Master – Ingénierie de trafic

Pile de protocole X2

Internet n'assure aucune garantie sur la façon dont il achemine les données, il fonctionne sur le mode "Best effort". Cependant, il existe des mécanismes qui permettent de séparer les différents flux circulants sur les réseaux.

Quand on parle de QOS, il s'agit de l'évaluation de propriétés liées aux performances du réseau et à la différenciation des garanties offertes en fonction du type de flux.

Principaux critères de qualité de service

Débit
Délai de bout en bout
Gigue : différence de délai maximale entre deux paquets (cas idéal=gigue nulle)
Pertes de paquets : ceci peut se produire soit à cause d'une congestion, soit à cause d'un délai trop important.
Disponibilité des équipements.

Priorisation du trafic

- La file des paquets entraîne des retards parce que les nouveaux paquets ne peuvent pas être transmis tant que les paquets précédents n'ont pas été traités.
- Si le nombre de paquets à mettre en file d'attente continue d'augmenter, la mémoire dans l'appareil se remplit et les paquets sont supprimés.

Bande passante, congestion, retard et jitter

- La congestion du réseau entraîne des retards.
- Le délai est le temps qu'il faut à un paquet pour voyager de la source à la destination.

- Le Jitter ou la gigue est la variation dans le retard des paquets reçus au fil du temps.

Perte de paquet

- En cas de congestion, les périphériques réseau tels que les routeurs et les commutateurs peuvent laisser tomber des paquets.
- La perte de paquets est une cause très fréquente de problèmes de qualité vocale sur un réseau IP.
- Dans un réseau bien conçu, la perte de paquets doit être proche de zéro.

Les ingénieurs réseau utilisent des mécanismes QoS pour classer les paquets vocaux pour zéro perte de paquets.

Caractéristiques du trafic

Tendances du trafic réseau

Les jeux vidéo et la visioconférence demandent une bonne fluidité, pas trop de perte et un temps de réponse faible. Ces applications ont donc besoin d'une bonne bande passante et de délais constants, un système qui limite les pertes et une gigue faible (variance dans les délais).

> **On considère qu'elles génèrent du trafic temps réel.**

Les applications Internet, la messagerie et le transfert de fichiers nécessitent que les données arrivent intactes et complètes. Il est nécessaire que le transport soit fiable et ces applications supportent un débit variable et elles n'imposent pas de délai très court pour le trafic.

> **On considère que le trafic est élastique car il s'adapte aux fluctuations.**

Les applications de streaming ont besoin de fluidité, cependant, elles acceptent d'attendre avant la diffusion. Il faut également limiter la perte et jouer sur la gigue en stockant une partie des données dans un cache avant de les utiliser.

Au début, le trafic est élastique, mais ensuite il est temps réel.

Sur internet le trafic double tous les 2 ans, et l'évolution est en forte croissance. Dans le top des applications les plus consommatrices sur internet, on trouve, les applications vidéo suivies des mails et du transfert de fichiers.

Les différents types de trafic

VOIX

- La voix est très sensible aux retards et aux paquets abandonnés ; il n'y a aucune raison de retransmettre la voix si des paquets sont perdus. 1% ou 2% de perte de données de voix en ligne ne sont pas trop gênants pour la qualité du service de VoIP, mais en revanche une variation fréquente de 100 ms sur le délai de transit est catastrophique et rend le service inutilisable.
- Les paquets vocaux doivent recevoir une priorité plus élevée que les autres types de trafic.
- La voix peut tolérer une certaine quantité de latence, de gigue et de perte sans aucun effet perceptible.

VIDÉO

- Par rapport à la voix, la vidéo est moins résistante à la perte et au volume plus élevé de données par paquet.
- La vidéo peut tolérer une certaine quantité de latence, de nervosité et de perte sans aucun effet notable.

DONNÉES

- Les applications de données qui n'ont aucune tolérance pour la perte de données, telles que les e-mails et les pages Web, utilisent TCP pour s'assurer que, si les paquets sont perdus en transit, ils seront mécontents.
- Le trafic de données est relativement insensible aux baisses et aux retards par rapport à la voix et à la vidéo.

Exigences	Data	Voice	Video
Bande passante	Élevé	Faible	Élevé
Délai	Pas applicable	moins de 150 ms	moins de 150 ms pour de la vidéo en temps réel
Gigue (variation de délai)	Pas	Faible	Faible
Perte de paquets	Moins de 5%	Moins de 1%	Moins de 1%
Disponibilité	Élevé	Élevé	Élevé
Sécurité	Élevé	Moyen	Faible ou moyen
Approvisionnementź Provisioning)	Effort moyen	Effort important	Effort moyen

Quels sont les besoins des applications ?

Les jeux vidéo et la visioconférence demandent une bonne fluidité, pas trop de perte et un temps de réponse faible. Ces applications ont donc besoin d'une bonne bande passante et de délais constants, un système qui limite les pertes et une gigue faible (variance dans les délais). On considère qu'elles génèrent du trafic temps réel.

Les applications Internet, la messagerie et le transfert de fichiers nécessitent que les données arrivent intactes et complètes. Il est nécessaire que le transport soit fiable et ces applications supportent un débit variable et elles n'imposent pas de délai très court pour le trafic. On considère que le trafic est élastique car il s'adapte aux fluctuations.

Les applications de streaming ont besoin de fluidité, cependant, elles acceptent d'attendre avant la diffusion. Il faut également limiter la perte et jouer sur la gigue en stockant une partie des données dans un cache avant de les utiliser. Au début le trafic est élastique mais ensuite il est temps réel.

Sur internet le trafic double tous les 2 ans, et l'évolution est en forte croissance. Dans le top des applications les plus consommatrices sur internet, on trouve, les applications vidéo suivies des mails et du transfert de fichiers.

Comment mesurer ?

Il s'agit à la fois de de critères **subjectifs** (QoE Quality Of experience – je suis content ou non du temps de réponse de mon application) ou de critères **mesurables** (nombre de paquets reçus vs nombre de paquets perdus)

La QOS doit permette d'offrir une garantie de services en fonction des besoins des applications.

Réservation flux par flux ou redimensionnement du réseau

Dans le cadre de la **réservation flux par flux** des ressources, les flux sont traités de façon individuelle et on leur attribue les ressources au plus près de leur besoin.

Par exemple, UserA et UserB prévoient une visioconférence de 14h00 à 15h00. Il faut donc trouver un chemin réseau permettant d'assurer la demande en termes de débit, de flux et de temps.

Cela impose de garder une information dans les routeurs pour indiquer pour chaque flux le traitement à appliquer. Cela augmente la taille des tables de routage car on ne peut pas agréger ces flux. De plus, à chaque paquet entrant dans le routeur, il faut parcourir la table pour savoir à quel flux il appartient et comment le traiter.

Ce système est référencé par IETF sous le nom de IntServ (flux par flux).

Avantages

Utilisation des ressources au plus proche des besoins.

Inconvénients

Complexité de la diversité des demandes à traiter, facturation difficile.

Dans le cas du **redimensionnement**, on regroupe les flux ayant les mêmes besoins pour réduire la complexité des traitements en cœur de réseau. Les ressources ne seront pas attribuées flux par flux mais on prévoie une quantité de ressources au préalables attribuées à un groupe (classe de service).

Pour cela, on fait une estimation de la quantité de ressources qui sera nécessaire. Comme on ne peut pas prédire avec précision les flux qui seront groupés, une mauvaise estimation peut conduire à sous dimensionner et les flux n'auront pas la totalité des ressources dont ils ont besoin.

Pour éviter cette situation, on aura tendance à surdimensionner le réseau. Ce système est référencé par l'IETF sous le nom DiffServ.

Avantages

Facturation plus simple, simplification du traitement dans les réseaux.

Inconvénients

Nécessite une bonne estimation des demandes à venir, besoin de surdimensionner les ressources, et avoir des équipements susceptibles de le gérer.

Techniques de mise en œuvre QoS

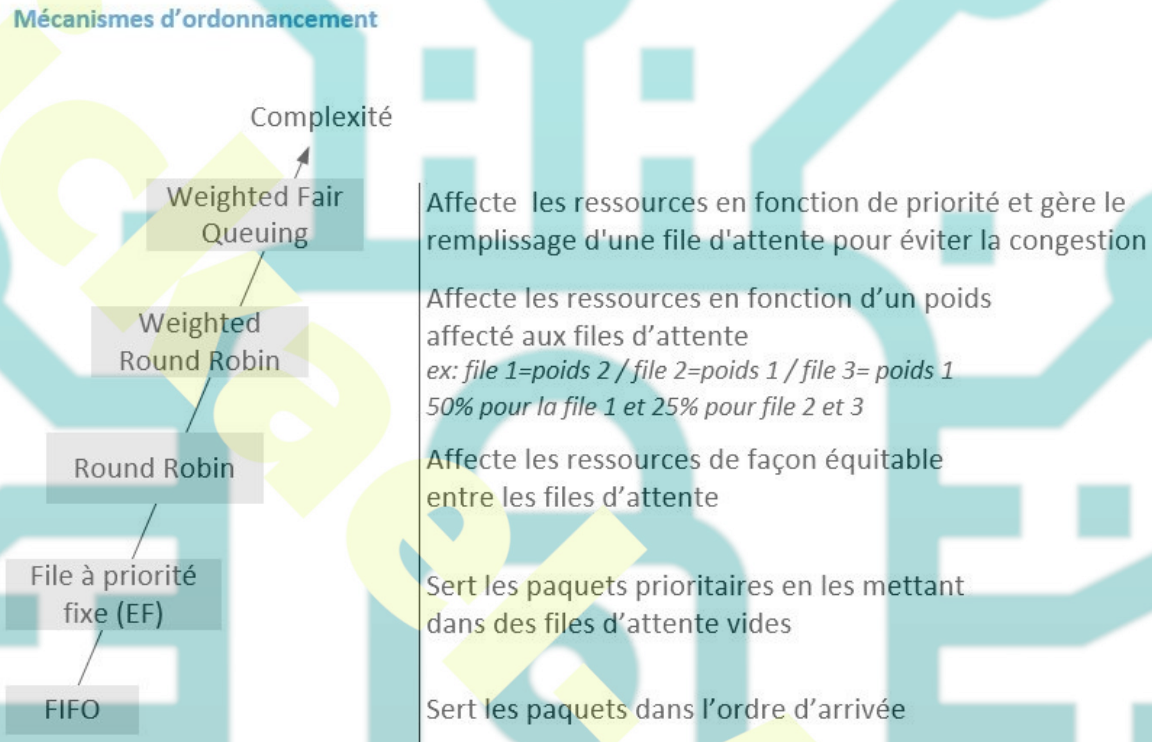
- **Par VLAN** , ce qui permet de donner une priorité à un VLAN par rapport à un autre ;
- **Par port UDP/TCP**, ce qui revient à donner la priorité à une application par rapport à une autre ;
- **Par adresse IP source ou de destination**, ce qui revient à donner la priorité à un équipement de réseau par rapport à d'autres (un serveur IPBX par exemple) ;
- **Par une interface d'entrée** dans le commutateur, ce qui revient à donner la priorité à un segment de réseau, ou un équipement ;
- **Par des informations de priorité** déjà présentes dans l'entête d'une trame ou d'un paquet entrant.

