Chapitre III: Mécanismes de la QoS

ChapII-Section B :Traffic shaping

ChapII-Section B :Traffic shaping

- Lorsqu'une connexion est établie, l'utilisateur et le sous réseau, c'est-à-dire le client et l'opérateur, se mettent d'accord sur un certain comportement du trafic pour un circuit.
- Cet engagement entre eux est souvent appelé le contrat de niveau de service, ou SLA (Service Level Agreement).

SLA = {

QoS requise,
Spécification de trafic,
Règles de traitement de paquets,

Coûts, pénalisation, bonus,
Aspects juridiques
...
}

ChapII-Section B :Traffic shaping

- Dans la pratique, cela se traduit par un client annonçant :

« Mon trafic est de ce type, pouvez-vous le transporter ? »

- Si l'opérateur est d'accord, il est confronté à un autre aspect du problème :

comment savoir si le client respectera ses engagements et que faire si ce n'est pas le cas ?

- Pour cela, il doit adopter une stratégie de surveillance du trafic (traffic policing)

ChapII-Section B :Traffic shaping

- La canalisation de trafic (traffic shaping), permet de réguler le trafic côté serveur, et non plus côté client.
- La canalisation du trafic concerne la régulation du rythme moyen auquel les données sont transmises, et par là même, des rafales.
- Cette technique limitent la quantité des données émises en une fois, mais pas le rythme auquel elles sont transmises.

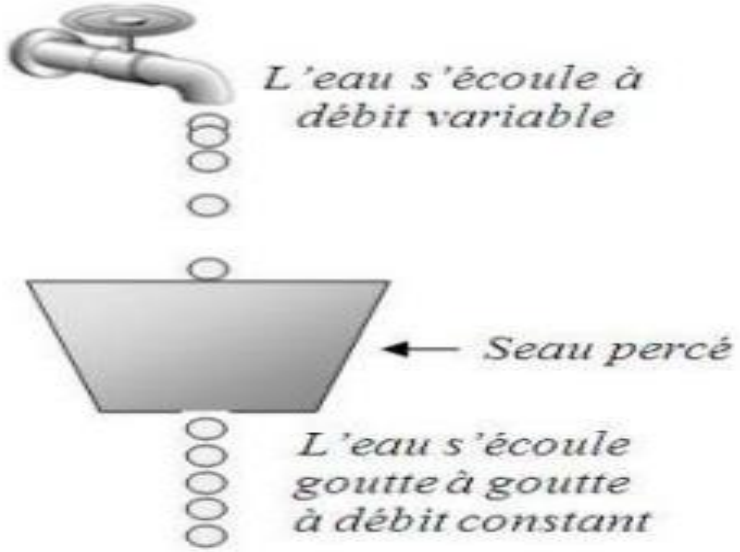
Chapitre II: mécanismes de la QoS

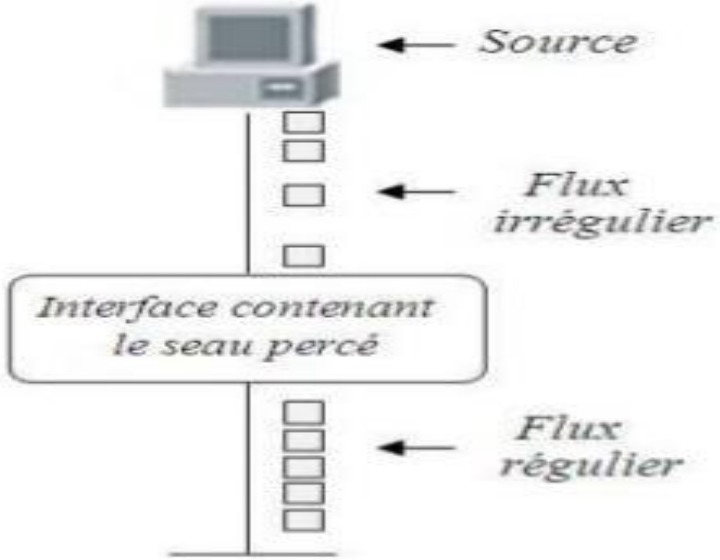
ChapII-Section B :Traffic shaping

Leaky bucket (seau percé)

Leaky bucket (seau percé)

- Imaginez un seau dont le fond est percé d'un petit trou.
- Quelle que soit la vitesse à laquelle vous le remplissez, la vitesse d'écoulement par le trou reste constante
- Lorsque le seau est plein, continuer de verser de l'eau provoque un débordement et une perte du liquide c'est-à-dire que l'eau perdue n'apparaîtra pas dans le flux de sortie qui passe par le trou





