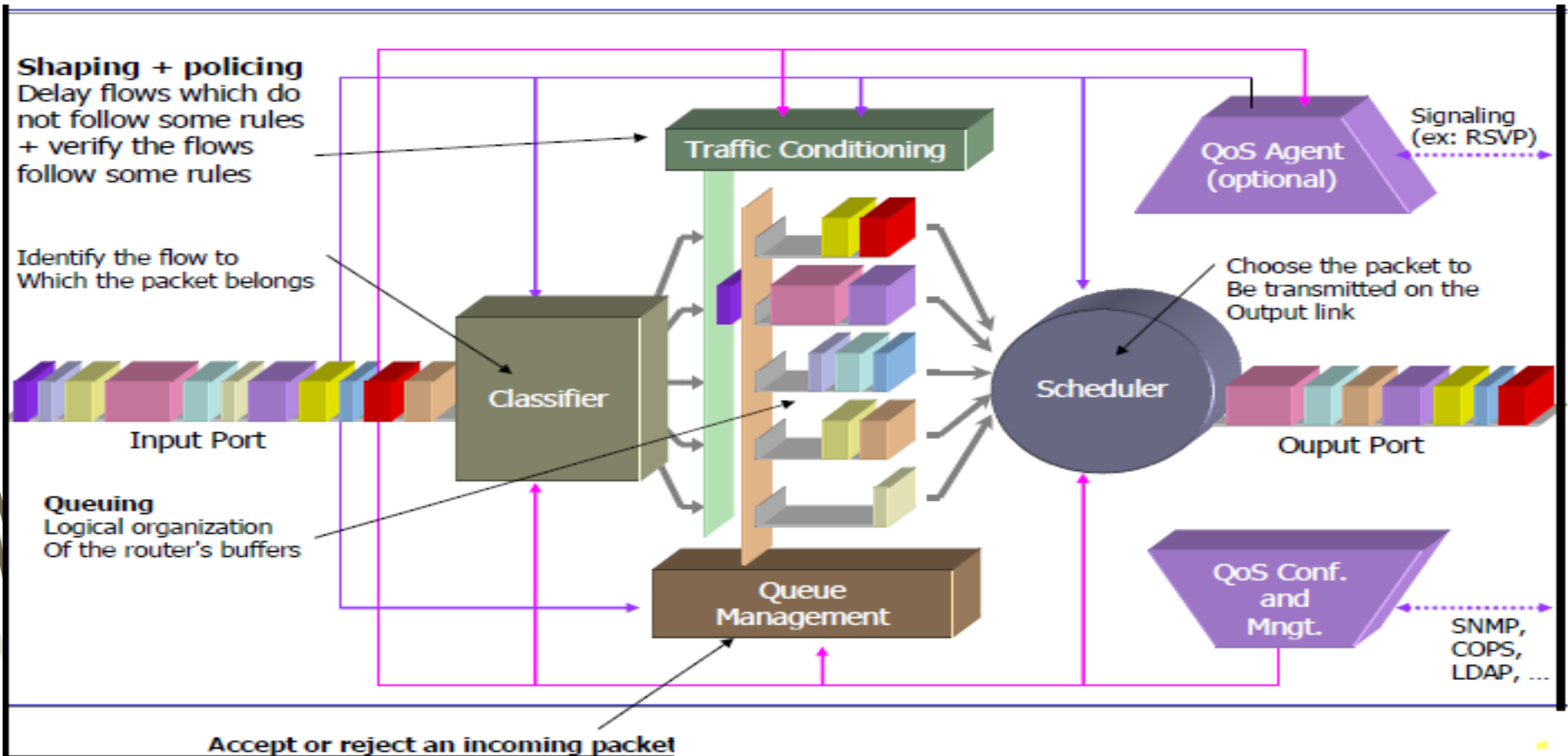


Chapitre 3: *Mécanismes de la QoS*

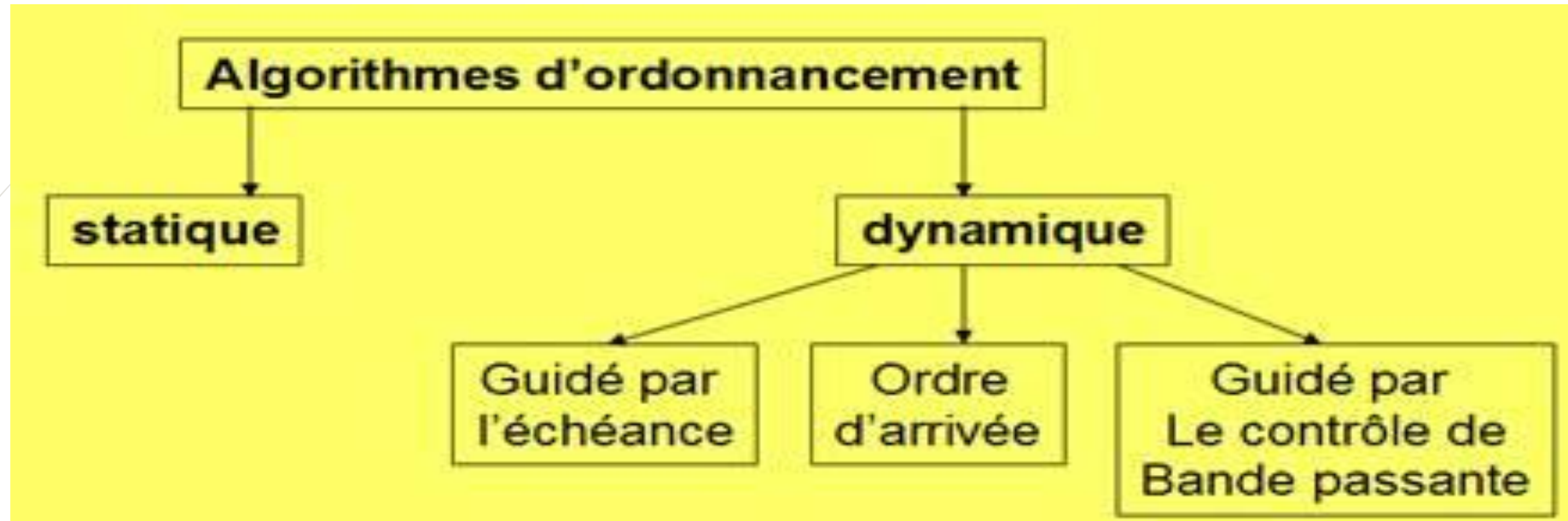
Vue globale

2



- ❑ Pour remédier aux problèmes de partage de ressources, des mécanismes d'ordonnancement sont implémentés dans les routeurs, ce qui permet d'administrer la circulation des applications sur le réseau en attribuant des parts de ressources, plus ou moins équitables selon l'algorithme utilisé
- ❑ Pour gérer les paquets reçus et les transmettre, un routeur devra forcément employer une ou plusieurs files d'attente
- ❑ L'ordonnancement et la gestion des files d'attente conditionne grandement les performances des routeurs et constitue un élément important de la QoS.

- ❑ Un routeur peut être vu comme une collection :
 - ✓ de processus d'entrée qui assemblent les paquets, vérifient leur intégrité (checksum) et fait de la classification suivant des critères basés sur l'adresse IP, les numéros de ports ou les types de protocoles encapsulés dans les datagrammes IP.
 - ✓ de processus de routage et de retransmission qui détermine l'interface de Destination