La priorité entre les files d'attente peut être traitée selon différents algorithmes, en fonction des implantations et des possibilités offertes par les constructeurs. Par exemple :

Priorité "stricte" : une file de priorité N est vidée entièrement avant les files de priorité N-1, c'est la technique dite de "Priority Queuing" ;
Priorité "à tour de rôle" : un paquet de chaque file d'attente est traité, à tour de rôle, c'est la technique dite du "Round Robin" ;
Priorité "pondérée" : le commutateur traite, par exemple, 8 trames de priorité 7, puis une trame de priorité 1, puis de nouveau 8 trames de priorité

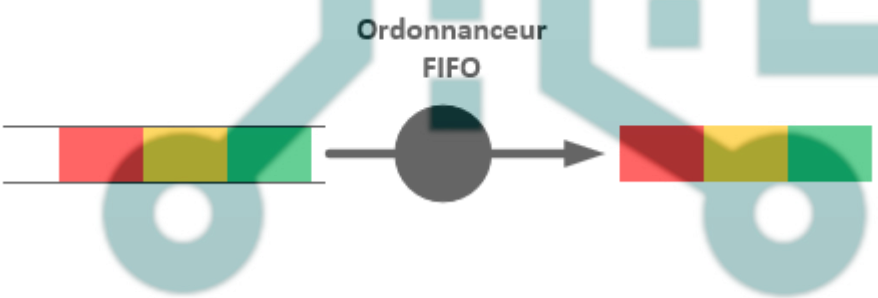
7, etc. Un poids est donc affecté à chaque file d'attente. C'est la technique du **WRR** (Weighted Round Robin).

Les différents mécanismes



FIFO

FIFO est l'ordonnanceur le plus simple : les paquets sont servis dans l'ordre de leur arrivée. Ne gère pas la différentiation de classe.



FIFO n'a pas de concept de priorité ou de classes de trafic et, par conséquent, ne prend aucune décision sur la priorité paquet.

FIFO, est la méthode la plus rapide et efficace pour les grands liens qui ont peu de retard et une congestion minimale.

Priorité stricte LLQ

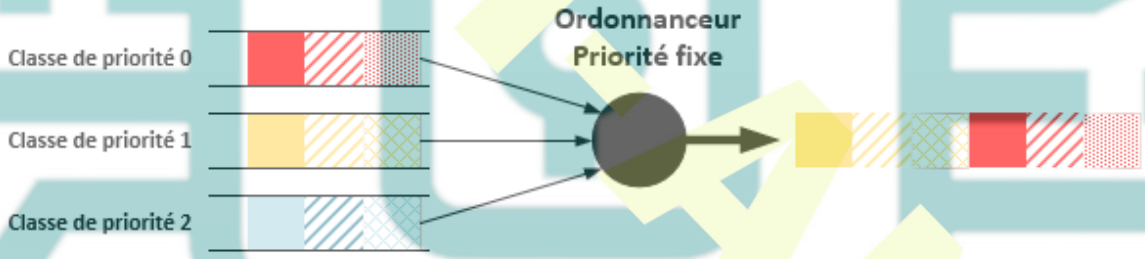
On associe une file à chaque classe et on sert les classes de plus grande priorité en premier.

Gère la différentiation de classe, mais les classes de faible priorité peuvent ne jamais être servies (cas de famine) ,

On peut garantir un délai maximal pour la classe de plus grande priorité, il n'y a pas d'isolation des flux et pas d'équité.

La bande passante attribuée aux paquets d'une classe détermine l'ordre dans lequel les paquets sont envoyés.

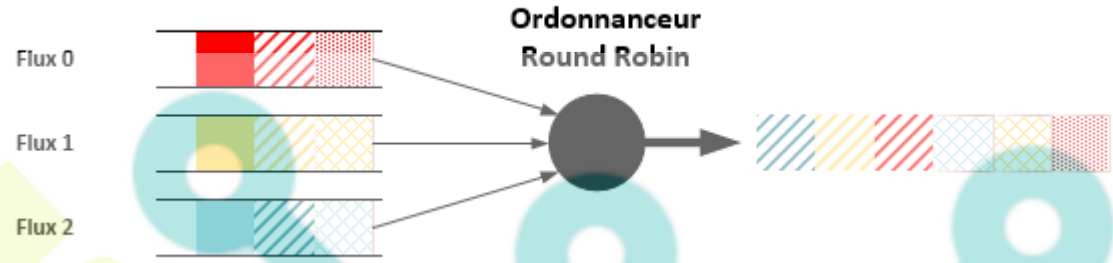
LLQ permet d'envoyer en premier des données sensibles aux retards telles que la voix.



Round Robin

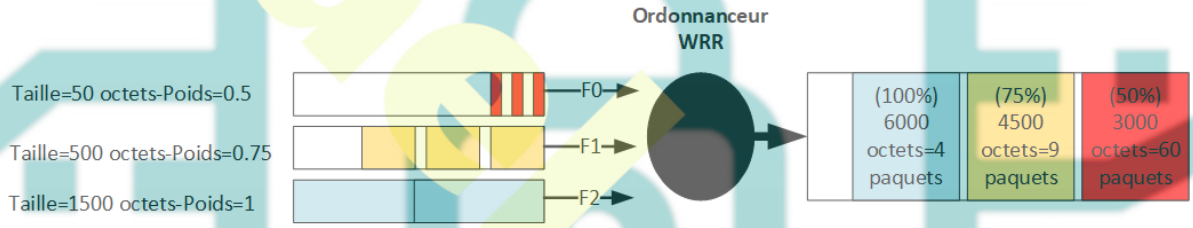
Les flux sont servis à tour de rôle.

On ne gère pas de différentiation de classe, équité stricte si les flux utilisent des paquets de même longueur. Pas de cas de famine.



Weighted Round Robin (WRR)

La file d'attente partage la bande passante entre ses classes en utilisant la technique du tourniquet pondéré. On peut associer des classes avec des files d'attente. Toutes les classes qui ont suffisamment de demandes obtiendront la bande passante proportionnellement au poids associé des classes.



1. Pour caractériser une classe, vous lui attribuez la bande passante, le poids et la limite maximale de paquets.
2. Vous spécifiez également la limite de file d'attente pour cette classe, qui est le nombre maximum de paquets autorisés à s'accumuler dans la file d'attente pour la classe.
3. Les paquets appartenant à une classe sont soumis aux limites de bande passante et de file d'attente qui caractérisent la classe.

Exemple

On prend 3 flux avec des tailles moyennes de paquet différentes (50, 500 et 1500) avec des poids initiaux (0.5, 0.75, 1). La capacité est de 6000 octets.

Le flux 0 pourra remplir la file à hauteur de 50% (3000 octets) soit l'équivalent de 3000/50=60 paquets
Le flux 1 pourra remplir la file à hauteur de 75% (4500 octets) soit 4500/500=9 paquets
Le flux 2 pourra remplir la file à hauteur de 100% (6000 octets) = 4 paquets

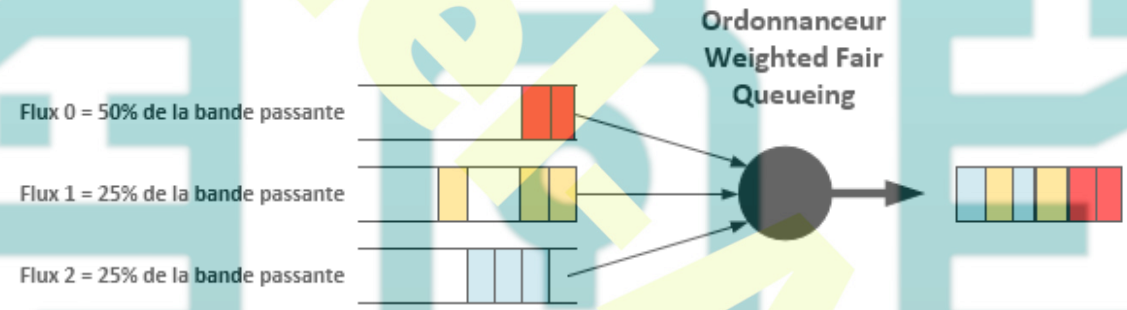
Weighted Fair Queueing

Dans un réseau de paquets, une seule connexion peut être servie à la fois et un paquet doit être transmis entièrement avant qu'un autre puisse l'être.

Il détermine le nombre de paquets qu'il est nécessaire d'ordonnancer à chaque cycle, au niveau de chaque file. Cette quantité est définie par un poids. Il peut supporter différentes tailles de paquets et ainsi assurer un traitement malgré l'hétérogénéité des flots.

Les paquets qui sont placés en tête de file sont servis selon le poids qui leur est attribué, et le traitement des files s'effectue cycliquement et séquentiellement.

Dans l'exemple suivant, la première file dispose d'un poids de 50%, supérieur aux autres, ce qui permet de pouvoir obtenir une transmission de plus d'un paquet issu d'une même file d'attente.



Les différentes implémentations

1. Routeurs de périphérie (edge device)

- Priorité stricte pour une file avec faible délai d'attente
- WFQ entre différentes classes de trafic
- FIFO à l'intérieur d'une classe entre flux

2. Routeurs de cœur (core device)

- Priorité stricte pour une file avec faible délai d'attente
- WRR (voire WFQ) entre différentes classes de trafic
- FIFO à l'intérieur de chaque classe entre flux

Gestion des files d'attente en cas de congestion

Token Bucket

Pour vérifier qu'un paquet est conforme ou pas, le routeur peut utiliser la méthode « Token Bucket » (seau à jeton) qui consiste à affecter les jetons du seau au paquet arrivant pour gérer les flux. L'algorithme du seau à jetons permet de contrôler le débit passant par un nœud d'un réseau.

Le seau se rempli à R jeton qui correspond au volume de trafic autorisé. Un seau autorise 2 niveaux de priorité, pour en avoir trois, il suffit d'ajouter un seau en série.

Le Token Bucket Filter (TBF) est un gestionnaire de mise en file d'attente. Il laisse passer les paquets entrants avec un débit n'excédant pas une limite fixée par le contrat.

L'implémentation TBF consiste en un tampon (seau), constamment rempli par des jetons, avec un débit spécifique (débit de jeton). Le paramètre le plus important du tampon est sa taille, qui correspond au nombre de jetons qu'il peut stocker.

Chaque jeton entrant laisse sortir un paquet de données de la file d'attente de données et le jeton est alors supprimé du seau.

1. Si les données arrivent avec un débit **égal** au débit des jetons entrants, chaque paquet a son jeton correspondant et passe la file d'attente sans délai.
2. Si les données arrivent avec un débit **plus petit** que le débit des jetons. Seule une partie des jetons est supprimée, de sorte que les jetons s'accumulent jusqu'à atteindre la taille du tampon. Les jetons libres peuvent être utilisés pour envoyer des données avec un débit supérieur au débit des jetons standard.
3. Si Les données arrivent avec un débit **plus grand** que le débit des jetons, cela signifie que le seau va bientôt manquer de jetons. Les paquets qui continuent à arriver sont éliminés.

Leaky bucket

Soit un seau percé en son fond : si le seau n'est pas vide alors le contenu s'y écoule avec un débit constant.

La taille du seau représente la quantité d'informations qui peut y être stockée, mesurée en nombre de paquets.

Lorsqu'un paquet arrive, s'il reste suffisamment d'espace dans le seau, il y est placé. Sinon, le seau déborde (paquet en excès) .

Les paquets en excès sont en général jetés. Ils peuvent aussi être mis en attente, ou marqués comme non-conformes avant d'être envoyés.

