

Exercices Dirigés

Unité d'enseignement RSX101

Réseaux et protocoles pour l'Internet -IP(2) QoS-

2019-2020

Ce support a été élaboré par l'équipe enseignante "Réseaux et protocoles", auteur principal M. Gressier Soudan.

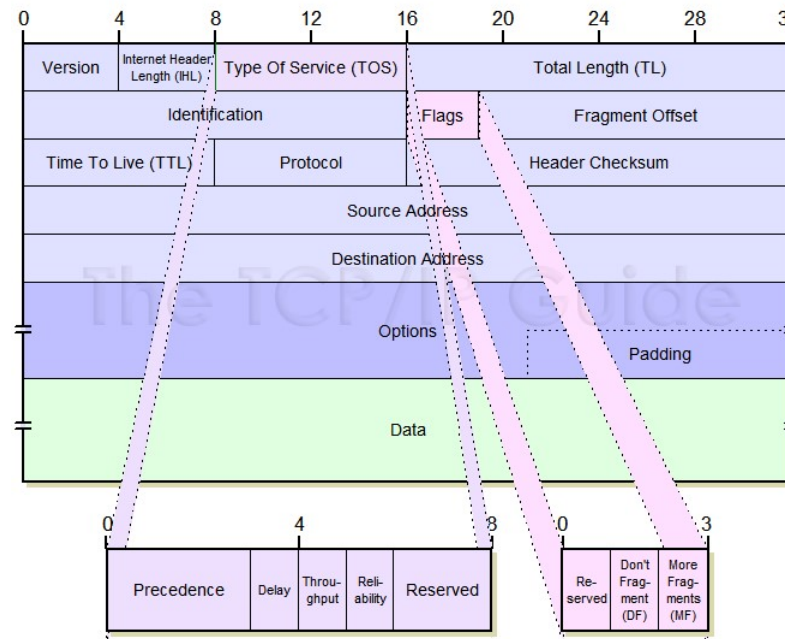
ED4•Couche Réseau : QoS & IP

Exercice 1 : Mise en œuvre de la QoS dans un routeur

On donne la structure d'une trame Ethernet :

Ethernet II				
Destination MAC 6 Bytes	Source MAC 6 Bytes	Type 2 Bytes	Data 46 – 1500 Bytes	Frame Check Sequence 4 Bytes

On donne la structure d'un datagramme IP dont son entête en détail, consulté le 23 décembre 2013, Source http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm :



Question 1 : On s'intéresse à une trace Wireshark qui est extraite d'un trafic Teams capturé en mars 2020. On s'attache en particulier à la trame 25 donnée ci-après. (question facultative, normalement c'est facile pour vous après UTC505)

not(ip.dsfield.dscp == 0)						
No.	Time	Source	Destination	Protocol	Length	Info
25	9.723396	192.168.1.10	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
> Frame 1962: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{...}						
✓ Ethernet II, Src: Sagemcom_61:2a:00 (a4:08:f5:61:2a:00), Dst: IPv4mcast_01 (01:00:5e:00:00:01)						
> Destination: IPv4mcast_01 (01:00:5e:00:00:01)						
> Source: Sagemcom_61:2a:00 (a4:08:f5:61:2a:00)						
Type: IPv4 (0x0800)						
Padding: 00000000000000000000000000000000						
✓ Internet Protocol Version 4, Src: 192.168.1.1, Dst: 224.0.0.1						
0100 = Version: 4						
.... 0110 = Header Length: 24 bytes (6)						
✓ Differentiated Services Field: 0x80 (DSCP: CS4, ECN: Not-ECT)						
1000 00.. = Differentiated Services Codepoint: Class Selector 4 (32)						
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)						
Total Length: 32						
Identification: 0x0000 (0)						
> Flags: 0x4000, Don't fragment						
Fragment offset: 0						
Time to live: 1						
Protocol: IGMP (2)						
Header checksum: 0x42ad [validation disabled]						
[Header checksum status: Unverified]						
Source: 192.168.1.1						
Destination: 224.0.0.1						
> Options: (4 bytes), Router Alert						
> Internet Group Management Protocol						
0000	01 00 5e 00 00 01 a4 08 f5 61 2a 00 08 00 46 80	..^.....a*...F.				
0010	00 20 00 00 40 00 01 02 42 ad c0 a8 01 01 e0 00	. .@... B.....				
0020	00 01 94 04 00 00 11 64 ee 9b 00 00 00 00 00 00d				
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00				

Voilà son extraction en hexadécimal :



RSX101

```

0000      01 00 5E 00 00 01 A4 08      F5 61 5d 1f 08 00 46 80
0010      00 20 00 00 40 00 01 02      42 AD C0 A8 01 01 E0 00
0020      00 01 94 04 00 00 11 64      EE 9B 00 00 00 00 00 00
0030      00 00 00 00 00 00 00 00      00 00 00 00

```

Attention la colonne la plus à gauche numérote les lignes et cette numérotation est hexadécimale.

- Délimiter l'entête du datagramme IP dans la capture en hexadécimal ci-dessous. Ne pas hésiter à utiliser des couleurs différentes pour que votre réponse soit facile à lire.

Retrouver les champs suivants dans la trace hexadécimale ci-dessus :

- Quelle est l'adresse Ethernet destination en **hexadécimal** ?
- Quelle est l'adresse Ethernet source en **hexadécimal** ?
- Quelle est la version du protocole IP en **décimal** ?
- Quelle est la longueur de l'entête IP en **décimal** ?
- Quelle sont les adresses IP source et destination en **hexadécimal** ?
- Quelle est la longueur totale du datagramme en **décimal** ?

Correction :

```

0000      01 00 5E 00 00 01 A4 08      F5 61 5d 1f 08 00 46 80
0010      00 20 00 00 40 00 01 02      42 AD C0 A8 01 01 E0 00
0020      00 01 94 04 00 00 11 64      EE 9B 00 00 00 00 00 00
0030      00 00 00 00 00 00 00 00      00 00 00 00

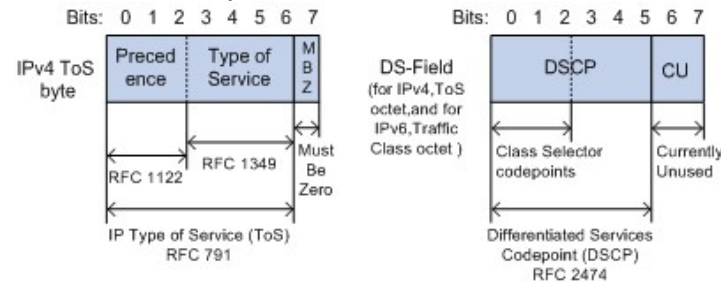
```

Attention les lignes contiennent 16 octets.

- Entête du datagramme IP dans la capture en hexadécimal ci-dessous.
- Quelle est l'adresse Ethernet destination en **hexadécimal** ? 01 00 5E 00 00 01
- Quelle est l'adresse Ethernet source en **hexadécimal** ? A4 08 F5 61 5d 1f
- Quelle est la version du protocole IP en **décimal** ? 4
- Quelle est la longueur de l'entête IP en **décimal** ? 6 $6 \times 4 = 24$ octets
- Quelle sont les adresses IP source et destination en **hexadécimal** ? C0 A8 01 01 soit 192.168.1.1 et E0 00 00 01 soit 224.0.0.1
- Quelle est la longueur totale du datagramme en **décimal** ? 00 20 soit $2 \times 16 = 32$ octets

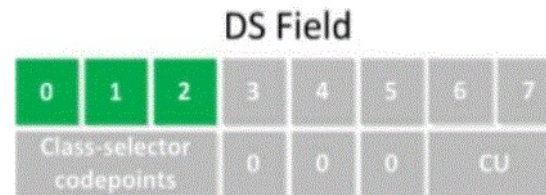


Question 2 : On se concentre sur la partie QoS (Quality of Service, Qualité de Service en français). La QoS est associée au 2^{ème} octet de l'entête IP. Dans la trame 25, il vaut 80 en hexadécimal ou 1000 0000 en binaire. Ce champ peut s'interpréter de deux façons différentes d'après la figure ci-après suivant les RFC qui servent à l'interpréter :



source https://techhub.hpe.com/eginfolib/networking/docs/switches/10500/cg/5200-1877_acl-qos_cg/content/470792642.htm (consultée le 18/08/2020)

<https://www.slideshare.net/NetworkersHome1/ip-precedence-dscp-values-quality-of-service-qos> (consultée le 18/08/2020) donne une interprétation plus fine de DSCP :



Pour chaque valeur de "CScodepoint" correspond un comportement particulier du routeur appelé PHB (Per Hop Behavior). On appelle ce PHB, une classe de service. La même source web donne un tableau des correspondances entre champ CS (Class-Selector) et Précédence du champ ToS :

Class selector name	DSCP value	IP precedence Value	IP precedence name
Default/CS0	000000	000	Routine
CS1	001000	001	Priority
CS2	010000	101	Immediate
CS3	011000	011	Flash
CS4	100000	100	Flash Override
CS5	101000	101	Critic/Critical
CS6	110000	110	Internetwork Control
CS7	111000	111	Network Control

Tableau 1. Définition des Class Selector.

As you can see, CS1 is the same as "priority" and CS4 is the same as "**flash override**". We can use this for compatibility between the "old" TOS byte and the "new" DS field.

The default PHB and these class-selector PHBs are both described in RFC 2474 from 1998.

Le champ DSCP évolue encore avec la RFC2597 qui définit 4 classes de PHB particuliers AF (Assured Forwarding PHB Group) selon la même source :



Avec les bits 3, 4 et 5, une probabilité d'élimination en cas de saturation de file de messages du routeur est spécifiée. Ce champ est défini dans le tableau ci-après (toujours d'après la même source web) :

Possible values that we can use:

Drop	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
	AF11	AF21	AF31	AF41
Medium	001100	010100	011100	100100
	AF12	AF22	AF32	AF42
High	001110	010110	011110	100110
	AF13	AF23	AF33	AF43

Class 4 has the highest priority. For example, any packet from class 4 will always get better treatment than a packet from class 3.

Some vendors prefer to use decimal values instead of AF11, AF32, etc. A quick way to convert the AF value to a decimal value is by using the $8x + 2y$ formula where X = class and Y = drop probability. For example, AF31 in decimal is $8 \times 3 + 2 \times 1 = 26$.

Tableau 2. Définition des PHB AF.

Est-ce que la classe CS4 indiquée par Wireshark, et qui appartient au tableau 1, se retrouve dans le tableau 2 ? Si oui sous quel AF ? Si non à votre avis pourquoi et dans quelle colonne AF devrait/pourrait-elle être placée ? Est-ce que c'est la même chose pour les autres classes CS du tableau 1 ?

Correction :

CS4 n'est pas dans le tableau 2. CS4 c'est 100000 d'après le tableau 1 et la trace. CS4 est donc plus proche de la colonne "Class 4" ou AF4x. Plus précisément, on pourrait faire l'hypothèse que la probabilité d'élimination des datagrammes de type CS4 est moins forte que AF41 si on essaie de rentrer dans la logique du tableau 2.

C'est la même chose pour les autres classes CS du tableau 1.

