Problème avec les outils précédents
  - Le débit des bursts n'est pas limité
  - Il faut donc les limiter en amplitude
- Autre possibilité : utiliser deux jeux de paramètres
  - Un pour le débit moyen (*sustainable rate*)
  - Un pour le débit crête (*peak rate*)



Notes :

---

---

---

---

---

---

---

---

## Un exemple : le TSpec de Intserv

### 2 Spécifications des besoins

- Un exemple : la modélisation du débit
  - Le problème
  - Un exemple de mise en œuvre : le token bucket
  - Un autre exemple de mise en œuvre : le GCRA
  - Burst instantanés ?
- Un exemple : le TSpec de Intserv

Notes :

---

---

---

---

---

---

---

---

## Un exemple : le TSpec de Intserv

Dans l'architecture IntServ, un trafic est caractérisé par la structure suivante

```
struct {  
    float r;  
    float b;  
    float p;  
    unsigned m;  
    unsigned M;  
} TOKEN_BUCKET_TSPEC;
```

- r** est le débit
- b** est la taille de burst
- p** est le débit crête (*peak rate*)
- m** est la taille min (un paquet coûte au moins m)
- M** est la taille max (un paquet plus gros est non conforme)

Notes :

---

---

---

---

---

---

---

---

## Comment partager les ressources ?

- ③ Comment partager les ressources ?
  - Maximiser le débit ?
  - Pourquoi pas un peu d'équité ?

Notes :

---

---

---

---

---

---

---

---

## Maximiser le débit ?

- ③ Comment partager les ressources ?
  - Maximiser le débit ?
  - Pourquoi pas un peu d'équité ?

Notes :

---

---

---

---

---

---

---

---



## Maximiser le débit ?

- Un lien offre un débit maximal lié à ses caractéristiques
  - Modulation, codage, ...
- Un objectif naturel est de tenter de l'utiliser au maximum
  - Transmettre dès qu'une trame est prête à être émise
  - Au travers des meilleures conditions de transmission
- C'est un choix dans le comportement de l'ordonnanceur
- Un tel comportement peut être très inéquitable
  - En particulier pour des clients dans de mauvaises conditions de transmission
- Une forme d'équité peut être souhaitable
  - Comment la mesurer ?

Notes :

---

---

---

---

---

---

---

---

## Pourquoi pas un peu d'équité ?

- ③ Comment partager les ressources ?
  - Maximiser le débit ?
  - Pourquoi pas un peu d'équité ?

Notes :

---

---

---

---

---

---

---

---

## Que signifie être équitable ?

- Distribution égale des ressources ?
  - Pas les mêmes conséquences pour chaque client
  - Voir l'exemple de l'ACM
- Fourniture de performances égales ?
  - Pas le même coût pour chaque client
  - Même raison, ...
- Équité ne veut pas dire égalité !
  - Tout le monde ne souhaite pas nécessairement la même qualité de service
  - Différents niveaux de QoS = différents tarifs
- Comment gérer le manque de ressource ?
  - Priver tout le monde dans les mêmes proportions ?
  - Satisfaire les moins exigeants ?
  - Satisfaire les plus chers ?
- Que faire du surplus en cas d'excès ?
  - Le répartir "équitablement" ?
  - ...
- Besoin d'outils de "mesure de l'équité"
  - Définition d'une forme d'équité

Notes :

---

---

---

---

---

---

---

---

## Max-min fairness

### Un odonancement est dit *max-min fair* . . .

. . . si on ne peut pas offrir plus de débit à un client sans en donner moins à un autre client qui n'était pourtant pas mieux servi.

