MI020 - Internet Nouvelle Génération

Module de M1 de la Spécialité Réseaux
(Mention Informatique)
2e semestre 2010 - 2011
Université Pierre et Marie Curie

Support de Cours n°3

Contrôle de trafic / QoS
BGP

Bénédicte LE GRAND
Prométhée SPATHIS

Qualité de Service dans les réseaux IP

Plan

- **Introduction**
  - Qu'est-ce que la QoS?
  - Le besoin de QoS
- **QoS : boîte à outils**
  - Classification, conditionnement de trafic, files d'attente …
  - Différentes approches pour la QoS
    - QoS garantie vs différenciée
- **Architectures de QoS**
  - Integrated Services (IntServ) et RSVP
  - Differentiated Services (DiffServ)

Qu'est-ce que la QoS ?

Contexte

- Réseau de paquets avec multiplexage asynchrone
  - Partage de bande passante dynamique
    - Multiplexage statistique, haute utilisation de la bande passante
  - Partage de ressources => contention
  - Contention => file d'attente FIFO
- Problèmes
  - Les flux de paquets sont remis en forme
    - Le délai introduit dans une rafale atteint par tous les flots
    - Pas d'isolation
  - Partage inéquitable quand forte contention
    - Différentes probabilités d'être jeté pour chaque flot
    - Bande passante allouée en fonction du niveau d'occupation de la file
  - Pas de différentiation des flots
    - Plusieurs utilisateurs peuvent demander des services différents
- Objectifs
  - Gérer la contention
  - En mode Best Effort : fair share
  - En service garanti : honorer les garanties de performance
    - Isolation et partage

Qu'est-ce que la QoS ?

- QoS (Qualité de Service) = attribut du service fourni par le réseau
  - Un réseau peut fournir différents modèles de service
  - Le client et l'opérateur réseau négocient un SLA (Service Level Agreement) basé sur des modèles de service rédéfinis
- QoS ≠ Performance
  - La performance caractérise le comportement du réseau
  - Le réseau devrait fournir les différents modèles de service **et** minimiser l'utilisation des ressources du réseau

Qu'est-ce que la QoS ?

- La QoS peut être évaluée selon différents critères:
  - Bande passante
  - Délai :
    - Délai de bout-en-bout
    - Variation du délai de bout-en-bout
  - Intégrité des données :
    - Taux de perte de paquets
    - Taux d'erreur des paquets
    - Taux de déséquencement des paquets
  - Fiabilité et disponibilité

Besoin de QoS dans les réseaux IP

Le besoin de QoS (1)

- Les applications ont besoin de QoS:
  - Les applications de voix, vidéo et multimédia constituent des flots de trafic temps réel qui ont des contraintes de délai.
  - Les applications commerciales basées sur des transactions client/serveur ont des contraintes fortes en terme de temps de réponse.
- Historiquement :
  - Applications vidéo et multimédia applications presque inexistantes dans le contexte de réseaux asynchrones
  - La voix avait son réseau optimisé, le PSTN (public switched telephone network), lui fournissait la QoS nécessaire.
  - Les applications commerciales avaient leurs réseaux avec des architectures ad-hoc telles que SNA (System Network Architecture) .

8

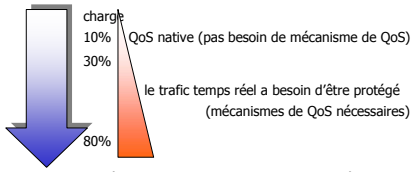
Le besoin de QoS (2)

- Aujourd'hui :
  - La voix est de plus en plus compressée et tend à être transmise dans des paquets de données plutôt que dans des réseaux à commutation de circuits
  - Développement des application vidéo et multimédia (vidéo-conférence, video-on-demand…)
  - Réseaux tout IP
- Les réseaux de données devraient pouvoir réaliser une intégration de service en fournissant à chaque application la QoS dont elle a besoin (concept de Next Generation Network)

9

Fournir la QoS

- La QoS a un sens quand le réseau est chargé :
  - Plus la charge du réseau est élevée et plus les mécanismes de QoS sont complexes.



  - Le niveau de déploiement de la QoS est directement lié aux aspects économiques :
    - Prix de la bande passante (réseaux d'accès ou de coeur)
    - Prix des équipements (impact de la QoS sur le hardware et le software)
    - services visés (contraintes de QoS constraints des applications, etc.)

10

Fournir de la QoS pour le trafic IP

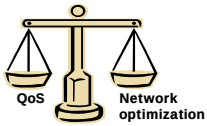
- Solutions possibles
  - **Sur-dimensionner les ressources du réseau**.
    - Dans ce cas, pas besoin de déployer des mécanismes de gestion de trafic complexes.
  - **Utiliser des fonctions de contrôle de la QoS** et charger le réseau à des niveaux plus élevés
    - Peut être implémenté
      - Au niveau IP
      - En utilisant IP sur ATM, la couche ATM fournissant les capacités de QoS à la couche IP

11

Pourquoi la QoS est-elle complexe ?

Le problème de la QoS

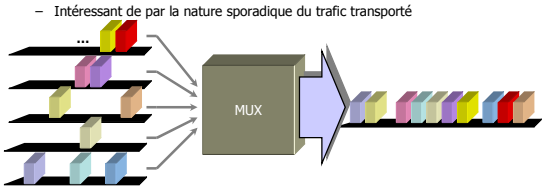
- QoS garantie => **allocation de ressources**.
- Optimiser l'utilisation des ressources réseau => **ressources partagées** entre tous les flots de données.
  - L'optimisation des ressources réseau est, en général, nécessaire pour réduire les coûts
- Compromis entre la QoS et l'optimisation des ressources réseau.
  - Ceci est effectué via le **multiplexage statistique** utilisé avec d'autres fonctionnalités afin de réaliser l'optimisation des ressources et la provision de QoS



13

Multiplexage statistique

- À un niveau microscopique (échelle de temps de la transmission de paquet)
  - Intéressant de par la nature sporadique du trafic transporté

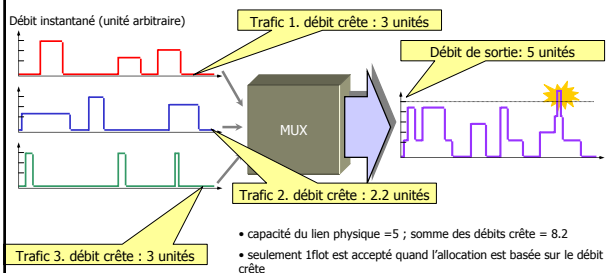


- À l'échelle de temps de la rafale (ex : ABT ATM Block Transfer)
- À l'échelle de temps du flot (flot / durée de connexion)

14

Multiplexage statistique (2)

- Illustration du compromis entre utilisation du réseau / QoS

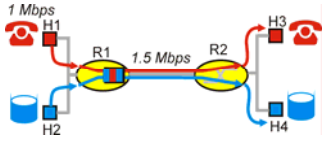


- capacité du lien physique =5 ; somme des débits crête = 8.2
- seulement 1flot est accepté quand l'allocation est basée sur le débit crête
- Tous les flots sont acceptés avec une allocation dynamique et le multiplexage statistique

15

Principes pour la garantie de QOS

- Exemple : téléphone IP 1Mbps, FTP 1Mbps ; partagent un lien à 1.5 Mbps.
  - Les rafales de FTP peuvent congestionner le routeur et causer des pertes audio
  - On veut donner la priorité à l'audio par rapport à FTP



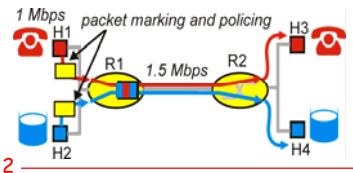
Principe 1

Marquage de paquets nécessaire pour que les routeurs puissent distinguer des classes différentes; nouvelle politique des routeurs pour traiter les paquets en fonction de leur classe

16

Principes pour la garantie de QOS (suite)

- Que se passe-t-il si les applications se comportent mal ? (ex: l'audio envoie plus de trafic que le débit déclaré)
  - Policing : contraindre les sources à adhérer aux allocations de bande passante
- Marquage et police en bordure du réseau
  - Similaire à l'UNI d'ATM (User Network Interface)



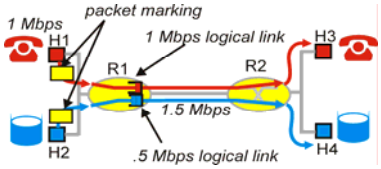
Principe 2

Protéger (*isoler*) les classes les unes par rapport aux autres

17

Principes pour la garantie de QOS (suite)

- Allocation de bande passante *fixée* (non partageable) à un flot : utilisation *inefficace* de la bande passante si le flot n'utilise pas ce qui lui a été alloué



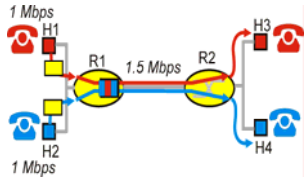
Principe 3

Tout en fournissant l'isolation, il faut utiliser les ressources aussi efficacement que possible

18

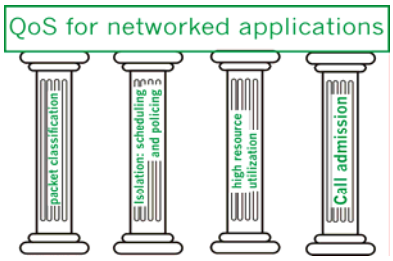
Principes pour la garantie de QOS (suite)

- *Fait réel :* on ne peut pas répondre à des demandes de trafic au-delà de la capacité du lien



Principe 4
Admission d'appel : un flot déclare ses besoins, le réseau peut bloquer l'appel (ex : signal occupé) s'il ne peut pas répondre à ces besoins

Résumé des principes de QoS



Regardons maintenant les mécanismes pour mettre tout ceci en oeuvre…

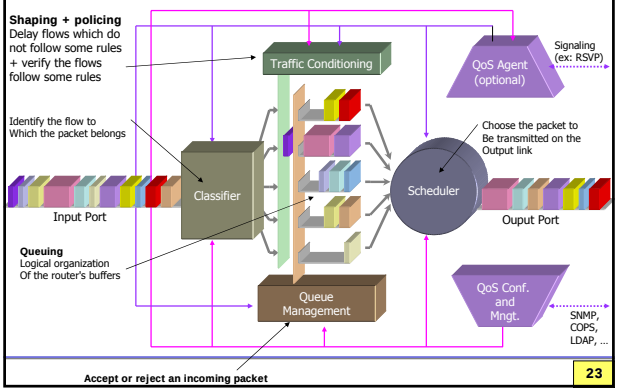
QoS : mécanismes

- **Protocole de réservation.** Pour indiquer le volume de ressources nécessaire (CPU, mémoire, bande passante) le long du chemin des données.
- **Classification.** Pour identifier le flot auquel appartient le paquet qui arrive
- **Contrôle d'admission.** Pour déterminer, pour chaque nouvelle réservation, si elle peut être acceptée ou non en fonction des ressources disponibles.
- **Fonction de police.** Pour vérifier si le volume de ressources réservé n'est pas dépassé par la source.
- **Fonction de mise en forme.** Pour retarder les flots qui ne suivent pas certaines règles.
- **Algorithmes d'ordonnancement.** Pour allouer une capacité de transmission sur une base paquet par paquet afin d'atteindre les objectifs de QoS pour chaque flot.
- **Gestion de files d'attente.** Pour jeter les paquets, en cas de congestion, selon le niveau de priorité des paquets

Plan

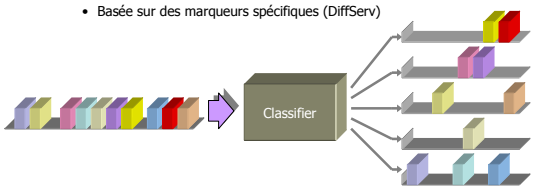
- **Introduction**
  - Qu'est-ce que la QoS?
  - Le besoin de QoS
- **QoS : boîte à outils**
  - Classification, conditionnement de trafic, files d'attente …
  - Différentes approches pour la QoS
    - QoS garantie vs différenciée
- **Architectures de QoS**
  - Integrated Services (IntServ) et RSVP
  - Differentiated Services (DiffServ)

Mécanismes de QoS : vue globale



Classification

- Classification du trafic
  - Besoin de classification :
    - Différents clients sur la même interface (chacun avec un SLA/TCA spécifique)
    - Différentes « classes de service » , etc.
  - Types de classification :
    - Basée sur les adresses IP (en-tête de niveau 3)
    - Basée sur les en-têtes de niveau supérieur (N°s de port TCP/UDP etc.)
    - Basée sur des marqueurs spécifiques (DiffServ)



Conditionnement de trafic

Nous aborderons :

- Le policing
- Le contrôle de flot
- Le contrôle de congestion
- L'équité
- Les Token/Leaky buckets
- Les TSpec
- Le contrôle d'accès
- La remise en forme (shaping)
- Les files d'attente
- …

25

Policing

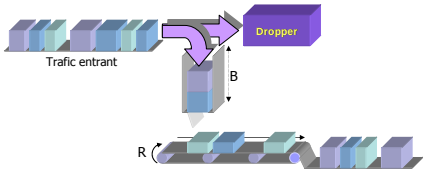
But : limiter le trafic pour ne pas dépasser les paramètres déclarés
Trois critères fréquemment utilisés :

- *Débit moyen (Long terme) :* combien de paquets peuvent être envoyés par unité de temps (à long terme)
  - Question cruciale : quel est l'intervalle de temps : 100 paquets par seconde ou 6000 paquets par minute ont la même moyenne !
- *Débit crête :* ex : 6000 paquets par minute en moyenne ; 1500 ppm de débit crête
- *Taille de rafale (Max.) :* nombre max. de paquets envoyés en une fois (sans silence intermédiaire)

26

Policing

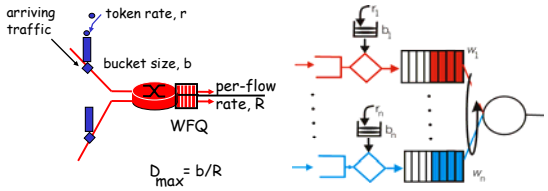
- Conditionnement de trafic
  - Nécessite des TCA (Traffic Conditionning Agreements)
  - De tels mécanismes incluent :
    - Des mesures de trafic (débit moyen, débit crête)
    - Packet Dropper (avec des politiques de rejet spécifiques)
    - Traffic Shaper
    - …
  - Exemple : Contrôle du débit crête avec un token bucket shaper (R,B)



27

Policing

token bucket, combiné avec WFQ pour imposer une borne supérieure au délai, i.e., *QoS garantie* !