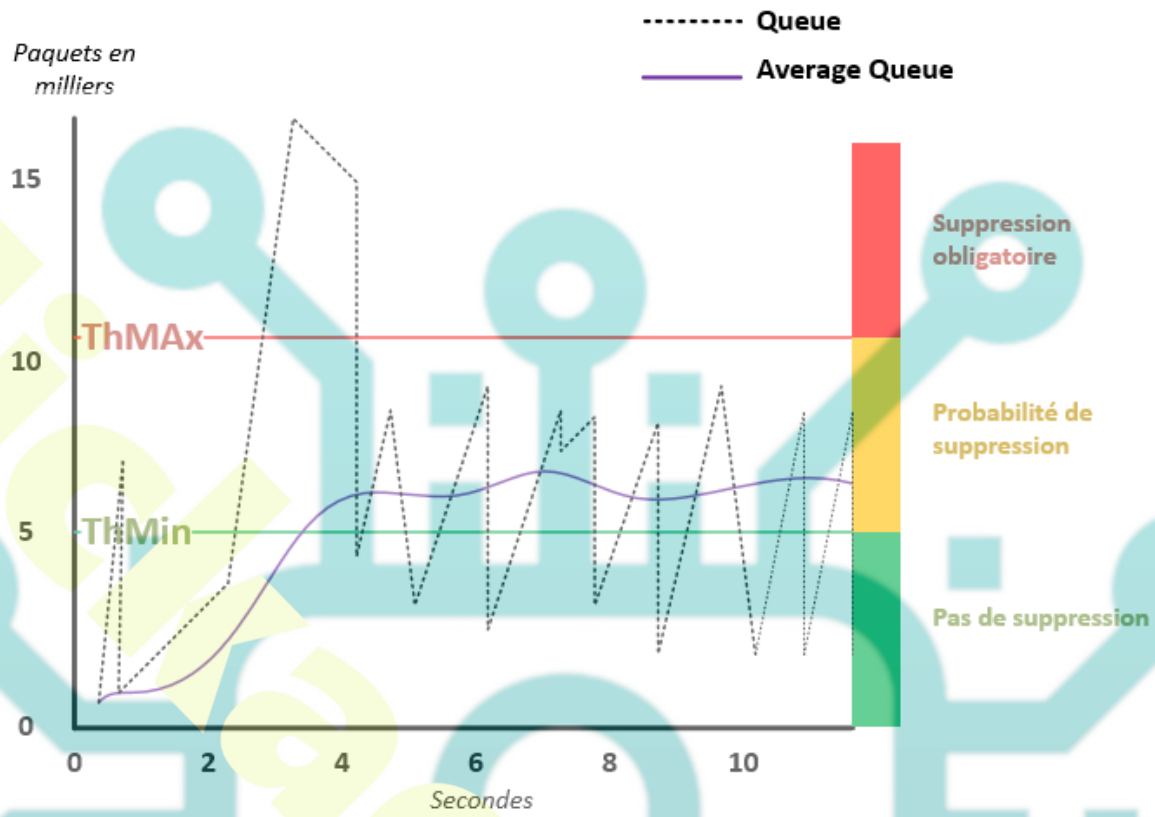
RED (Random Early Discard)

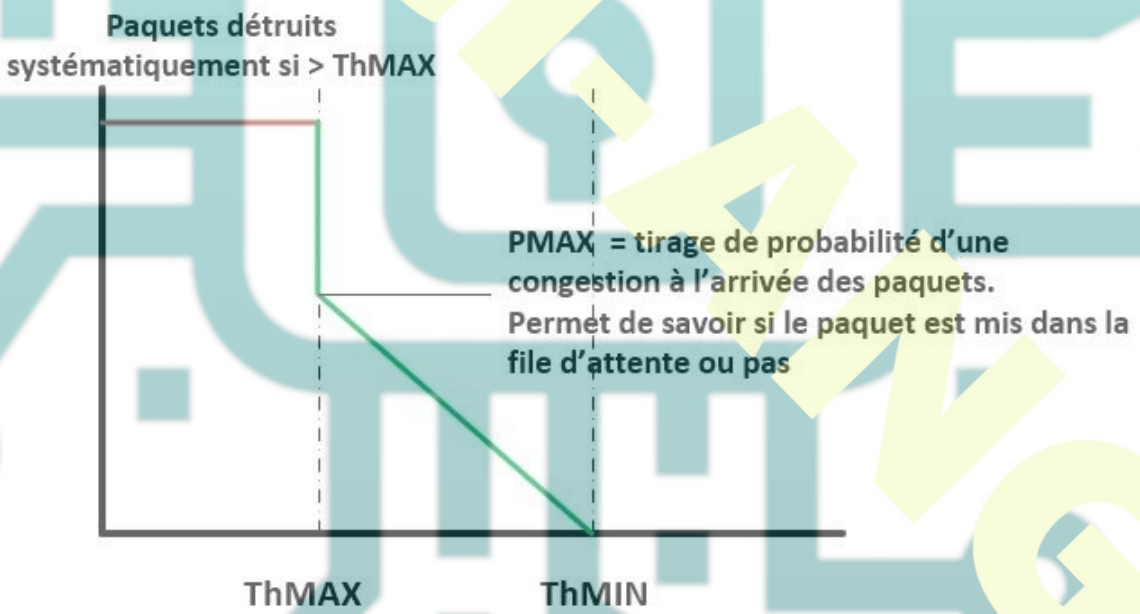
Permet de ralentir le remplissage d'une file d'attente en détruisant des paquets avant l'arrivée de la congestion. Ce système est prévu pour 2 niveaux de priorité.

En effet, les paquets arrivent souvent en rafale au routeur, ce qui cause leur suppression si la file d'attente est pleine. Ce mécanisme qui permet de garantir que le buffer ne sera jamais rempli, afin de toujours pouvoir accepter et traiter les paquets.

L'algorithme RED se divise en deux parties : l'estimation de la taille moyenne de la file (avg) et la décision de supprimer ou de ne supprimer un paquet arrivant.

RED décide de jeter un paquet sortant avec une certaine probabilité. Pour cela, RED va utiliser deux variables minth (minimum threshold) et maxth (maximum threshold) qui représente un intervalle de taille pour la file.

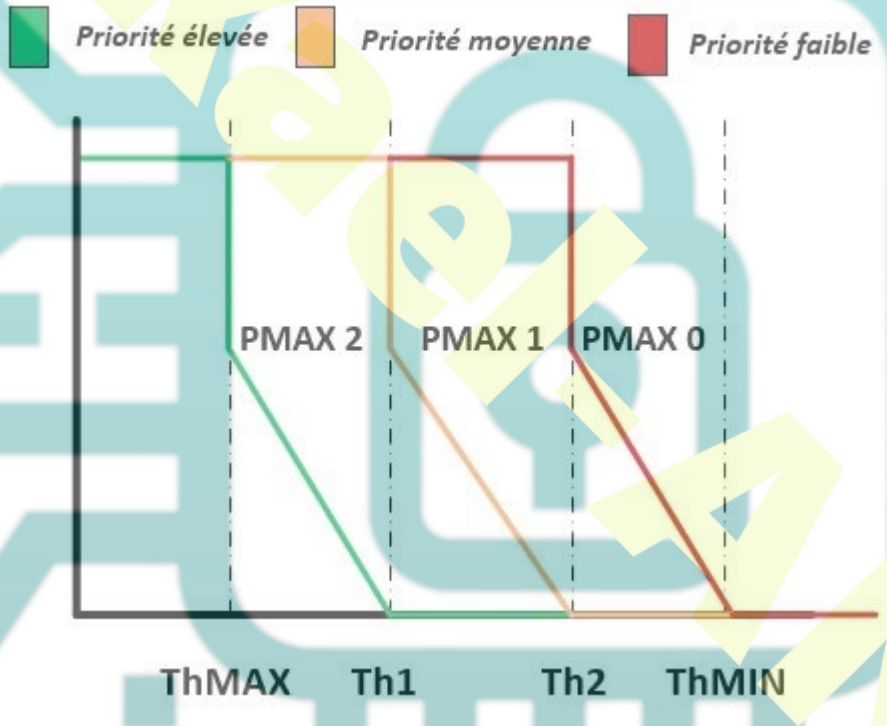

Queue
Average Queue
Paquets en milliers
15
10
5
0
ThMAx
ThMin
Suppression obligatoire
Probabilité de suppression
Pas de suppression
0
2
4
6
8
10
Secondes


Paquets détruits systématiquement si > ThMAX
PMAX = tirage de probabilité d'une congestion à l'arrivée des paquets.
Permet de savoir si le paquet est mis dans la file d'attente ou pas
ThMAX
ThMIN

Bande passante	BW théorique (Kbps)	BW physique (Kbps)	Seuil minimum	Seuil maximal
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

Valeurs recommandées basées sur la bande passante du lien

WRED (Weighted Random Early Discard) permet à 3 RED de s'exécuter en parallèle.



Cette technique impose 7 paramètres (ThMAx, Th1, Th2, ThMin, Pmax2, Pmax1, Pmax0).

Architecture DiffServ

Le principe est de gérer en amont la complexité pour que l'exécution soit plus simple.

Les classes de service

Une application choisie une des classes de service fournie par l'administrateur de réseau pour envoyer son trafic.

Les paquets sont marqués pour permettre aux routeurs de connaitre le traitement à effectuer sur ce dernier.

DiffServ propose des PHB (Per Hop Behaviour – comportement par traitement/marquage)

Les classes de service doivent être valables de bout en bout, c'est à dire que les valeurs doivent être standardisées (RFC 2474) et les routeurs doivent comprendre le marquage associé.

Marquage des paquets

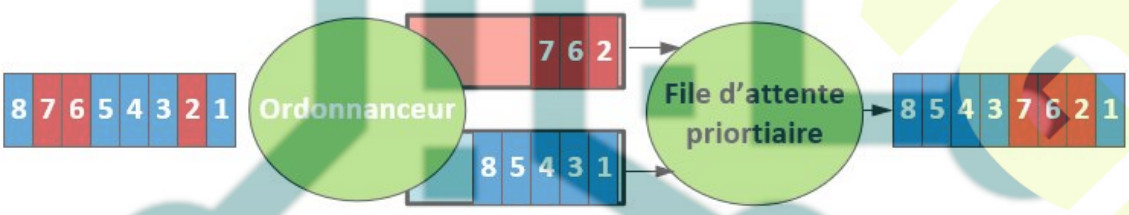
Le marquage peut être soit effectué sur l'application elle-même, soit effectué sur un routeur proche de la source via des ACL (équipement, firewall) Ces ACL sont couplées par les routeurs avec les politiques de QOS.

Dans IPv4, le marquage se fait au niveau du troisième champs d'en-tête DSCP (DiffServ Code Point)
Dans IPv6, le marquage est situé dans le champs Traffic Class et Flow Label

1. Best Effort– Traite tous les paquets qui n'ont pas besoin de garantie de performance.

2. EF (Expedited Forwarding) – Traite les paquets en imposant une forte priorité (l'ordonnanceur du routeur va placer les paquets dans des files d'attente vides en priorité fixe).

Il ne faut pas trop de paquets EF, on les restreints via un contrôle d'admission.



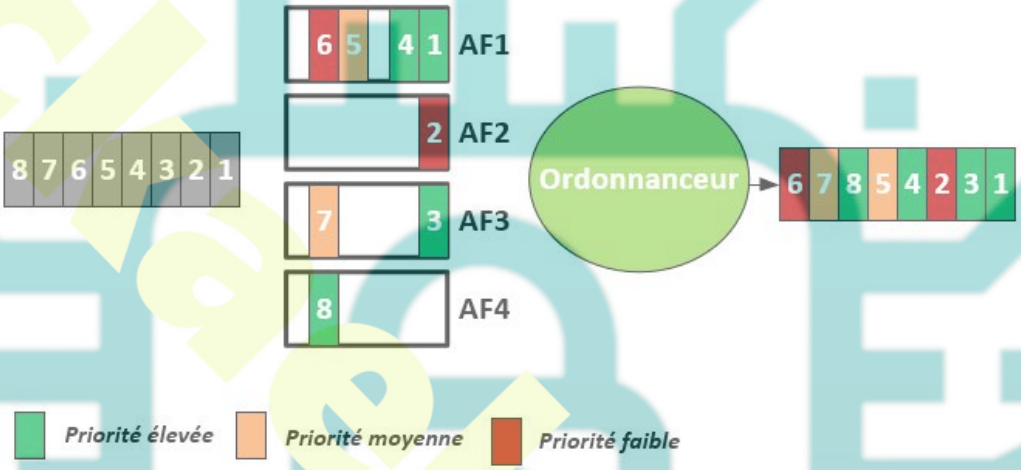
AF (Assured Forwarding) – Les garanties sont plus faibles que EF (perte de paquets, délai) mais meilleures que Best Effort.