- Le *round robin* à taille de paquet constante est *max-min fair*
  - Tout accroissement de débit d'un client se fait au détriment d'un autre, qui n'est pas mieux servi
- Si les paquets ne sont pas de taille constante, ce n'est plus le cas
- Proposition de techniques dites *Fair Queuing*, par exemple
  - On partage équitablement
  - On enlève le surplus de ceux qui ont trop
  - On le répartit équitablement entre les autres

Notes :

---

---

---

---

---

---

---

---

## Proportional fairness

- Pourquoi ne pas chercher un compromis ?
  - Essayer de bien utiliser le débit disponible
  - Fournir un service minimal à tous les utilisateurs
- Associer à chaque client une priorité
  - Inversement proportionnelle à son coût
  - Par exemple un client dans de mauvaises conditions "coûte cher"
- Assez largement décliné dans la littérature [11]

Notes :

---

---

---

---

---

---

---

---

## Le critère de Jain

- Recherche d'un critère [10]
  - Indépendant de la taille de la population
  - Indépendant de l'échelle des mesures
  - Compris dans [0, 1]
  - Continu

- Définition du critère suivant

$$J(x_1, \dots, x_n) = \frac{(\sum_{k=1}^n x_i)^2}{n \times \sum_{k=1}^n x_i^2}$$

- Permet d'évaluer une forme d'équité dans le comportement d'un algorithme

Notes :

---

---

---

---

---

---

---

---

# Deux architectures pour le monde IP

- 4 Deux architectures pour le monde IP
  - L'architecture *IntServ*
  - L'architecture *DiffServ*

Notes :

---

---

---

---

---

---

---

---

# L'architecture *IntServ*

- 4 Deux architectures pour le monde IP
  - L'architecture *IntServ*
    - Présentation générale
    - Les éléments de base
    - L'architecture
    - Les classes de service
    - Les paramètres de QoS
    - Une architecture réaliste ?
  - L'architecture *DiffServ*

Notes :

---

---

---

---

---

---

---

---

# Présentation générale

- 4 Deux architectures pour le monde IP
  - L'architecture *IntServ*
    - Présentation générale
    - Les éléments de base
    - L'architecture
    - Les classes de service
    - Les paramètres de QoS
    - Une architecture réaliste ?

Notes :

---

---

---

---

---

---

---

---

## L'architecture *IntServ*

- Groupe de travail de l'IETF [8]
- Deuxième moitié des années 90
- Constat : possibilité d'intégrer des services sur un réseau paquet
  - Audio
  - Vidéo
  - Temps-réel
  - Données
- Définition des interfaces et signalisation en ce sens

Notes :

---

---

---

---

---

---

---

---

## Les objectifs *IntServ*

- Garantir des paramètres de QoS par flot
  - Un flot = une communication
  - Garanties fermes de bout en bout
- Pour cela, définition et spécification
  - D'une architecture générale [3]
  - D'outils de spécification des paramètres de trafic ( $T_{Spec}$ ) et des paramètres de QoS ( $R_{SPEC}$ )
  - D'un protocole de signalisation (RSVP [4])
  - De comportements associés aux classes de service proposées [18, 13]

Notes :

---

---

---

---

---

---

---

---

## Les objectifs du groupe de travail

Trois grands axes

### Définition des services

Définir et documenter un modèle de service fourni aux applications

### Définition des interfaces

Expression des besoins applicatifs, des informations fournies aux routeurs et des contraintes sur le réseau local.

### Validation des routeurs

Définir des tests (comportementaux) permettant de valider la capacité d'un routeur à supporter le modèle.

Notes :

---

---

---

---

---

---

---

---

## Les éléments de base

### ✦ Deux architectures pour le monde IP

- L'architecture *IntServ*
  - Présentation générale
  - Les éléments de base
  - L'architecture
  - Les classes de service
  - Les paramètres de QoS
  - Une architecture réaliste ?

Notes :

---

---

---

---

---

---

---

---

## Les éléments de base

### Notion de flux de données

Ensemble des paquets entre une source (IP/port) et une destination.

### Réservation de ressources

Réquisition des moyens nécessaires à la garantie d'un service voulu sur un flux déterminé.

Notes :

---

---

---

---

---

---

---

---

## Les classes de trafic *IntServ*

Trois grandes classes de trafic

- Guaranteed delay [13]
  - Garantie de délai de bout en bout
  - Garantie de débit
  - Garantie de (non) pertes
- Controlled load [18]
  - Équivalent à un best-effort sur un réseau peu chargé
- Best effort
  - Traditionnel

Notes :

---

---

---

---

---

---

---

---

## L'architecture

### Deux architectures pour le monde IP

#### ● L'architecture *IntServ*

- Présentation générale
- Les éléments de base
- L'architecture
- Les classes de service
- Les paramètres de QoS
- Une architecture réaliste ?