28

Problème

- Transfert de fichier
  - Le flot de paquet représente le fichier
  - Le débit dépend de la capacité du récepteur et la capacité du réseau
- Comment choisir le débit ?
  - Trop faible : temps perdu
  - Trop élevé : risque d'introduire de la congestion
- Vidéo à la demande
  - Débit variable
- Comment réserver des ressources réseau ?
  - Trop : faible utilisation des ressources
  - Trop peu : risque d'introduire de la congestion

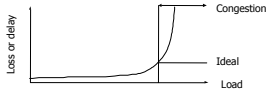
29

Contrôle de flot

- Définition
  - Ensemble de techniques de contrôle de trafic et de congestion pour obtenir un service réseau satisfaisant pour l'utilisateur
- Contrôle
  - Contrôle de trafic
    - Ensemble d'actions de contrôle effectuées par le réseau pour éviter la congestion
    - Actions préventives
  - Contrôle de congestion
    - Ensemble d'actions pour restaurer un fonctionnement normal du réseau en cas de congestion
    - Actions réactives : quand la congestion est proche ou installée
  - Objectifs du contrôle
    - Maintenir la qualité du transfert pour les flots de service garanti
    - Équité entre les flots BE
    - Utilisation optimale des ressources
      - Liens
      - Files d'attente

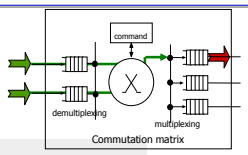
30

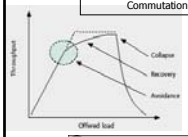
Congestion

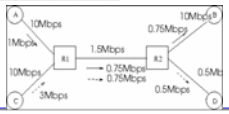
- Définition générale
  Situation où le réseau ne peut pas satisfaire un service pour les applications
  - Quand ?
    $\sum Demand(t) > Resource(t)$
    

  - Causes
    - Fluctuations de trafic imprévisibles
    - Problèmes du réseau
- Solutions
  - Augmenter les ressources
    - BP, Buffers …
  - Contrôler les demandes

Conséquences de la congestion





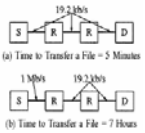


- Contention gérée par les files
  - Sérialisation des demandes
  - La file est la première protection face à l'augmentation de λ
- Sévère contention λ >> μ
  - Buffers pleins
  - Pertes
- Conséquences
  - Les temps de transfert des paquets augmente
  - Perte de paquets
    - Transfert de données : retransmission des paquets perdus et risque de retransmissions abusives dues aux délais importants
  - "Congestion collapse"
    - Gâchis de ressources réseau
    - Ressources utilisées par un paquet avant qu'il soit détruit sont perdues
    - Augmentation de la charge => diminution du débit

(Mauvaises) solutions possibles

- Augmenter la taille du buffer
  - Augmentation du délai, retransmission activée par le transport
    - Durée de vie finie du paquet
  - Gâchis de ressources, le paquet retransmis est jeté
  - Pire qu'un petit buffer

- Augmenter la bande passante
  - Ajouter un lien à haut débit
    - Augmente le temps de transfert
  - Un réseau possède des liens hétérogènes



- Conséquence
  - La congestion est un problème dynamique qui requiert des solutions dynamiques
    - Contrôler les demandes
    - Gérer les ressources

Critères de performance

- Efficacité
- Best-effort : équité
  - Partage équitable des ressources
- QoS : performance garantie
  - Fournir des garanties et les respecter
    - Utiliser des réservations
  - Expression via des paramètres de QoS
    - BP
    - Délai
    - Gigue
    - Perte

=> Il faut isoler les utilisateurs

Équité

- Allocation d'un système partagé
- Enjeu quand certaines demandes ne sont pas satisfaites
- Résulte d'une optimisation
- Métrique
  - Fair index[1]
    - Quantifie l'équité dans l'allocation de la BP en excès entre les clients qui partagent un lien

$$f(x_1,x_2,\ldots,x_n)=\frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n.\sum_{i=1}^{n} x_i^2}$$

≤1 toujours,
=1 s'il y a équité

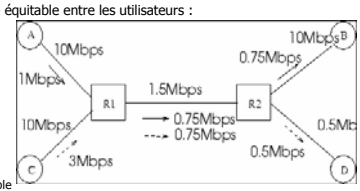
où xi est le débit en excès du ième client

- Portée
  - Globale : niveau réseau => action au niveau rafale
  - Locale : niveau routeur => action au niveau paquet

1 defined in R. Jain, A. Durresi and G. Babic, Throughput Fairness Index: An explanation, ATM Forum/99-0045, Feb. 1999

Équité locale et globale

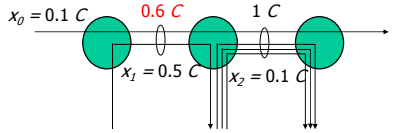
- Considérons un partage équitable entre les utilisateurs :



- Résultat
  - Localement équitable
  - Globalement inéquitable
- Éviter le congestion collapse
  - Un flot doit avoir une allocation correspondant au goulot d'étranglement
    - Limiter le débit par source
    - Échelle temporelle d'action : RTT
  - sinon
    - Utilisation inefficace des ressources réseau
    - Du point de vue des sources, la performance n'est pas satisfaisante

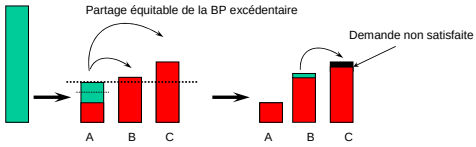
Partage équitable

- Le partage équitable n'est pas une bonne solution
- Trouver un équilibre entre efficacité et équité



Max-min Fair Share

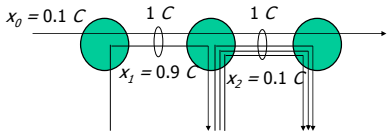
- Satisfaire un maximum de flots
- Principe
  - Satisfaire ceux qui demandent moins
  - Aucun nœud n'obtient plus que ce qu'il demande
  - Ceux qui ne sont pas satisfaits obtiennent un partage équitable



- Définition
  - Personne ne peut obtenir plus aux dépens de quelqu'un qui a déjà moins

Max-min Fairness

- Augmente l'efficacité du réseau
  - x_1 a une plus grande allocation sans gêner les autres flots.



Goulot d'étranglement

- Définition

  Un lien est goulot d'étranglement pour une source si et seulement si
  1. Le lien est saturé
  2. La source s sur ce lien a le débit maximum de toutes les sources sur ce lien

- Théorème
  - Une allocation X est max-min fair ssi chaque source a un goulot d'étranglement

Isolation

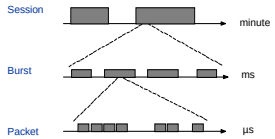
- Protection
  - Isoler chaque flot
- L'équité apporte la protection
  - Chaque source a sa propre allocation équitable

Action de contrôle

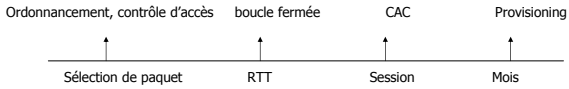
- Échelle de temps
  - Paquet
  - RTT
  - Session
  - Long terme
- Granularité
  - Flot applicatif
  - Agrégation de flot
- Localisation
  - Source
    - Couche transport ou application
    - Action basée sur une notification
  - Routeur
    - En bordure ou au cœur
    - Police (vérifier que le trafic respecte certaines règles)
    - Priorités : disciplines de files d'attente et de service

Échelle de temps du contrôle

- Modèle de trafic



- Actions



Modes de contrôle

- *Boucle ouverte*
  - La source décrit son débit désiré
  - Le réseau accepte l'appel
  - La source respecte le débit
  - Actions préventives
  - Contrôle de trafic
- *Boucle fermée*
  - La source adapte le taux de service en fonction d'une notification de feedback
  - Latence dans la propagation des notifications => erreurs
  - Actions réactives
  - Contrôle de congestion
- *Hybride*
  - La source demande un débit minimum
  - Mais elle peut envoyer plus si c'est possible

Boucle ouverte

- Principe
  - Performance prévisible si les flots sont contraints
  - Flot contraint + réservation = performance garantie
- Pour des application capables de décrire leur comportement futur
  - Média continus
- 2 phases
  - Établissement d'appel
  - Transmission de données
- Établissement d'appel
  - Négocier les ressources (entre source et réseau)
    - Décrire le trafic : descripteur de flot
  - Le réseau accepte ou refuse : contrôle d'admission
  - Contrat de trafic : réservation
    - Si la source se conforme au contrat, le réseau garantit qu'il n'y a pas de congestion
- Transmission de données
  - La source envoie du trafic conformément à la description
  - Le réseau contrôle la conformité (contrôle d'accès)
    - Régulation : mise en forme
    - Surveillance: Dropping
  - Actions du réseau pour fournir la QoS au flot
    - Priorités (ordonnancement et files d'attentes)

Évaluation

- Contrôle pour les applications critiques
  - La QoS peut être fournie
  - Pas de congestion
    - Pas de perte dans le réseau
    - Gigue et temps de réponse minimisés
  - Difficile à mettre en œuvre en mode sans connexion
  - Sous-utilisation des ressources
    - Mauvaise caractérisation de la source
    - Les sources ne sont pas toujours actives
- Difficulté à choisir
  - Le descripteur de flux
  - Comment admettre / rejeter un flot
  - La politique d'ordonnancement et de file d'attente

Descripteur de flot

- Définition :
  - Ensemble de paramètres qui représentent le comportement de la source
  - Description du trafic envoyé
- Principe
  - Décrire le « pire » comportement
  - Définir une enveloppe
- Utilisation
  - Contrôle d'admission : contrat de trafic
  - Contrôle d'accès : régulation, surveillance.

Débit crête

- Principe
  - Débit maximal auquel les données peuvent être envoyées
  - Intervalle temporel minimum entre les paquets

$$D_{\max} = \frac{L}{T}$$ L : taille du paquet

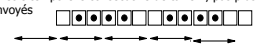
- Mise en forme (shaping)
  - Démarrer un temporisateur quand un paquet est envoyé
  - Possible d'envoyer un autre paquet quand la tempo expire
- Évaluation
  - Limite extrême, sensible aux variations

Débit moyen

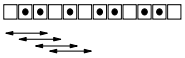
- Principe
  - Débit sur une période de temps
    - réduire la sensibilité aux variations
  - Volume de données A envoyé sur une période de temps T

$$D = \frac{A}{T}$$

- Descripteurs
  - Fenêtres
    - Pour chaque fenêtre temporelle consécutive de taille T, pas plus de A bits peuvent être envoyés



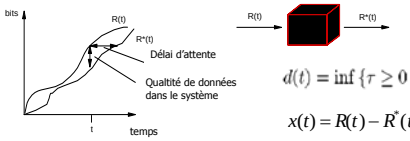
    - Dépendent de l'état initial
  - Fenêtre glissante
    - Pour chaque période T, pas plus de A bits peuvent être envoyés



    - Trop complexe à un débit élevé

Modèle fluide

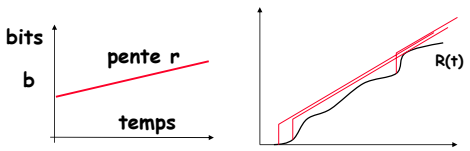
- Modèle mathématique
  - Utiliser un modèle parfait
    - Fonction continue
    - Pas de discontinuité introduite par les paquets
  - Étudier le flot isolé des autres flots
    - Lien dédié
- Courbes d'arrivée et de départ



$$d(t) = \inf\{\tau \geq 0 : R(t) \leq R^*(t+\tau)\}$$

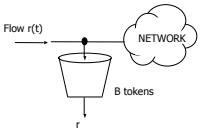
$$x(t) = R(t) - R^*(t)$$

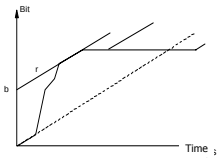
Flot contraint : enveloppe



Leaky Bucket

- Paramètres
  - r: rate (débit)
  - b: taille du buffer
- Algorithme
  - Token = droit d'envoyer 1 bit
  - Le seau se vide à un débit de token contraint (r)
  - Le seau ne doit pas déborder
  - Quand il y a débordement, le flot n'est pas conforme
    - L'action sur les données non-conformes dépend de la politique.



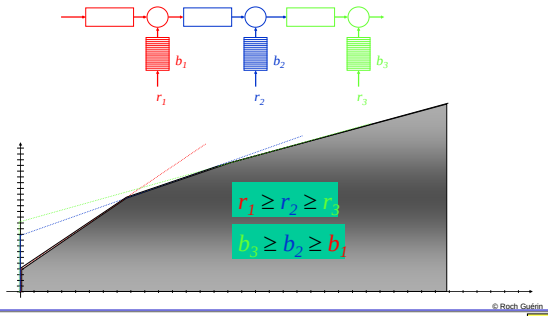


Leacky Bucket

- Utilisation
  - Média continus : descripteur approximant le débit moyen et la sporadicité
  - Média discrets : détermine une enveloppe obtenue après régulation
- Évaluation
  - Limite la taille des rafales
  - Algorithme simple
  - Difficile de choisir b et r
  - Pour les média discrets, consiste en la mise en forme au débit r

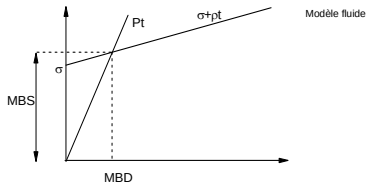
Concaténation de Buckets

- Plusieurs seaux peuvent être concaténés pour contrôler plusieurs débits
  - Le débit diminue au fur et à mesure des seaux



Token bucket

- Token Bucket
  - Ajout d'un régulateur au débit crête



MBS = Maximum Burst Size

$$MBS = \sigma \frac{P}{P-\rho} > \sigma$$

MBD = Maximum Burst Duration

$$MBD = \frac{MBS}{P} = \frac{\sigma}{P-\rho}$$

55

Leaky bucket et Token Bucket

- Leacky bucket
  - Contrôle de conformité
    - Débit moyen
  - Mise en forme : couplée avec une file d'attente
  - Surveillance : effacer les paquets non conformes.
- Token Bucket
  - Leaky bucket avec un régulateur de débit crête
  - Pour contrôler
    - Le débit moyen
    - Le débit crête
    - La taille maximum de rafale (Maximum burst size)
  - Utilisé pour le TSpec dans IntServ (présenté plus loin)