 - ✓ de processus de sortie qui fait de la gestion des files d'attente, de l'ordonnancement et transmettent les paquets vers le nœuds suivants.
- ❑ Chaque processus opère sur un paquet à la fois et travaille de manière asynchrone vis à vis des autres.
- ❑ La liaison entre les processus d'entrée, de routage et de sortie se fait via des files d'attente : file d'attente en entrée (Ingress interface) et file d'attente en sortie (Egress interface)



- ❑ Dans un algorithme *d'ordonnancement statique* les priorités des applications sont fixes et invariantes au cours du temps. Un message moins prioritaire n'est servi que si tous les messages plus prioritaires en attente sont tous servis.
- ❑ Dans un *algorithme guidé par l'échéance ou guidé par le contrôle de bande passante* : Cette catégorie d'ordonnancement permet de partager dynamiquement des ressources en respectant le critère de l'échéance ou la bande passante.

Chapitre II: mécanismes de la QoS

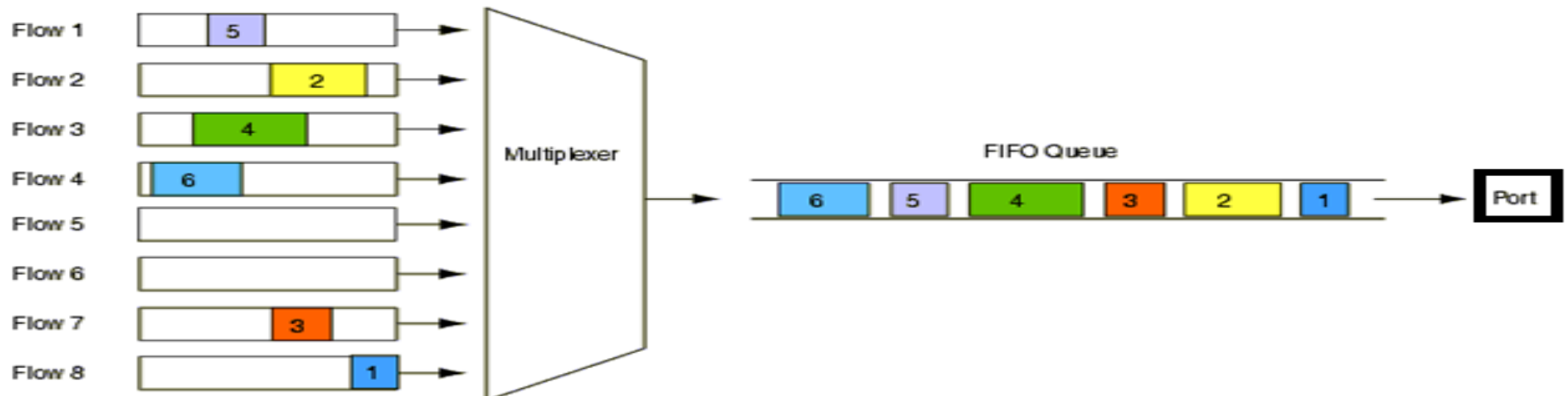
ChapII-Section A :Ordonnancement et gestion des files d'attente