Notes :

---

---

---

---

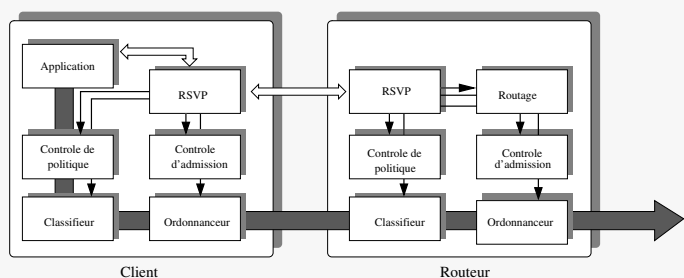
---

---

---

---

## L'architecture *IntServ*



Notes :

---

---

---

---

---

---

---

---

## Les classes de service

### Deux architectures pour le monde IP

#### ● L'architecture *IntServ*

- Présentation générale
- Les éléments de base
- L'architecture
- Les classes de service
- Les paramètres de QoS
- Une architecture réaliste ?

Notes :

---

---

---

---

---

---

---

---

## La classe de trafic *Guaranteed*

- Ce qu'elle garanti aux applications [13]
  - Un délai borné et aucune perte due à la congestion pour les paquets conformes
- Pour cela, bien sûr, l'application s'engage à respecter son  $TS_{spec}$ 
  - Notamment en termes de taille de paquet (pas de fragmentation)
- Chaque routeur doit donc s'assurer qu'il introduira un délai borné
  - Défini par un modèle *token bucket*

Notes :

---

---

---

---

---

---

---

---

## La classe de trafic *Controlled load*

- Ce qu'elle "garanti" aux applications [18]
  - Une grande proportion des paquets seront transmis (un taux d'erreur proche de celui des supports)
  - La plupart des paquets subiront un temps de transfert ne dépassant pas de beaucoup le temps le plus faible
- Pour cela, bien sûr, l'application s'engage à respecter son  $TS_{spec}$ 
  - Les paquets hors profil seront traités différemment
  - Par exemple au même titre que le trafic "*best effort*"
- Les routeurs doivent donc accepter ou non les communications en fonction
  - Des  $TP_{sec}$  des nouvelles communications
  - Des ressources disponibles
  - Des  $TS_{spec}$  ou du comportement observé des communications en cours

Notes :

---

---

---

---

---

---

---

---

## Les paramètres de QoS

- ④ Deux architectures pour le monde IP
  - L'architecture *IntServ*
    - Présentation générale
    - Les éléments de base
    - L'architecture
    - Les classes de service
    - Les paramètres de QoS
    - Une architecture réaliste ?

Notes :

---

---

---

---

---

---

---

---

## Les spécifications IntServ

Deux ensembles de paramètres clairement identifiés

- La spécification des trafics (TSPEC)
  - Défini ce à quoi s'engage le client (l'application)
  - Outil commun à toutes les classes de trafic
- La spécification des besoins (RSPEC)
  - Défini ce que demande le client
  - Spécifiques à chaque classe de trafic

Notes :

---

---

---

---

---

---

---

---

## La spécification des trafics

- Les caractéristiques de trafic (TSPEC) [14, 15]
  - Token bucket  $r, b$
  - Débit crête  $p$
  - Taille maximale de paquet  $M$
  - Taille minimale de paquet  $m$  (tout paquet est traité comme si sa taille était au moins  $m$ )
- Les unités sont l'octet et la seconde
- Valeurs extrêmes très larges
- Outil commun à toutes les classes de trafic

Notes :

---

---

---

---

---

---

---

---

## La spécification de la classe de trafic *Guaranteed*

- Les exigences de QoS sont exprimées au travers de deux paramètres
  - Un débit maximal  $R \geq r$
  - Une différence de délai  $S$  entre le délai acceptable et celui fourni à un débit  $R$
- Chaque routeur doit alors
  - Vérifier la disponibilité de ses ressources
  - Évaluer le retard (maximal) qu'il va induire
    - Une partie constante évaluant un pire cas ( $C$ )
    - Une partie fonction du débit ( $D$ )
    - Il s'agit en fait d'une marge d'erreur par rapport à un modèle fluide
- Le destinataire reçoit la somme des paramètres  $C$  et  $D$  et peut évaluer le délai maximal de bout en bout

Notes :

---

---

---

---

---

---

---

---



## Une architecture réaliste ?

- 4 Deux architectures pour le monde IP
  - L'architecture *IntServ*
    - Présentation générale
    - Les éléments de base
    - L'architecture
    - Les classes de service
    - Les paramètres de QoS
    - Une architecture réaliste ?