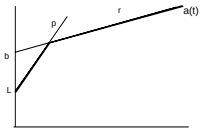
56

Application du TSpec

- Attention
  - Un flot de paquet n'est pas un liquide
  - Paquets transmis au débit du médium
  - Les paquets sont de taille variable
- TSpec (spécification de trafic)
  - Le TSPEC(r,b,p,L) détermine une courbe d'arrivée

$$a(t) = \min(L + pt, b + rt)$$

  - équivalent à la conjonction de 2 leaky buckets

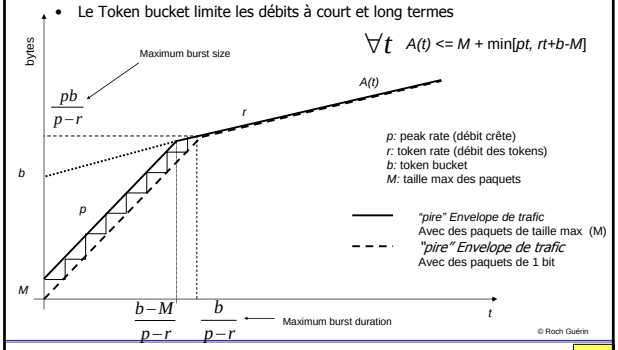


57

Le *Dual* leacky Bucket

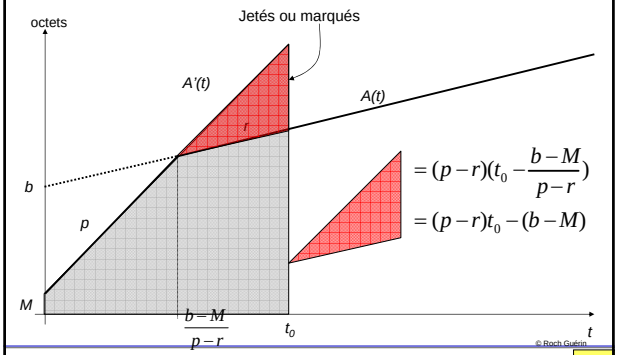
- Le Token bucket limite les débits à court et long termes

$\forall t$ $A(t) <= M + \min[pt, rt+b-M]$



58

Mode Dropping/Marquage



$$= (p-r)(t_0 - \frac{b-M}{p-r})$$

$$= (p-r)t_0 - (b-M)$$

59

Contrat de trafic

- Pour définir les caractéristiques d'un flot et les imposer
- La source doit respecter les caractéristiques annoncées
- Le réseau doit maintenir la QoS demandée
- Établi durant la phase de négociation

- Éléments du contrat de trafic
  - Paramètres de QoS de l'utilisateur
    - Service attendu
    - Ensemble de paramètres
      - Négociables : taux de perte, temps de transfert max…
      - Non négociables : taux d'erreur par paquet…
    - Descripteur de flot
      - Caractérise le trafic de la source
      - Comprend : le descripteur de trafic de la source, la définition de conformité

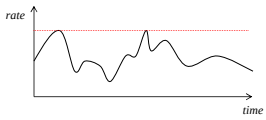
60

Contrôle d'admission

- Limite le nombre de flots dans le réseau
- Vérifie que le réseau a suffisamment de ressources pour accepter un nouveau flot
- Mise en œuvre
  - Analyser la demande dans le contrat de trafic
  - Réserver des ressources
- **Contrôle d'admission pour CBR :** $L + \rho_i \leq C$
  - Test pour un appel i et une charge déjà admise L
  - Simple
  - Si cela échoue, la requête est re-routée, retardée ou effacée
- **Contrôle d'admission pour BE**
  - Jamais rejeté
- **Contrôle d'admission pour VBR**
  - Caractéristiques de VBR
    - rafales (les débits crête et moyen sont différents)
    - Politiques de réservation
      - Débit crête : ressources gâchées
      - Débit moyen : paquets jetés pendant les rafales (probablement)
    - Plusieurs types de réservation (débit crête, débit moyen …)

Contrôle d'admission au débit crête

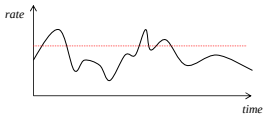
- Avantages
  - Délai et perte nuls
  - Simple
    - calcul
    - L'ordonnancement FIFO est OK
- Inconvénients
  - Gâche la bande passante



  - Le débit crête peut augmenter après être passé par plusieurs ordonnanceurs
    - Introduit de la gigue

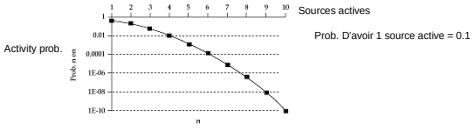
Contrôle d'admission : LBAP (*Linear Bounded Arrival Process*)

- Principe
  - Description de la source avec LBAP
  - Utilise un ordonnaceur qui réserve
    - BP ≥ débit moyen
    - Un buffer pour MBS
- Avantages
  - Plus efficace pour la BP que l'allocation au débit crête
  - Utilisé pour donner des garanties de délai, perte et BP
    - Dans GS
- Inconvénients
  - Un délai court peut nécessiter une réservation supérieure au débit crête
  - Le pire cas peut être très improbable => gâchis de BP
  - Complexe



Contrôle d'admission : garanties statistiques

- Principe
  - Repose sur le multiplexage statistique
  - Lorsque le nombre de flots N augmente, la probabilité qu'ils soient tous actifs diminue



  - Pour des valeurs élevées de N, le débit total est quasiment constant
  - Contrôle sur la notion de BP équivalente
- Avantages : utilisé pour fournir des garanties statistiques pour le délai
- Inconvénients
  - Suppose que les sources sont indépendantes
  - Complexe

Contrôle d'admission : MBAC (Measurement-based Admission Control)

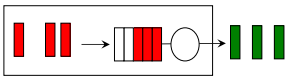
- Pour les flots qui ne peuvent pas être décrits ou varient de manière imprévisible
- Principe
  - Mesure la charge réelle
  - Nouveau flot décrit par son débit crête
  - Admission
    - crête + moyen ≤ capacité
  - Le flot est décrit par le débit moyen quand actif
  - Utilisé par CL
- Inconvénients
  - Suppose que le futur et le passé sont similaires
  - Durée de la période de mesure ?
  -

Contrôle d'accès

- Problèmes
  - La charge peut dépasser la demande si
    - Elle est mal caractérisée
    - On a une mauvaise connaissance du trafic
- Objectifs
  - Vérifier que le trafic est conforme à la déclaration
  - Éviter la congestion
- Principe
  - Vérification
    - De la part du réseau : que chaque flot est conforme au descripteur
      - En utilisant un leaky bucket
      - Représente un ensemble d'actions (de la part du réseau) pour surveiller et contrôler le trafic
- Actions
  - Surveillance : détecter si un paquet est conforme ou non à une spécification
  - Régulation (shaping)
    - Réduire la sporadicité du trafic ou changer les caractéristiques du trafic
  - Marquage
    - Préférence pour jeter les paquets parmi les paquets non-conformes

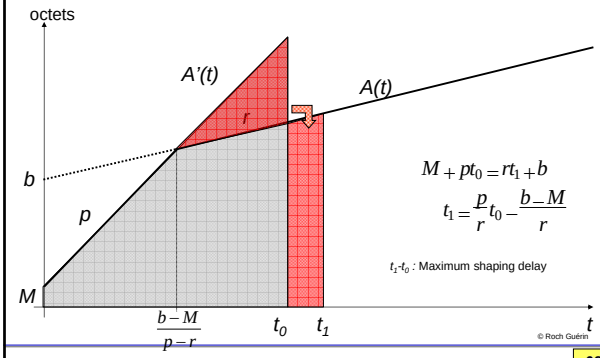
Shaping

- Définition
  - Modifier le flot temporellement pour s'assurer qu'un flot envoyé sur le réseau ne dépasse pas certaines limites
  - Limites en termes de
    - Débit crête, débit moyen, taille maximum de rafale
- Principe
  - Adapter le flot en fonction d'un contrat / profil établi
  - Si la file est pleine, des paquets sont perdus
    - => les paramètres de trafic doivent être correctement définis pour éviter un dropping excessif
- Objectifs
  - Contraindre le flot à des caractéristiques connues
    - Pour fournir des garanties de QoS
  - Réduire la sporadicité



Interface de sortie

Shaping Mode

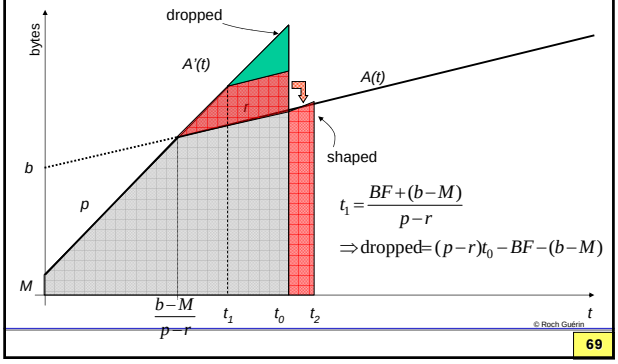


$$M + pt_0 = rt_1 + b$$

$$t_1 = \frac{p}{r}t_0 - \frac{b-M}{r}$$

t_1-t_0 : Maximum shaping delay

Shaping Mode – Taille du Buffer (BF)

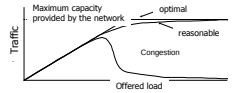


$$t_1 = \frac{BF + (b-M)}{p-r}$$

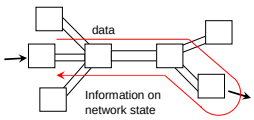
$$\Rightarrow \text{dropped} = (p-r)t_0 - BF - (b-M)$$

Boucle fermée : boucle de contrôle

- Utilisation
  - La source ne peut pas décrire son trafic ou le réseau ne peut pas réserver de ressources
  - Les ressources sont insuffisantes pour la source
    - La source peut envoyer plus que ce qu'elle a
- Objectifs
  - Éviter le congestion collapse (réduction de la quantité de trafic qui parvient à traverser le réseau)
  - Équité (partage équitable de ressources)
  - Contrôler la performance obtenue



- Contrôle réactif
  - La source envoie une notification



Boucle de contrôle

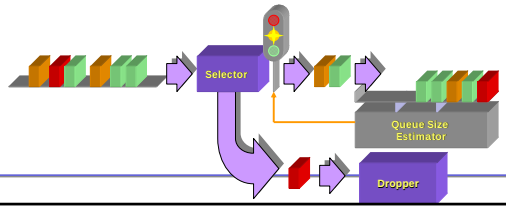
- Principe
  - Évaluer les ressources disponibles
  - Notification de la source
    - Implicite ou explicite
  - Contrôler le débit de la source
    - fenêtre, etc.
- Inconvénients
  - Requiert la coopération de la source de trafic
  - En général, un émetteur doit pouvoir contrôler son débit et le réduire pour réduire la congestion
- Exemple :
  - TCP
    - Notification implicite et fenêtre de congestion
    - Notification implicite et contrôle par fenêtre.
- évaluation
  - Pas efficace pour les applications temps réel
  - Utilisation optimale des liens

Classification des gestions de files d'attente

- Nombre
- Par couleur, par circuit virtuel, par flot ou globale
- Multiple ou unique
- seuil : single ou multiple
- Différents types
  - SAST : Single Accounting, Single threshold
  - SAMT: Single Accounting, Multiple threshold
  - MAST Multiple Accounting, Single threshold
  - MAMT Multiple Accounting, Multiple threshold

Files d'attente

- Gestion des files d'attente
  - Pourquoi ?
    - Des mécanismes préventifs (conditionnement de trafic, ordonnancement) permettent d'éviter l'apparition de congestion, avec un certain niveau de confiance. Cependant, à de courtes échelles temporelles, la congestion pourraît apparaître, auquel cas les files d'attente grossissent. Des procédés curatifs sont alors nécessaires pour protéger les paquets "privilégiés".
  - Une gestion active de files d'attente (Active Queue Management) peut utiliser des priorités de pertes spécifiques (jaunes, vertes, rouge)



Files d'attente

- Q ? Quand jette-t-on des paquets ?
  - Quand le buffer est plein (tail drop)
  - Quand l'occupation du buffer augmente trop (RED)
- Q ? Quels paquets doit-on jeter ?
  - Le paquet arrivant (tail drop : mais est-ce que ce paquet est responsable de la congestion ?)
  - Un autre paquet parmi le flot de paquets arrivants
    - Ceci pourrait aider les algorithmes de contrôle de congestion
  - Un paquet d'un flot quelconque
  - Le paquet en tête de file
    - Pourrait améliorer les performances de TCP

Files d'attente

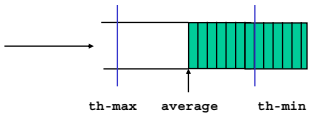
- Objectifs
  - Ajouter de la discrimination dans le réseau
  - Résoudre les problèmes de TCP
    - Le contrôle de congestion de bout-en-bout est inéquitable
    - UDP n'est pas équitable avec TCP
    - Le DropTail entraîne une synchronisation des sources TCP
    - Le Droptail pénalise le trafic sporadique
    - Difficile d'identifier la nature de la congestion (temporaire ou non)
- Principe
  - Gérer les buffers des routeurs
  - Actions
    - Jeter des paquets (RED, …)
    - Notification (ECN)

Files d'attente : RED (Random Early Detection)

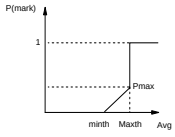
- Idée
  - Agir avant la congestion et réduire le débit de la source
  - Seuil pour commencer à jeter des paquets
- Objectifs
  - Éviter la synchronisation des sources TCP
  - Augmenter la charge sur les liens
  - Ne pas pénaliser les flots sporadiques
  - Éviter de jeter des rafales

RED (Random Early Detection)

- Algorithme :
  - 2 seuils (Min_Th, Max_Th) pour la file d'attente
  - Taille moyenne de la file (avg)
  - Effacer (aléatoirement) des paquets de la file



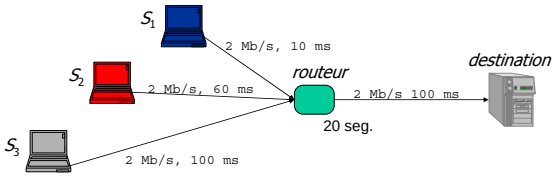

```
Pour chaque paquet reçu
    Calculer la taille moyenne de la file avg
    si minth ≤ avg ≤ maxth
        Calculer la proba p(avg)
        Avec la proba (avg)
            marquer/jeter le paquet
    Sinon si axth < avg
        Jeter le paquet
```



RED

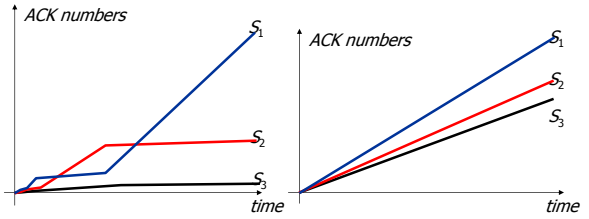
- Evaluation
  - Adapté à TCP
  - Jeter un paquet => éviter la congestion
  - La probabilité de choisir un flot est proportionnelle au débit du flot
  - Difficile à paramétrer
    - Dépend du trafic : dynamique
- Amélioration : marquage
  - Réduit le nombre de paquets jetés
  - Réduit le temps pour détecter la congestion
  - Limites :
    - Les paquets marqués peuvent être jetés
    - Certaines sources peuvent ne pas prendre le marquage en compte
- RED amélioré
  - Identifier les flots "gourmands"
    - Supprimer l'inéquité UDP
  - Réguler les flots
    - Équité et protection avec des buffers
    - maintenir 1 état par flot

Exemple de réseau pour RED



- Exemple de réseau avec 3 sources TCP
  - Différents temps de propagation des liens
  - Files limitées sur le lien (20 paquets)

Débit sans / avec RED



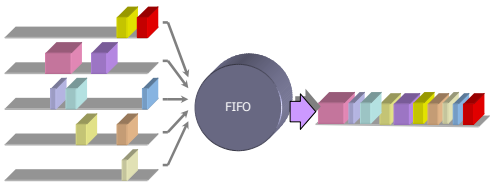
sans RED avec RED

Ordonnancement (Scheduling)

- Nous étudierons :
  - FIFO
  - Round robin
  - Fair queuing
  - Weighted Fair queuing
  - Virtual Colock
  - Generalized Processor Sharing
  - …

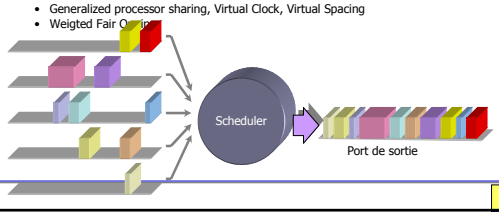
Ordonnancement

- Dans l'Internet
  - Ordonnancement : FIFO
  - Discipline de file d'attente : Drop Tail



Ordonnancement

- Scheduling : choisir les prochains paquets à envoyer sur le lien
- Ordonnanceur
  - Permet de servir une classe de service avec un débit spécifique ou une priorité spécifique
- Ordonnancement FIFO (first in first out) : envoyer par ordre d'arrivée dans la file
- Exemples :
  - Round Robin, Weighted Round Robin (ex : 10% or, 30% argent, 55% Bronze, 5% Controle), Deficit Round Robin
  - Priorité stricte (décrite ci-dessous)
  - Generalized processor sharing, Virtual Clock, Virtual Spacing
  - Weigted Fair Q…



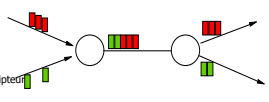
Ordonnancement : Work Conserving ou non ?