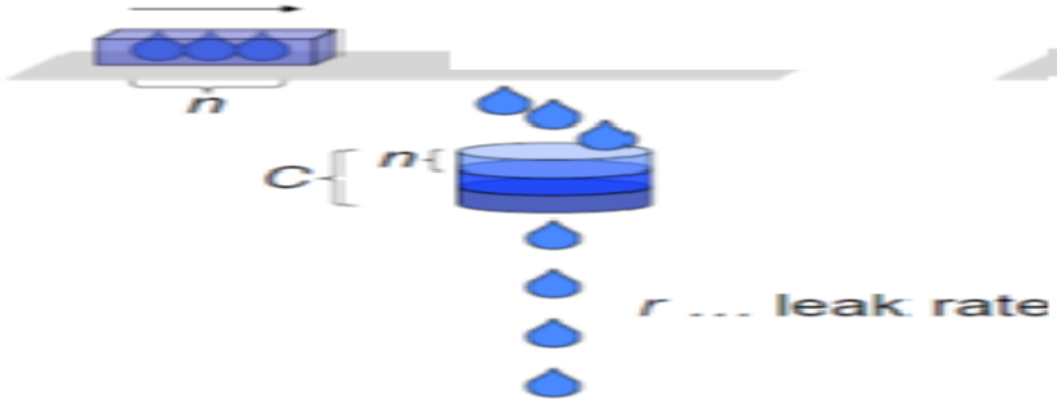
Leaky bucket (seau percé)

- L'implémentation de l'algorithme du seau percé est simplement une file d'attente de taille fixe.
- Un paquet qui arrive est accepté si la file n'est pas pleine, sinon, il est éliminé.
- Si un ou plusieurs processus sur l'hôte tentent d'envoyer un paquet lorsque le nombre maximal admis est déjà atteint, le nouveau paquet est supprimé.

☹ L'algorithme du seau percé impose ***un écoulement rigide à un débit moyen***, quelles que soient la sporadicité et la longueur des rafales. Pour certaines applications toutefois, il est parfois préférable d'accélérer l'écoulement lorsque de grandes rafales arrivent. Il faudrait donc disposer d'un algorithme plus souple, de préférence qui ne perde jamais de données, comme l'algorithme du seau à jetons (token bucket)

Leaky bucket (seau percé)

- Le seau est avec capacité ***C*** fuit à débit fixe ***r***
- Le seau ne doit jamais déborder
- Si le seau est vide, il cesse de fuir
- Un paquet est conforme, si la quantité d'eau, ***n***, peut être ajouté au seau sans provoquer de débordement;
- Un paquet non conforme si aucune eau n'est ajoutée au seau



Leaky bucket (seau percé)

Exercice « Leaky Bucket »

Un ordinateur dispose de 19,5 Mo à envoyer sur un réseau et transmet les données en rafale à 6 Mbps. La vitesse de transmission maximale entre les routeurs du réseau est de 4 Mbps. Si la transmission de l'ordinateur est mise en forme à l'aide d'un seau percé , quelle capacité du doit le seau contenir pour ne pas supprimer de données?