On pourrait réécrire le tableau 2 de cette façon :

Drop	Class 1	Class 2	Class 3	Class 4
No Drop ?	001 000	010 000	011 000	100 000
	CS1	CS2	CS3	CS4
Low	001 010	010 010	011 010	100 010
	AF11	AF21	AF31	AF41
Medium	001 100	010 100	011 100	100 100
	AF12	AF22	AF32	AF42
High	001 110	010 110	011 110	100 110
	AF13	AF23	AF33	AF43

Plus difficile de positionner CS5, CS6 et CS7 dans ce tableau. CS0 c'est best effort ou default.

Globalement, c'est un peu ce qu'on retrouve dans le tableau donné un peu plus loin dans le texte de l'exercice.

Question 3 : On s'intéresse à la relation entre classes et types d'applications pour configurer les files de messages d'un routeur. **(3 points)**
 À l'ensemble des classes données dans les deux tableaux ci-dessus, il faut en ajouter deux autres. Best Effort (BE) ou Default (DF) donne la marque 000 000 en binaire. L'autre classe, c'est Expedited Forwarding (EF) qui vaut 101110 en binaire ou 46¹ en décimal.

¹ Passer de 101110 à 46 en décimal n'est pas immédiat. Pour trouver 46, il faut ajouter '00' devant 101110. En effet, 00101110 donne 2E en hexadécimal, qui fait bien 46 en décimal.

Peer-to-peer media-sharing applications (Kazaa, Morpheus, Grokster, Napster, iMesh, and so on), gaming applications (Doom, Quake, Unreal Tournament, and so on), and any entertainment video applications.

QoS Values Calculator v3

CoS = Class of Service

DSCP = Differentiated Services Code Point

ToS = Type of Service

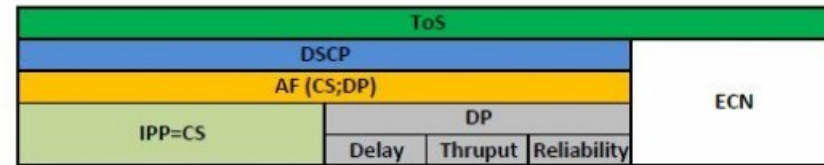
AF = Assured Forwarding

IPP = IP Precedence

CS = Class Selector

DP = Drop Probability

ECN = Explicit Congestion Notification



	8th bit	7th bit	6th bit	5th bit	4th bit	3rd bit	2nd bit	1st bit
ToS	128	64	32	16	8	4	2	1
DSCP	32	16	8	4	2	1		
CoS=IPP	4	2	1					

Application	CoS=IPP	AF	DSCP	ToS	ToS HEX	DP	8th bit	7th bit	6th bit	5th bit	4th bit	3rd bit	2nd bit	1st bit
Best Effort	0	0	0	0	0		0	0	0	0	0	0	0	0
Scavenger	1	CS1	8	32	20		0	0	1	0	0	0	0	0
Bulk Data	1	AF11	10	40	28	Low	0	0	1	0	1	0	0	0
	1	AF12	12	48	30	Medium	0	0	1	1	0	0	0	0
	1	AF13	14	56	38	High	0	0	1	1	1	0	0	0
Network Mgmt.	2	CS2	16	64	40		0	1	0	0	0	0	0	0
Transaction Data	2	AF21	18	72	48	Low	0	1	0	0	1	0	0	0
	2	AF22	20	80	50	Medium	0	1	0	1	0	0	0	0
	2	AF23	22	88	58	High	0	1	0	1	1	0	0	0
Call Signaling	3	CS3	24	96	60		0	1	1	0	0	0	0	0
Mission-Critical	3	AF31	26	104	68	Low	0	1	1	0	1	0	0	0
Streaming Video	3	AF32	28	112	70	Medium	0	1	1	1	0	0	0	0
	3	AF33	30	120	78	High	0	1	1	1	1	0	0	0
	4	CS4	32	128	80		1	0	0	0	0	0	0	0
Interactive Video	4	AF41	34	136	88	Low	1	0	0	0	1	0	0	0
	4	AF42	36	144	90	Medium	1	0	0	1	0	0	0	0
	4	AF43	38	152	98	High	1	0	0	1	1	0	0	0
Voice	5	CS5	40	160	A0		1	0	1	0	0	0	0	0
	5	EF	46	184	B8		1	0	1	1	1	0	0	0
Routing	6	CS6	48	192	C0		1	1	0	0	0	0	0	0
	7	CS7	56	224	E0		1	1	1	0	0	0	0	0

Version:

v2 - ToS in HEX added

v3 - Applications description and DSCP 0 added

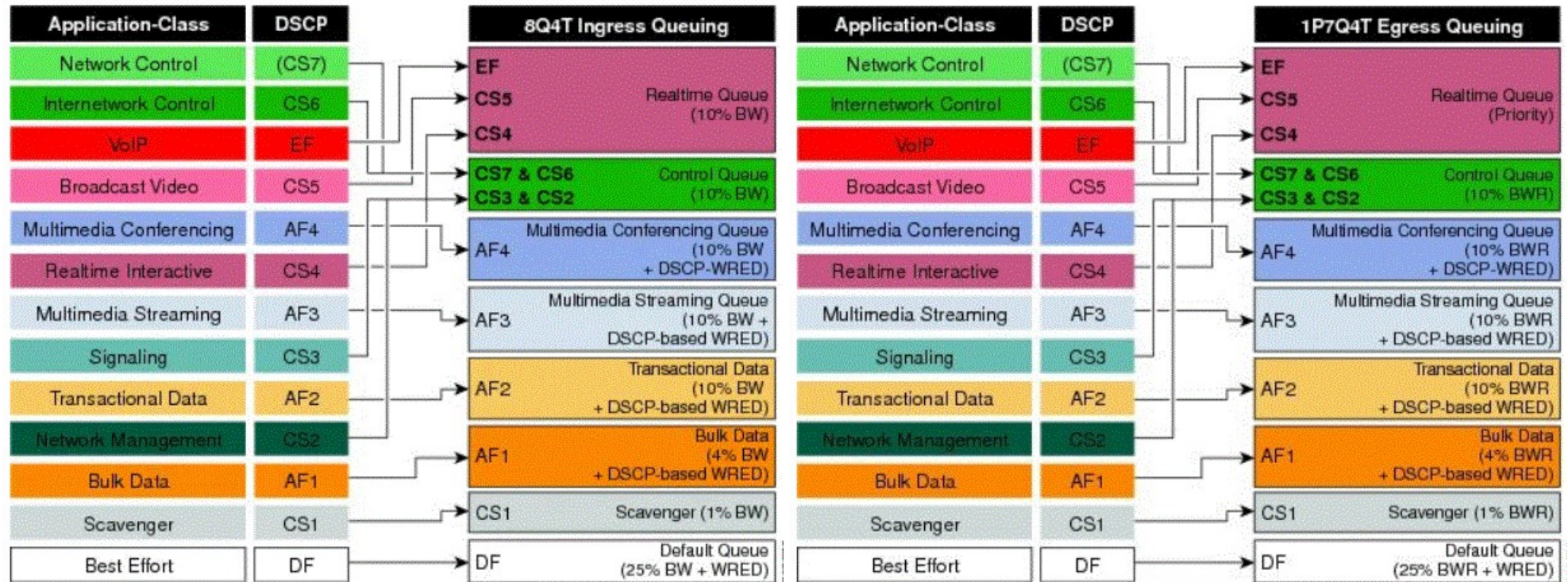


Relation entre les différents marquages dans le champ ToS/DSCP, et aussi dans celui de la couche liaison IEEE802.1Q/p (CoS)

source : <http://www.netcontractor.pl/blog/?tag=dscp> (consultée le 18/08/2020)



Les files de messages associées aux interfaces réseau d'un routeur, dans le cadre d'une gestion de QoS, ont deux fonctions : la mise en œuvre de l'ordonnancement des messages pour le respect du SLA client de bout en bout, et, la politique d'évitement de congestion des files. On associe un modèle de gestion de file, par classification, pour les datagrammes qui sont entrants²(ingress) et un pour les datagrammes qui sont sortants (egress) :



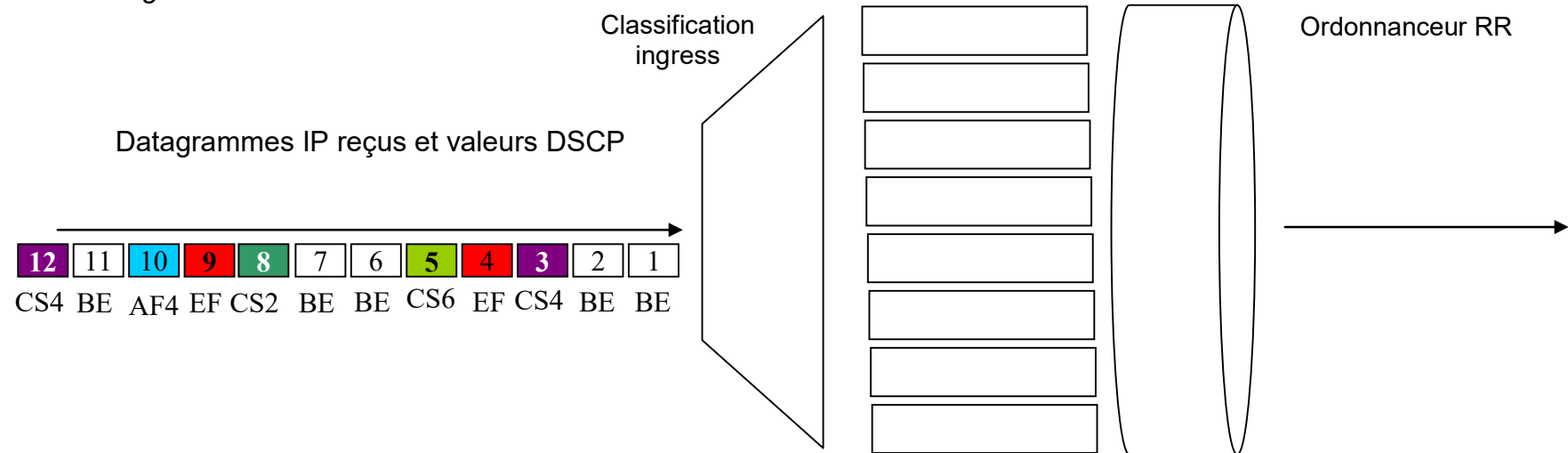
Légendes :

- BW : Bandwidth, débit cible
- WRED : Weighted Random Early Detection, un algorithme de gestion de congestion des files dans un équipement
- 8Q4T : 8Q-8 files sans gestion de priorités; 4T-4 seuils peuvent être définis par file
- 1P7Q4T : 1P-une file avec priorité ; 7Q-7 files sans gestion de priorités, 4T-4 seuils peuvent être définis par file