- Work-conserving
  - Jamais oisif quand des paquets attendent
  - Loi de conservation
    - Délais constants, l'ordonnancement alloue les délais

$$\sum_{i=1}^{N} \rho_i d_i = Cst \qquad \text{With } \Sigma \rho_i \leq 1$$

- Non work-conserving
  - Peut être oisif même si des paquets attendent
  - Retarde le service quand le paquet est éligible

- Evaluation
  - Conservation
    - Modifie la forme du trafic
    - Augmentation de la taille des buffers
  - Non conservation
    - Conserve la gigue
    - Limite le besoin des buffers
    - Force la source pour respecter le descripteur



Ordonnancement : FIFO

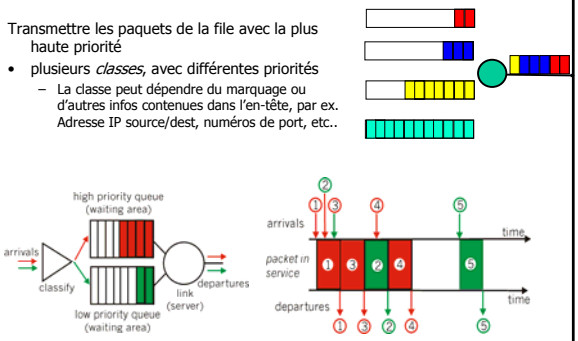
- Ordonnancement par défaut de l'Internet
- Partage de bande passante
  - proportionnel à la charge offerte
- Pas d'isolation
  - Flot adaptatif / non adaptatif
  - Flot temps réel / non temps réel

85

Ordonnancement : ordonnancement par priorité

Transmettre les paquets de la file avec la plus haute priorité

- plusieurs *classes*, avec différentes priorités
  - La classe peut dépendre du marquage ou d'autres infos contenues dans l'en-tête, par ex. Adresse IP source/dest, numéros de port, etc..



86

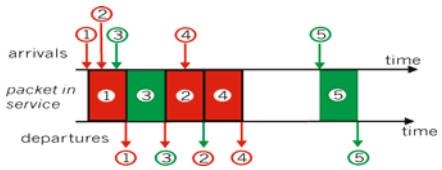
Ordonnancement : ordonnancement par priorité

- Avantages
  - Simple
- Inconvénients
  - Les files de faible priorité risquent de ne pas avoir de service
  - Protection unidirectionnelle
    - Trafic urgent non influencé par le BE
  - Pas de garantie de BP ou de délai

87

Ordonnancement : ordonnancement round robin

- Parcourt les files des différentes classes de manière cyclique, en servant 1 paquet de chaque classe (s'il y en a)
- WRR : Weighted Round Robin
- Evaluation
  - Faible complexité, isolation des flots, équité entre les paquets (paquets de taille variée => flots avec des gros paquets ont une plus grosse part), difficile d'allouer de la BP précisément (garantie de délai et de BP ?)



88

Ordonnancement : Fair Queuing

- Émule le round robin par bit pour chaque flot
- Principe
  - Estampille temporelle sur chaque paquet avec le temps de transmission du dernier bit
  - Sert les paquets avec la fin de service la plus proche

- Soit R(t) le nombre de cycles de service
- Soit l'estampille temporelle de fin de service F_i^k pour le paquet k du flot ii

$$F_i^k = S_i^k + L_i^k$$

$$S_i^k = \max\left(F_i^{k-1}, A_i^k\right)$$

Début du service

Arrival

$$A_i^k = R(d_i^k)$$

Scheduling state

- Un flow est actif (backlogged) tant que

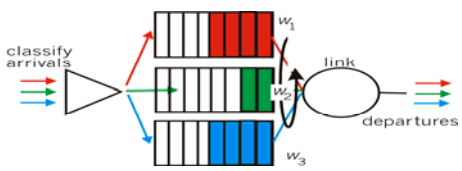
$$F_i^k > R(t)$$

- La durée d'un cycle est proportionnelle au nombre de flots actifs

89

Ordonnancement : Weighted Fair Queuing

- Round Robin généralisé
- Chaque classe obtient une quantité de service pondérée à chaque cycle



90

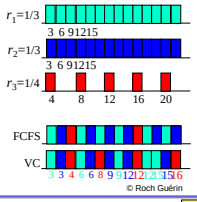
Ordonnancement : Virtual Clock

- Émulation TDM
  - timestamp de fin de service dans TDM
  - Le Timestamp dépend du débit alloué
- Pour chaque flot i, un débit moyen : Ri
-

$$VC_i^k = S_i^k + \frac{L_i^k}{R_i} \quad \text{with} \quad S_i^k = \max\left(VC_i^{k-1}, A_i^k\right)$$

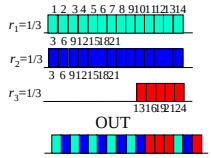
$$A_i^k = a_i^k$$

  - Ordre de service
    - Ordre des timestamps



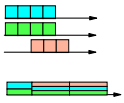
Ordonnancement : Virtual Clock

- Évaluation
  - Garantie de délai : flot régulé
  - Pénalise l'utilisation de rafales
    - Si un flot a envoyé une rafale, il se peut qu'il doive rester oisif pendant une longue période
  - Inéquitable
    - Utilise le temps réel



Ordonnancement : GPS (Generalized Processor Sharing)

- Équité Max-Min
- Principe
  - Modèle de service
    - Unité de service infinitésimale (tous les flots servis en même temps)
    - Cas idéal, infaisable



  - Service reçu proportionnel au poids du flot

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} = \frac{\phi_i}{\phi_j}$$

ϕ nombre réel positif

Intervalle de temps

Ordonnancement : GPS (Generalized Processor Sharing)

- g_i: taux de service pour le flot I de poids ϕ_i

$$g_i = \frac{\phi_i C}{\sum_{j=1}^{N} \phi_j} \qquad \text{Si} \quad \sum_{j=1}^{N} \phi_j \le 1$$

Garantie de BP absolue

  - Service équitable pour :

$$\phi_i = \frac{1}{N}$$

- Délai maximum
  - Si tous les flots i sont régulés (σ_i,ρ_i)

$$D_i = \frac{\sigma_i}{g_i} \le \frac{\sigma_i}{\rho_i} \quad \Rightarrow \quad g_i \ge \rho_i$$

- Évaluation
  - Garantie de BP
  - Fair share de la bande passante excédentaire
  - Work conserving
  - Un système réel utilise des paquets
    - Les services de paquets doivent approximer le modèle fluide
    - Le service d'un paquet induit une inéquité

Ordonnancement : PGPS-WFQ

- Principe
  - Détermine le temps de fin de service d'un paquet s'il avait été servi avec GPS
    - Émulation de GPS
  - Similaire au fair queuing mais notion de temps virtuel
  - **Temps virtuel : temps de service en GPS**
    - Plus rapide ou plus lent en fonction du nombre de flots actifs
    - Temps virtuel = temps réel quand tous les flots sont actifs
    - En temps virtuel, le service reçu par chaque flot reste constant
    -

$$A_i^k = V(a_i^k)$$

$$pas = \frac{1}{\sum_{i=1}^{N} \phi_i}$$ — Nb de flots actifs

  - Timestamp de fin de service

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i} \quad \text{with} \quad S_i^k = \max\left(F_i^{k-1}, A_i^k\right)$$

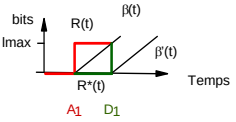
Ordonnancement : PGPS-WFQ

- Caractéristiques
  - Délai du paquet p en PGPS comparé à GPS

$$F_{PGPS}^p - F_{GPS}^p \le \frac{l_{max}}{C}$$

  - Le délai additionnel du au service de paquet est au plus l_{max}
    - Graphe de service

$$\beta'(t) = \left[\beta(t) - l_{max}\right]^+$$



- Pour un flot régulé (σ_i, ρ_i)
  - Délai de bout en bout
    - Pour m noeuds

$$D_i = \frac{\sigma_i}{g_i} + \frac{(m-1)L_{max}}{g_i} + \sum_{j=1}^{m} \frac{P_{max}}{C_j}$$

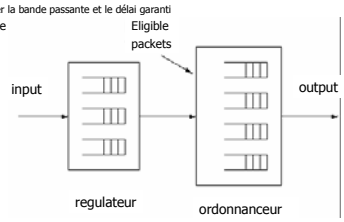
Ordonnancement : PGPS-WFQ

- Évaluation
  - Difficile de maintenir le temps virtuel de GPS en mode paquet
    - Mis à jour constamment
  - Calcul complexe de V(t) quand un nouveau paquet arrive
    - Effet de cascade

  - 1 état par flot
  - Nombreux algorithmes pour simplifier le calcul de V(t)

Ordonnancement : Service à débit contrôle

- Idée
  - Séparer la bande passante et le délai garanti
- Architecture



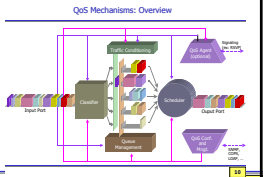
- Régulateur
  - Retient les paquets jusqu'à ce qu'ils soient éligibles
- Ordonnanceur
  - Choisit l'ordre des paquets éligibles
    - Indépendemment du débit

Ordonnancement : propriétés des contrôleurs de débit

- garanties
  - Débit
  - Délai
  - Contrôle de la gigue
  - sporadicité
- Mais
  - Pas d'utilisation de la bande passante en excès
  - Non-Work conserving

Provisionnement de la QoS

- Le provisionnement de la QoS requiert la définition de :
  - Service Level Agreement (SLA), qui contient en particulier :
    - Les cibles de QoS (délai borné ? Faibles pertes ? Etc.)
    - Le trafic concerné (tout le trafic du client, seulement la VoIP etc…)
  - Traffic Conditoning Agreement (TCA)
    - Conditions d'application du SLA (ou clauses du contrat), qui peuvent imposer en particulier des restrictions sur le volume de trafic qui peut être accepté
- Le contrat de trafic (SLA/TCA) peut alors être implémenté en utilisant :
  - Le contrôle d'admission (basé sur le TCA)
  - La réservation de ressources (basée sur le SLA)
    - Gestion de file d'attente
    - Réservation de bande passante
    - …



Plan

- **Introduction**
  - Qu'est-ce que la QoS?
  - Le besoin de QoS
- **QoS : boîte à outils**
  - Classification, conditionnement de trafic, files d'attente …
  - Différentes approches pour la QoS
    - QoS garantie vs différenciée
- **Architectures de QoS**
  - Integrated Services (IntServ) et RSVP
  - Differentiated Services (DiffServ)

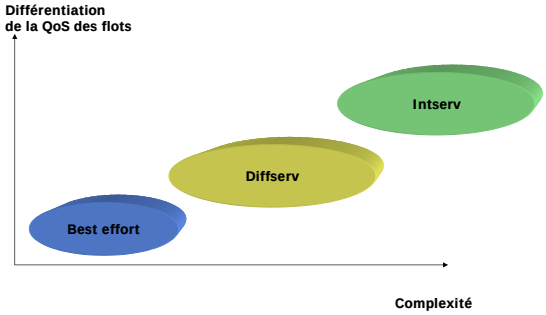
QoS différenciée / garantie (1)

- **La QoS garantie requiert la réservation de ressources** et les fonctionnalités de contrôle associées.
- Un autre type de QoS existe : la QoS différenciée.
  - Les flots sont agrégés en classes de trafic.
  - Pas de valeurs explicites données aux paramètres de QoS de bout en bout, seulement les "priorités relatives" entre les classe de trafic sont gérées dans les noeuds du réseau
  - Un paquet d'une classe de trafic avec une priorité temporelle supérieure devrait être traité "avant" le trafic avec une priorité temporelle inférieure (ordonnancement)
  - Le choix de rejet devrait dépendre de la priorité de perte de la classe

QoS différenciée / garantie(2)

- La QoS différenciée est **plus simple** à implémenter.
- Pour être supportée, elle a besoin :
  - D'une manière de reconnaître le niveau de priorité de chaque paquet de données et
  - D'algorithmes d'ordonnancement et de files d'attentes orientés priorité
- Malgré tout, c'est moins efficace que la QoS garantie en cas de congestion des ressources de plus haute priorité
- Le volume de trafic de plus haute priorité doit être contrôlé afin de s'assurer que l'approche Diffserv reste efficace

Services



Architecture IntServ

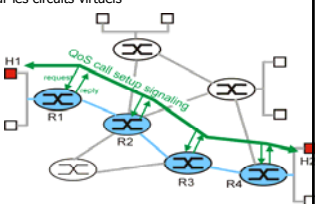
Services intégrés de l'IETF

- Architecture pour fournir des garanties de QoS dans des réseaux IP pour des sessions applicatives individuelles (flots)
  - Avec une emphase spécifique sur les applications multimédia (voix, vidéo)
- Réservation de ressources : les routeurs maintiennent des états des ressources allouées, des demandes de QoS
- Admettent / refusent de nouvelles requêtes :

Question : est-ce qu'un flot nouvellement arrivé peut être admis avec des garanties de performance sans violer les garanties de QoS des flots déja admis ?