Il existe 4 classes de services AF et 3 niveaux de priorité par classe.

La classification des flux en fonction du volume qui peut se mesurer au niveau du débit moyen.

On peut également déterminer la classification en fonction du « Profile Meter » qui détermine si la partie du trafic non conforme au contrat peut être modifiée.

Un paquet peut être marqué en basse priorité, retardé ou détruit.

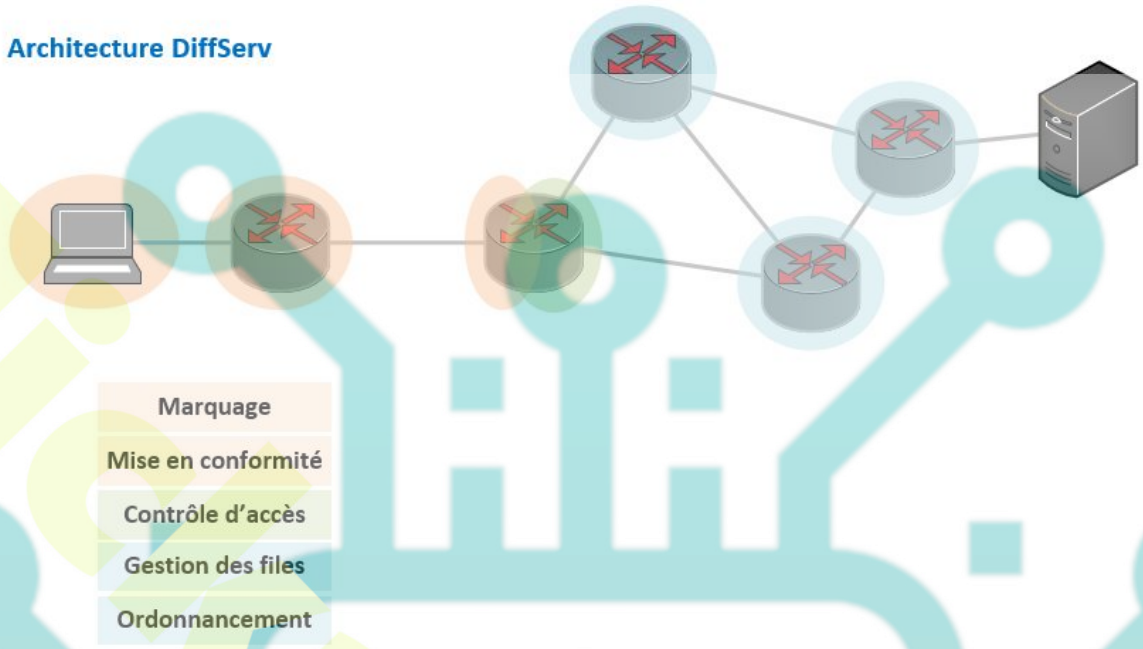


Pour vérifier qu'un paquet est conforme ou pas le routeur peut utiliser la méthode Token Bucket (seau à jeton) qui consiste à affecter les jetons du seau au paquet arrivant pour contrôler les flux. Lorsque le seau est vide les paquets suivants sont considérés comme ne respectant plus le contrat et ils sont alors affectés à une priorité plus basse ou supprimer ou en attente de remplissage du seau.

Le seau se rempli à R jeton qui correspond au volume de trafic autorisé. Un seau autorise 2 niveaux de priorité, pour en avoir trois, il suffit d'ajouter un seau en série.

Mise en œuvre de DiffServ

Les systèmes Windows et Linux peuvent utiliser la différenciation des services via les firewalls ou directement au niveau des applications et plus généralement les services mail, visioconférence…



L'application peut faire le marquage et la mise en conformité et si elle n'est pas capable de le faire, le routeur le plus proche peut s'en occuper.

Les **routeurs** de cœur mettent en œuvre des fonctions de répartition de la bande passante et essayent d'éviter les congestions en gérant le remplissage des files d'attente. Ils exécutent également la politique de QOS mise en place par l'administrateur et gèrent l'ordonnancement.

A l'entrée du réseau de l'opérateur la vérification de la mise en conformité est importante. En effet, si le paquet ne correspond pas, on ne se met pas en conformité avec le SLA de l'opérateur.

L'opérateur peut décider pour les paquets **AF** de laisser passer un volume un peu supérieur au contrat, mais il marquera ces paquets comme étant prioritaires à la perte.

Si le paquet est **EF** il fera un contrôle d'admission (détruire les paquets non conformes ou les faire attendre) en utilisant le seau à jeton par exemple.

Si le routeur le permet, l'administrateur pourra affecter une **file d'attente** différente pour les classes de trafic, ce qui lui permettra au routeur d'isoler les paquets d'une même classe et de leur attribuer le comportement prévu.

Si les équipements ne disposent que de 2 files d'attente, on regroupe les flux mais pas les flux EF avec les flux BE (best effort). Cependant, on peut mélanger AF et BE via un mécanisme WRED.

NB. les flux EF ne doivent pas être mélangés avec les autres flux (AF et BE)

L'ordonnanceur est le dernier mécanisme à paramétrer sur les routeurs de cœur (attention, les algorithmes peuvent varier selon les constructeurs) .

Si le routeur possèdent une file d'attente à priorité fixe alors elle sera réservée au trafic EF.

Si ce n'est pas le cas, il faut utiliser une file classique en surdimensionnant le lien pour EF. La logique consiste à vider le plus souvent possible la file EF.

Résumé

La qualité de la transmission réseau est affectée par la bande passante des liens entre la source et la destination, les sources de retard que les paquets sont acheminés vers la destination, et la nervosité ou la variation du retard des paquets reç us.

Sans mécanismes QoS en place, les paquets sont traités dans l'ordre dans lequel ils sont reç us. En cas de congestion, les paquets sensibles au temps seront supprimés avec la même fréquence que les paquets qui ne sont pas sensibles au temps.

Les paquets vocaux ne nécessitent pas plus de 150 millisecondes (ms) . La gigue ne devrait pas être plus de 30 ms, et la perte de paquets vocaux ne devrait pas être supérieure à 1%. Le trafic vocal nécessite au moins 30 Kb/s de bande passante.

Les paquets vidéo ne nécessitent pas plus de 400 millisecondes (ms) . La gigue ne devrait pas être plus de 50 ms, et la perte de paquet vidéo ne devrait pas être plus de 1%. Le trafic vidéo nécessite au moins 384 Kb/s de bande passante.

Pour les paquets de données, deux facteurs influent sur la qualité de l'expérience (QoE) pour les utilisateurs finaux :

- Les données proviennent elles d'une application interactive ?
- La mission de données est-elle essentielle ?

Les quatre algorithmes de file d'internet

- **FIFO**
  Les paquets sont transmis dans l'ordre dans lequel ils sont reçus.

- **WFQ**
  Les paquets sont classés dans différents flux en fonction des informations d'en-tête, y compris la valeur ToS.
- **CBWFQ**
  Les paquets sont attribués à des classes définies par l'utilisateur en fonction de correspondances avec des critères tels que les protocoles, les ACL et les interfaces d'entrée. L'administrateur réseau peut attribuer la bande passante, le poids et la limite maximale de paquets à chaque classe.
- **LLQ**
  Les données sensibles aux retards telles que la voix sont ajoutées à une file d'attente prioritaire afin qu'elles puissent être envoyées en premier (avant les paquets dans d'autres files d'attente).

LES TROIS MODÈLES DE FILE D'ATTENTE :

- **Meilleur effort (Best effort)**
  Il s'agit du modèle de file d'étude par défaut pour les interfaces. Tous les paquets sont traités de la même manière. Il n'y a pas de QoS.
- **Services intégrés (IntServ)**
  IntServ fournit un moyen de fournir le QoS de bout en bout dont les applications en temps réel ont besoin en gérant explicitement les ressources réseau pour fournir QoS à des flux de paquets utilisateur spécifiques, parfois appelés microflows.
- **Services différenciés (DiffServ)**
  DiffServ utilise une approche QoS souple qui dépend des périphériques réseau qui sont mis en place pour desservir plusieurs classes de trafic chacune avec des exigences QoS variables. Bien qu'il n'y ait pas de garantie QoS, le modèle DiffServ est plus rentable et évolutif qu'IntServ.

LES OUTILS QOS COMPRENNENT LES ÉLÉMENTS SUIVANTS :

- **Classification et marquage**
  Classification détermine la classe de trafic à laquelle appartiennent les paquets ou les cadres. Le marquage signifie que nous ajoutons une valeur à l'en-tête du paquet. Les appareils recevant le paquet examinent ce champ pour voir s'il correspond à une stratégie définie.
- **Évitement de la congestion**
  Les outils d'évitement de la congestion surveillent les charges de trafic du réseau afin d'anticiper et d'éviter la congestion. Comme les files d'attente se remplissent jusqu'au

seuil maximum, un petit pourcentage de paquets sont supprimés. Une fois le seuil maximal franchi, tous les paquets sont supprimés.

- **Mise en forme et maintien de l'ordre**
  La mise en forme conserve les paquets excédentaires dans une file d'attente, puis planifie l'excédent pour la transmission ultérieure sur des incréments de temps. La mise en forme est utilisée sur le trafic sortant. La police baisse ou remarque l'excès de trafic. Les services de police s'appliquent souvent à la circulation entrante.

L'ingénierie de trafic

Grâ ce à l'ingénierie, on va chercher à obtenir un QOS en agissant sur plusieurs points.

- La fiabilisation des communications en proposant plusieurs chemins d'un point A à un point B (chemin de secours et/ou répartition de charges). On peut ainsi agir sur la fiabilisation des chemins et sur la congestion.
- Le traitement des flux différentiés en séparant les flux UDP et TCP, en attribuant un chemin particulier au flux vidéo, en créant un tunnel pour les données de collecte vers d'autres opérateurs (transit, peering)
- La réduction des coûts en éteignant les équipements superflus lorsque que le trafic est faible en reroutant le trafic vers un nombre réduit de tunnels.

Le cycle de l'ingénierie de trafic comprend les étapes suivantes :

- Le dimensionnement
- La configuration
- La surveillance
- Le monitoring
- Le diagnostic

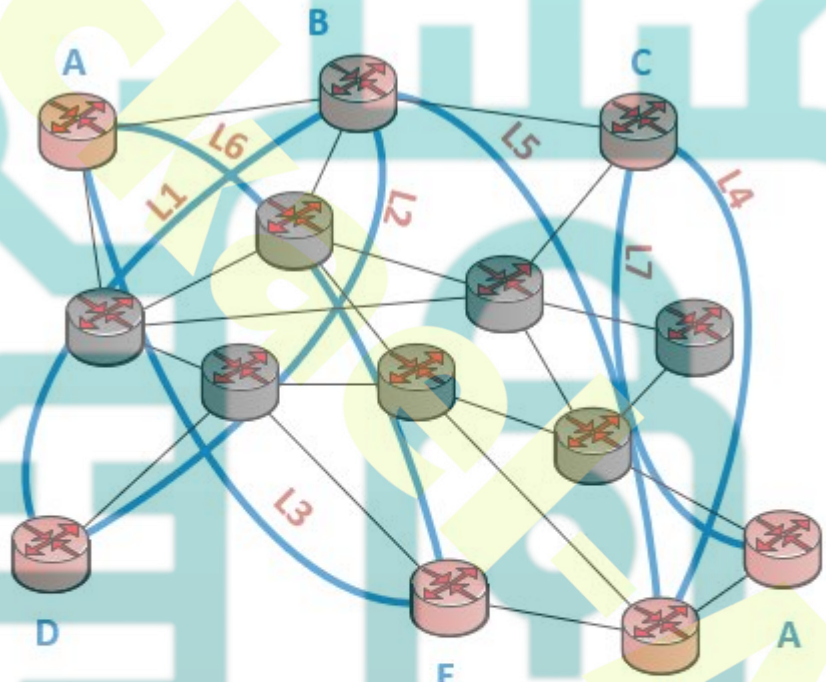
Ce cycle est renouvelé continuellement pour gérer l'arrivée de nouveaux flux et de nouveaux besoins.