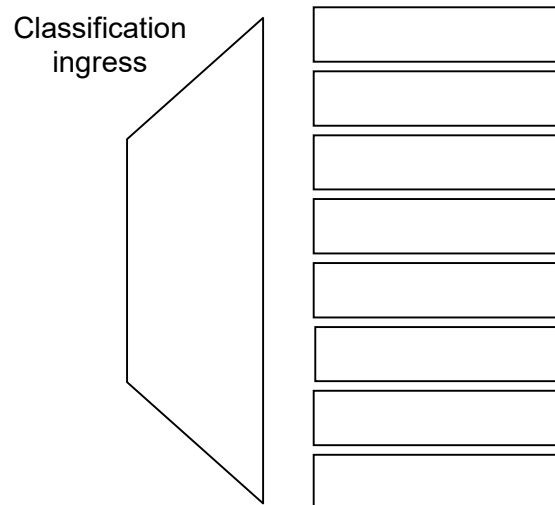
² La source des 2 figures est <https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/qoscampuscat6500sup2taag.html> (consultée le 18/08/2020)

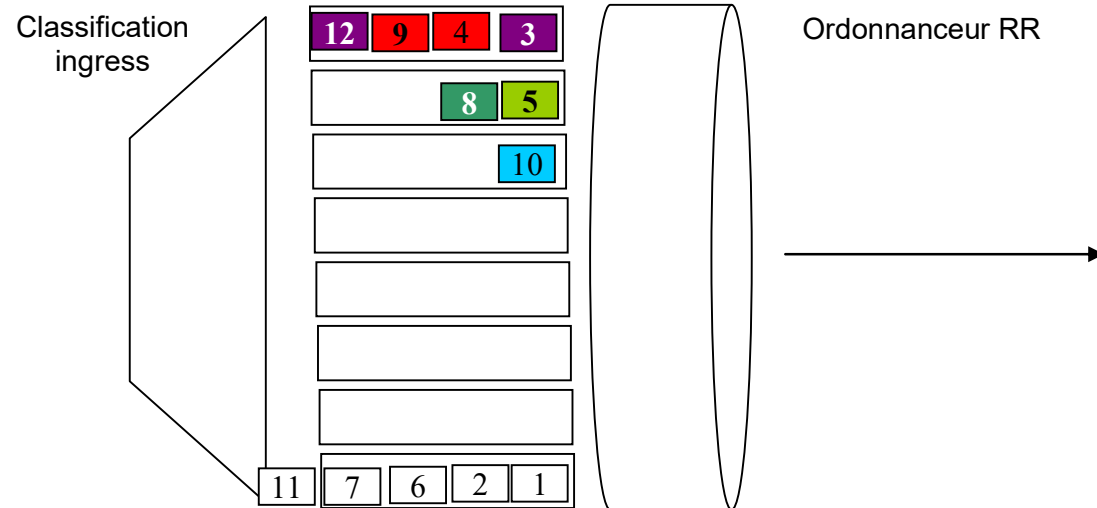
On s'intéresse à l'ordonnancement résultant sur les files d'entrées et de sortie sur un flux de datagramme traversant le routeur. Sur la file d'entrée on ordonnance avec une politique WRR (Weighted Round Robin).

Soit le Flux de datagrammes IP suivant sur la carte en entrée :

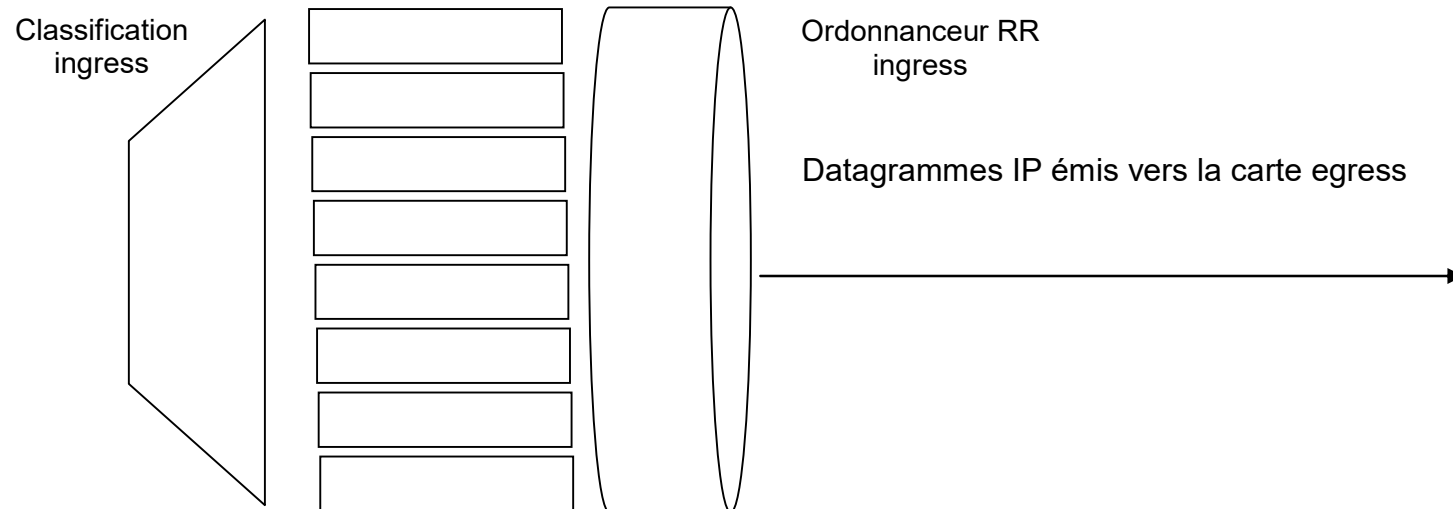


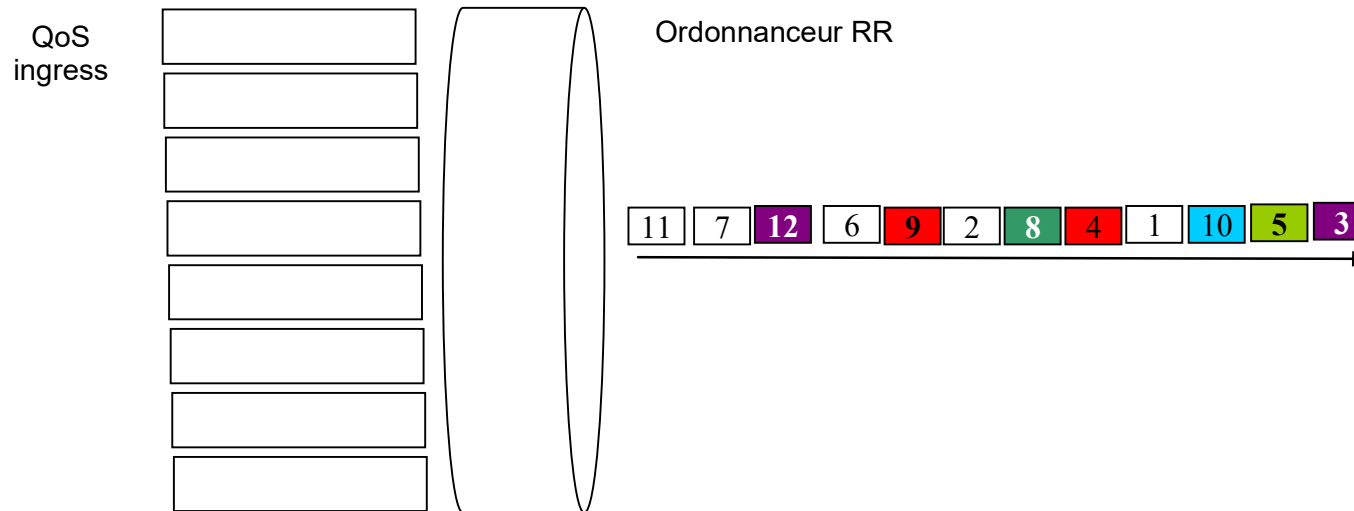
Donner la classification obtenue dans les 8 files conformément au modèle **ingress** présenté ci-dessus en complétant le schéma ci-après. (0,5point)



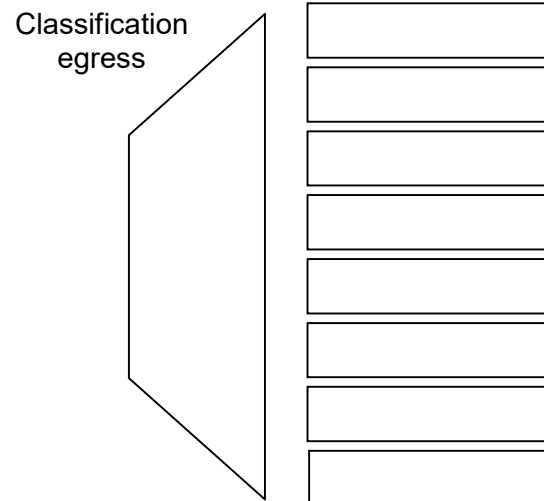
Correction :

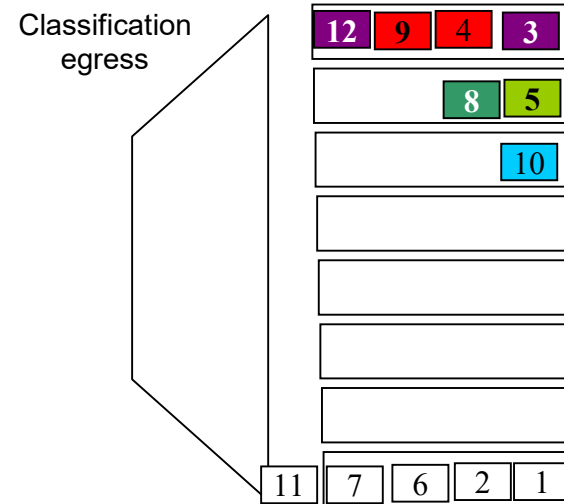
Donner le flux des datagrammes IP sortant de l'ordonnanceur RR (Round Robin) du modèle **ingress** en complétant le schéma ci-après. On supposera que l'ordonnanceur s'exécute une fois les différentes files remplies par le flux arrivant.