Architecture Intserv/RSVP

- Qui ?
  - 2 groupes de travail à l'IETF
- IntServ (Integration Services)
  - Modèle pour fournir un ensemble de services
    - À une session applicative
    - Utilise la réservation des ressources et les états dans les routeurs
  - Composants pour fournir ces services
  - Démarré en 1994, inspiré des travaux sur les circuits virtuels
- RSVP (Resource Reservation Protocol)
  - Protocole de signalisation (RFC 2205)
  - Informe le réseau des besoin applicatifs
  - Requiert des réservations IP
  - Vu comme LE protocole de réservation d'Intserv



IntServ : caractéristiques principales

- Granularité fine (isolation des flots) : 1 SLA/TCA par flot
  - Flot = source @, dest @, numéros de port
  - Description de flot
    - Spécification de trafic, QoS demandée, règles pour identifier le flot
- Contrôle d'admission
  - Vérifier que la QoS requise peut être produite sans déranger les flots existants
- Classification (classifieur de paquets)
  - En fonction de la QoS requise
- Ordonnancement
  - Pour respecter la QoS demandée
- Protocole de signalisation (RSVP)
  - Alloue les ressources nécessaires pour le service.
- Paramètres de QoS de granularité fine disponibles
- Réservations dynamiques : plan de contrôle avec un protocole de signalisation spécifique (RSVP)
- QoS de bout-en-bout (grâce au plan de contrôle)

Mécanismes du routeur

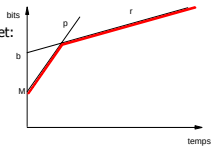
- Contrôle d'admission
  - Décide si un flot peut être accepté
    - Une session peut décrire son flot
      - Classe et niveau de service
      - Descripteur de flot
    - Chaque routeur vérifie que l'admission peut être assurée
      - En fonction de la demande
      - En fonction des ressources disponibles
    - COMMENTAIRE : différent du contrôle d'accès (*policing*)
- Pour les paquets
  - Classification
    - Associe un paquet à la réservation appropriée
  - Discipline de service
    - Ordonnancement
    - Équivalent au contrôle d'accès

Description de flot

- FlowSpec : décrit les caractéristiques du flot
  - Trafic envoyé
  - Service désiré
    - Delai
    - débit
- FilterSpec : identifie la (les) sources (s)
  - IPv4 : adresse Source et numéro de port
  - IPv6: adresse Source address et identifiant de flot
- Session : identifie la (les) destination(s)
  - Adresse Destination, protocol ID, numéro de port

FlowSpec

- Tspec : TSPEC= (r, b, p, m, M)
  - r: (débit moyen) mean rate
  - b: sporadicité (burstiness)
  - p: débit max (peak rate)
  - M: taille max de paquet (MTU)
  - m: taille min de paquet
    - Les paquets plus petits que m sont considérés de taille m => pénalise les petits paquets
  - Définit l'enveloppe du trafic envoyé
  - Implémenté sous forme de token bucket:



- Rspec: service requis
  - controlled-load : pas de valeur
  - Garanti : délai maximum

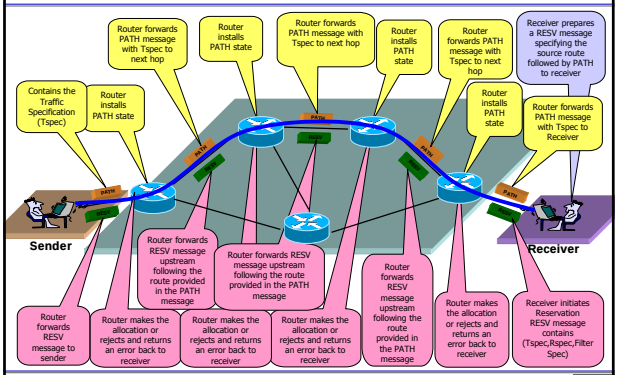
RSVP

- **Protocole de signalisation** pour la réservation de ressources sur un chemin donné
- Un flot utilise RSVP pour **demander une qualité de service (QoS) spécifique** de la part du réseau,
  - RSVP transporte la requête à travers le réseau,
  - À chaque noeud, RSVP essaie de faire une **réservation de ressources** pour le flot
- Protocole Soft State
  - Les terminaux **rafraîchissent** périodiquement le soft state de la réservation
  - En l'absence d'un rafraîchissement, l'état RSVP dans les routeurs expirera et sera effacé
- Protocole de signalisation orienté récepteur (paradigme multicast)
  - **Flots de données unidirectionnels**. Deux réservations indépendantes pour un flot bidirectionnel
- Conçu pour les flots multicast (1 vers plusieurs)
  - RSVP peut être utilisé à la fois pour des flots **unicast et multicast**
  - RSVP est un protocole de réservation **orienté récepteur**.
    - Les messages de réservation sont envoyés par les récepteurs et fusionnent alors qu'ils progressent dans l'arbre multicast
- Différents styles de réservation
- Adaptation aux changements de route
- De bout en bout
- Indépendant du service
- **Aucun mécanisme de routage n'est implémenté dans RSVP**.
- Pas de contrôle d'admission, pas d'autorisation, pas de discipline de service (réservation = algorithme local)
- RSVP fonctionne au-dessus d'IP (IPv4 et IPv6). Il peut aussi fonctionner sur UDP

Messages RSVP

- 2 principaux types de messages:
  - **PATH** (de la source vers les récepteurs)
    - Transporte les caractéristiques du trafic de la source et l'identification du flot
    - Fixe le chemin inverse pour les messages RESV
    - Collecte les informations sur l'état du réseau (BP disponible, délai de propagation)
    -
  - **RESV** (des récepteurs vers la source)
    - Envoyé sur demande du récepteur

RSVP et IntServ



Tspec, Rspec : Traffic, Receiver Specification

Le message *Path*

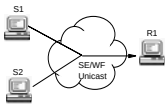
- Le message *Path* est le message envoyé par une source d'un arbre de distribution multicast aux récepteurs de l'arbre
- Les principales fonctionnalités du message *Path* sont :
  - Transporte la **description du trafic que la source va envoyer**
  - Transporte la **description de la capacité des noeds le long du chemin**. Chaque noeud supportant RSVP insère ou modifie l'information contenue dans ce champ en fonction de ses possibilités
  - Détermine le **chemin retour d**es messages *Resv* messages que les récepteurs enverront à réception du message *Path*. Ce chemin inverse est aussi maintenu dans le soft state des routeurs du chemin
- Transporte aussi l'ADSPEC
  - Mis à jour à chaque routeur et utilisé pour calculer le délai de bout-en-bout
  - Contient le nombre de saut, la BP sur le chemin, un délai fixe, la MTU, C_{tot}, D_{tot}

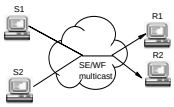
Le message *Resv*

- Envoyé par les récepteurs d'un arbre de distribution multicast pour effectuer une réservation de ressources dans le réseau
- **Suit le chemin inverse** tracé par le message *Path*
- **Fusionne aux embranchements** de l'arbre multicast
- Plusieurs **styles de réservation** existent quand il y a plusieurs sources dans l'arbre multicast
  - Wildcard Filter (WF) : plusieurs émetteurs wildcarded partagent la même réservation (un seul parle à la fois)
  - Fixed Filter (FF) : réservations distinctes par émetteur (vidéo)
  - Shared Explicit (SE) : liste explicite des émetteurs qui partagent la même réservation
- Les paramètres de réservation **dépendent du service intégré demandé**
- **Transporte le RSpec**
  - **Peut être différent du TSpec**

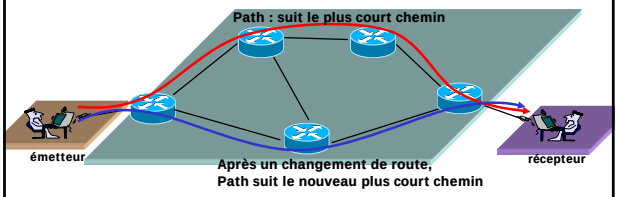
Styles de réservation

- Déterminent les règles de fusion des messages RESV
- Dédiée
  - Une réservation par émetteur (*Fixed Filter*)
  - Fusionne les réservations pour une session
    - Par émetteur
- Partagée
  - Une réservation pour un ensemble d'émetteurs
    - tous (*Wilcard Filter*)
    - Identification explicite (*Shared Explicit*)
  - Réservation conforme à la demande la plus haute de la session

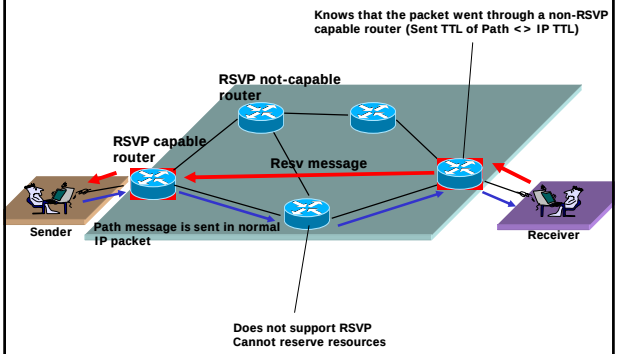




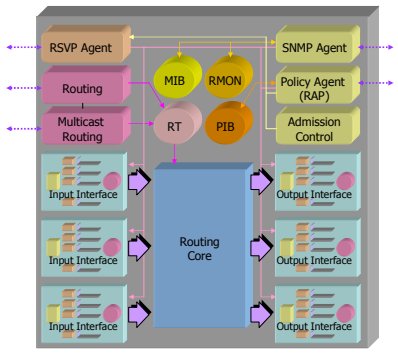
RSVP et changements de routes



Nuages non-RSVP

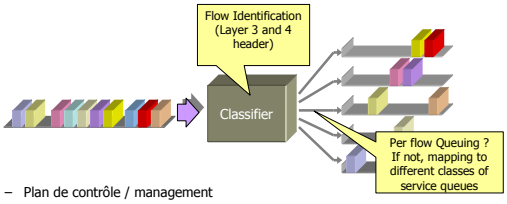


IntServ : Vue globale



IntServ : Complexité

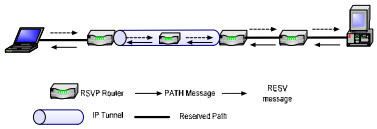
- Questions d'implémentation
  - Isolation des flots avec une granularité fine



  - Plan de contrôle / management
    - Les routeurs traditionnels manipulent seulement des agrégats de route (mas des micro-flots)
    - Dans IntServ, un protocole de signalisation est requis
    - Architecture de police aussi nécessaire pour le contrôle d'accès
      - Contrôle d'accès local (basé sur les ressources disponibles)
      - Décisions de police centralisées

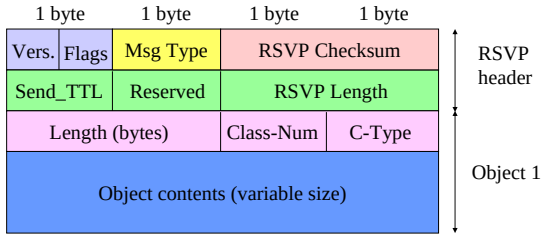
Le concept de session de service

- Une session est explicitement définie par une **adresse destination** (unicast ou multicast), un **protocole de transport** et un **numéro de port destination**
  - <session> = <DstAddr, PID, DstPort>
  - Problèmes avec les tunnels IP



- Le **session id** permet de corréler les flots de contrôle RSVP d'une session donnée (flots *Path* and *Resv*) et identifie un contexte particulier (incluant les réservations soft et la route inverse) pour cette session à chaque noeud du chemin

Messages RSVP
format commun



Facile de définir de nouveaux messages

MessagesRSVP
Champs de l'en-tête

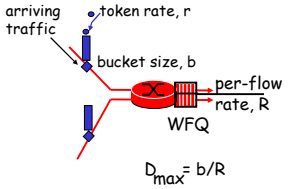
- **Vers** : actuellement, RSVP version 1
- **Flags** : par défini pour l'instant
- **Msg Type** :
  - 1=Path
  - 2=Resv
  - 3=PathErr
  - 4=ResvErr
  - 5=PathTear
  - 6=ResvTear
  - 7=ResvConf
- **RSVP Checksum** : checksum calculé sur la totalité du message RSVP
- **Send_TTL** : utilisé pour détecter les noeuds non-RSVP
- **RSVP Length** : longueur totale (en octets) du message RSVP
- Pour les objets
  - **Length** détermine la longueur de l'objet
  - **Class-Num** identifie l'objet
  - **C-Type** identifie les sous-classes de l'objet (ex : applicable à Ipv4 ou IPv6)
- Le nombre d'objets et leur type dépend du message RSV

QoS Intserv : Modèles de service [rfc2211, rfc 2212]

2 classes de service sont disponibles :

Best Effort
Service IP tradionnel. Utilisé par défaut
(ce service ne requiert pas l'utilisation de RSVP)

Service Controlled load
"une qualité de service se rapprochant de la QoS que le même flot recevrait de l'élément d'un réseau non chargé"



Guaranteed service:
- Arrivée de trafic dans le pire des cas : source policée par un leaky-bucket
- *Limite* simple (prouvable mathématiquement) sur le délai
- Classe prévue pour fournir un service à parte faible / nulle et un délai limité aux applications sensibles telles que les applications temps réel

Evaluation

- L'architecture Int-serv/RSVP introduit des aspects de QoS dans le domaine IP sans nécessiter de nouvelle technologie (ex : ATM) ⇒ c'est un important **facteur de continuité**
- Int-serv/RSVP est basé sur les flots ⇒ ceci lui confère une **bonne granularité** mais n'est **pas très scalable**
  - Int-serv/RSVP sera difficile à utiliser chez un gros ISP ou un backbone (il faut de l'agrégation)
  - Cela peut être utilse dans l'Intranet d'une grosse entreprise où l'on peut avoir de la congestion
- Aujourd'hui, RSVP devient un protocole de signalisation générique pour établir n'importe quel flot pour
  - MPLS, SDH, layer 4

Architecture DiffServ

Services différenciés de l'IETF

Problèmes avec Intserv:

- Passage à l'échelle : signalisation, maintien d'états par flot au niveau des routeurs difficile avec un nombre élevé de flots
- Modèles de Service fluides : Intserv possède seulement 2 classes. On veut aussi des classes de service « qualitatives »
  - Distinction de services relatifs : platine, or, argent…

- MAIS
  - Besoin de rester pragmatique étant donnés la capacité des routeurs actuels et les problèmes d'échelle de l'Internet