L'ingénierie de trafic adapte le routage en fonction des besoins de l'administrateur. Elle doit prendre en compte un certain nombre de contraintes comme le **CBR** (Constraint Base Routing) ou routage contraint qui permet de jouer sur les routes en fonction de choix politiques (chemin le plus court, le chemin ne doit pas passer par tel AS, délai, perte, bande passante)

L'ingénierie impose la connaissance de la topologie, des protocoles de routage utilisés et du trafic géré par le réseau.

Matrice de trafic

On peut la représenter par un tableau à 2 dimensions en indiquant les nœuds entrants et sortants en ligne/colonne et les liens en données comprenant le délai, le débit, les pertes…



SORTIES \ ENTREES	A	B	C	D	E	F	G
A					L6		
B				L1			
C							
D		L2					
E	L3						
F							
G							

Cas d'usage

Définition de la charge du réseau

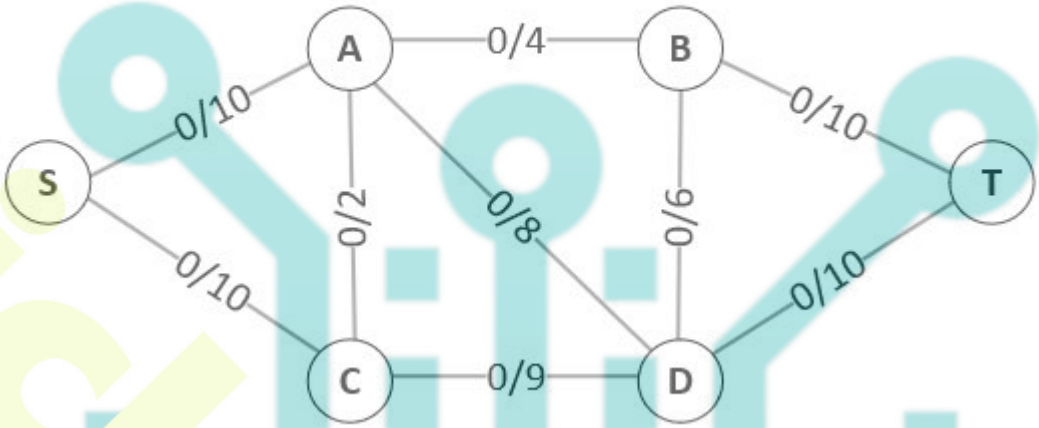
Le principe est de charger au maximum tous les chemins du réseau entre une source et une destination. Pour cela on utilise la théorie des graphes (Ford Fulkerson) pour le problème de flot maximum.

L'ensemble du réseau **R**=(**G,c,s,t**)
Le graphe **G**=(V,E) **V** étant l'ensemble des sommets et **E** l'ensemble des arcs du graphe.
c étant la capacité (bande passante nominale du lien ou une métrique de délai, de perte), **s** la source et **t** la destination

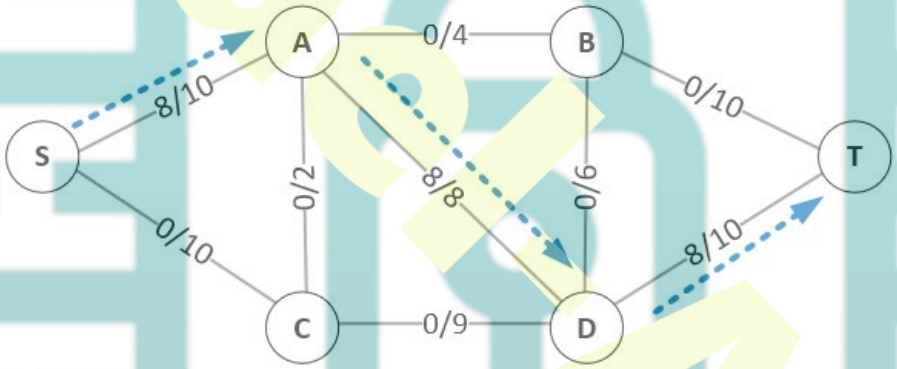
La capacité d'un arc (ou arête)
e ∈ **E** associé à une capacité **c(e)** ≥**0** avec **c : E** → **R+**

Le flot
Fonction **f** qui associe e avec une quantité **f(e)** qui passe par **e** pour aller de **s** vers **t**