FIFO, First In, First Out

FIFO

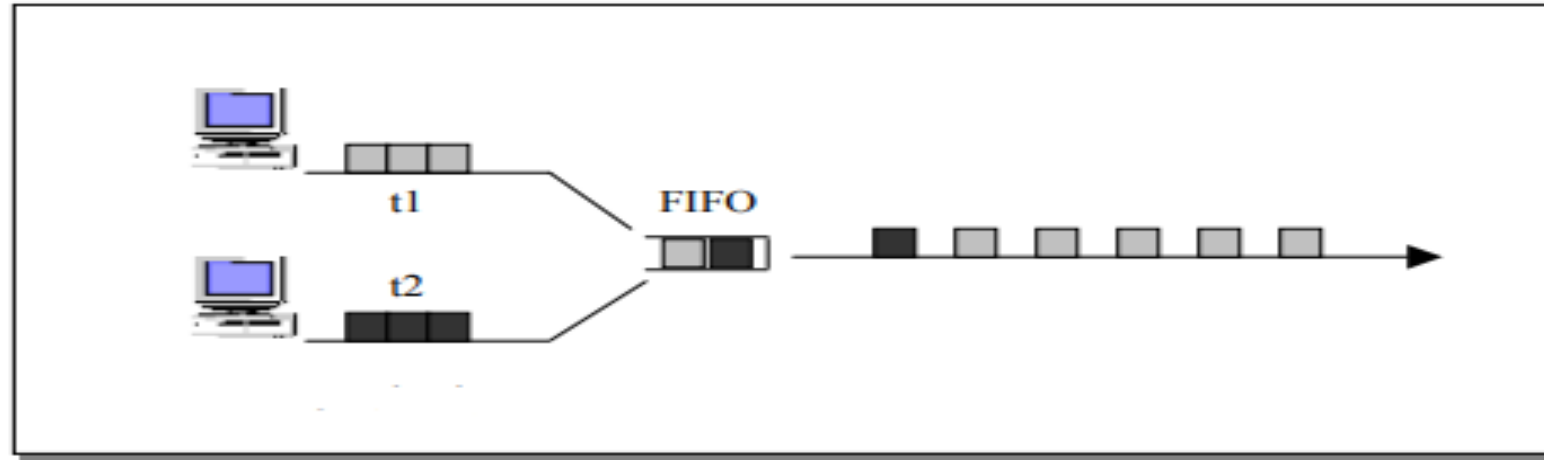
7

- ❑ La file d'attente FIFO « premier arrivé, premier servi » est l'ordonnancement par défaut, le plus simple qui soit.
- ❑ Les paquets sont mis dans la file de sortie et servis dans l'ordre avec lequel ils ont été reçus par le ou les interfaces d'entrée.
- ❑ Le routeur est muni de huit interfaces d'entrée associés à huit files d'attente FIFO. Les paquets arrivant dans ces files d'attente seront ordonnancés dans une file d'attente unique de sortie suivant leurs temps d'arrivée dans le retour.



- ❑ ☺ Il est le plus rapide au point de vue de la vitesse de transmission des paquets, étant donné qu'elle n'effectue aucun traitement sur ceux-ci. En effet, il ne consomme pas beaucoup de CPU au niveau du système d'exploitation du routeur.
- ❑ ☺ Cet ordonnanceur est suffisant dans un réseau à forte capacité car on peut considérer que les files restant presque toujours vides, les délais sont alors faibles.
- ❑ ☹ Par contre, dans le cas d'une rafale, la file d'attente peut se retrouver en débordement et les paquets arrivés après la rafale peuvent être jetés. Dans ce cas, les paquets jetés le sont de manière indifférenciée, sans prise en compte du type de trafic auquel ils correspondent.

Exemple FIFO



Scénario utilisant l'ordonnancement FIFO

- ❑ Le premier client utilise des applications élastiques, par exemple un transfert de fichiers (FTP) ; le second participe, par exemple, à une séance de vidéoconférence.
- ❑ La première machine commence ses requêtes à l'instant $t1$, tandis que la deuxième utilise son application à l'instant $t2$, tel que $t1$ inférieur à $t2$.
- ❑ Les paquets envoyés par l'application FTP se présentent à l'entrée de la file d'attente avant les paquets de vidéoconférence et occupent fortement la bande-passante disponible.

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

Priority Queuing ,PQ

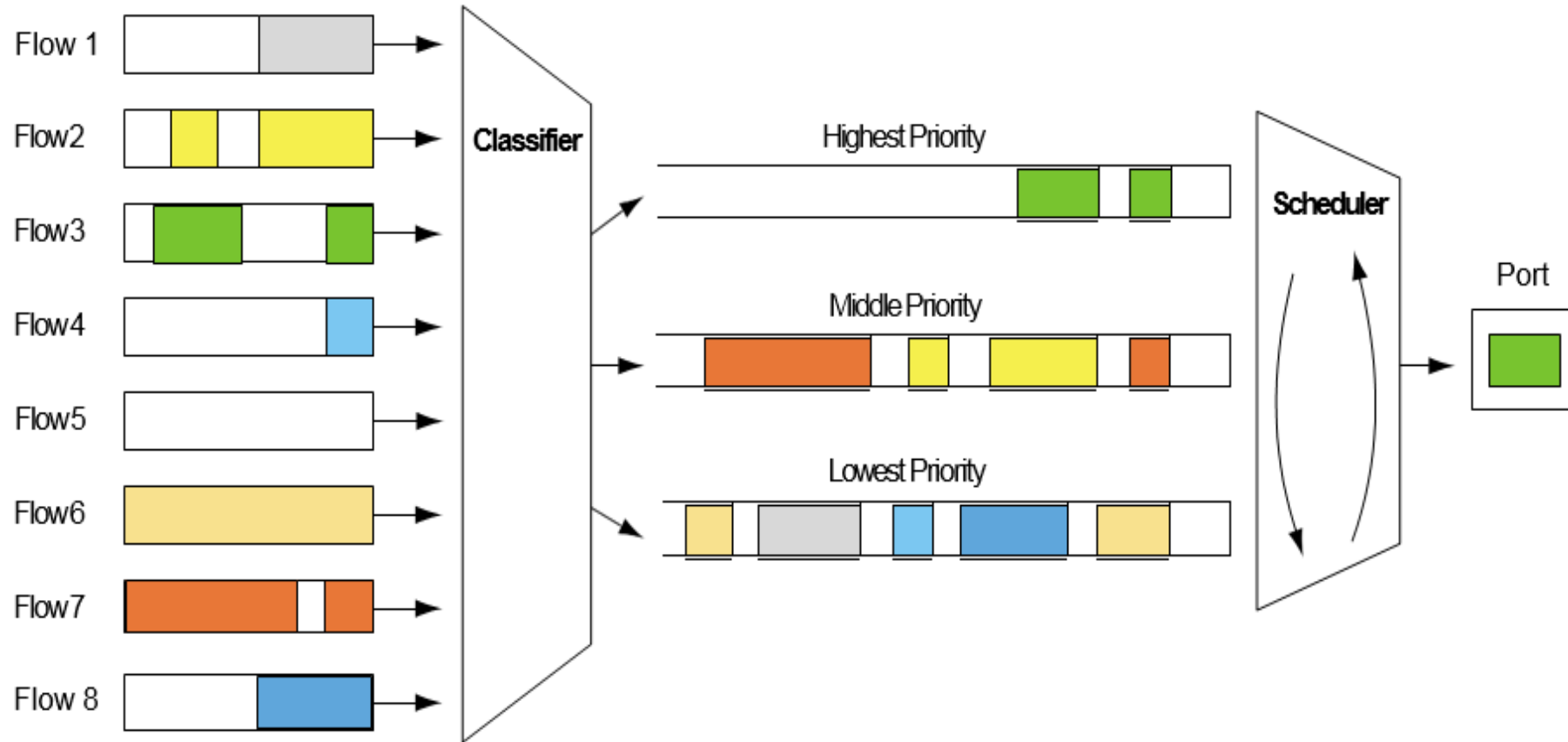
Priority Queuing : PQ

PQ utilise un processus de classification pour ordonner les paquets provenant des flux vers différentes files, qualifiées par une priorité. On distingue ainsi quatre niveaux de priorité:

- ✓ La file de plus haute priorité contiendra les paquets jugés par le classificateur comme étant les plus prioritaires, c'est-à-dire les paquets « gourmands » qui demandent une disponibilité de ressources suffisantes. Ces paquets appartiendront par exemple à des flots de type «hard real-time ».
- ✓ Les applications de type temps-réel seront classées vers les files de priorité moyenne
- ✓ les applications adaptatives seront plutôt orientées vers la priorité normale ou faible. A l'intérieur de ces files à priorité, le traitement est effectué selon l'ordonnancement FIFO. C'est à la sortie que se présente l'ordonnancement par priorité.

Priority Queuing : PQ

12



Avantages ☺

- ❑ simplicité, rapidité et faible coût au niveau du système d'exploitation routeur,
- ❑ les priorités absolues permettent de privilégier de manière absolue un trafic par rapport à un autre.

Priority Queuing : PQ

13

Inconvénients ☹️

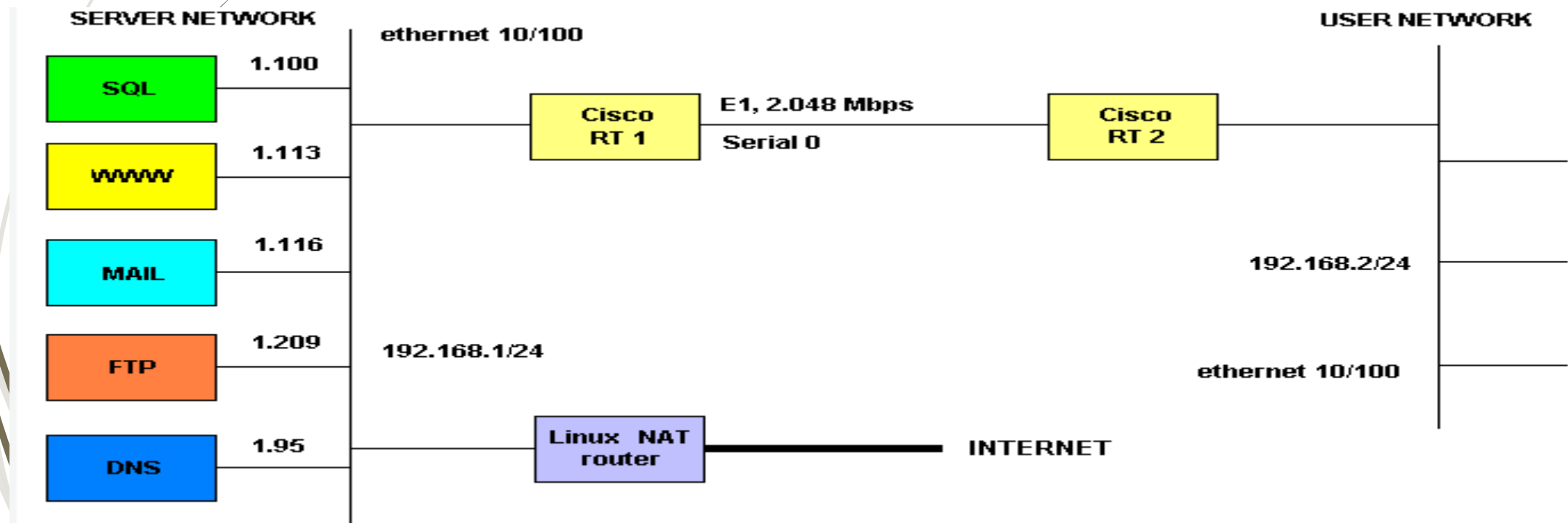
- ❑ Les priorités absolues peuvent causer le problème de famine se pose lorsque l'ordonnancement statique effectué par l'administrateur va aiguiller trop de paquets dans la file " Priorité élevée « par rapport aux autres files. Dans ce cas, le routeur ne routera que les paquets de la file " Priorité élevée " en délaissant ceux des autres files, d'où la "famine" pour les autres files. Il faudra donc s'assurer de ne pas "faire trop passer" de paquets dans la file "Priorité élevée ".
- ❑ Il y a seulement 3 niveaux de priorité, comment faire alors pour classer plus de 4 classes de trafic ?

Priority Queuing : PQ

14

Exemple PQ

Quatre machines serveurs sont connectées au même segment Ethernet, lui même connectant un routeur Cisco RT1 permettant d'aller vers le réseau des utilisateurs des services installés dans les quatre serveurs. Un deuxième routeur Linux NAT est installé pour sortir vers le réseau Internet.



Exemple PQ

Les services fournis ne sont pas de la même importance pour les utilisateurs. C'est pourquoi, l'administrateur a configuré le routeur Cisco RT1 de la manière suivante :

- ☐ les paquets arrivant du serveur SQL sont les paquets les plus prioritaires. En effet, c'est ce service qui est considéré comme le plus important pour les utilisateurs.
- ☐ Le service Web est considéré important mais en seconde position.
- ☐ Les paquets arrivant du serveur MAIL sont plus prioritaires que ceux arrivant du serveur FTP.
- ☐ Le service transfert de fichiers est considéré comme le moins prioritaire de tous les services existants. Il est donc moins prioritaire que le service de messagerie électronique.