Vue globale de l'approche DiffServ

- Fonctions simples dans le cœur du réseau, fonctions relativement complexes au niveau des routeurs de bordure (ou les hôtes)
- Ne définissent pas de classes de service, fournissent des composants fonctionnels pour construire des classe de service

- Propriétés principales
  - Faible granularité : pas de manipulation de micro-flots. QoS offerte à des agrégats de trafic.
    - Simple classification des flots
    - Nombre d'états limité pour la gestion de la QoS (par rapport à IntServ)
  - Complexité en bordure du réseau.
    - Marquage de trafic au niveau du nœud entrant.
    - L'implémentation DiffServ ne devrait nécessiter que des mises à jour simples au niveau des routeurs de cœur de réseau
  - Pas de plan de contrôle. Pas de protocole de signalisation. La QoS est offerte au moyen d'un dimensionnement de réseau adéquat

Vue globale de l'approche DiffServ

- Mode hiérarchique pour la gestion des ressources
  - Intra-domaine
    - Responsabilité des ISP : configuration et dimensionnement des classes
  - Inter-domaine
    - Contrat de service
      - Entre 2 domaines DS
      - Entre un client et un domaine DS
    - Contrat :
      - Caractérisation du trafic pour le client
      - Garantie de QoS pour le fournisseur
- Service du réseau
  - Police de conditionnement
    - Modifier le trafic pour qu'il respecte les règles (contrat)
  - PHB : comportement d'un routeur
    - Par paquet
    - Service d'un élément de réseau
- DiffServ ne décrit pas
  - Les mécanismes pour les PHBs
  - Le nombre de classes
  - Les caractéristiques des classes

La bordure du réseau

- Bordure
  - Hôtes supportant DiffServ
  - Premier routeur du domaine DS
- Classification
  - Marquage à l'entrée en fonction du SLA
- Conditionnement
  - Actions sur le trafic entrant
  - Contrôle d'accès
    - Limite la quantité de trafic à l'intérieur d'une classe
    - Évite la congestion

Le cœur du réseau

- Forwarding
  - Comportement appliqué à un paquet en fonction du marquage (qui identifie la classe de service)
- Dimensionnement
  - Pour fournir de la QoS, la classe doit être provisionnée

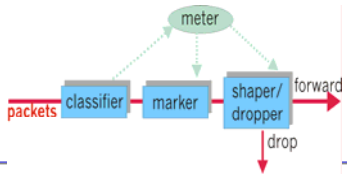
Classification

- En bordure
  - Identifier les paquets en fonction du SLA
  - Classification multi-champs
    - Marquage dans le paquet IP

- Au coeur
  - Classification basée sur le marquage
  - Marquage = index d'un comportement
- Identification de classe
  - Champ "octet DS"
    - Au lieu du ToS(IPv4) ou classe (IPv6)
  - Valeur du DS codepoint
    - Identifie le comportement dans un routeur

Conditionnement

- Action décrite dans le contrat
- Actions
  - marqueur
    - Modifie le *codepoint* => le PHB
    - Pour allouer un niveau de priorité différent
  - métreur
    - Vérifie que le flot est conforme à un profil
  - dropper
    - Détruit un datagramme, modifie les caractéristiques sémantiques d'un flot
  - shaper
    - Modifie les caractéristiques temporelles des flots



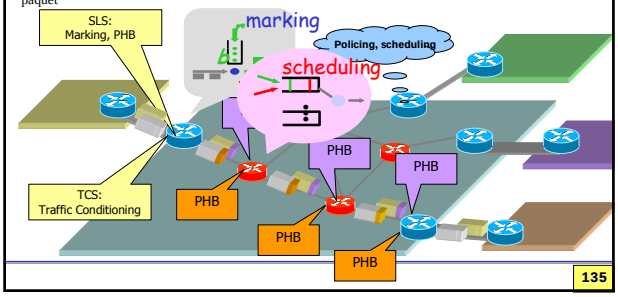
Architecture DiffServ (1)

Routeur de bordure :

- Management du trafic par flot
- marque les paquets (champ DS) comme in-profile et out-profile
- PHB : Per Hop Behavior associé au champ DS du paquet

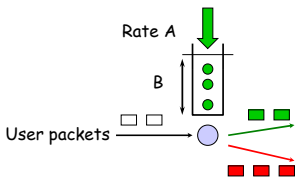
Routeur de coeur :

- Management de trafic par classe
- buffering et scheduling basés sur le marquage en bordure
- Préférence données aux paquets in-profile



Marquage de paquet par le routeur de bordure

- Profil : débit pré-négocié A, taille de bucket B
- Marquage de paquet en bordure basé sur un profil par-flot



Utilité possible du marquage :

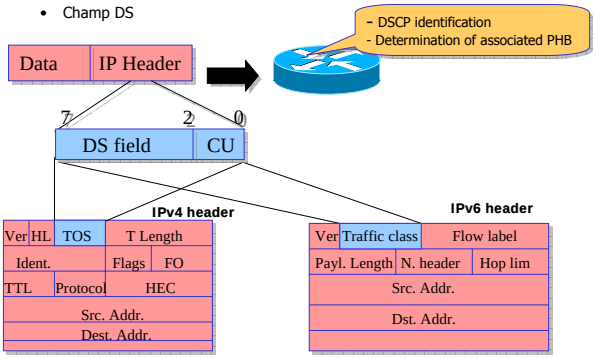
- **Marquage basé sur la classe** : paquets de classes différentes marqués différemment
- **Marquage intra-classe** : partie conforme du flot marquée différemment de la partie non-conforme

Classification et Conditionnement

- Le paquet est marqué dans le champ Type of Service (TOS) en IPv4, et Traffic Class en IPv6
- 6 bits utilisés pour le Differentiated Service Code Point (DSCP) qui déterminent le PHB que le paquet recevra
- 2 bits sont pour l'instant inutilisés

Champ DiffServ en IPv4 / IPv6

- Champ DS



Forwarding (PHB)

- Le PHB résulte en des performances de forwarding différentes observables (mesurables)
- Le PHB ne spécifie pas quels mécanismes utiliser pour assurer le comportement PHB requis
- Exemples:
  - LA classe A obtient x% du lien sortant pendant des intervalles de temps d'une longueur spécifique
  - Les paquets de la classe A partent avant les paquets de classe B

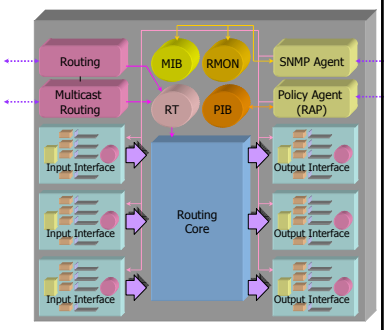
Forwarding (PHB)

- **Pas d'état dans les routeurs** !!!
- Services traditionnels
  - « Best-Effort » (BE) PHB.
- Expedited Forwarding: Le débit de départ d'un paquet d'une classe est égal ou dépasse un débit spécifié
  - « Service Premium » (pour les flots interactifs)
  - PHB avec peu de pertes, délai garanti.
  - Lien logique avec un débit minimum garanti
  - Pas de trafic hors profil
- Assured Forwarding: 4 classes de trafic avec pour chacune une bande passante minimale garantie
  - Chacune avec 3 partitions de drop preference
  - 4 priorités temporelles (ne pouvant pas être modifiées par le réseau), et 3 priorités spatiales (pouvant être modifiées par le réseau)
  - Assurance ≠ garantie
  - Du trafic opportuniste (hors profil) peut être admis

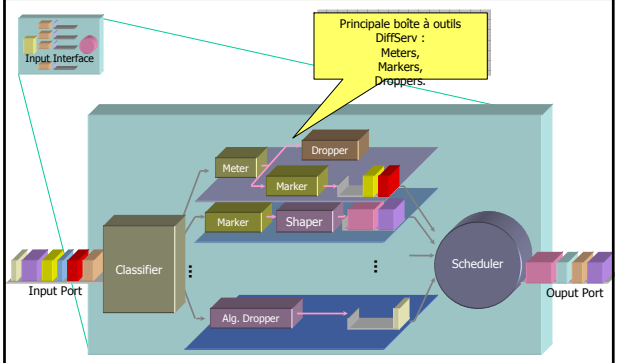
Architecture de routeur DiffServ (1)

différentiation de service à l'intérieur des interfaces

La complexité des interfaces dépend du vendeur (pas de standard)



Architecture de routeur DiffServ (2)



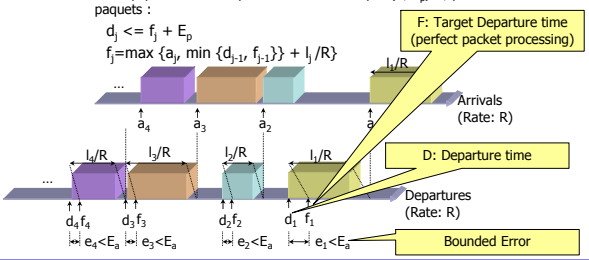
Boîte à outils DiffServ

EF PHB : Définition

- Expedited Forwarding
  - DSCP: 101110
  - Classe unique sans drop precedence disponible
- Définition :
  - Un équipement EF compliant est caractérisé par (R, E_p) si, pour tous les paquets :

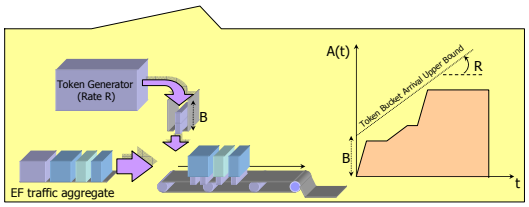
$d_j <= f_j + E_p$

$f_j = \max \{a_j, \min \{d_{j-1}, f_{j-1}\}\} + l_j/R\}$



EF PHB : limites de QoS

- Avec la définition précédente, le délai est limité par D avec
  D= B/R + E_p
  - R: débit de service EF sur l'interface de sortie,
  - Ep : terme d'erreur par rapport au traitement idéal,
  - B: profondeur du Token Bucket (de débit R) qui correspond à l'arrivée des paquets (pour l'agrégat EF)

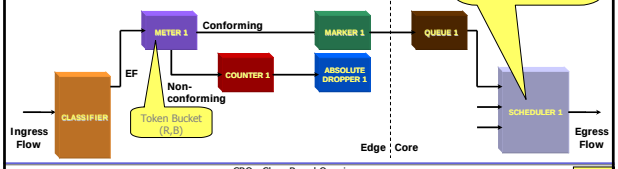


- Limites sur la gigue :
  J= B/R + $E_{p\ variable}$ avec: $E_p = E_{p\ fixed} + E_{p\ variable}$.

EF PHB : Implémentation

- Implémentation
  - Conditionnement de trafic :
    - Peak Rate Metering : un token bucket est l'outil de conditionnement de trafic le plus approprié par définition de EF.
    - Comme aucune drop precedence n'est disponible, le trafic hors profil doit être soit jeté soit « déclassé » (best-effort ou bulk traffic)
  - PHB
    - Pas de management de file d'attente spécifique requis
    - L'EF est par essence basé sur l'ordonnancement

Strict Priority:
Max Priority for EF
Other (CBQ, WFQ, ...):
Minimum Bandwidth share reserved for EF aggregate.



CBQ : Class-Based Queuing

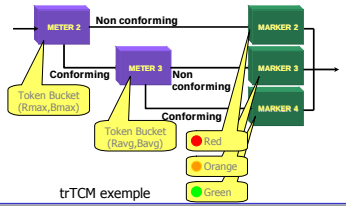
AF PHB : Définition

- Assured Forwarding PHB Group
  - 4 classes de services, avec Class 1 ≥ Class 2 ≥ Class 3 ≥ Class 4
  - 3 Drop precedence disponibles

Drop precedence	Class 1	Class 2	Class 3	Class 4
Low (Yellow)	001010	010010	011010	100010
Medium (Orange)	001100	010100	011100	100100
High (Red)	001110	010110	011110	100110

AF PHB : Implémentation

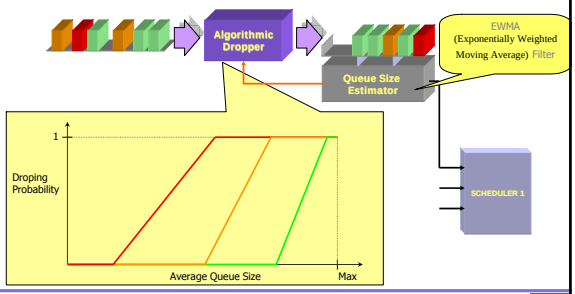
- Conditionnement de trafic
  - Possibilité de contrôler la quantité de trafic qui entre dans le réseau dans une classe spécifique avec une drop precedence spécifique.
  - Exemples :
    - srTCM (Single Rate Two Color Marker)
    - trTCM (Two Rate Three Color Marker)
    - tswTCM (Time Sliding Window Three Color Marker), ...



trTCM exemple

AF PHB : Implémentation

- Gestion de file d'attente :
  - « Active Queue Management » obligatoire.
    - Ex: RIO (RED In Profile/Out Profile)



Evaluation (1)

- L'architecture Diff-serv est relativement **immédiate** avec un certain nombre de composants (PHBs) permettant de construire une large variété de services différenciés
- Une certaine **complexité** demeure concernatn
  - L'implémentation des traffic conditioners au niveau des nœuds de bordure
  - Le choix et la configuration des algorithmes d'ordonnancement à l'intérieur des nœuds
- L'approche DiffServ pour la provision de QoS pour le trafic IP est une méthode concurrente à d'autres approches telles que
  - Int-serv/RSVP
  - MPLS
  - Marquage par priorité relative (champ IP Precedence)

Evaluation (2)

- Diff-serv est
  - **Plus scalable** que Int-serv/RSVP mais moins granulaire
  - **Plus simple** à implémenter que MPLS étant donné que Diff-serv est encore basé sur le routage tandis que MPLS est basé sur les étiquettes (càd que DiffServ a moins de contraintes sur le hardware des nœuds du réseau)
  - une **extension** aux traditionnelles approches de priorité / marquage de service déjà utilisées dans certaines parties de l'Internet
- Diff-serv peut aussi être **complémentaire** à d'autres architectures :
  - Diff-serv peut être utilisé pour agréger des flots Int-serv/RSVP flows au cœur du réseau
  - Diff-serv peut utiliser MPLS comme une technologie intra-domaine alternative où un BA peut être mappé à un chemin commuté par étiquette spécifique à travers le réseau

Résumé : améliorer la QOS dans les réseaux IP

Jusqu'ici : "faire le mieux du best effort"

Futur : Internet de nouvelle génération avec des garanties de QoS

- RSVP : signalisation pour les réservations de ressources
- Differentiated Services : garanties différentielles
- Integrated Services : garanties fermes

Routage inter-domaine

Plan

- **Introduction**
- Routage intra-domaine
- Routage inter-domaine
  - BGP

« Philosophie » Internet

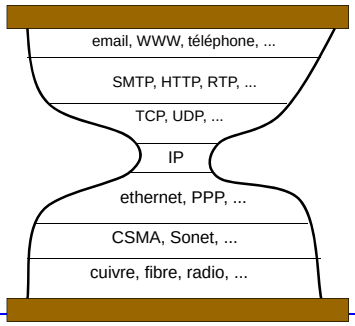
- Intelligence aux extrémités du réseau
- Forwarding des paquets le plus rapide possible
- Pas d'état par paquet / par flot dans le réseau
- Pas de contrôle d'accès

Tenter de concevoir et de faire évoluer le réseau en suivant ces principes

Qu'est-ce que l'Internet ?