Correction :

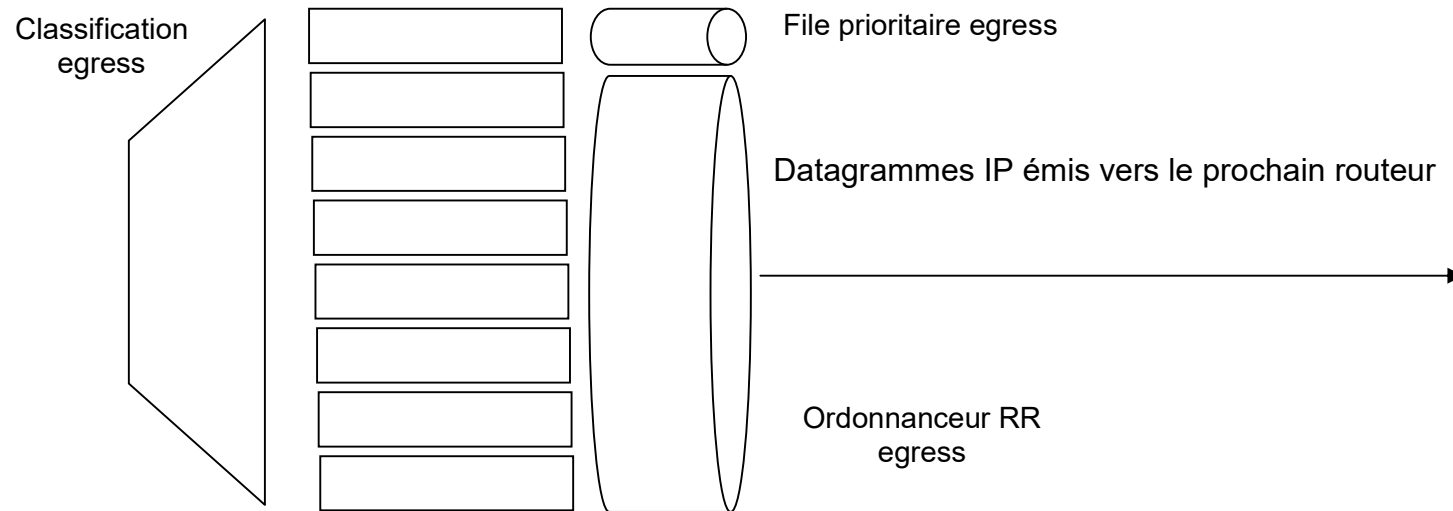
On suppose que le flux de datagrammes IP se dirige vers la carte de sortie qui met en œuvre le modèle egress. Remplir le schéma ci-après correspondant à la classification des datagrammes.

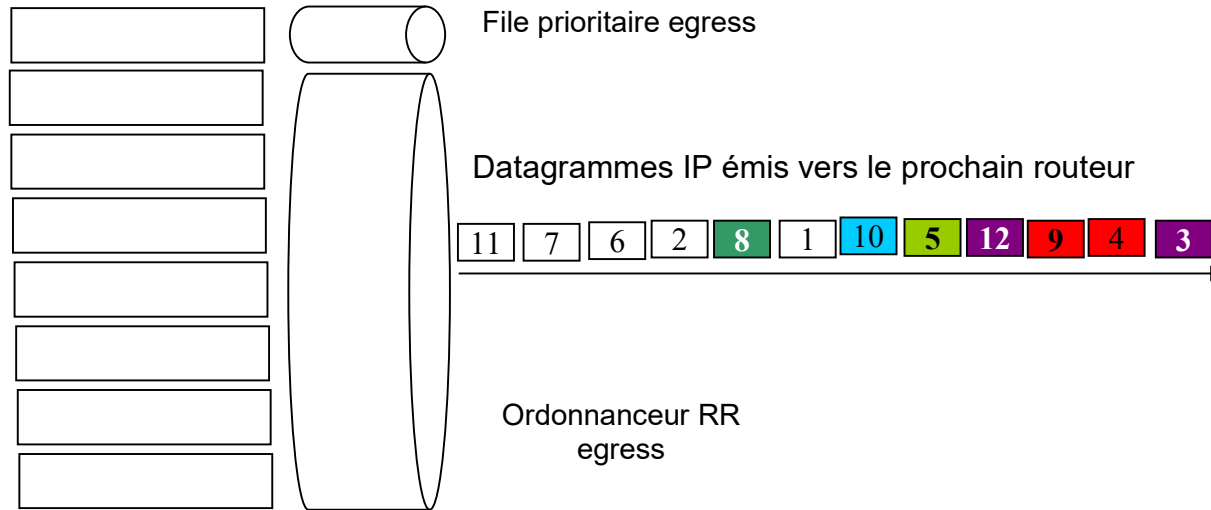


Correction :

C'est la même que la classification ingress.

Donner le flux des datagrammes IP sortant de l'ordonnancement du modèle **egress** en complétant le schéma ci-après. On supposera que l'ordonnanceur s'exécute une fois les différentes files remplies par le flux arrivant.



Correction :

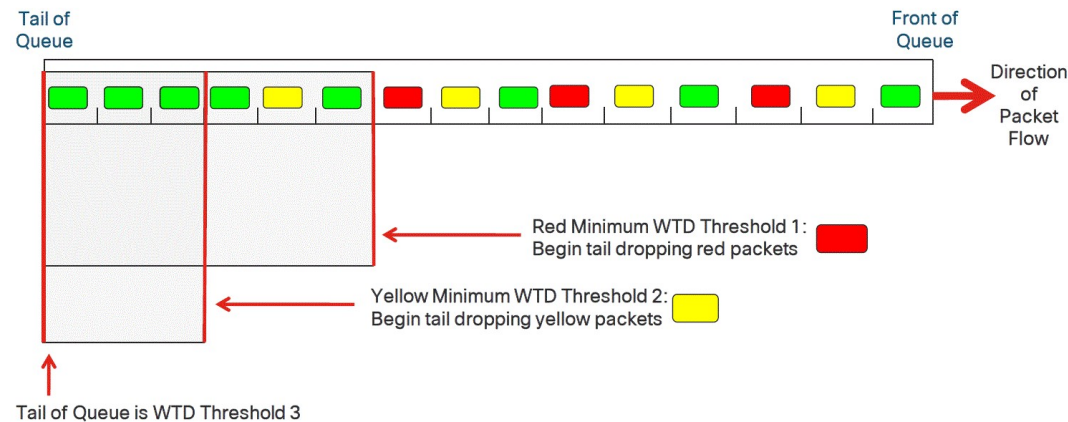
Cet ordonnancement semble plus conforme à l'acheminement de datagrammes avec des contraintes de délai fortes.

Question 4 : Comparaison de politiques de contrôle de congestion des files de messages des équipements.

Cisco propose par défaut deux politiques de gestion de la congestion : Weighted Tail Drop (WTD), et WRED. Les deux stratégies sont rappelées ci-dessous. Comparer les en donnant les avantages et les inconvénients que vous y voyez.

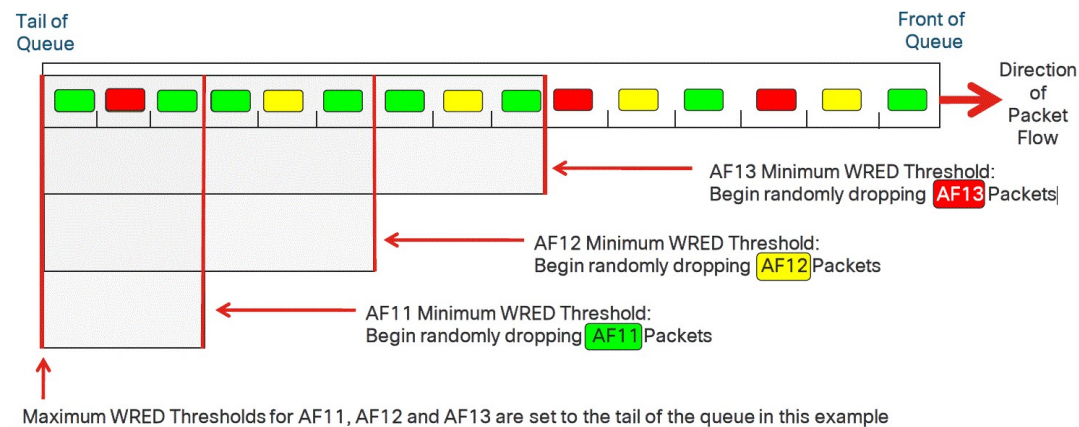
Weighted Tail Drop (WTD) Operation

3T WTD Example



Weighted Random Early Detect (WRED) Operation

3T WRED Example



Correction :

La politique Weighted Tail Drop est radicale pour la catégorie éliminée. Si ces messages contiennent des informations protocolaires susceptibles de libérer des ressources (messages en attente) comme un ACK, il va y avoir des réémissions donc de potentielles nouvelles congestions à venir.

Bien sûr, il faut avoir en tête que les messages éliminés appartiennent à des associations client-serveur différentes.

La politique WRED, dans la mesure où les messages jetés sont choisis aléatoirement dans la catégorie éliminée, est moins violente et permet de faire jouer de façon moins systématique et donc plus répartie le mécanisme d'évitement de congestion de TCP et de redémarrage ralenti (Congestion Avoidance & Slow Start). A priori, il faut appliquer cette politique de façon plus anticipée dans les files de messages.

Exercice 2 : Voix et QoS

On utilise un codage G711, plus connu sous le nom : MIC (Modulation par Impulsion et Codage) ou encore PCM (Pulse Code Modulation). On rappelle que la voix numérisée correspond à l'envoi d'un échantillon de 8 bits toutes les 125 micro-secondes, en complet isochronisme. Soit 8000 échantillons par seconde ou encore 64K³b/s au niveau du codeur de son (DSP, Digital Signal Processor). C'est donc un trafic temps réel qui est très contraint.

CISCO fournit le tableau indicatif suivant qui relie codage et débit sur Internet avec quelques caractéristiques sur le flux voix sur IP correspondant :

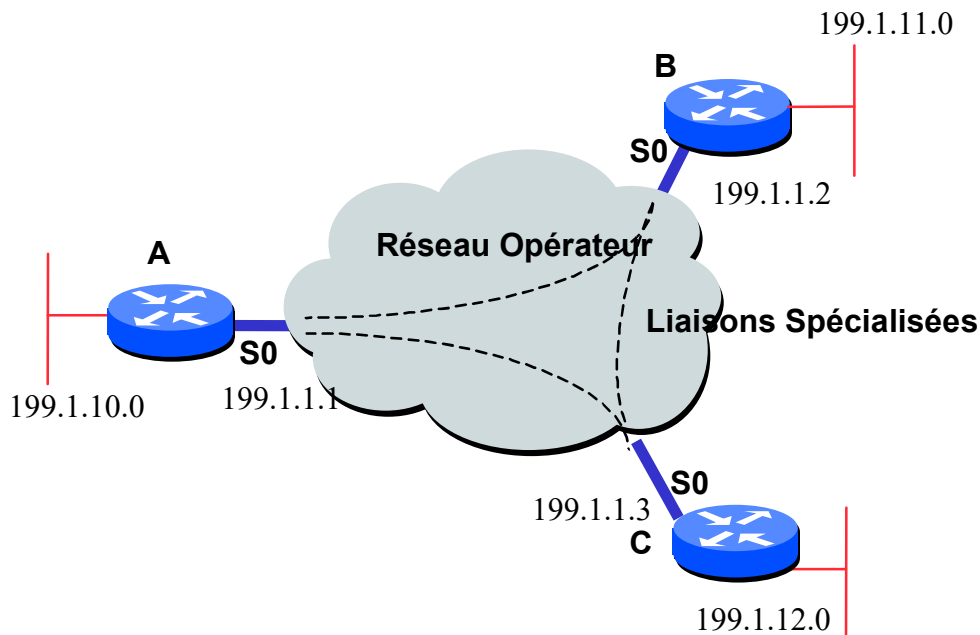
Informations Codec				Calculs de la bande passante					
Codec et débit binaire (Kbps)	Taille d'échantillon Codec (octets)	Intervalle d'échantillon Codec (ms)	Note moyenne d'opinion	Taille de la charge utile vocale (octets)	Taille de la charge utile vocale (ms)	Packets par seconde (PPS)	Bande passante MP ou FRF.12 (Kbps)	Bande passante avec cRTP MP ou FRF.12 (Kbps)	Bande passante Ethernet (Kbps)
G.711 (64 Kbps)	80 octets	10 ms	4.1	160 octets	20 ms	50	82,8 Kbps	67,6 Kbps	87,2 Kbps
G.729 (8 Kbps)	10 octets	10 ms	3.92	20 octets	20 ms	50	26,8 Kbps	11,6 Kbps	31,2 Kbps
G.723.1 (6,3 Kbps)	24 octets	30 ms	3.9	24 octets	30 ms	33.3	18,9 Kbps	8,8 Kbps	21,9 Kbps
G.723.1 (5,3 Kbps)	20 octets	30 ms	3.8	20 octets	30 ms	33.3	17,9 Kbps	7,7 Kbps	20,8 Kbps
G.726 (32 Kbps)	20 octets	5 ms	3.85	80 octets	20 ms	50	50,8 Kbps	35,6 Kbps	55,2 Kbps
G.726 (24 Kbps)	15 octets	5 ms			20 ms	50	42,8 Kbps	27,6 Kbps	47,2 Kbps
G.728 (16 Kbps)	10 octets	5 ms	3.61	60 octets	30 ms	33.3	28,5 Kbps	18,4 Kbps	31,5 Kbps
G722_64k (64 Kbps)	80 octets	10 ms	4.13	160 octets	20 ms	50	82,8 Kbps	67,6 Kbps	87,2 Kbps
ilbc_mode_20 (15.2Kbps)	38 octets	20 ms	NA	38 octets	20 ms	50	34.0 Kbps	18,8 Kbps	38.4 Kbps
ilbc_mode_30 (13.33Kbps)	50 octets	30 ms	NA	50 octets	30 ms	33.3	25,867 Kbps	15.73 Kbps	28,8 Kbps