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

RR, Round Robin

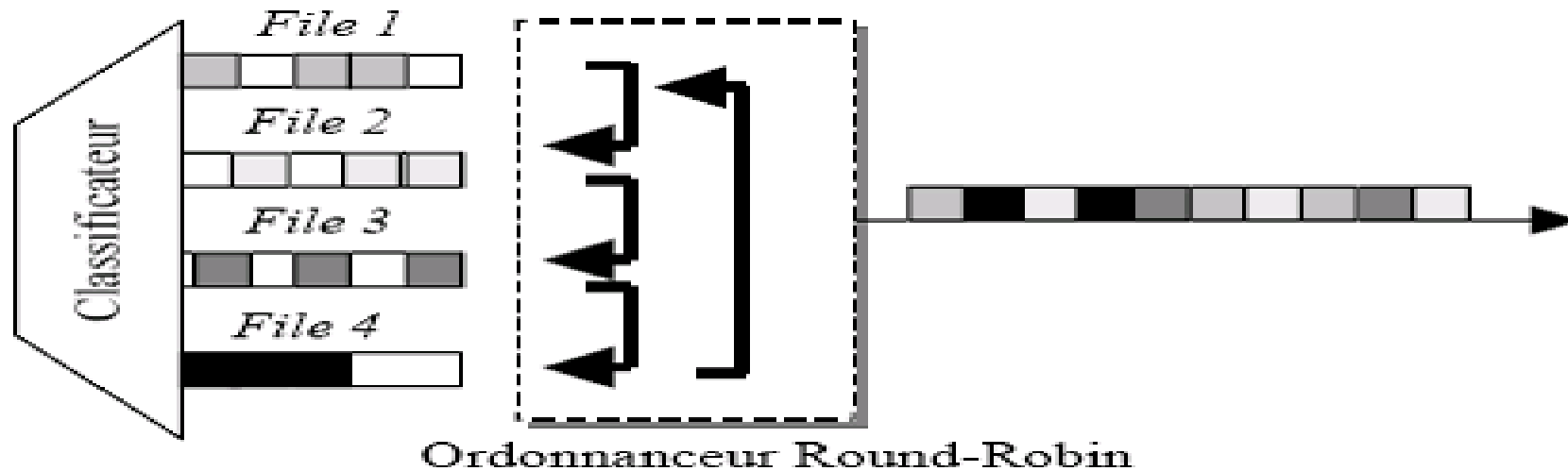
RR, Round Robin

- ☐ Dans la discipline Round Robin les paquets sont classés par flux par le système et insérés dans une file d'attente particulièrement dédiée à chaque flux.
- ☐ Les flux sont servis en tourniquet (Round Robin) paquet par paquet.
- ☐ Si la file contient un paquet, ce dernier sera servi. Par contre, si aucun paquet ne se trouve à ce moment, l'ordonnanceur passe directement à la file suivante.
- ☐ Donc, les files d'attente vides sont sautées par l'ordonnanceur et revisitées au tour suivant.

RR, Round Robin

18

Exemple RR



- ❑ Après classification des flux vers quatre files d'attente, l'ordonnanceur va examiner la première file. A sa tête, on remarque qu'il ne se présente aucun paquet. L'algorithme incrémente alors son indexation pour examiner la seconde file : un paquet est présent et est donc envoyé sur la sortie du lien.
- ❑ L'ordonnanceur passe à la file suivante et effectue le même traitement : un paquet se trouvant à la tête de la file, va être transmis sur le lien. Enfin, la file 4 est examinée, mais aucune information n'est encore présente, l'ordonnanceur va boucler son indexation vers la première file d'attente

Avantages 😊

- ❑ L'algorithme RR ne discrimine aucune file et essaye d'effectuer par conséquent le partage de bande passante le plus équitablement possible. elle fournit un niveau de service minimum à chaque flux indépendamment du comportement des autres flux. ➡ RR permet de résoudre le problème de la
- ❑ En effet, les files contenant des paquets deux fois plus volumineux que ses concurrentes se verront attribuer, raisonnablement, deux fois plus de bande passante.

Inconvénients ☹️

- ❑ La discipline de service RR n'est pas destinée à servir des flux ayant des besoins différents en terme de bande passante. Son objectif se limite à fournir la même portion de la bande passante à tous les flux.
- ❑ La garantie équitable du même taux de bande passante à tous les flux n'est possible que lorsque tous les paquets ont la même taille. En effet, les flux de paquets de large taille auront plus de bande passante que les flux de taille de paquets plus petite.
- ❑ La discipline de service RR est sensible à l'ordre d'arrivée des paquets. En effet, si un paquet arrive à une file d'attente vide immédiatement après avoir été visitée par l'ordonnanceur, alors ce paquet doit attendre la fin de service de tous les paquets des autres files d'attente avant d'être transmis.

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

WRR, Weighted Round Robin

WRR, Weighted Round Robin

22

- ❑ L'ordonnanceur parcourt les différentes files et les sert en fonction du poids associé à chacune d'elles. Cet ordonnanceur reste simple.
- ❑ L'équité est vérifiée si les paquets des différentes files ont la même longueur.
- ❑ Dans le cas contraire, ce sont les files avec les paquets de plus grandes tailles qui seront favorisées.

Paramètres:

- L_i est la taille moyenne des paquets du flux i
- ϕ_i : poids du débit i ($\phi_i > 0$)



Calculez le nombre de paquets à servir à chaque tour

$$x_i = \phi_i / L_i$$

WRR, Weighted Round Robin

Avantages WRR 😊

- ❑ WRR garantit que toutes les classes de service ont accès à au moins une certaine quantité configurée de bande passante réseau pour éviter une pénurie de bande passante
- ❑ WRR permet un contrôle approximatif du pourcentage de bande passante du port de sortie alloué à chaque classe de service

Inconvénients WRR ☹

- ❑ Les files avec les paquets de plus grandes tailles qui seront favorisées

Exemple WRR

On considère un lien à 45 Mb/s utilisé par 500 connexions ayant des paquets de taille fixe égale à 500 octets. 250 connexions ont un poids de 1 et 250, un poids de 10, calculer la durée d'un tour ?

Correction

On considère un lien à 45 Mb/s utilisé par 500 connexions ayant des paquets de taille fixe égale à 500 octets. 250 connexions ont un poids de 1 et 250, un poids de 10.