- Suite de protocoles IP, ainsi que les mécanismes et applications liés (standardisés par l'IETF)
- Réseau d'interconnexion international constitué d'ISPs, de réseaux d'entreprise et de campus, etc.

Le sablier Internet (Deering@IETF)



Internet – Interconnexion de réseaux

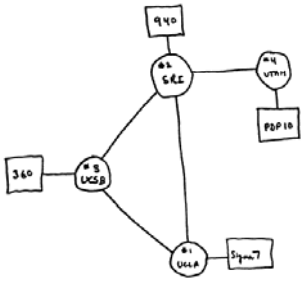
- Protocole Internet
  - Transmet un paquet unique d'un hôte vers un autre
  - Le paquet inclut l'adresse IP de l'émetteur et du récepteur
  - Les paquets peuvent être perdus, retardés ou déséquencés
  - Les hôtes gèrent la retransmission et le reséquencement des paquets
- Adresse IP
  - Adresses IP sur 32 bits divisées en octets (12.34.158.5)
  - Allouées aux institutions sous la forme de blocs continus ou préfixes
  - 12.34.158.0/24 est un préfixe sur 24 bits avec 2^8 adresses IP
  - Le routage des paquets IP est basé sur les préfixes

Caracteristiques de l'Internet

- L'Internet est
  - Décentralisé (confédération souple de pairs)
  - Auto-configuré (pas de registre de topologie global)
  - Sans état (informations limitées dans les routeurs)
  - Sans connexion (pas de connexions fixées entre les hôtes)
- Ces attributs contribuent
  - Au succès de l'Internet
  - À sa croissance rapide
  - … et à la difficulté à le contrôler !

Il était une fois…

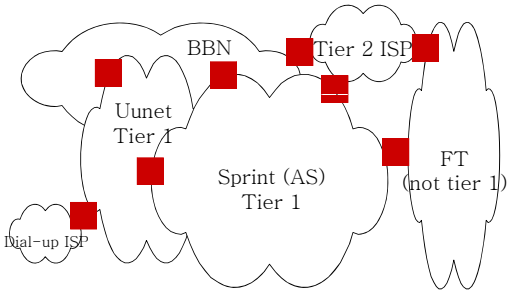
- Voilà ce qu'était Internet !



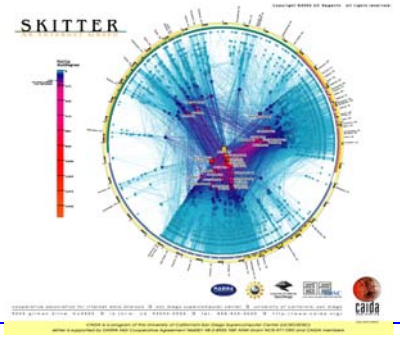
Architecture de l'Internet

- Divisé en Systèmes Autonomes (AS)
  - Régions correspondant à des contrôles administratifs distincts (~6000-7000)
  - Ensemble de routeurs et de liens gérés par une seule institution
  - Fournisseur de service, entreprise, université, …
- Hiérarchie de Systèmes Autonomes
  - Fournisseur de tier-A, de grande taille, backbone à l'échelle nationale
  - Fournisseur de taille moyenne avec un backbone plus petit
  - Petit réseau géré par une unique université ou entreprise
- Interaction entre Systèmes Autonomes
  - Topologie interne non partagée entre AS
  - … mais les AS voisins interagissent pour coordonner le routage

L'Internet



Connectivité des ISPs



Routage à travers l'Internet

- Intra-domaine
  - Un AS détermine le chemin d'un paquet à l'intérieur de son domaine
  - Utilisation d'un protocole (à état des liens)
  - Performances : question essentielle
- Inter-domaine
  - Les AS échangent des informations d'atteignabilité
  - Basé sur des politiques
  - Pas nécessairement le plus court chemin

Routage à travers l'Internet

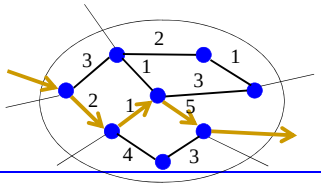
- Protocoles
  - Interior Gateway Protocols (IS-IS, OSPF, RIP, IGRP)
  - Exterior Gateway Protocols (BGP)
- Fonctionnement
  - IGP : trouve le "meilleur" chemin à travers un AS
  - BGP : Définit des règles de relations entre AS

Routage inter-domaine (entre différents AS)

- Les AS s'échangent des information d'atteignabilité
- Politiques locales pour la sélection de chemin (lequel utiliser ?)
- Politiques locales pour la propagation de route (Qui informer ?)
- Politiques configurées par l'opérateur du réseau de l'AS

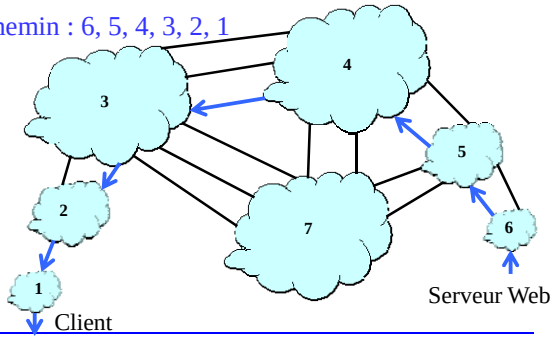
Routage intra-domaine (à l'intérieur d'un AS)

- Les routeurs échangent des informations pour apprendre la topologie
- Les routeurs déterminent le "prochain saut" pour atteindre les autres routeurs
- Sélection de chemin en fonction du poids des liens (plus court chemin)
- Poids des liens configurés par l'opérateur du réseau d'AS
- … pour gérer le flux de trafic



Systèmes Autonomes (AS)

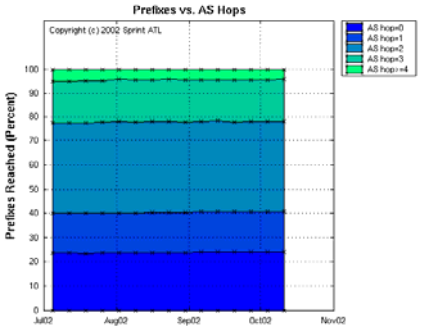
Chemin : 6, 5, 4, 3, 2, 1



Tiers

- Les ISPs de Tier 1 sont ceux qui ne louent pas de fibre et n'achètent pas de transit auprès d'autres ISPs
  - FT, DT, BT ne sont pas des Tiers 1
  - Le "Peering" ne s'applique qu'à des ISPs de même niveau
- Classification de la topologie Internet à partir des traversées d'AS : les tiers 1 sont en haut de l'arbre
  - Les Tier 1s ont un chemin plus court vers le reste de l'Internet

Atteignabilité Internet de Sprint



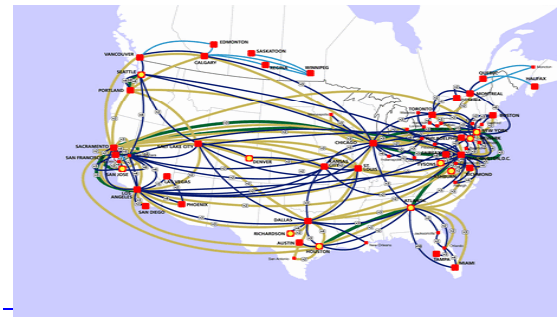
Backbones de Tier-1 actuels

- Technologies
  - IP over SONET (POS)
  - IP over ATM
  - IP over MPLS
- Topologies
  - Nombreux petits PoPs
  - Relativement peu de gros PoPs

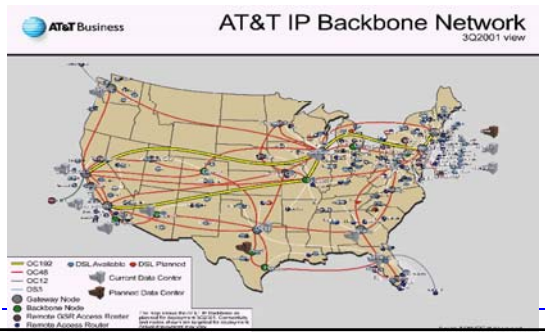
Topologies

- UUNet : MPLS/IP/ATM avec des petits PoPs
- AT&T : IP over Sonet avec beaucoup de petits PoP. Bientôt MPLS.
- Sprint : IP over Sonet avec des PoPs plus gros et moins nombreux

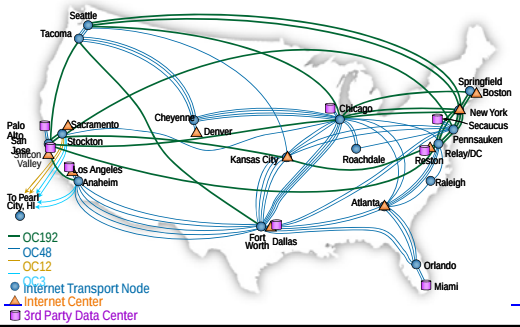
UUNet en Amérique du Nord



Backbone IP d'AT&T



Backbone IP de Sprint

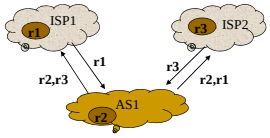


Clients, Fournisseurs, Pairs

- Public peers : NAP publics, pas de transit, disparaissent petit à petit (aux US).
- Peers : "pairs" (AT&T, UUNET, etc.) : pas de transit, peering gratuit s'ils sont assez gros, prix en fonction du volume sinon.
- Clients (tier 2, CDN, etc.) : transit, prix en fonction du lien.

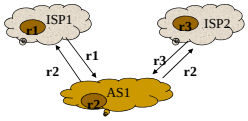
AS de Transit / Non transit

•***AS de transit*** : transporte du trafic de transit
  •Annonce ses propres routes ET les routes apprises par d'autres AS



AS de Transit / Non transit

•***AS Non-transit*** : ne transporte pas de trafic de transit
  • annonce ses propres routes et les routes provenant du transit (clients)
  • ne propage pas les routes apprises des fournisseurs et des non transit (pairs)



Routage inter-domaine

Historique

- Popularité
  - Au départ, BGP était assez peu connu, et utilisé par un petit nombre de gros IPSs
  - En 1995 (début de la popularité du Web), le nombre d'organisations utilisant BGP a énormément augmenté.
- 2 raisons pour la croissance de son usage et de son intéret :
  - Croissance significative du nombre d'ISPs;
  - Naissance d'organisations dont le succès dépend de la connectivité
- CIDR (Classless Inter-Domain Routing) introduit en 1995

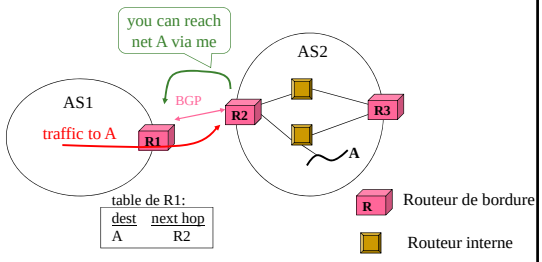
CIDR : Classless Inter-Domain Routing

- Format d'adresse <adresse IP /préfixe P>.
  Le préfixe concerne les P premiers bits de l'adresse IP.
- Idée : *utiliser l'agrégation* – fournir du routage pour un grand nombre de clients en annonçant un préfixe commun.
  - Ceci est possible parce que l'adressage est de nature hiérarchique
- Résumer l'information de routage réduit la taille des tables de routage, mais permet de maintenir la connectivité.
- L'agrégation est essentielle pour le passage à l'échelle et la "survie" de l'Internet

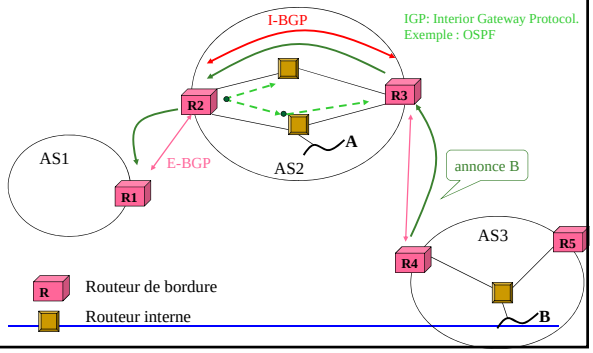
CIDR : matching du plus grand préfixe

- Puisque des préfixes de longueur arbitraires sont autorisés, il peut y avoir des préfixes recouvrants.
- Exemple :
  un routeur entend 124.39.0.0/16 d'un voisin
  et 124.39.11.0/24 d'un autre voisin
- Le routeur transmet le paquet en fonction de l'information de forwarding la plus précise, appelée *longest prefix match*
  - Un paquet avec l'adresse de destination 124.39.11.32 sera transmis en utilisant l'entrée /24 de la table.
  - Un paquet avec l'adresse de destination 124.39.22.45 sera transmis en utilisant l'entrée /16.

Objectif : partage d'information de connectivité

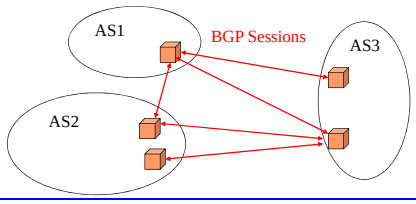


Protocoles de routage



Sessions BGP

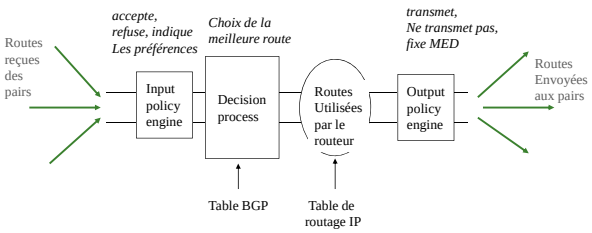
- Un routeur peut participer à plusieurs sessions BGP.
- *Initialement* … un noeud annonce TOUTES les routes qu'il veut faire connaître à ses voisins (il peut y en avoir >50K)
- *Pendant* … seulement informer les voisins des changements



4 messages de base

- *Open* :
  établit une session BGP (utilise le N° de port TCP 179)
- *Notification* :
  Rapporte les conditions non usuelles
- *Update* :
  Informe un voisin de nouvelles routes devenues actives
  Informe un voisin de nouvelles routes devenues inactives
- *Keepalive* :
  Informe un voisin que la connexion est toujours valide

Processus de routage



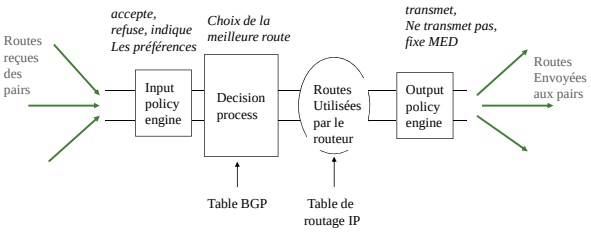
Configuration et politique

- Un noeud BGP n'a pas à accepter toutes les routes apprises de ses voisins. Il peut sélectivement accepter et rejeter des messages.
- Un noeud BGP sait quelles routes partager avec ses voisins. Il peut n'annoncer qu'une portion de sa table de routage à un voisin
- *Quoi accepter* de ses voisins et *quoi partager* avec ses voisins est déterminé par la politique de routage, spécifiée dans le fichier de configuration des routeurs.