Notes :

---

---

---

---

---

---

---

---

## Les difficultés de *IntServ*

- Traitement flot par flot
  - États transitoires (*soft state*)
  - Signalisation
  - Files d'attente individuelles
- Traitement reposant sur les routeurs
  - Cœur du réseau
  - Difficultés de déploiement

### Problème majeur

- Quelle capacité de passage à l'échelle ?

Notes :

---

---

---

---

---

---

---

---

## L'architecture *DiffServ*

- 1 Introduction
- 2 Spécifications des besoins
- 3 Comment partager les ressources ?
- 4 Deux architectures pour le monde IP
  - L'architecture *IntServ*
  - L'architecture *DiffServ*
    - Les principes
    - Les éléments de base
    - Les classes de trafic
    - Les domaines
    - L'allocation des ressources
    - Les fonctions
    - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## Les principes

### 4 Deux architectures pour le monde IP

- L'architecture *DiffServ*
  - Les principes
    - Les éléments de base
    - Les classes de trafic
    - Les domaines
    - L'allocation des ressources
    - Les fonctions
    - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## L'approche *DiffServ*

Groupe de travail de l'IETF [7] [12] [2]

### Objectifs

- Architecture à QoS supportant les facteurs d'échelle
- Gestion de l'hétérogénéité
- Définir des blocs de base garantissant une architecture évolutive
- Contrainte : garder une architecture simple

### Idées de base

- Différenciation des services
  - Qualité de service assurée sur la base de *classes de service*, pas de flux individuels
- Hiérarchisation des fonctions des routeurs
  - Pas de traitement lourd dans le cœur du réseau

Notes :

---

---

---

---

---

---

---

---

## Les principes de *DiffServ*

### Mise en œuvre des idées de base

- Les routeurs de frontière réalisent la classification (éventuellement complexe) et le conditionnement du trafic
- Les routeurs de cœur traitent un nombre limité de classes de trafic

### Gestion de l'hétérogénéité

- Notion de domaine
  - Chaque domaine (réseau) peut appliquer les idées de base "indépendamment" de ses voisins
- Possibilité de reclassification
  - La classification peut être refaite à chaque entrée dans un nouveau domaine

Notes :

---

---

---

---

---

---

---

---

## DiffServ et les facteurs d'échelle

Modération de l'impact des facteurs d'échelle.

### Notion de classe de service

- Ensemble limité de besoins

### Per Hop Behavior

- Comportement global par classe de service

Notes :

---

---

---

---

---

---

---

---

## Les éléments de base

### 4 Deux architectures pour le monde IP

- L'architecture *DiffServ*
  - Les principes
  - Les éléments de base
  - Les classes de trafic
  - Les domaines
  - L'allocation des ressources
  - Les fonctions
  - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## Les classes de trafic

### Constitution par marquage

- A l'entrée du domaine (réseau)
- Dans le champ DSCP (ex-TOS d'IP)

### Traitement par classe

- Traitement commun au sein d'une classe
- Traitement simple et rapide
- Un DSCP implique un PHB

Notes :

---

---

---

---

---

---

---

---

## Les per hop behaviors

### Définition de quelques PHBS

- Expedited Forwarding
  - Ou premium
- Assured Forwarding
  - Plusieurs sous-classes
- Best Effort
  - Toujours présent (*default* PHB)

Notes :

---

---

---

---

---

---

---

---

## Les classes de trafic

### 4 Deux architectures pour le monde IP

- L'architecture *DiffServ*
  - Les principes
  - Les éléments de base
  - Les classes de trafic
  - Les domaines
  - L'allocation des ressources
  - Les fonctions
  - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## La classe EF