- Chaque paquet dure $500 * 8 / 45 \text{ Mb/s} = 88.8 \text{ microsecondes}$
- Durée d'un tour = $(250 * 1 + 250 * 10) * 88.8 = 244.2 \text{ ms}$
- Cette durée ne permet pas d'avoir des flux audio ou vidéo

Conséquence : WRR est une discipline efficace pour des paquets de petites tailles avec des durées de tour petites (c'est le cas d'ATM par exemple (53 octets))

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

DRR, Deficit Round Robin

DRR, Deficit Round Robin

- DRR essaye de résoudre le problème de l'équité de WRR.
- Il introduit un compteur de déficit lié à chaque file. A chaque tour, ce compteur est incrémenté d'un quantum q_K pour la classe K .
- Si la valeur du compteur devient plus grande que la taille du premier paquet dans la file, alors ce paquet sera envoyé au réseau et le compteur de la file sera réduit de la valeur de la taille du paquet. Cette valeur sera sauvegardée pour le prochain tour.
- Dans le cas où la somme du déficit et du quantum est inférieure à la taille du paquet, le déficit est incrémenté du quantum et sauvegardé, mais le paquet ne sera pas émis. L'ordonnanceur passera ensuite à la prochaine file.
- Quand l'ordonnanceur se présente devant une file vide, le déficit de cette dernière est initialisé à zéro afin de garder l'équité.

DRR, Deficit Round Robin

27

Algorithme DRR

Associer un compteur $C[k]$, initialisé à 0, à chaque queue k , Lorsque la connexion k est visitée par DRR

- ❑ Si la queue k est non vide

$$D[k] = C[k] + q_k(\text{quantum}) ;$$

- Si $\text{Taille}(\text{tetequeue}[k]) \leq D[k]$

Le paquet est transmis;

$$D[k] = D[k] - \text{taille du paquet transmis};$$

- Si $\text{Taille}(\text{tetequeue}[k]) > D[k]$

Passer à la queue suivante;

- ❑ Si la queue k est vide $D[k] = 0$

DRR, Deficit Round Robin

28

Exercice DRR

$C(k)=0$, Quantum $q_k=500$

				D(k)
Flux 1	<input type="text"/>	<input type="text"/>	750 300	<input type="text"/>
Flux 2	<input type="text"/>	<input type="text"/>	200 600	<input type="text"/>
Flux 3	<input type="text"/>	<input type="text"/>	400 950	<input type="text"/>

Calculer 2 tours et déterminer la valeur de Déficit $D(k)$ à chaque fois pour chaque flux?

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

FQ, Fair Queuing (FQ)

FQ, Fair Queuing (FQ)

30

- ❑ Le principe se base tout d'abord sur le classement des flots selon leur caractéristiques, puis sur l'utilisation de plusieurs files d'attente dédiées à ces flots.
- ❑ L'ordonnancement est effectué sur toutes les files contenant des paquets, de manière séquentielle, selon le mécanisme Round-Robin décrit antérieurement.

❑ **Max-min Fair Share**

- Satisfaire un maximum de flots
- Principe
 - Satisfaire ceux qui demandent moins
 - Aucun noeud 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

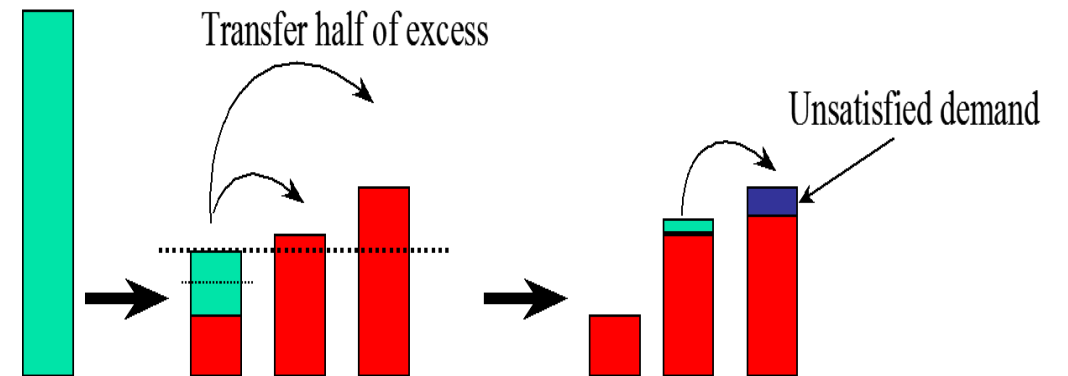
FQ, Fair Queuing (FQ)

31

❑ Max-min Fair Share

N flux partagent une liaison de débit C . Le flux f souhaite émettre au débit $W(f)$, et se voit attribuer le débit $R(f)$.

1. Choisissez le flux, f , avec le plus petit débit demandé.
2. Si $W(f) < C / N$, alors définir $R(f) = W(f)$.
3. Si $W(f) > C / N$, alors définir $R(f) = C / N$.
4. Réglez $N = N - 1$. $C = C - R(f)$.
5. Si $N > 0$, passez à 1.



Avantages FQ 😊

- ❑ L'intérêt d'un tel algorithme est que la séparation des flots vers différentes files permet d'avoir des traitements indépendants.
- ❑ Si un flot désire obtenir plus de ressources que la part qui lui est attribuée, seul ce flot sera affecté sans engendrer d'impact négatif sur les performances des autres files d'attente.

Inconvénients FQ ☹

- ❑ FQ ne peut appliquer exactement l'équité que dans le cas où tous les paquets ont la même taille. Or ce scénario dispose d'une très faible probabilité de se produire réellement ; c'est pourquoi l'équité n'est pas utilisée par l'algorithme dans son sens le plus absolu. La compétitivité de flots temps-réel sur un réseau qui implémente Fair Queuing ne pourra en aucun cas garantir le délai ni le débit qui sont les principaux besoins (respectifs) de ces flots.