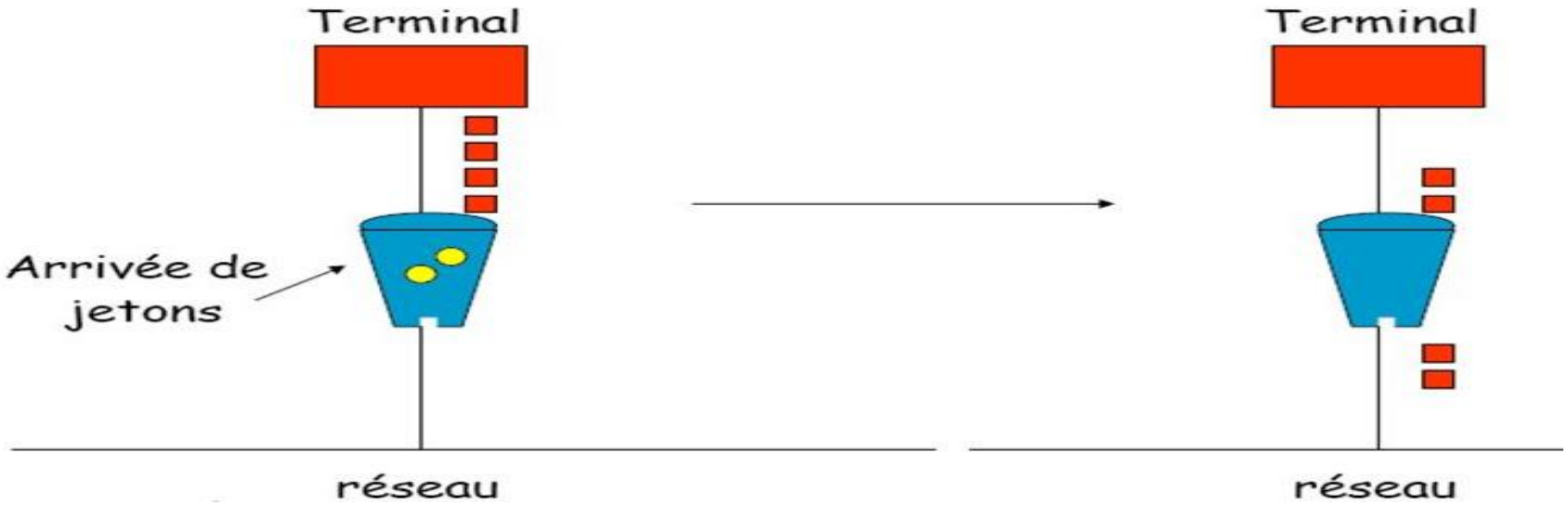
Chapitre II: mécanismes de la QoS

ChapII-Section B :Traffic shaping

Token bucket (seau à jetons)

Token bucket (seau à jetons)

- Dans cet algorithme, le seau percé contient des jetons générés par une horloge à un rythme d'un jeton toutes les ΔT secondes.
- Imaginons un seau avec 2 jetons et 4 paquets en attente de transmission. Pour qu'un paquet soit transmis, il doit capturer et détruire un jeton.
- 2 des 4 paquets sont passés à travers, mais les deux autres sont bloqués et doivent attendre la génération de deux jetons supplémentaires.



Token Bucket Algorithm

1. Un jeton est ajouté à chaque fois.
2. Le seau peut contenir au maximum des b-tokens. Si un jeton arrive lorsque le seau est plein, il est rejeté.
3. Lorsqu'un paquet de **m** octets est arrivé, **m** jetons sont supprimés du seau et le paquet est envoyé au réseau.
4. Si moins de **m** jetons sont disponibles, aucun jeton n'est retiré des seaux et le paquet est considéré comme non conforme. Le paquet non conforme peut être mis en file d'attente pour une transmission ultérieure lorsqu'une quantité suffisante de jetons a été accumulée dans le seau.

Token Bucket Algorithm

- S :la longueur de rafale (Burst length (sec)).
- M: est débit de sortie maximal(Maximum output rate (b/sec))
- C: est la capacité maximale seau (Token bucket capacity (*bits)*)
- $\boldsymbol{\rho}$: le débit d′arrivée (Token arrival rate (b/sec))

- Le rafale de sortie contient un maximum de $\boldsymbol{C + \rho S}$ bits
- Le nombre de bits dans une rafale de vitesse maximale d'une longueur de $\boldsymbol{S}$ secondes est égale $\boldsymbol{MS}$.
- Par conséquent, nous avons: $\boldsymbol{C + \rho S = MS}$

Exercice « Token Bucket »

Considérons qu'un réseau frame relay ayant une capacité de 1 Mb de données arrive à un débit de 25 Mbps.

Le taux d'arrivée des jetons est de 2 Mbps et la capacité du seau est de 500 Kb avec un débit de sortie maximal de 25 Mbps.

1. La longueur de rafale.
2. Temps de sortie total.

Leacky bucket

1. Dans l'algorithme Leaky bucket, lorsque le bucket est plein, les données ou les paquets sont supprimés.
2. Leaky bucket envoie des paquets à un débit constant.
3. Dans l'algorithme Leaky bucket, les paquets sont transmis en continu.
4. Il n'enregistre aucun jetons.
5. L'algorithme Leaky bucket est plus restrictif que l'algorithme Token bucket.

VS

TokenBucket

1. Dans l'algorithme du seau à jetons, si le seau est plein, les jetons sont rejetés et non les paquets.
2. Le compartiment à jetons peut envoyer de grandes rafales de paquets à un rythme plus rapide.
3. Dans l'algorithme du compartiment à jetons, les paquets ne peuvent être transmis que lorsqu'il y a suffisamment de jetons.
4. Il enregistre le jeton pour la rafale de transmission de paquets.
5. L'algorithme du compartiment à jetons est moins restrictif que l'algorithme du compartiment Leaky.