### La classe *Expedited Forwarding* [9] [5]

- Service garanti indépendamment du trafic non EF
  - Débit garanti
  - Délai, gigue, taux de perte faibles
- File d'attente spécifique
- Applications temporellement contraintes
- Intuitivement, le trafic EF doit toujours obtenir au moins le débit  $R$  prédéfini
- Concrètement, un paquet ne doit pas subir de retard supplémentaire à celui obtenu au débit  $R$  supérieur à une constante

Notes :

---

---

---

---

---

---

---

---

## La classe AF

### La classe *Assured Forwarding*

- Garantie d'un taux de perte maximal
- Plusieurs classes  $AF_i$ 
  - Débit min et taux de perte max
  - Découpées en trois priorités de perte  $AF_{i,j}$
- Décrite dans une RFC [6]

Notes :

---

---

---

---

---

---

---

---

## Les domaines

### 4 Deux architectures pour le monde IP

- L'architecture *DiffServ*
  - Les principes
  - Les éléments de base
  - Les classes de trafic
  - **Les domaines**
  - L'allocation des ressources
  - Les fonctions
  - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## Notion de domaine

- Un domaine
  - Un ensemble de routeurs administrés de façon cohérente
- À la frontière d'un domaine
  - Un autre domaine
  - Un client
- Conditions de traitement dans le domaine
  - Négociées dans un SLA [2, 17]
    - Qui intègre en particulier un TCA (*Traffic Conditioning Agreement*) définissant les règles de conditionnement à appliquer à une classe de trafic
  - Mise en place d'un contrat entre le domaine et le client ou un autre domaine

Notes :

---

---

---

---

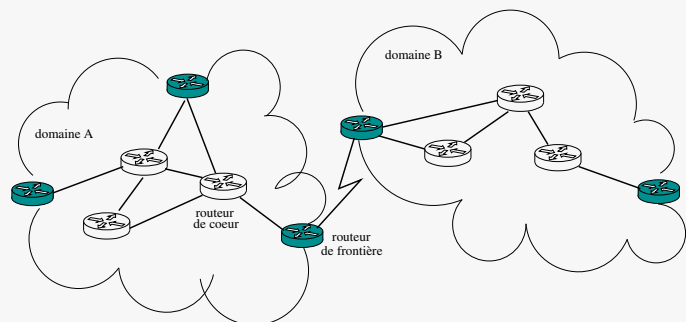
---

---

---

---

## Notion de domaine (2)



Notes :

---

---

---

---

---

---

---

---

## Notion de domaine (3)

La notion de domaine permet

- De différencier les routeurs
  - Cœur
  - Frontière
- De hiérarchiser la gestion
  - ISP, AS, ...
- De mettre en place des PHB différents entre domaines
  - Par re-marquage
- De supporter le passage à l'échelle
- Un déploiement incrémental

Notes :

---

---

---

---

---

---

---

---

## L'allocation des ressources

### Deux architectures pour le monde IP

- L'architecture *DiffServ*
  - Les principes
  - Les éléments de base
  - Les classes de trafic
  - Les domaines
  - L'allocation des ressources
  - Les fonctions
  - Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

## L'allocation des ressources

- Comment allouer les ressources à une classe de trafic ?
  - Les contrats de trafic sont négociés entre un domaine et un client/domaine
  - Ils sont spécifiés au travers de SLA intégrant en particulier des TCA
  - L'architecture *DiffServ* ne spécifie aucun mécanisme d'allocation de ressource
- Différentes options
  - De proche en proche, par exemple à l'aide de RSVP
  - Par une entité supérieure, un *Bandwidth Broker*
  - Par le plan de gestion

Notes :

---

---

---

---

---

---

---

---

## Les fonctions

- 4 Deux architectures pour le monde IP
  - L'architecture *DiffServ*
    - Les principes
    - Les éléments de base
    - Les classes de trafic
    - Les domaines
    - L'allocation des ressources
    - Les fonctions
    - Bilan de *DiffServ*

Notes :