source ajoutée après la séance d'exercice :

https://www.cisco.com/c/fr_ca/support/docs/voice/voice-quality/7934-bwidth-consume.html (consultée le 19/10/2020 20h30)

³ attention, ici le Kb correspond bien à 1000 bits et pas à 1024 bits.

Après une étude de dimensionnement, on évalue à en moyenne deux communications simultanées entre 2 sites pendant les jours ouvrables.



Les liaisons spécialisées supportées par la technologie de l'opérateur doivent être vues comme des tuyaux de transmission dédiés. On pourrait imaginer que la technologie MPLS est mise en œuvre pour supporter les liens entre sites.

Question 1

L'entreprise cherche à se doter d'une architecture tout IP. Quels seraient les paramètres d'une communication Voix, on vous demande de renseigner les attributs suivants :

- Latency (latence)
- DelayVariation (gigue)
- ServiceType (type de service)
- MaxSduSize (taille maximum d'un datagramme)
- MinimumPolicedSize (taille minimum d'un datagramme)

On rappelle que la transmission de la voix utilise les protocoles IP, UDP et RTP (un protocole de description de trafic temps réel) qui ont des entêtes respectivement de 20 octets, 8 octets, 12 octets. A titre indicatif, on considèrera qu'un paquet RTP regroupe 80 échantillons de voix codés sur 8 bits (1 octet).

On propose d'associer une classe de trafic temps réel : EXPEDITED FORWARDING pour le trafic de type voix.

On ne considère pas d'autres types d'informations pour ce problème.

Correction :

On va remplir les champs d'une demande SLS (Service Level Specification) :

- **Latency** : 150 ms (c'est le délai max pour qu'une conversation téléphonique soit confortable)

- **DelayVariation** : 0 (pas de gigue admise sur du trafic voix en G711, ou encore MIC)
- **ServiceType** : EXPEDITED FORWARDING (EF)
- **MaxSduSize**⁴ : 120 octets qui se décomposent en 80 (charge utile) + 20 (IP) + 8 (UDP) + 12 (RTP), les SduSize se calculent en datagramme ni au-dessus ni au-dessous
- **MinimumPolicedSize** : 120 octets, les datagrammes sont de longueur fixe

Le nuage Réseau à circuits virtuels est remplacé par un domaine Internet à qualité de service de type DiffServ géré par un opérateur réseau. Chaque routeur de l'entreprise est raccordé à un routeur d'entrée de l'opérateur.

Question 2

Quel avantage y a-t-il à placer un mécanisme de lissage (shaping) dans le routeur de sortie de l'entreprise ? Même question à propos du routeur d'entrée de l'opérateur ?

Correction :

Le lissage en sortie de l'entreprise permet de rendre le trafic conforme au contrat de qualité de service souscrit par l'entreprise et évite les litiges.

Le lissage en entrée d'un opérateur permet de contrôler le trafic entrant et d'éviter d'accepter un flux non-conforme qui risquerait de consommer des ressources dans le réseau de l'opérateur voire de le congestionner.

Les deux peuvent se cumuler sans contradiction.

Question 3

Donner la règle associée à la transmission d'un trafic WEB du site A vers le site C dans la table de vérification de conformité du trafic dans le routeur d'entrée "en face de A" chez l'opérateur. On indique que le trafic WEB sert à la consultation des données du système d'information de l'entreprise, il est donc important mais n'est pas prioritaire en cas de surcharge. L'adresse IP du serveur Web est 199.1.12.60, il est sur le port 80.

On rappelle la suite d'informations contenue dans une entrée de cette table :

IP Source	Port Source	protocole	IP Destination	Port Destinât	Classe Priorité	PHB	Ovrflow PHB
-----------	-------------	-----------	----------------	---------------	-----------------	-----	-------------

⁴ La SDU est le datagramme IP. Les paramètres de trafic sont comptabilisés au niveau de la couche IP... ni au-dessus ni en-dessous. C'est pour ça qu'on comptabilise les entêtes des couches au-dessus d'IP.

Correction :

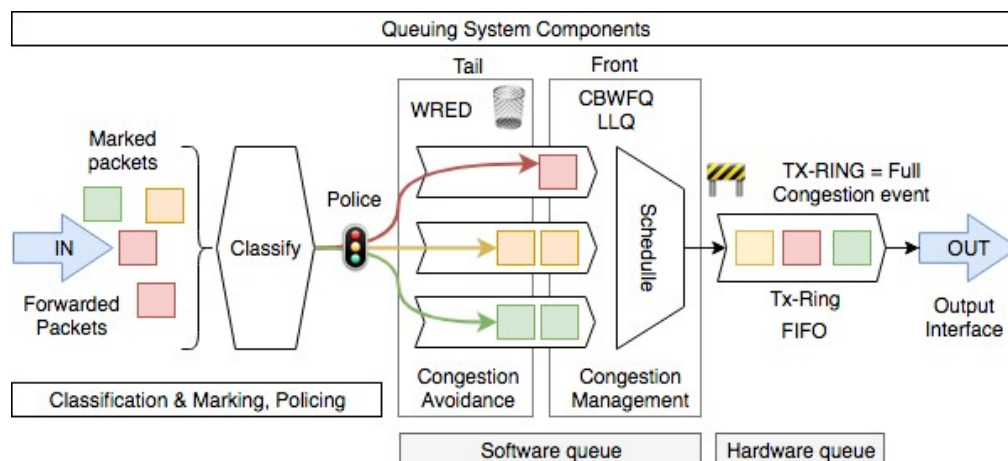
IP Source	Port Sour	prot o	IP Destination	Port Dest	Classe Prior	PHB	Ovrflow PHB
*	*	TCP	199.1.12.60	80	3	AF-argent	AF-bronze
199.1.12.60	80	TCP	*	*	3	AF-argent	AF-bronze

La première ligne correspond au trafic navigateur qui va vers le serveur Web. La seconde correspond au trafic retour. Les * indiquent toute machine.

Pour Ovrflow PHB on aurait pu mettre aussi BE.

Exercice 3 : Réseau IP et QoS

Pour vous remettre en contexte, on rappelle une vue plus globale d'un routeur. L'architecture interne d'un routeur combine différents mécanismes : classification, politiques de gestion de files de datagrammes, gestion de congestion, gestion mémoire des organes d'émission :



source : <https://learningnetwork.cisco.com/thread/122029> consultée le 4/11/2019

- TX-Ring est un buffer de mémoire géré de façon circulaire (cf cours de structure de données), d'où le nom "ring", associé au processeur d'interface et qui est géré en FIFO pour la sortie des datagrammes. On est dans la partie très basse d'un routeur, il peut être géré en hardware, le paquet ne bénéficie plus de traitements de QoS
- LLQ : Low Latency Queuing qui combine PQ (Priority Queuing) et CBWFQ (Class Based Weighted Fair Queuing), la figure venant de l'équipementier CISCO, il semble que ces deux politiques soient souvent mises en oeuvre.
- WRED : Weighted Random Early Detection, politique d'élimination de paquet quand une surcharge est détectée.

Question 1

On s'intéresse à la Qualité de Service, QoS (QoS, Quality of Service en anglais).

Rappeler les paramètres de performance qui interviennent dans la gestion de la QoS et qu'on associe souvent à un contrat appelé Service Level Agreement (SLA) en anglais. En particulier, qu'est-ce que la gigue ?

Correction :

Les paramètres de QoS orientés performance sont :

- Débit (throughput en anglais)

- Latence (latency en anglais)
- Gigue (jitter en anglais)
- Taux de perte (loss rate en anglais)
- Taux d'erreur suivant certains fournisseurs et certains contexte (error rate en anglais)

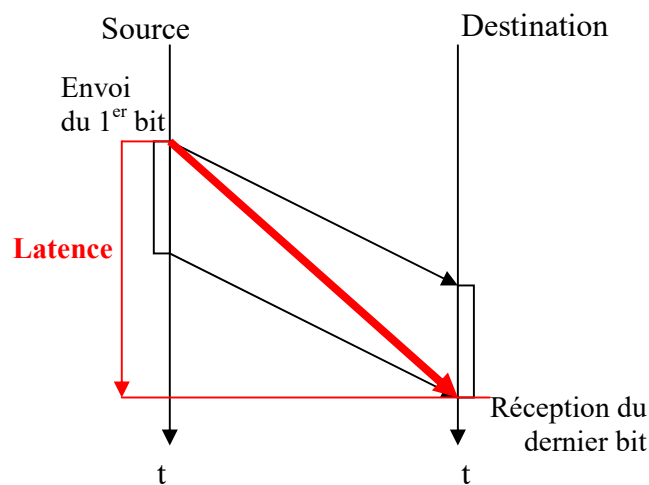
On ajoute aussi le taux de disponibilité, mais ce n'est pas un paramètre de performance.

Sur des réseaux sans fil on ajoute parfois :

- Taux de déconnexion

La gigue correspond à la variation de la latence (Delay Variation serait une meilleure terminologie, plus précise, elle est donnée dans la RFC3393). Pour lever toute ambiguïté, les définitions sont issues de RFCs.