Valeur 1 = flot
Valeur 2 = capacité

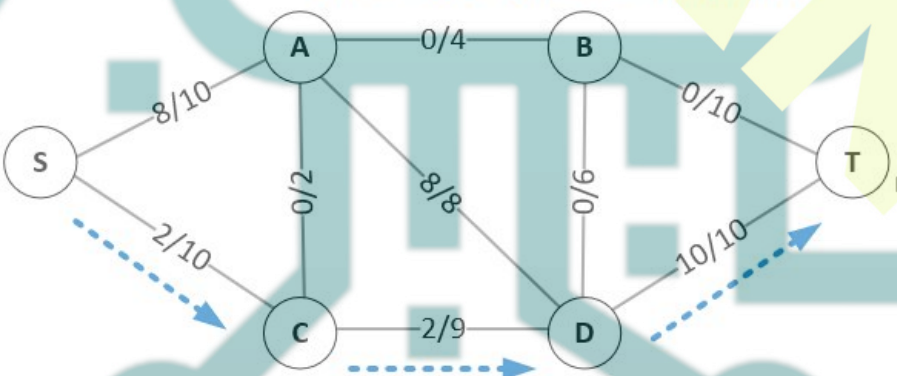


Choix du chemin avec un flot de 8



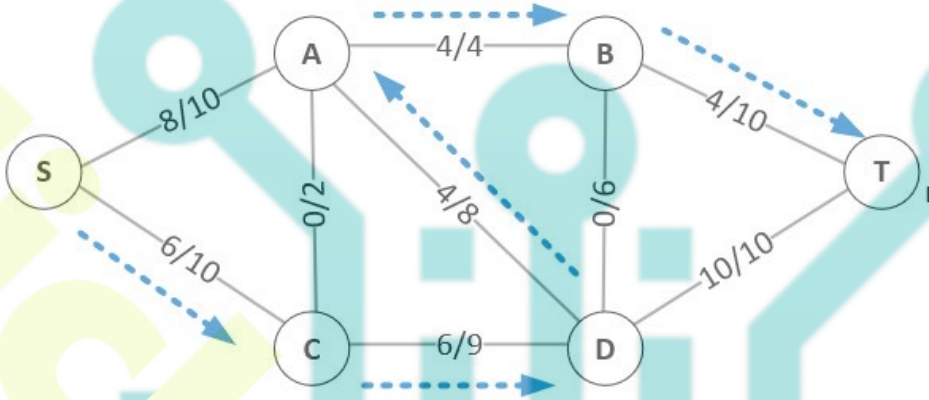
Valeur nécessaire 8

Choix du chemin avec un flot de 2 parce que D-T est le goulot d'étranglement



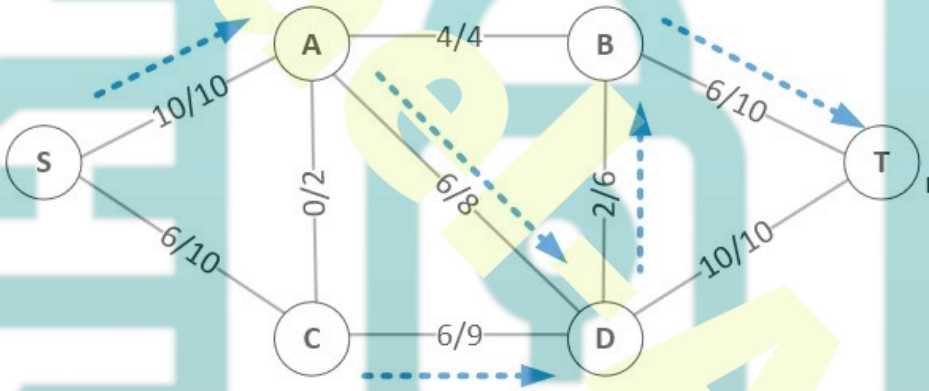
Valeur nécessaire 10

Choix du chemin avec un flot de 4 par un autre chemin

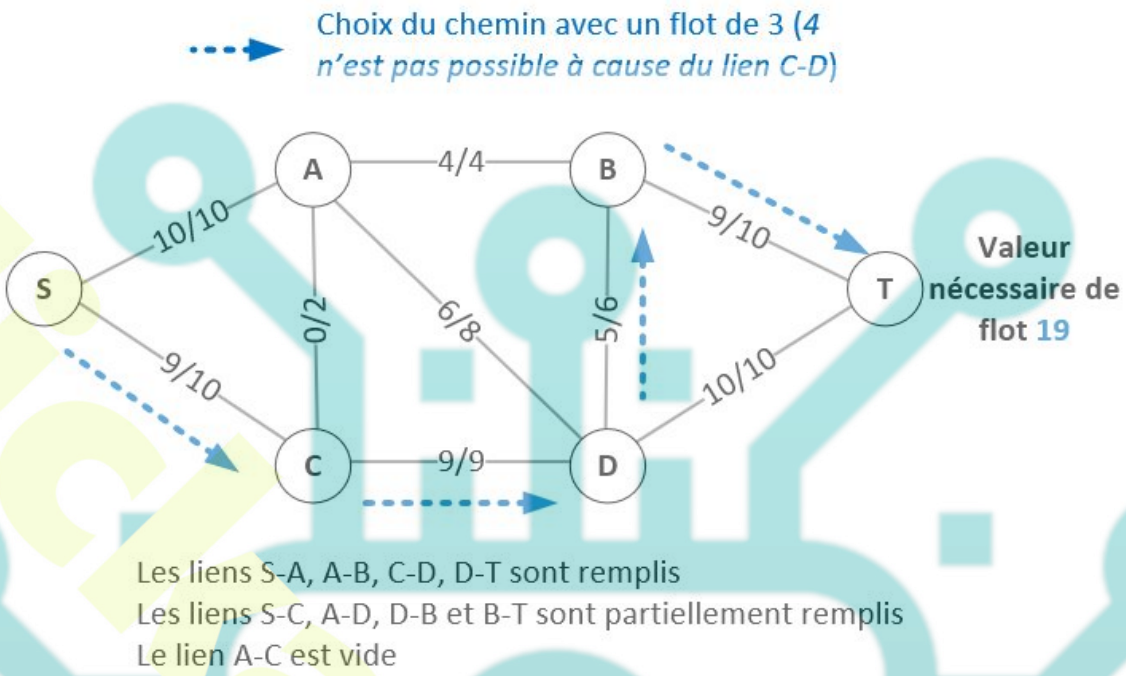


Valeur nécessaire 14

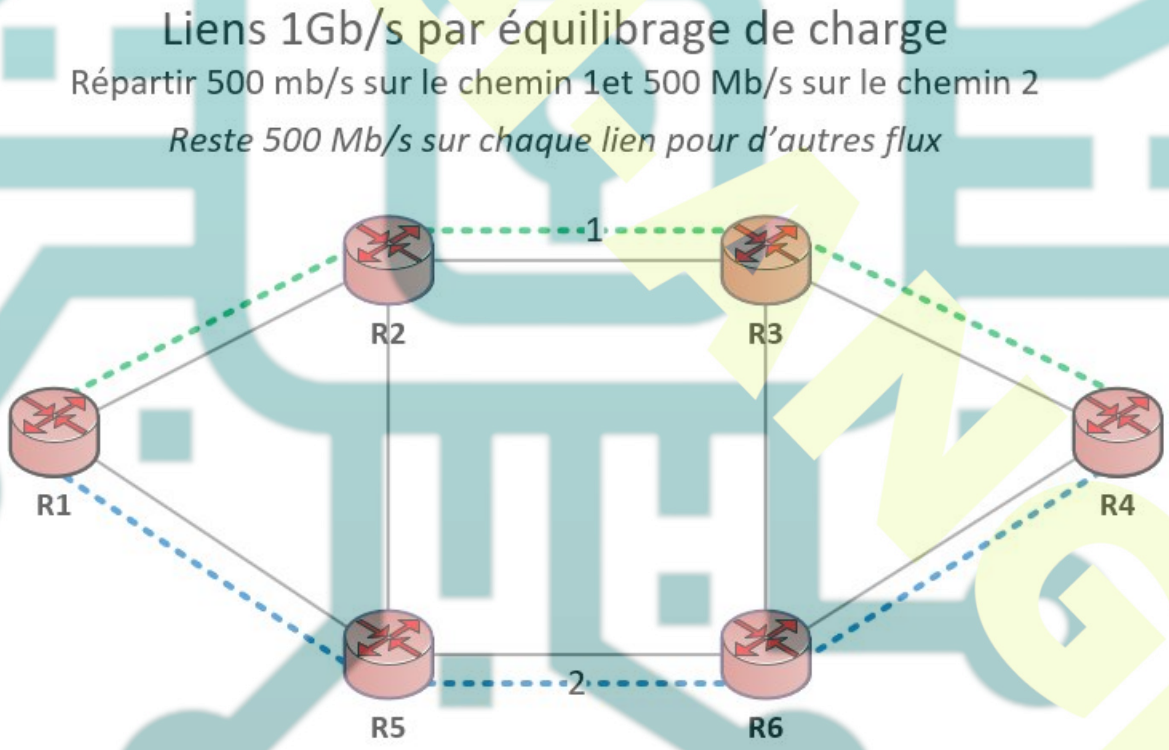
Choix du chemin avec un flot de 2 par un autre chemin *(A-B n'est plus possible D-T non plus)*



Valeur nécessaire 16



Équilibrage de charges dans l'ingénierie de trafic



Exemple de 3 flux demandant une réservation de bande passante

A veut 1/3, B veut 1/4 et C veut 2/3

Le problème est que l'on ne peut pas répondre à la demande puisqu'elle dépasse la

capacité.

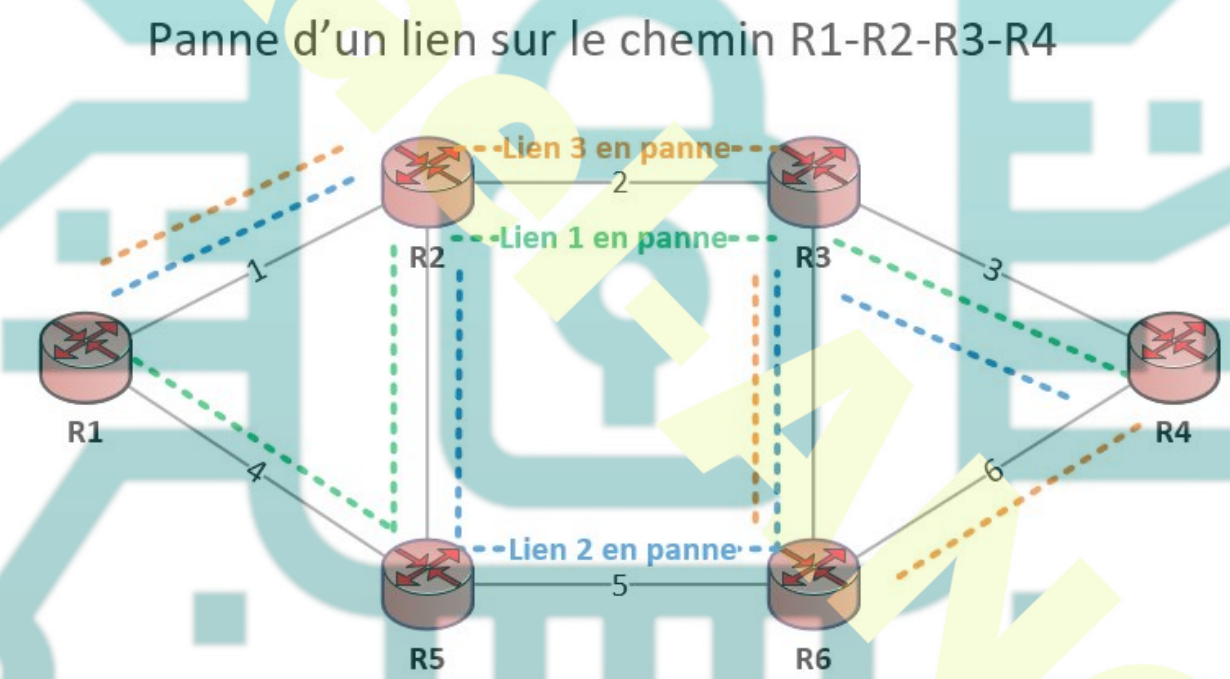
On veut que la bande passante soit répartie équitablement

L'algorithme va donc diviser la bande passante par le nombre de demandes. Il va ensuite trier les demandes en tenant compte de leur besoin par ordre croissant.

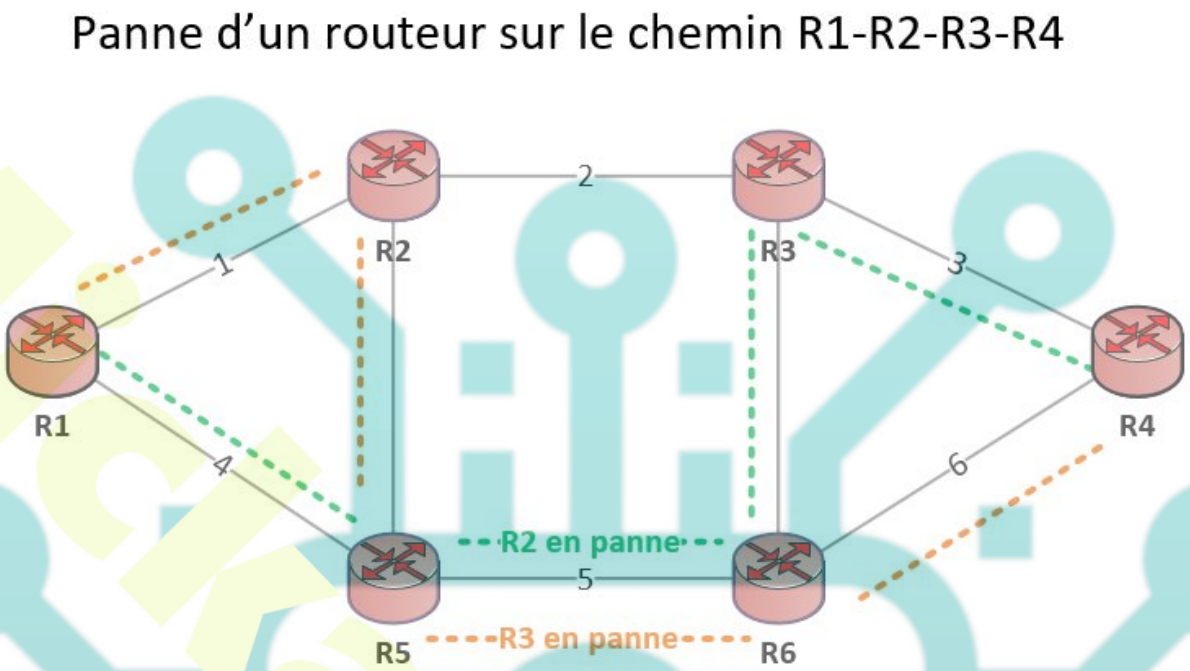
Il commence à satisfaire **B** en lui donnant 1/3 au lieu d'un quart (il restera donc 1/3-1/4= 1/12 de bande inutilisée)

Puis, A et C obtiennent 1/3 + 1/24 de la bande passante (1/12 restant divisé par 2) **A** va rendre 1/24 à **C** qui obtient en fin de compte 1/3 + 1/12 (5/12)

Gestion des pannes d'un lien



Gestion des pannes d'un routeur (2 liens)



Gestion globale de la panne d'un chemin

Dans ce cas prévoir un autre chemin pour R1-R2-R3-R4, c'est à dire R1-R5-R6-R4 et inversement.

Si on utilise la répartition de charge + la tolérance aux pannes sur des liens 1Gb/s, on peut affecter 500Mb/s au chemin R1-R2-R3-R4 et 500Mb/s au chemin R1-R5-R6-R4.

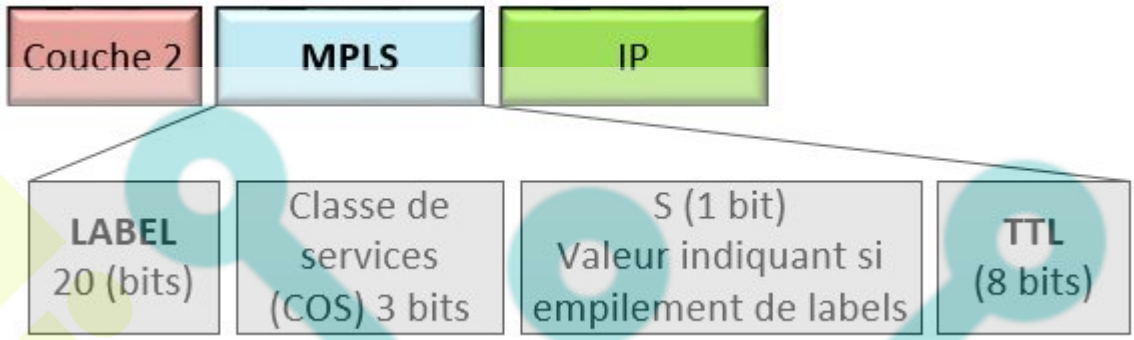
En cas de panne, le lien de secours supportera la charge (ses 500Mb/s + les 500Mb/s de surplus).

Les FEC MPLS

Les paquets IP entrant sur le réseau MPLS sont associés à une FEC (Forwarding Equivalent Class) . Des paquets appartenant à une même FEC suivront le même chemin et auront la même méthode de forwarding.

Une FEC va définir comment un paquet sera acheminé à travers tous le réseau MPLS. En IP, la classification d'un paquet dans une FEC est faite sur chaque routeur, à partir de l'IP destination.

En MPLS, le choix d'une FEC peut se faire sur plusieurs paramètres (adresse IP source, destination et paramètre de QoS (débit, délai) .



La classification des paquets s'effectue à l'entrée du réseau MPLS, par les Ingress LSR. A l'intérieur du backbone MPLS, les paquets sont labellisés, et aucune reclassification n'a lieu.

Chaque LSR affecte un label local, qui sera utilisé en entrée, pour chacune de ses FEC et le propage à ses voisins.

Les LSR voisins sont appris grâce à l'IGP. L'ensemble des LSR utilisés pour une FEC, constituant un chemin à travers le réseau, est appelé Label Switch Path (LSP) . Il existe un LSP pour chaque FEC et les LSP sont unidirectionnels.

Distribution des labels

MPLS utilise les informations de routage fournies par des protocoles comme, OSPF, BGP, pour connaitre le prochain saut pour une FEC donnée.