---

---

---

---

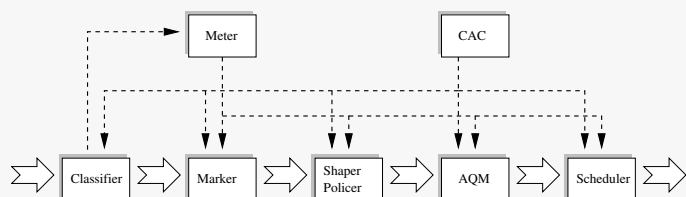
---

---

---

---

## Les fonctions



Présence et localisation dépendantes

- De l'architecture
- De l'entité dans l'architecture

Notes :

---

---

---

---

---

---

---

---

### 4 Deux architectures pour le monde IP

#### • L'architecture *DiffServ*

- Les principes
- Les éléments de base
- Les classes de trafic
- Les domaines
- L'allocation des ressources
- Les fonctions
- Bilan de *DiffServ*

Notes :

---

---

---

---

---

---

---

---

### Les plus

- Plus "réaliste" qu'*IntServ*
- Granularité plus large
- Prise en compte de l'hétérogénéité

### Les moins

- Quel plan de contrôle ?
  - Signalisation inter-domaine
  - Déploiement de politique au sein d'un domaine
  - COPS, RSVP, ...
- Quelle gestion des ressources ?
  - Distribuée (quelle pertinence ?)
  - Centralisée (*Bandwidth Broker*)
- Quelle garantie par flux ?

Notes :

---

---

---

---

---

---

---

---

- [1] F. Baker and G. Fairhurst.  
IETF Recommendations Regarding Active Queue Management.  
RFC 7567 (Best Current Practice), July 2015.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, and Z. Wang January.  
RFC 2475 - an architecture for differentiated service.  
Informational, IETF, December 1998.
- [3] R. Braden, D. Clark, and S. Shenker.  
Integrated services in the internet architecture : an overview.  
Technical report, Internet Engineering Task Force, United States,  
1994.
- [4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin.  
RFC 2205 : Resource reservation protocol (rsvp) – version 1  
functional specification.  
Standards track, IETF, September 1997.
- [5] B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec,  
W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis.  
RFC 3246 : An Expedited Forwarding PHB (Per-Hop Behavior).

Notes :

---

---

---

---

---

---

---

---



Technical Report 3246, March 2002.

- [6] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski.  
RFC 2597 : Assured forwarding phb group.  
Standards track, IETF, June 1999.
- [7] IETF, <http://www.ietf.org/html.charters/OLD/diffserv-charter.html>.  
*Differentiated Services (diffserv) charter*.
- [8] IETF, <http://www.ietf.org/html.charters/OLD/intserv-charter.html>.  
*Intergated Services (intserv) charter*.
- [9] V. Jacobson, K. Nichols, and K. Poduri.  
RFC 2598 : An expedited forwarding phb.  
Standards track, IETF, June 1999.
- [10] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al.  
A quantitative measure of fairness and discrimination.  
*Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984.*
- [11] F P Kelly, A K Maulloo, and D K H Tan.

Notes :

Rate control for communication networks : shadow prices, proportional fairness and stability.  
*Journal of the Operational Research Society*, 49(3) :237–252, 1998.

- [12] K. Nichols, S. Blake, F. Baker, and D. Black.  
Definition of the differentiated services field (DS field) in the ipv4 and ipv6 headers.  
RFC 2474, Internet Engineering Task Force, December 1998.
- [13] S. Shenker, C. Partridge, and R. Guerin.  
RFC 2212 : Specification of guaranteed quality of service.  
Standards track, IETF, 1997.
- [14] S. Shenker and J. Wroclawski.  
General Characterization Parameters for Integrated Service Network Elements.  
Technical Report 2215, September 1997.
- [15] S. Shenker and J. Wroclawski.  
Network Element Service Specification Template.

Notes :

Technical Report 2216, September 1997.

- [16] Andreas Vogel, Brigitte Kerhervé, Gregor v. Bochmann, and Jan Gecsei.  
Distributed multimedia applications and quality of service : a survey.  
In *CASCON '94 : Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research*, page 71.  
IBM Press, 1994.
- [17] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser.  
RFC 3198 : Terminology for Policy-Based Management.  
Technical Report 3198, Internet Engineering Task Force, November 2001.
- [18] J. Wroclawski.  
RFC 2211 :Specification of the Controlled-Load Network Element Service.  
Technical Report 2211, IETF, September 1997.

Notes :