FQ, Fair Queuing (FQ)

33

❑ *Exemple Max-min Fair Share*

Calculez l'allocation max-min fair pour les flots A, B, C, D et E, quand leurs demandes sont 2, 3, 4, 4, 5, et la taille des ressources est 15

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

WFQ, Weighted Fair Queuing (WFQ)

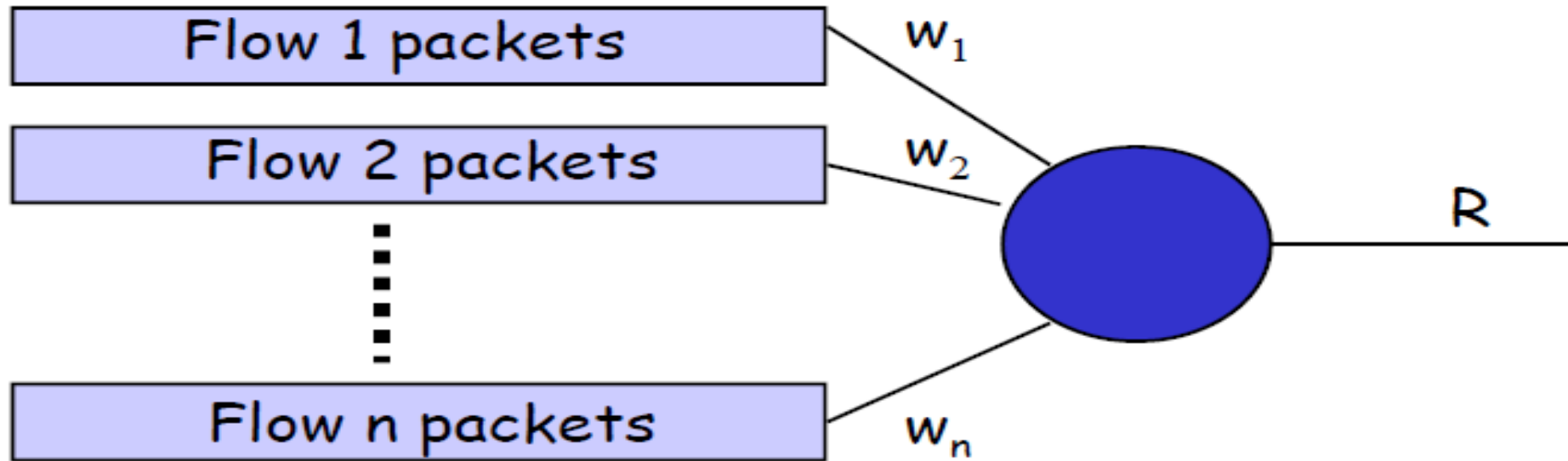
WFQ, Weighted Fair Queuing (WFQ)

35

- ❑ Dérivé de l'ordonnancement Fair Queuing, le Weighted Fair Queuing a été implémenté de manière à déterminer 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 », d'où la notion de « weighted ». Cette notion a été proposée pour pouvoir allouer différentes parts de bande-passante aux flots, selon leur besoin.
- ❑ WFQ est implémenté de manière à pouvoir supporter différentes tailles de paquets et ainsi assurer un traitement malgré l'hétérogénéité des flots.

WFQ, Weighted Fair Queuing (WFQ)

36



- Chaque flux i reçoit un poids w_i
- Le débit attribué pour chaque flux i est

$r_i = R * w_i / (w_1 + w_2 + ... + w_n)$ Avec R est la débit de la bande passante bps

WFQ, Weighted Fair Queuing (WFQ)

37

Avantages WFQ 😊

- ❑ Chaque flux obtiendra donc une file d'attente dynamique entre lesquelles les priorités seront respectées.
- ❑ Le WFQ est conçu pour limiter les efforts de configuration manuelle et pour gérer automatiquement les conditions de trafic du réseau.
- ❑ WFQ protège chaque classe de service en garantissant un niveau minimum de bande passante du port de sortie indépendamment du comportement des autres classes de service.

Inconvénients WFQ ☹

- ❑ WFQ implémente un algorithme complexe qui nécessite la maintenance d'une quantité significative d'état de classe par service et d'analyses itératives d'état à chaque arrivée et départ de paquet.
- ❑ La complexité des calculs a un impact sur l'évolutivité de WFQ lors de la tentative de prise en charge d'un grand nombre de classes de service sur des interfaces haut débit.

□ *Exemple WFQ*

Calculez l'allocation WFQ pour les flots A, B, C, D et E, quand leurs demandes sont 2, 3, 4, 4, 5, leurs poids sont 2.5, 1, 0.5, 1, 2 et la taille des ressources est 15