Pour la création des labels et de LSP, on peut utiliser comme le routage la méthode statique manuelle avec le même manque de souplesse que pour le routage. On peut également s'appuyer sur des protocoles de types BGP, RSVP et LDP (Label Distribution Protocol) .

Dans le cas de BGP et LDP, on ne gère que des flux Best Effort. BGP peut diffuser les labels lorsqu'il distribue les routes IP et les VPN

La distribution implicite de labels aux LSR est réalisée grâce au protocole LDP.

LDP définit une suite de procédures et de messages utilisés par les LSR pour s'informer mutuellement du mapping entre les labels et le flux. Les labels sont spécifiés selon le chemin saut par saut défini par l'IGP (Interior Gateway Protocol) dans le réseau. Chaque nœud doit donc mettre en œuvre un protocole de routage de niveau 3, et les décisions de routage sont prises indépendamment les unes des autres.

LDP est bidirectionnel et permet la découverte dynamique des nœuds adjacents grâce à des messages Hello échangés par UDP. Une fois que les 2 nœuds se sont découverts, ils établissent une session TCP.

Cette distribution permet de s'assurer que le mapping FEC label est bien cohérent dans tous les Routeurs. Il existe 2 protocoles de distribution, compatibles IPv4 et IPv6 :

- **CR-LDP** : Constraint-based Routed Label Distribution Protocol
- **RSVP-TE** : ReSerVation Protocol – Traffic Engineering

Opération sur la pile de labels

Le LSR peut utiliser un **SWAP** qui a pour conséquence de remplacer la valeur de sommet de piles si la table de commutation demande son remplacement.

Lorsque la colonne LABEL OUT est sur la valeur **POP**, le LSR fait passer le label suivant en tête de pile. Si le POP n'est suivi d'aucun autre label, le LSR supprime l'en-tête MPLS et transmet le paquet à la couche 3 (IP) .

Le LSR peut utiliser le **PUSH** pour insérer une pile de labels, ce que fait un Edge LSR lorsque qu'un paquet IP entre sur le réseau et qu'il doit lui affecter un label MPLS.

Le LSR d'entrée ajoute l'en tête entre la couche 2 et la couche 3.

Le LSR de sortie enlève l'en-tête MPLS et décrémente le TTL pour la compatibilité avec IP.

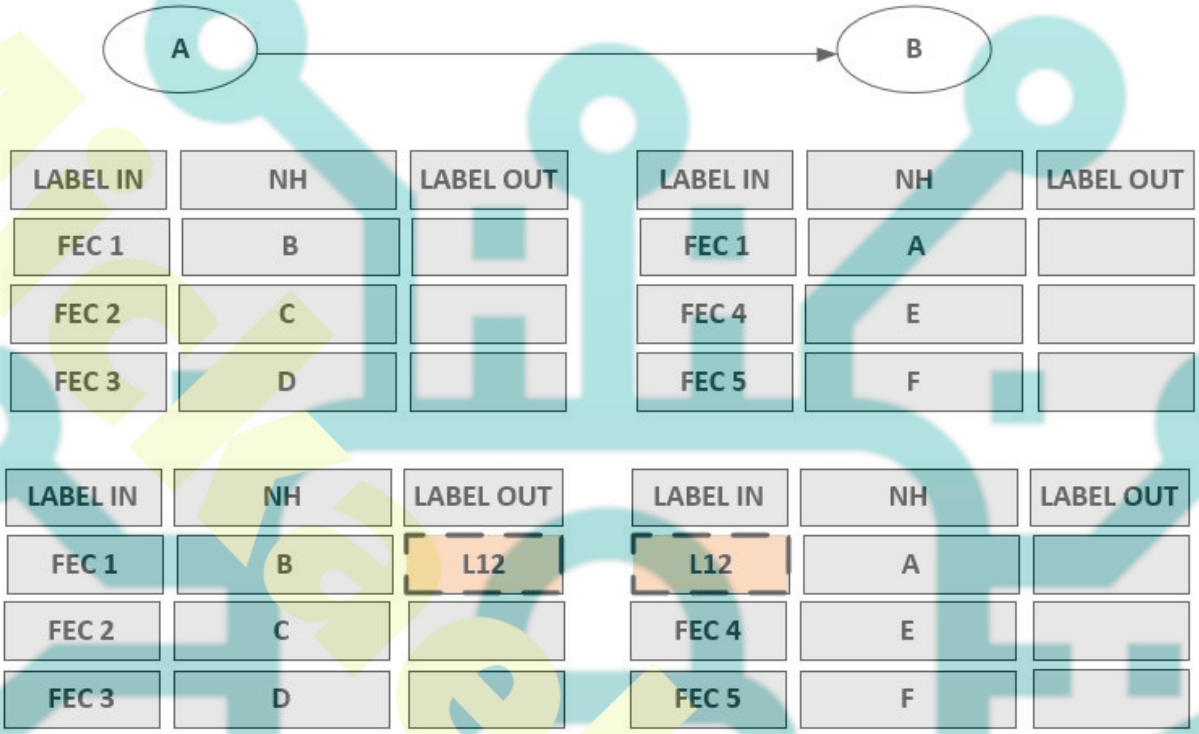
Établissement des chemins

Chaque LSR gère une table de commutation et pour chaque FEC créées par l'administrateur le LSR va créer 2 associations.

- La première est entre la FEC et le prochain saut décidée au niveau 3 par les protocoles de type OSPF, BGP grâce à la table de routage.
- La deuxième est entre la FEC et le label. Le LSR créé le label et le lie à la FEC, puis, il distribue le label à son voisin. Chaque équipement MPLS va devoir créer sa table de commutation en associant un label à chaque FEC.

La valeur de label choisie entre 2 liens doit être équivalente et c'est l'équipement en aval qui décide de la valeur et qui en informe son voisin. Dans une FEC associant un trafic

allant de A vers B, c'est le nœud B en aval qui décide de prendre une valeur qui n'est pas utilisée ou réservée.



Encapsulation des étiquettes dans les VPI/VCI (ATM)

Le format exact d'une étiquette, et la manière avec laquelle cette étiquette est ajoutée au paquet, dépend de la technologie utilisée au niveau de la couche liaison de données.

Une étiquette peut correspondre à un VPI/VCI ATM (Virtual Path Identifier / Virtual Channel Identifier) . Pour d'autres types de couches de niveau 2, comme par exemple Ethernet ou PPP (Point-to-Point Protocol) , l'étiquette est directement ajoutée au paquet de données dans une entête MPLS « shim », qui est placée entre les entêtes des couches 2 et 3.

Dans le cas d'un réseau MPLS sur ATM, l'étiquette correspond à un VPI/VCI ATM.

Extension TE des protocoles

Présentation

Un protocole d'ingénierie de trafic prend en compte des métrique beaucoup plus subtiles que les protocoles de routage classiques.

- Ils peuvent s'adapter dynamiquement en prenant en compte la QOS (bande passante disponible, les délais, les pertes ou un coût).
- Il permet de diffuser les valeurs de ces métriques dans le réseau en gérant une base de données qui stocke ces valeurs (**TED** – Traffic Engineering Database)

Extension TE des protocoles

Pour intégrer le QOS, les protocoles classiques ont été étendus (OSPF-TE, ISIS-TE) .

MPLS a également été étendu dans une version d'ingénierie de trafic avec le protocole MPLS-TE.

Cette extension permet de prendre en compte les métriques de la TED dans le calcul des chemins et de configurer ces chemins grâce au protocole RSVP.

Le principe est d'agréger des flux dont le comportement est similaire et de les insérer dans un même LSP.

Un LSP unidirectionnel est appelé un **TRUNK**

MPLS-TE doit résoudre 3 problèmes :

- Faire la correspondance entre les paquets et les classes d'équivalence (FEC)
- Choisir les TRUNKS en fonction des FEC
- Faire le lien entre les TRUNKS et la topologie physique du réseau (LSP)

Le TRUNK (tunnel)

Un TRUNK est un ensemble d'attributs (sélection de chemin, priorité, préemption, résilience ou de politique de trafic)

Pour permettre l'ingénierie de trafic, le protocole de routage interne (IGP) doit être un protocole à état de liens car on doit déterminer le chemin à emprunter par le tunnel, les routeurs doivent avoir la connaissance complète du réseau.

Pour choisir le meilleur chemin correspondant aux critères de QOS, l'algorithme OSPF a été modifié pour tenir compte de ces contraintes. Cet algorithme est appelé PCALC et permet donc routage contraint.

L'algorithme **PCALC** est le suivant :

- Supprimer les liens qui ne disposent pas de la bande passante suffisante ;
- Supprimer les liens qui ne correspondent pas à l'affinité demandé ;
- Exécuter l'algorithme de Dijkstra sur la topologie restante (avec les métriques de l'IGP) ;

L'établissement d'un tunnel, après exécution de l'algorithme PCALC, est réalisé grâce au protocole RSVP-TE

1. La préemption
Imaginons que le chemin A qui est le plus prioritaire subisse une panne sur l'un de ses liens. Il faut donc trouver un chemin de secours. Cependant, celui-ci peut être utilisé par un autre chemin B, dans ce cas, il faut rerouter le chemin B également.

2. La résilience
Elle permet de définir la conduite à tenir en cas de défaillance (un TRUNK doit-il être rerouté ? le chemin de secours doit-il garantir la même bande passante ?)

Associer MPLS-TE et DiffServ

Pour associer les comportements des deux protocoles, le marquage DiffServ doit être associé au champs TC dans MPLS.

Le champ TC valant 3 bits, il ne permet que 8 comportements.

MPLS n'impose pas d'algorithme pour les chemins contraints, ni pour la réservation de ressource ou l'établissement de TRUNK.

RSVP

L'IETF à lui choisi **RSVP** (Ressource Reservation protocol) pour gérer cette fonctionnalité. RSVP est un protocole de signalisation (pas de données).

A l'origine, RSVP a été défini pour faire la réservation de ressource dans IntServ.

RSVP-TE permet de :

- Créer un LSP le long d'une route explicite
- Etablir un LSP en distribuant des labels
- Définir les besoins en bande passante des liens qui forment un LSP

Etablissement d'un tunnel

Il utilise les routes définies dans le protocole de routage pour envoyer des messages **PATH** entre la source et la destination demandant à chaque nœud traversé s'il dispose des capacités nécessaires à une réservation de ressource donnée.

Si le message **PATH** arrive à destination, cela veut dire que l'on a trouvé un chemin satisfaisant la demande.

Contenu des messages PATH

- **Session –** LSP transportant de l'IPv4 ou IPv6 ?
- **ERO** – EXPLICIT_ROUTE (par quels LSR veut-on passer ?)
- **RRO** – ROUTE_RECORD (quels LSR ont été traversés ?)
- **Style** – Quelle sorte de réservation ? Avec QoS sur le lien ? Etc.
- **Session attribute** – LSP_TUNNEL_RA ou LSP tunnel

La destination envoie le message **RESV** comme accusé réception.

Pour obliger le message **RESV** à passer par le même chemin, on indique dans le message **PATH** l'expéditeur et le chemin à utiliser.