Latence (RFC2679):

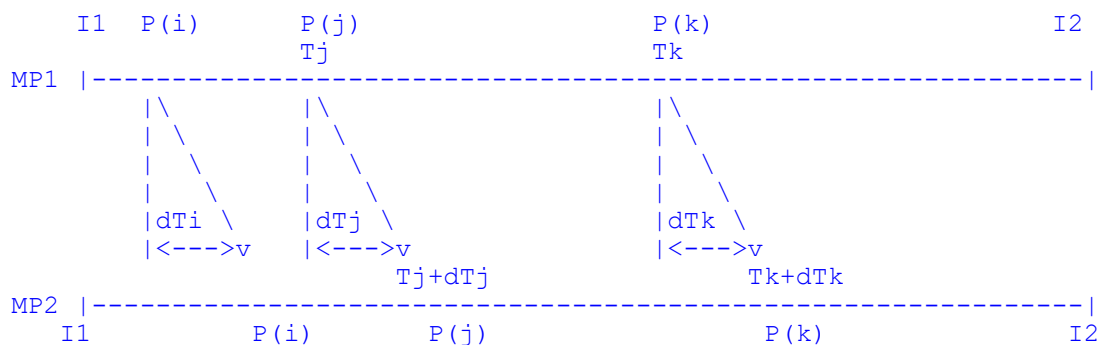


Il reste difficile de mesurer cette quantité, car elle nécessite une horloge de référence. Souvent on fabrique cette horloge de référence à partir d'une synchronisation d'horloge : par GPS, par serveur de temps NTP (Network Time Protocol). La RFC2679 donne une méthode de calcul de cette latence.

Issu de la RFC 3393 p6 :

Soient $I1$ et $I2$ deux instants de mesure, $MP1$ et $MP2$ deux points de mesure. On suppose que les paquets $P(j)$ et $P(k)$ sont sélectionnés.

ddT , c'est la valeur d'IP Packet Delay Variation ($ipdv$) par rapport à la latence qu'on cherche. Son calcul s'effectue par rapport à un flux de datagrammes (paquets) où on sélectionne une paire de paquets pour faire le calcul d' $ipdv$. $ipdv$ est la différence entre les latences de deux paquets sélectionnés.



$$\begin{aligned}\text{Latence}_j &= T_j + dT_j - T_j \\ \text{Latence}_k &= T_k + dT_k - T_k \\ ddT &= |dT_k - dT_j|\end{aligned}$$

En effet, certains trafics sont insensibles à la variation de la latence, comme le transfert de fichiers, au contraire de la voix au format MIC (Multiplexage Impulsion et Codage) encore appelé G711 et qui correspond à un échantillonnage de 8000 Hertz soit un débit de 64 kb/s.

Le principe de la métrologie passive est d'étudier un trafic après sa capture en un point précis d'un réseau (par exemple un routeur, on a vu en cours que l'équipement routeur est un élément clef de l'architecture Internet). Les mesures ainsi obtenues sont quantitatives et peuvent être vues de deux façons différentes : sous la forme de flux et sous la forme d'entité réseau. On peut préciser la nature des mesures pour chacune de ces formes, globalement on trouve :

- Au niveau des flux, on les caractérise comme dans un routeur à QoS, et on y ajoute quelques informations de volume.
- Au niveau de la partie réseau, on caractérise par :
 - L'utilisation des liens (nombre de bits ou de datagrammes par seconde). Un routeur a plusieurs interfaces ;
 - Le nombre de datagrammes éliminés ;
 - Le taux d'erreurs ;
 - Des mesures au niveau des équipements comme la charge du processeur, l'utilisation de la mémoire, la température.

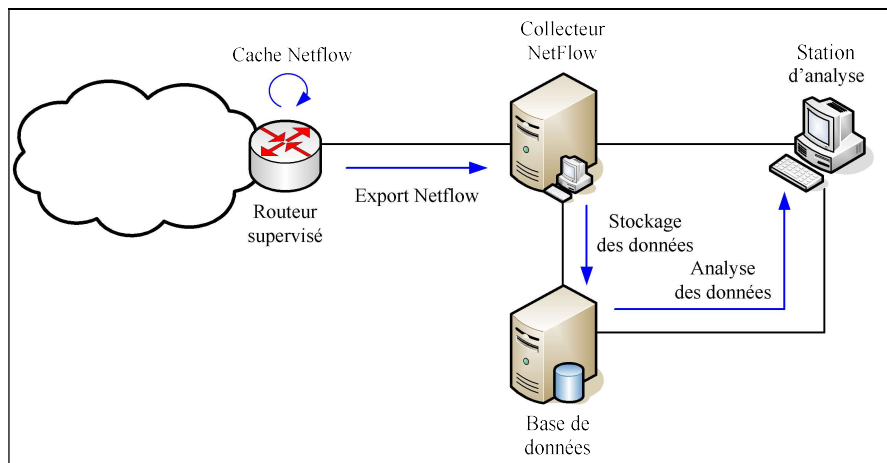
Question 2 : Gestion de la QoS : caractérisation de flux

Le protocole Netflow⁵ est un protocole de métrologie passive développé par l'entreprise CISCO. Il est composé de trois éléments :

- L'équipement supervisé. Il a pour rôle de créer le cache (appelé « Cache Netflow »), d'effectuer d'éventuelles agrégations et d'exporter les enregistrements vers le serveur Netflow.
- Le serveur Netflow encore appelé collecteur. Il est localisé sur une machine tierce qui collecte les enregistrements envoyés par les équipements supervisés. Selon sa configuration, le collecteur va réaliser une première phase de traitement en effectuant des filtres et des agrégations puis procéder au stockage des enregistrements obtenus dans une base de données de type SQL.
- L'application d'analyse. Son objectif est de pouvoir présenter les données collectées sous une forme synthétique à l'exploitant : rapports de synthèse, affichage de statistiques et génération d'alertes.

Ci-après un schéma synthétique illustre cette architecture.

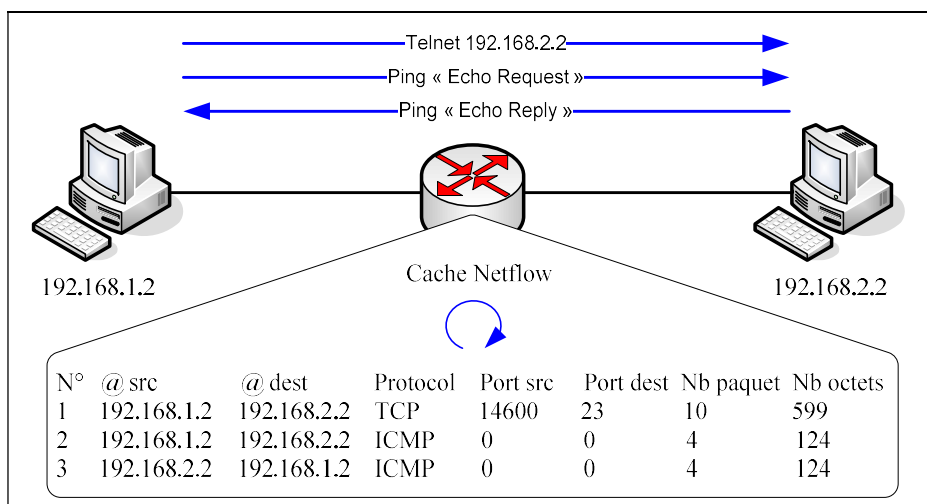
⁵ Le comportement et les caractéristiques de Netflow ne sont pas rigoureusement utilisés dans l'exercice. Elles ont plutôt inspiré l'exercice. Etre plus rigoureux par rapport à Netflow aurait complexifié le problème au-delà du nécessaire. J'espère que les étudiants excuseront les libertés prises et que l'entreprise CISCO aussi.



Les routeurs sur lesquels est activé Netflow, scrutent le trafic des flux en transit qui les traversent. Netflow caractérise les flux par plusieurs informations :

- L'adresse IP source ;
- L'adresse IP de destination ;
- Le port UDP ou TCP source ;
- Le port UDP ou TCP de destination ;
- Le protocole (TCP, UDP, ICMP) ;
- Le champ type de service (TOS : Type Of Service) ;
- L'index de l'interface d'entrée/sortie vers un lien réseau de l'équipement.

Ci-dessous, un exemple des flux capturés par un cache Netflow :



2.1 Comment identifie-t-on un flux dans un routeur à QoS ? Remarque : sur un hôte, un flux est identifié de la même façon que sur un routeur, mais c'est moins stratégique pour un hôte.

Correction :

Un flux est identifié par une adresse IP source, un numéro de port source, une adresse IP destination, un numéro de port destination, et un protocole transporté (tcp ou udp généralement).

2.2 Un flux est une vision microscopique des datagrammes qui traversent un routeur. Une classe de service est une vision macroscopique du trafic qui traverse un routeur. Comment identifie-t-on une classe de trafic dans un routeur à QoS ?

Correction :

On a les mêmes informations que ci-dessus en entier ou en partie, pour les champs non renseignés on utilise un joker (*). On y ajoute la valeur du champ Type of Service (ToS)-Differentiated Service Code Point (DSCP) qui indique une classe de trafic et les privilèges de routage associés.

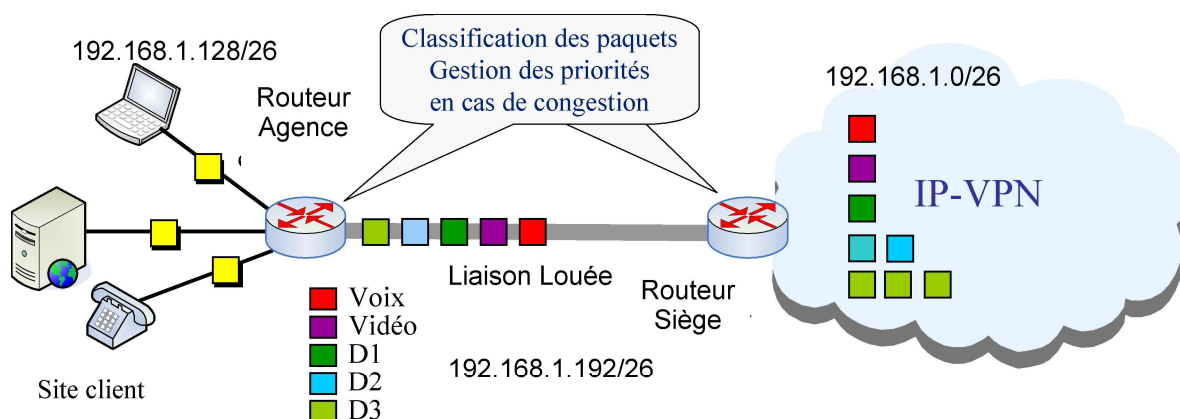
2.3 Le cache Netflow requiert de la puissance processeur supplémentaire pour pouvoir effectuer son travail. Quelle recommandation feriez vous avant d'installer un cache Netflow sur un routeur ?

Correction :

Il faut prévoir qu'une partie de la puissance du routeur puisse être consacrée à l'exécution du protocole de métrologie en lui-même. Cette puissance ne pourra pas en effet être consacrée à l'exercice des tâches de routage. La recommandation serait donc d'installer Netflow sur un routeur qui n'est pas au maximum de son utilisation. Compte tenu des informations fournies, il n'est pas possible de recommander un taux d'occupation maximum du processeur.