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

CQ, Custom Queuing

CQ, Custom Queuing

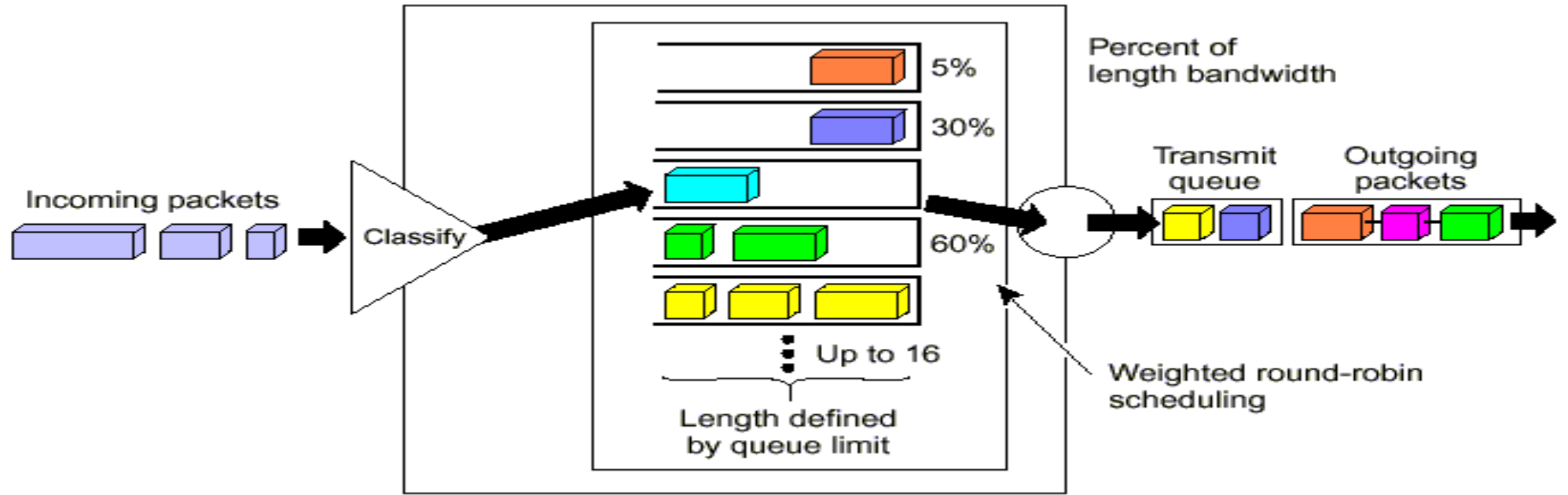
40

- Le Custom Queuing, appelé aussi Class Based Queuing (CBQ) est une variante intéressante des files d'attente avec privilèges.
- Plutôt que d'utiliser des files pondérées et d'y attribuer un certain degré de priorité, le Custom Queuing réserve un certain pourcentage de la bande passante disponible dans une interface en fonction de chaque type de trafic.
- Le critère n'est cette fois pas le degré de priorité mais la demande en bande passante.

CQ, Custom Queueing

41

Exemple CQ



Chez Cisco, le CQ utilise 16 files d'attente pour une interface de sortie. Elles sont ordonnancées statiquement avec des priorités relatives calculées sur une demande de garantie de bande passante. L'administrateur va en effet configurer la taille de chaque file et l'ordonnanceur servira chaque file à tour de rôle (round robin) en transmettant au maximum autant d'octets que configuré dans la file en cours. Il en résultera une affectation de la bande passante par file, par exemple :

file 1 = 5% de la bande passante , file 2 = 20% de la bande passante, file 3 = 30 % de la bande passante

CQ, Custom Queuing

42

Avantages CQ ☺

- ❑ L'intérêt majeur du Custom Queuing est la prise en compte de priorités en fonction d'une demande de bande passante garantie, de plus ce mode de fonctionnement reste simple et rapide.

Inconvénients CQ

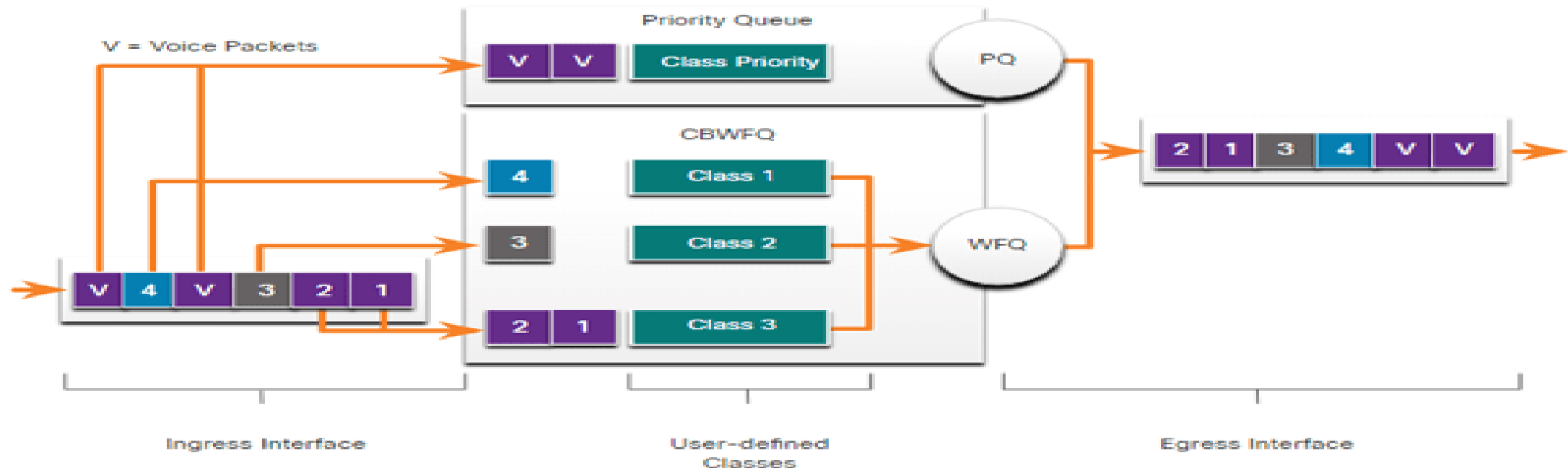
- ❑ Avec 16 files, comment faire pour bien répartir les trafics dans chaque file et comment répartir la bande passante ?
- ❑ La configuration nécessite de spécifier la taille en octet des files
- ❑ Ce mode de fonctionnement est coûteux au niveau du système d'exploitation du routeur

Chapitre II: mécanismes de la QoS

ChapII-Section A :Ordonnancement et gestion des files d'attente

CBWFQ + LLQ, Class-Based Weighted Fair Queuing + Low Latency Queuing

CBWFQ + LLQ, Class-Based Weighted Fair Queuing + Low Latency Queuing



- ❑ L'ordonnancement des paquets dans les files est dit "fair" + "weighted" + "class-based" + "low latenced"

CBWFQ + LLQ, Class-Based Weighted Fair Queuing + Low Latency Queuing

- ❑ *fair et weighted* sont les mêmes que précédemment, class-based : l'administrateur a la possibilité d'ordonnancer lui-même en entrée un trafic donné dans une file "utilisateur". Puis par configuration, il pourra spécifier la bande passante allouée à chaque file utilisateur et ainsi assurer une bande passante minimum garantie à ces flux utilisateurs,
- ❑ *low latenced* : la présence d'une nouvelle file dite "low latency" propose une priorité absolue pour les trafics qui y seront orientés. Ce qui veut dire que les paquets en entrée qui seront aiguillés dans cette le se verront attribuer une priorité absolue et seront donc servis les premiers en sortie sans mise en concurrence avec les autres les (c'est la le "ultra prioritaire").