Filtrage en amont - Input Policy Engine

- Le filtrage en entrée contrôle le trafic en sortie
  - Les filtres routent les mises à jour reçues des autres pairs
  - Filtrage basé sur les préfixes IP, l'AS-PATH, la communauté
  - Refuser un préfixe signifie que BGP ne veut pas atteindre ce préfixe via le pair qui a envoyé l'annonce
  - Accepter un préfixe signifie que le trafic vers ce préfixe peut être transmis au pair qui a envoyé l'annonce
- Manipulation d'attributs
  - Positionnement d'attributs pour les routes acceptées
  - Exemple : spécifier LOCAL_PREF pour établir des priorités entre plusieurs pairs qui peuvent atteindre une destination donnée

Processus de routage



Processus de décision BGP

1. Choisir la route avec le plus haut LOCAL-PREF
2. S'il y a plus d'1 route, sélectionner la route ayant l'AS-PATH le plus court
3. S'il y a plus d'1 route, sélectionner selon le type ORIGIN le plus petit, avec *IGP < EGP < INCOMPLETE*
4. S'il y a plus d'1 route, sélectionner la route ayant le MED le plus petit
5. Sélectionner le chemin de coût minimum vers le NEXT HOP en utilisant des métriques IGP
6. S'il y a plusieurs chemins internes, utiliser l'ID du routeur BGP pour trancher.

ATTRIBUTS BGP

LOCAL PREF

- Utilisé pour indiquer des préférences parmi plusieurs chemins pour un même préfixe n'importe où dans l'Internet.
- Valeurs supérieures préférées
- Échangé uniquement entre pairs IBGP. Local aux AS.
- Souvent utilisé pour sélectionner un point de sortie spécifique pour une destination particulière

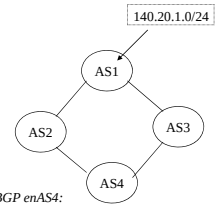


Table BGP enAS4:

Destination	AS Path	Local Pref
140.20.1.0/24	**AS3 AS1**	**300**
140.20.1.0/24	**AS2 AS1**	**100**

ATTRIBUTS BGP

- AS-PATH :
  - Liste d'AS à travers lesquels l'annonce pour un préfixe est passée
  - Chasque AS ajoute son N° d'AS à l'attribut AS-PATH lors de la transmission d'une annonce
  - Utile pour détecter et prévenir les boucles

Prefix	***Next hop***	***AS Path***
128.73.4.21/21	232.14.63.4	1239 701 3985 631

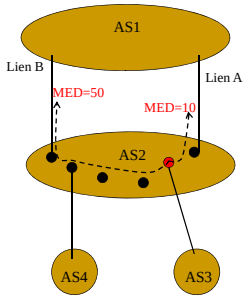
- ORIGIN :
  - Qui est à l'origine de l'annonce ? Où un préfixe a-t-il été *injecté* dans BGP?
  - IGP, EGP ou Incomplete (souvent utilisé pour les routes statiques)

ATTRIBUTS BGP

MED

Multi-Exit Discriminator

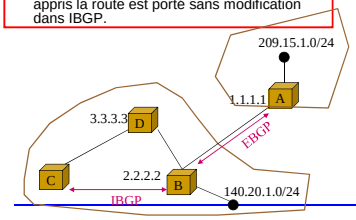
- Quand des AS sont interconnectés par 2 liens (ou plus)
- L'AS annonçant le préfixe fixe MED
- Permet à l'AS2 d'indiquer sa préférence
- L'AS recevant le préfixe utilise MED pour sélectionner le lien



ATTRIBUTS BGP

NETX HOP

- Pour une session EBGP, NEXT HOP = adresse IP du voisin qui a annoncé la route.
- Pour les sessions IBGP sessions, si la route à une origine interne à l'AS, NEXT HOP = adresse IP du voisin qui a annonce la route
- Pour les routes don't l'origine est externe à l'AS, le NEXT HOP du noeud EBGP qui a appris la route est porté sans modification dans IBGP.



BGP Table at Router C:

Destination	Next Hop
140.20.1.0/24	2.2.2.2
209.15.1.0/24	1.1.1.1

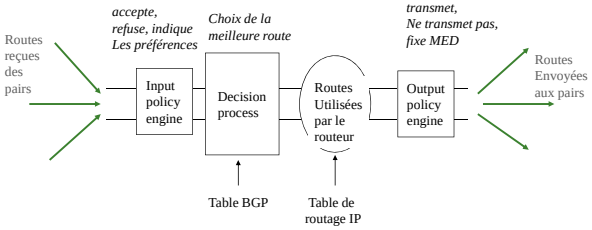
IP Routing Table at Router C:

Destination	Next Hop
140.20.1.0/24	2.2.2.2
2.2.2.0/24	3.3.3.3
3.3.3.0/24	Connected
209.15.1.0/24	1.1.1.1
1.1.1.0/24	3.3.3.3

Processus de décision BGP

1. Choisir la route avec le plus haut LOCAL-PREF
2. S'il y a plus d'1 route, sélectionner la route ayant l'AS-PATH le plus court
3. S'il y a plus d'1 route, sélectionner selon le type ORIGIN le plus petit, avec *IGP < EGP < INCOMPLETE*
4. S'il y a plus d'1 route, sélectionner la route ayant le MED le plus petit
5. Sélectionner le chemin de coût minimum vers le NEXT HOP en utilisant des métriques IGP
6. S'il y a plusieurs chemins internes, utiliser l'ID du routeur BGP pour trancher.

Processus de routage



Filtrage en aval -Output Policy Engine

- Le filtrage de sortie contrôle le trafic entrant
  - Transmettre une route signifie que les autres peuvent choisir d'atteindre un préfixe en passant par nous
  - Ne pas transmettre une route signifie que les autres doivent utiliser un autre route pour atteindre le préfixe
  - Cela peut dépendre sur le fait qu'il s'agit d'un pair E-BGP ou I-BGP
  - Exemple : si ORIGIN=EGP et que vous êtes un AS non-transit et que le pair BGP n'est pas client, alors pas de transmission
- Manipulation d'attributs
  - Fixe les attributs tels que AS_PATH et MED

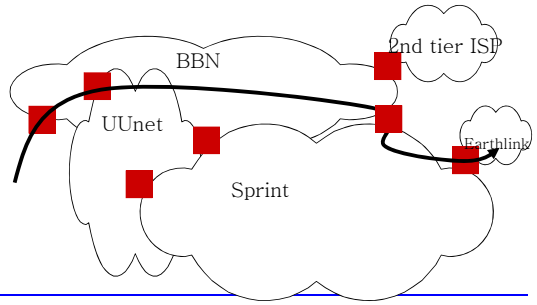
Annonce d'un préfixe

- Lorsqu'un routeur annonce un préfixe à l'un de ses voisins BGP :
  - L'information est valide jusqu'à ce qu'un routeur annonce explicitement qu'elle n'est plus valide
  - BGP ne nécessite pas le rafraîchissement de l'information
  - Si le noeud A annonce un chemin pour un préfixe au noeud B, alors B peut être sûr que A lui-même utilise ce chemin pour atteindre la destination.

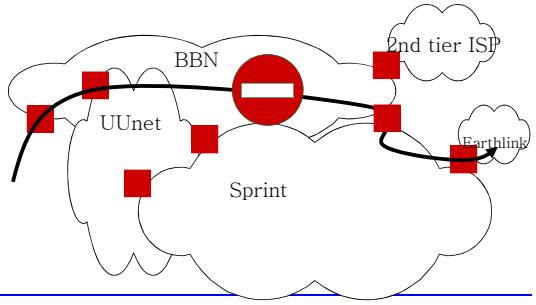
Quelques pratiques de routage intéressantes

- Routage Hot Potato
- Multi-homing
- Filtrage et dampening de route
- Agrégation de route

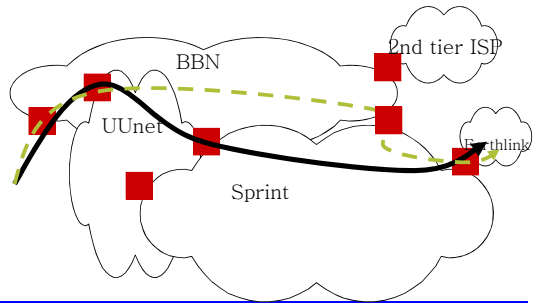
Routage Hot potato



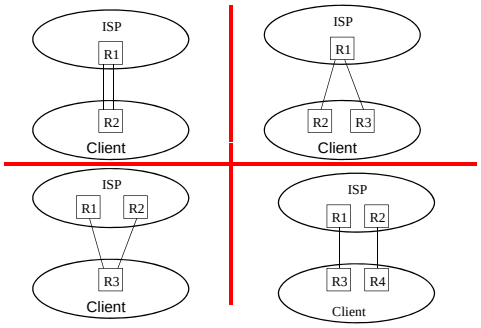
Routage Hot potato



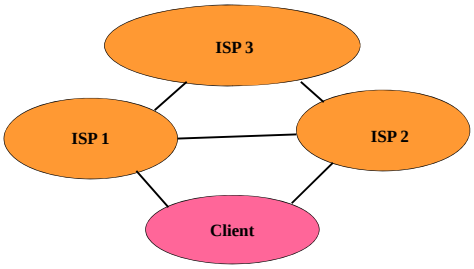
Routage Hot potato



Multihoming vers un même fournisseur



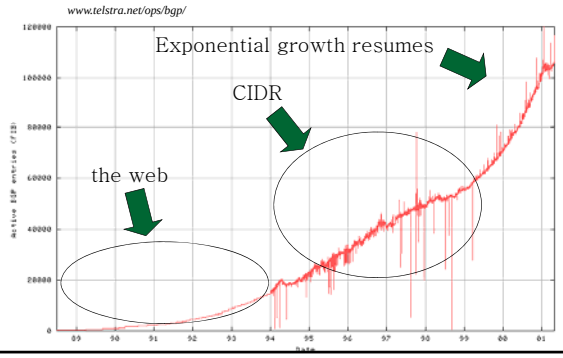
Multihoming vers plusieurs fournisseurs



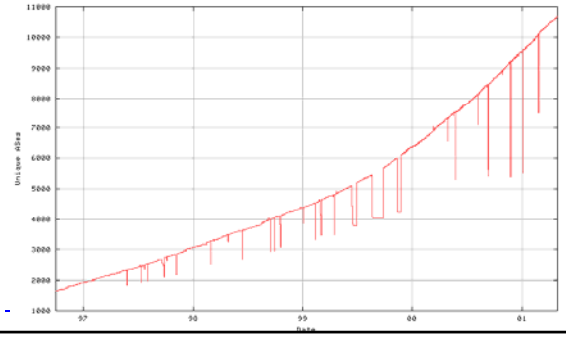
Motivations pour le multihoming

- Partage de charge
- fiabilité
- Indépendence vis-à-vis des fournisseurs

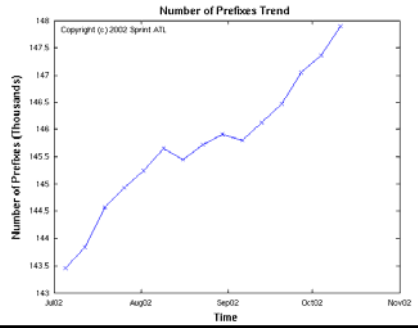
Croissance des tables BGP (1989-2001)



Croissance du nombre d'AS



Taille des tables BGP



Tendances de BGP

- chaque AS annonce de plus petits intervalles d'adresses
- /24 est le préfixe qui croît le plus dans la table (dans l'absolu).
- /24 - /31 est l'aire qui connait la plus grande croissance relative (essentiellement à cause du NAT)

Pratiques communes : filtrage

- Mécanisme par lequel on accepte/rejette une mise à jour de routage entrant/sortant
- Le filtrage peut s'effectuer selon
  - Le numéro de réseau (en utilisant des listes de préfixes)
  - L'AS-PATH (en utilisant des listes d'accès)

Pratiques communes : agrégation

- Lorsqu'un préfixe est contenu dans un autre, annoncer seulement le plus grand
- Exemple : un ISP a le bloc d'adresses 128.0.0.0/8 et son client a 128.32.0.0/16. L'ISP annonce seulement le bloc /8.
- Aide à contrôler la taille de la table de routage

Pratiques communes : agrégation

- Éviter les trous noirs : si un ISP annonce un bloc d'adresses mais ne contient pas tous les sous-blocs, alors ces sous-blocs manquants ne seront pas joignables.
- Limites de l'agrégation
  - multihoming
  - Un même AS peut avoir des blocs d'adresses non continus
  - Réticence à renuméroter l'espace d'adressage

Pratiques communes : Dampening

- Utilisé pour contrôler l'instabilité de la route
- Repérer le nombre de fois qu'une route a changé sur une période de temps
  - Les routes qui varient beaucoup sur une courte période de temps sont supprimées (non annoncées)
  - Une fois que les routes arrêtent de changer pendant suffisamment longtemps, elles sont rétablies (ré-annoncées)

Causes de l'asymétrie du routage

- Moins de 10% du routage est symétrique
  - Routage Hot potato
  - Pratiques BGP
  - Multi-homing
  - Équilibrage de charge

Résumé

- L'Internet est stable
  - Grande disponibilité et atteignabilité
  - Grande performance
- Mais il est fragile
  - Explosion BGP
  - Peu de contrôle sur ce qui se passe à l'intérieur
  - Résoudre les problèmes manuellement devient difficile
  - Chaque ISP est différent

Références

- Données fournies par :
  - Christophe Diot
  - Jeniffer Rexford
  - Matthias Grossglauser
  - Kavé Salamatian