Question 3 : DiffServ

Le logiciel NetFlow sert à étudier les flux d'une entreprise. L'analyse de ces flux permet, après une analyse détaillée de paramétrer des routeurs à QoS suivant le modèle DiffServ. L'étude révèle un schéma et une typologie des flux résumés dans la figure ci-dessous :

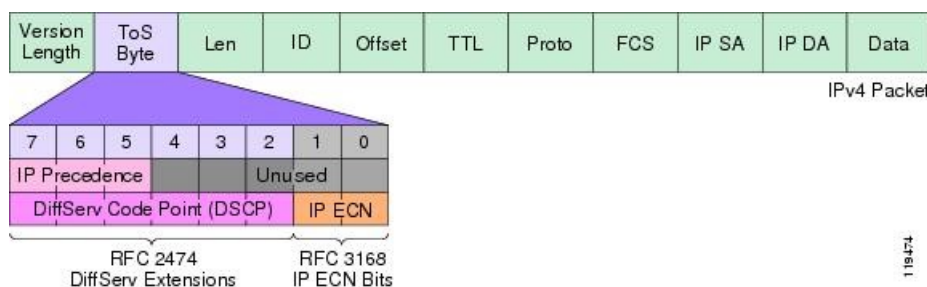


- La **voix** correspond à 15% du trafic global sur le lien observé. C'est du trafic **temps réel**.
- La **vidéo** correspond à 15% du trafic sur ce même lien. C'est du trafic **temps réel** aussi.
- **D2** correspond à du trafic de type "données" orienté **ERP** (Enterprise Resource Planning), soit 30%. C'est un trafic vital pour l'entreprise mais sans contraintes temps réel.
- **D3** correspond à du trafic de type "données" comme du **transfert de fichiers** (ftp essentiellement). Ce trafic correspond à 10% du trafic de l'entreprise.

- **D1** correspond à du **trafic web** pour un niveau de 30%, il n'est pas prioritaire.

On se sert du champ DSCP (Differentiated Service Code Point), ex champ TOS (Type Of Service) dans le datagramme pour identifier son appartenance aux différentes classes de trafic. Le champ DSCP porte un code. Les codes sont repartis entre 4 classes de service AF (Assured Forwarding ou Expédition Assurée), une classe EF (Expedited Forwarding ou Expédition Accélérée) et une classe BE (Best Effort).

Le champ DSCP (IPv6) et sa mise en correspondance avec une partie du champ TOS (IPv4) pour une utilisation homogène sont rappelés ci-dessous :



source : https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/QoSIntro.html consultée le 04/11/2019

L'entreprise après avoir fini l'étude de son architecture, configure la table de gestion des classes de service d'un de ses routeurs de la façon suivante :

	Politiqu	IP Source	Port Src	prot	IP Dest	Port Dest	Class	PHB	Overflow PHB
1	Voix	192.168.1.0/26	16384 – 32768	17	192.168.1.0/26	16384 – 32768	1	EF	AF-or
2	Vidéo	192.168.1.128/26	970-974	17	192.168.1.0/26	970-974	2	EF	BestEff
3	D1	*	*	6	192.168.1.0/24	80	5	BE	Drop
4	D1	192.168.1.0/24	80	6	*	*	5	BE	Drop
5	D2	*	4000-6000	6	192.168.1.60	*	4	AF-or	AF-argent
6	D2	192.168.1.60	*	6	*	4000-6000	4	AF-or	AF-argent
7	D3	192.168.1.61	20-25	6	*	*	3	AF-arg	AF-bronze
8	D3	*	*	6	192.168.1.61	20-25	3	AF-arg	AF-bronze

Quelques clefs pour comprendre la table :

- Les préfixes :
 - EF, Expedited Forwarding,
 - AF, Assured Forwarding,
 - BE, Best Effort,
 - Drop, le paquet est éliminé
- Les classes de service, du plus prioritaire (moins susceptible d'être éliminé en cas de congestion) au moins prioritaire : EF (temps réel) > AF-or > AF-argent > AF-bronze > BE (classe de base, on fait au mieux).
- Les classes de services sont paramétrées par trafic unidirectionnel, à moins qu'une spécification particulière s'applique à chacun des sens de transfert d'information comme dans la première ou la deuxième ligne de la tableau de propagation (Forward Information Base).
- Quand le trafic PHB (per hop behavior) passe en surcharge, il est déclassé tel que décrit dans la colonne OverflowPHB (dépassement de capacité)
- Le protocole 17 correspond à udp, et le protocole 6 correspond à tcp
- "*" veut dire tout numéro de port ou toute adresse IP

Le routeur de l'entreprise reçoit un datagramme avec les informations suivantes :

- source 192.168.1.35,
- port source 20000
- destination 192.168.1.133,
- port destination 18000,
- protocole transporté 17

Quelle classe de trafic lui est-elle appliquée ?

Que se passe-t-il en cas de surcharge, quelle classe lui est-elle appliquée dans ce cas ?

Correction :

L'adresse source 192.168.1.35, et le port source 20000 correspondent à la ligne 1 de la table des politiques de QoS appliquées par le routeur. L'adresse destination 192.168.1.133, et le port destination 18000, correspondent aussi. Donc c'est bien la ligne 1 qui s'applique. C'est donc un trafic de type VoIP.

En mode normal, ce trafic bénéficie de la classe de service la plus favorable : ExpForw et en cas de surcharge, la classe de service est dégradée à AF-or.

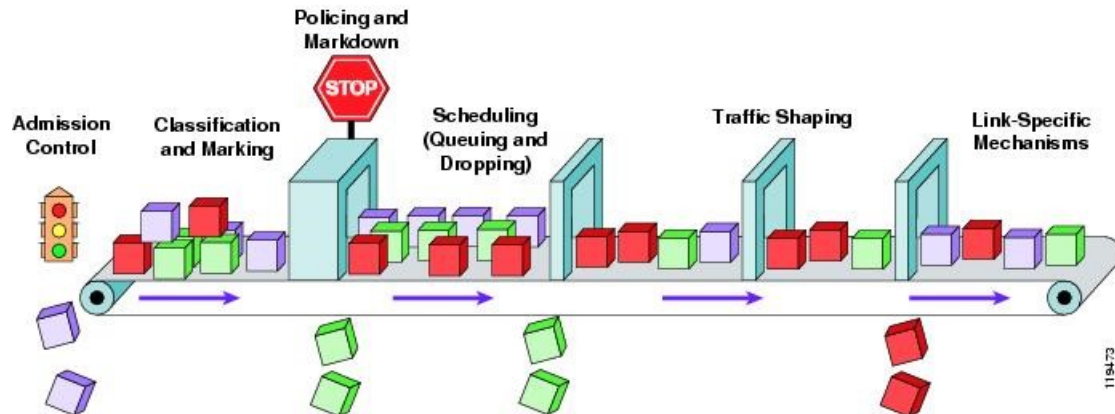
Que se serait-il passé si le port destination avait été 15000 au lieu de 18000 ? Le flux correspondant au datagramme n'aurait correspondu à aucune des classes spécifiées dans la table. Deux solutions alors:

- Le datagramme est évalué non-conforme, il est rejeté (logique de contrôle d'admission).
- Le datagramme est évalué non-conforme, mais il est accepté et dégradé en BestEffort.

C'est plutôt la première option qui est la bonne.

Question 5 : Ordonnancement des paquets dans les files de message

On rappelle l'enchainement des opérations pour les paquets dans un routeur.



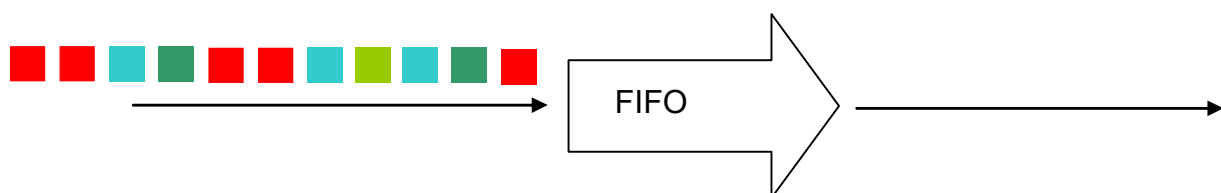
source : https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/QoSIntro.html consultée le 04/11/2019

On regarde plus en détail l'effet de politiques de gestion de files de paquet dans le routeur.

On considère une suite de datagramme, dont la couleur représente le type comme dans la figure représentant les trafics de l'entreprise. On les rappelle ici :

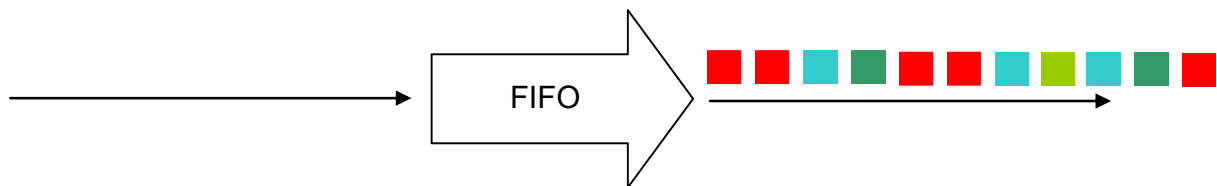
- **Rouge (VoIP)**, voix, contrainte de latence et de gigue (**vital**)
- **Violet (Vidéo conférence)**, vidéo conférence, contrainte de latence et de gigue (**essentiel** quand on le met en œuvre, mais non vital), si ça ne marche pas, on peut toujours faire une réunion téléphonique
- **Vert foncé (D1 Web)**, trafic web, **peu important**
- **Bleu (D2 ERP)**, trafic de données type ERP (entreprise Resource Planning) **vital** pour l'entreprise mais pas de contraintes temps réel strictes.
- **Vert clair (D3 FTP)**, trafic de données de **moyenne importance** type transfert de fichiers par ftp pour l'entreprise

5.1 On suppose que la file de paquets, la partie logicielle et pas la partie matérielle est gérée, est gérée en **FIFO**, comment sorte les paquets reçus ? Compléter le dessin ci-dessous.