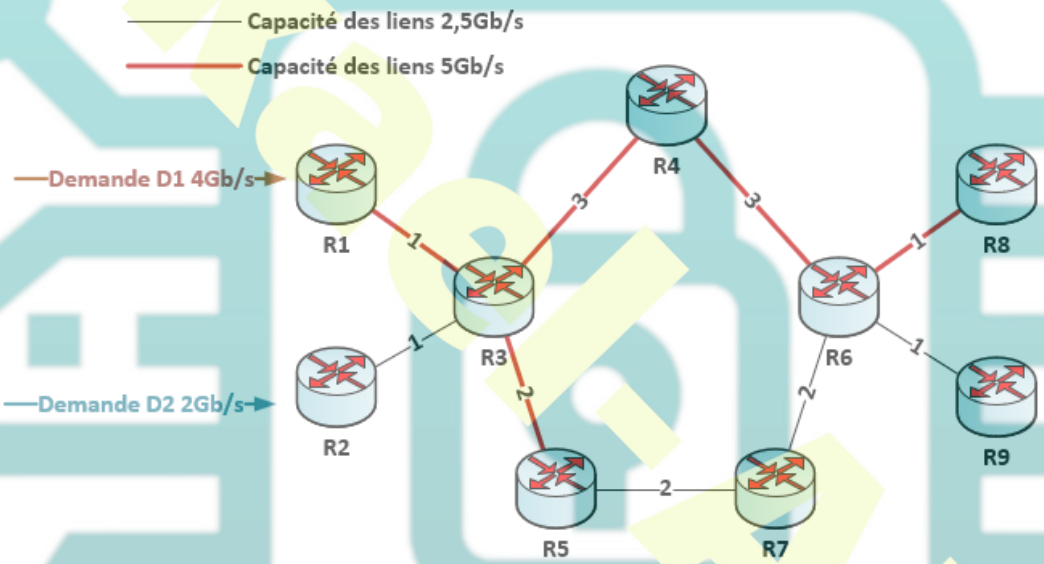
1. Dans le message PATH, on fait une demande de label pour une FEC donnée (caractéristiques d'ingénierie de trafic désirées). Cela indique aux routeurs traversés qu'il faut mettre en place un label pour le LSP.

2. La destination choisit le label pour la FEC et renvoie ce label dans le message RESV. Lorsque la source reçoit le message RESV, le chemin est paramétré.

Exemple de routage contraint avec le protocole MPLS-TE pour acheminer les flux.

Pour choisir les LSP, l'administrateur met en place une stratégie de routage contraint CSPF (Constrained Shortest Path First) dont le principe est d'assurer que les plus courts chemins respectent un ensemble de contraintes.

La contrainte à respecter est la bande passante requise par la demande.



Le plus court chemin calculé pour satisfaire la contrainte de bande passante de la demande D1 passera par R1-R3-R4-R6-R8 et les labels devront être distribués par RSVP-TE.

Pour la demande D2, Le plus court chemin calculé sera R2-R3-R5-R7-R6-R9

Automatisation de l'ingénierie de trafic

Pour les flux exigeants, il faut proposer une convergence en moins de 200ms. Evidemment, les protocoles peuvent permettre la convergence rapide, mais le passage à l'échelle peut être compliqué à mettre en œuvre.

SDN

Les Software Defined Networking cherche à automatiser la configuration des réseaux.

Le but est de pouvoir rendre le réseau programmable (initialisation, contrôle et gestion) et dynamique (grâce aux interfaces ouvertes).

C'est un ensemble de techniques visant à faciliter l'architecture, la livraison et l'opération de services réseaux de manière déterministe, dynamique et pouvant être déployés à grande échelle.

La technique consiste à séparer le plan de contrôle du plan de données. Le SDN met en œuvre la virtualisation du réseau ce que lui permet d'automatiser la surveillance, de gérer l'ingénierie de trafic et de se protéger contre les attaques de déni de service.

SDN vise également à réduire les coûts d'investissement (CAPEX) et les frais opérationnels (OPEX) en centralisant le contrôle des équipements et en automatisant le dimensionnement.

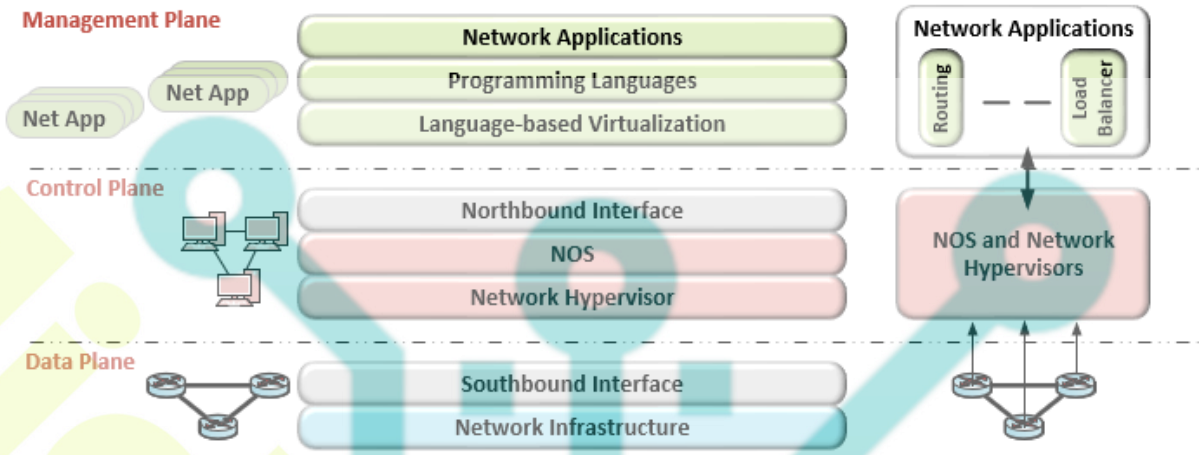
Les propriétés de SDN sont la flexibilité et la puissance, une architecture programmable, une abstraction entre le plan de contrôle et le plan données.

Le SDN s'adapte à tous les types de réseaux et peut être utilisé par exemple pour passer de IPv4 à IPv6 en créant automatiquement les tunnels.

Le SDN est adaptable (il se réorganise en fonction des changements du réseau), il est capable de transporter des micro-flux et des agrégats.

SDN met en œuvre la propriété d'élasticité, c'est-à-dire qu'il adapte les éléments du réseau en fonction des besoins de l'administrateur (ajout de nœuds, économie d'énergie en éteignant les équipements non utilisés, réadapter les requêtes à destination d'un serveur qui vient de migrer dans un Datacenter.

Les architectures AWS, Open Stack et Windows Azure s'intègrent au SDN sur le plan de contrôle et de données.



Le Plan de Gestion (Management Plane)

1. La couche Applications Réseaux (Network Applications)

Elle met en œuvre le contrôle logique qui sera traduit en commandes pour être implémentées dans le « Data Plane », afin de dicter les marches à suivre. Les commandes de routage vont définir un chemin d'un point A à un point B, puis vont charger le contrôleur de mettre en place les règles de transferts pour les équipements traversés. Ainsi le réseau défini par l'application peut être déployé sur n'importe quel réseau traditionnel.

Les applications SDN gèrent : le trafic d'ingénierie, la mobilité et sans fils, la mesure et la surveillance, la sécurité et la fiabilité des données et la mise en réseau.

2. La couche de Programmation (Programming languages)

Créer des abstractions afin de simplifier la tâche de programmation des équipements de transmission. Elle permet la modularité et la réutilisabilité des programmes du code dans le plan réseau, au niveau du Control Plane et favorise la virtualisation de réseau.

3. La couche virtualisation basée sur le langage (Language-based Virtualization)

Cette abstraction simplifie le développement et le déploiement des applications de réseau complexes, tels que la sécurité avancée et d'autres services en agrégeant les configurations.

Le Plan de Contrôle ou Contrôleur (Control Plane)

Le contrôleur est en fait un logiciel qui se substitue au logiciel de commande inclus dans chaque équipement réseau. Il fournit une interface de programmation au réseau. Les fonctions de base du contrôleur se résument en trois catégories : gérer la commutation et le routage des trames en appliquant des règles prédéfinies, effectuer cette tâche

dynamiquement et en fonction des besoins en capacité, enfin, pouvoir être programmé, afin de les exécuter à des moments déterminés par l'administrateur, en fonction des exigences métier par exemple.

1. L'interface **Southbound** permet la communication entre le contrôleur SDN et les nœuds de réseau (routeurs et routeurs physiques et virtuels) afin que le routeur puisse découvrir la topologie du réseau, définir les flux réseau et implémenter les requêtes relayées via les API northbound.

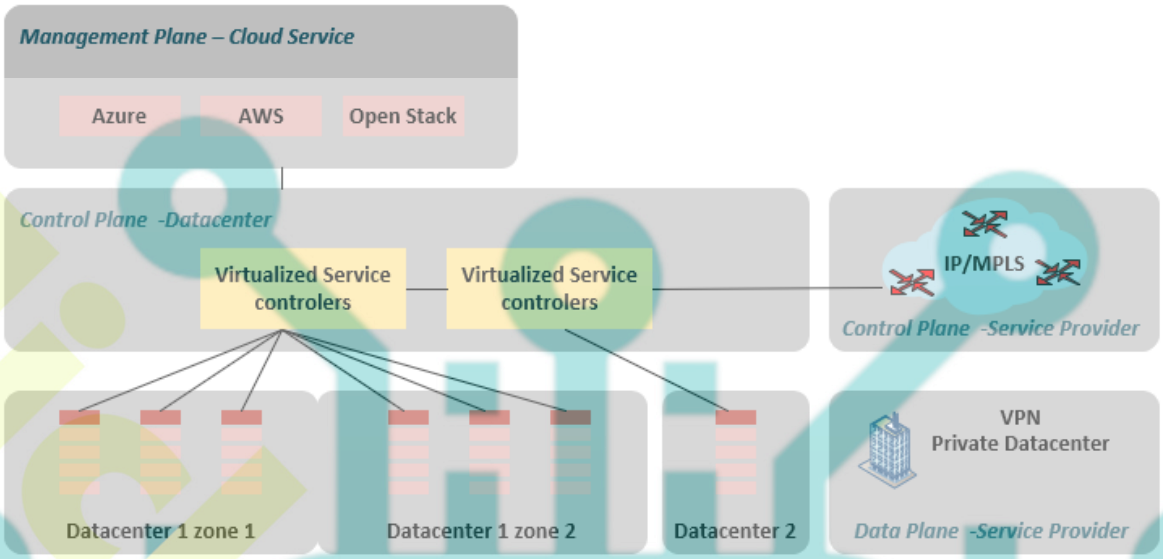
2. L'interface **Northbound** décrit la zone de communication prise en charge par le protocole entre le contrôleur et les applications ou les programmes de contrôle de couche supérieure.

Dans un datacenter d'entreprise, les fonctions des APIs Northbound incluent des solutions de gestion pour l'automatisation et l'orchestration, ainsi que le partage de données exploitables entre systèmes. Les fonctions des APIs orientées vers le sud incluent la communication avec la matrice de commutation, les protocoles de virtualisation de réseau ou l'intégration d'un réseau informatique réparti

1. Le system d'exploitation (Network Operating Systems NOS)
Comme avec les systèmes d'exploitation classiques, le but du NOS est de fournir des couches d'abstractions, des services et des API communes. Le NOS diffuse des informations sur l'état et la topologie du réseau et la détection des périphériques. Il permet donc de définir une politique sur le réseau, le gestionnaire de réseau n'a plus besoin de s'occuper des équipements de distribution de données et de routage.

2. Couche hyperviseurs du réseau (Network Hypervisors)
Elle permet à différentes machines virtuelles de partager la même ressource matérielle. Dans le Cloud IaaS, chaque utilisateur peut avoir ses propres ressources virtuelles à partir de calcul de stockage. L'objectif est de permettre à plusieurs réseaux logiques de partager la même infrastructure physique en fournissant une couche d'abstraction permettant à des réseaux multiples et divers de coexister.

Exemple de SDN


Management Plane – Cloud Service
Azure
AWS
Open Stack
Control Plane -Datacenter
Virtualized Service controlers
Virtualized Service controlers
IP/MPLS
Control Plane -Service Provider
Datacenter 1 zone 1
Datacenter 1 zone 2
Datacenter 2
VPN
Private Datacenter
Data Plane -Service Provider