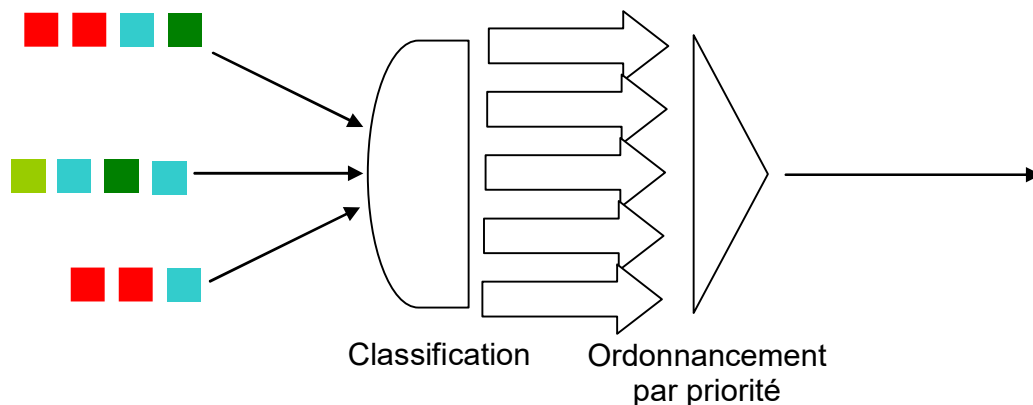


Correction :

Les paquets, avec la politique FIFO, sortent dans l'ordre où ils sont rentrés. On a donc :

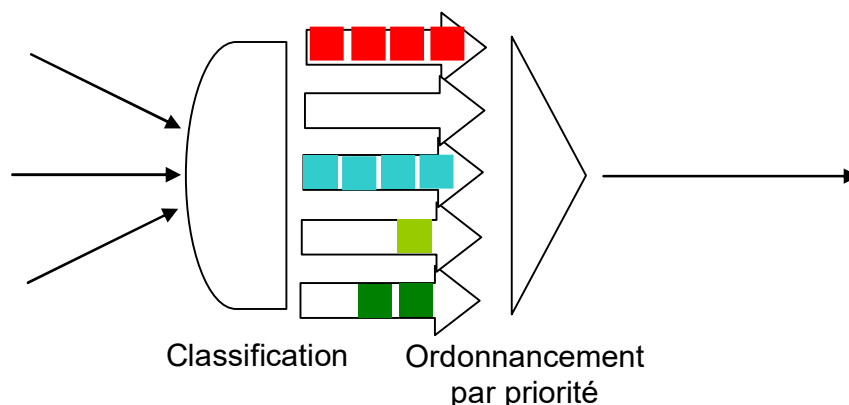


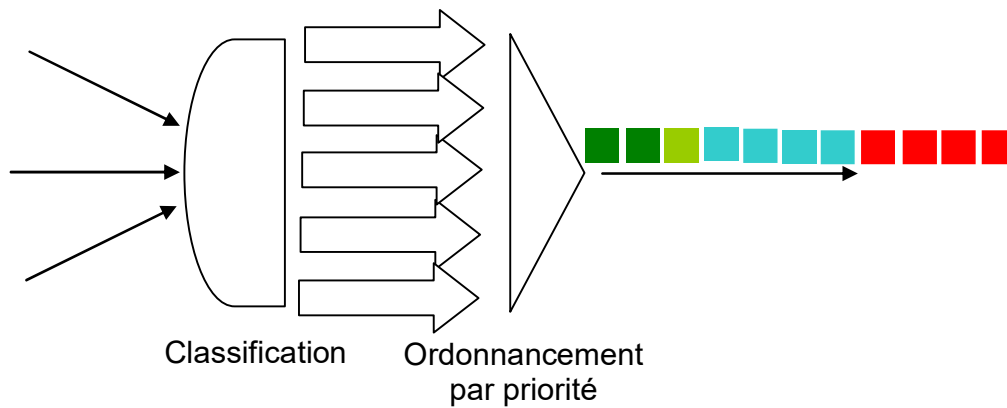
5.2. On utilise la politique par **priorité**, Priority Queuing (PQ). Reprendre la table de gestion du trafic QoS avec toutes les informations sur les priorités. Donner les paquets tels qu'ils sortent du routeur sur le lien entre l'agence et le réseau principal de l'entreprise. On fait l'hypothèse qu'il y a une classification sur le routeur.

**Correction :**

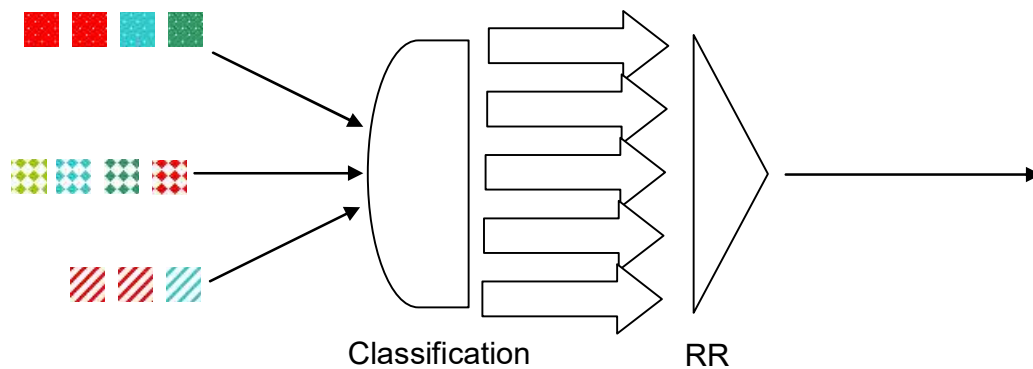
On suppose que compte tenu des temps de traitement matériel, tous les paquets se retrouvent en même temps dans le système de classification et sont donc rangés dans les files de paquets à ordonnancer pour être émis.

Le trafic VoIP est prioritaire. Il va passer en premier. Il n'y a pas de trafic vidéo. Puis D2 est prioritaire, il va sortir juste après. D3 est un trafic d'importance moyenne, il passe après D2. D1 n'est pas prioritaire, il sort en dernier.



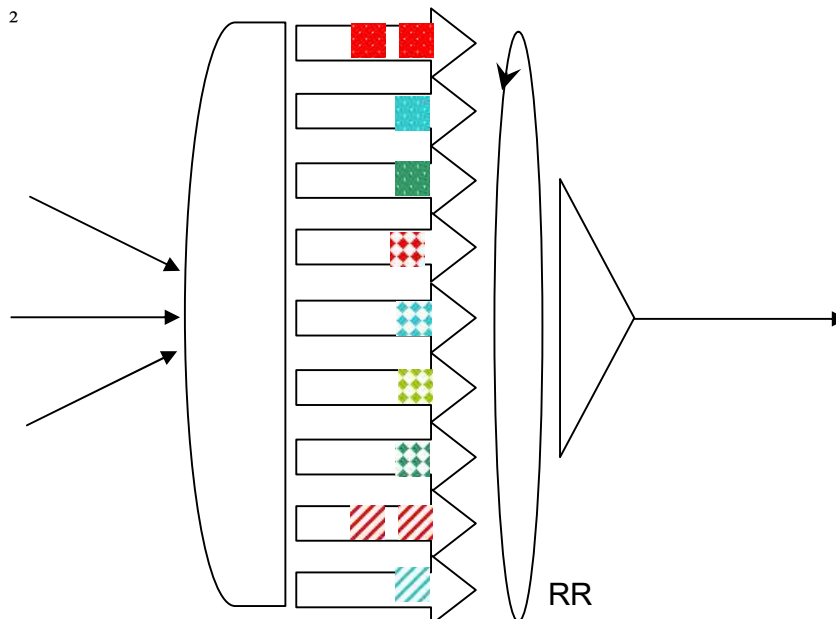


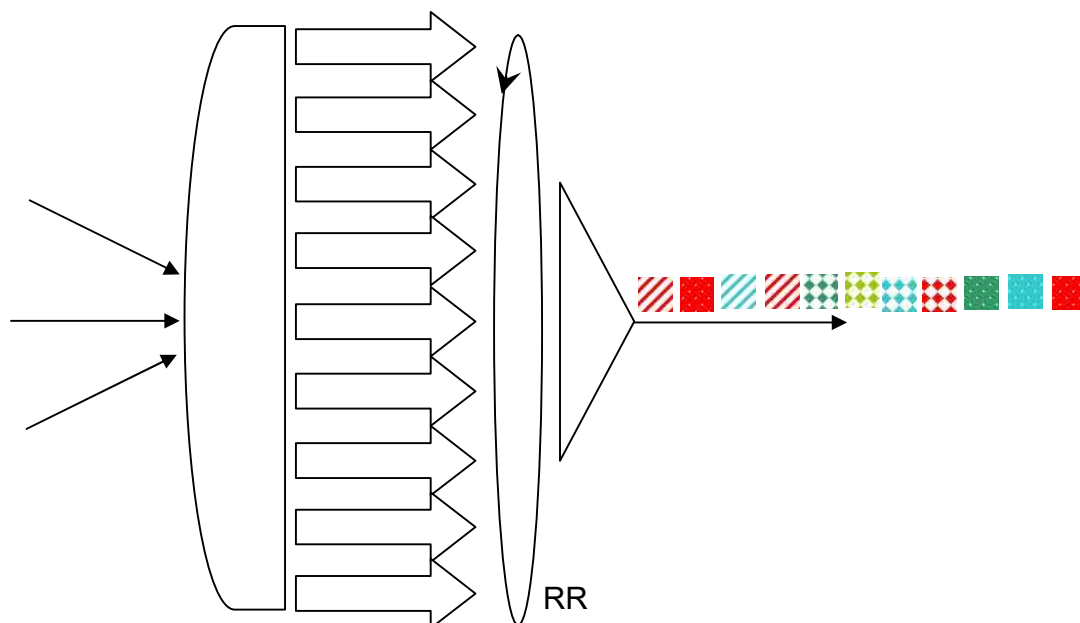
5.3 Même chose avec **Round Robin** (RR). On suppose que les flots de datagrammes venant des 3 liens sont différents car ils sont identifiés par l'adresse IP source, le port source, le protocole, l'adresse IP destination, et le port destination. RR s'applique par rapport aux flots et non par rapport aux classes.



Correction :

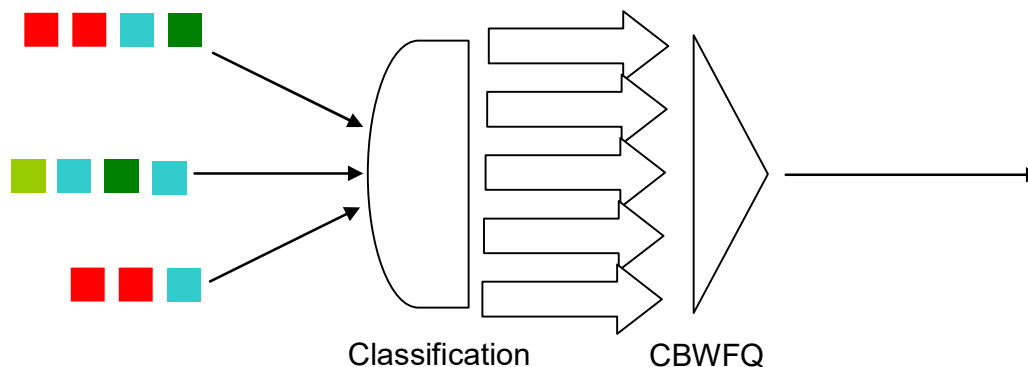
Compte tenu de l'identification de chaque flot, on a 9 flots au total qui traversent la classification. Avec RR, chaque file est servie à son tour.





Attention, on pourrait dire que Fair Queing (FQ) est identique à RR... ce n'est pas le cas, FQ travaille à l'échelle du bit de chaque paquet d'une file. Un bon synonyme pour FQ serait bit-by-bit RR. Il existe une version byte-by-byte RR semble-t-il. Dans la littérature, on trouve que FQ consomme beaucoup de ressources à cause de sa gestion bit par bit ou byte par byte. Le terme FQ apparaîtrait en 1985 d'après Wikipedia (https://en.wikipedia.org/wiki/Fair_queuing) comme une évolution de RR.

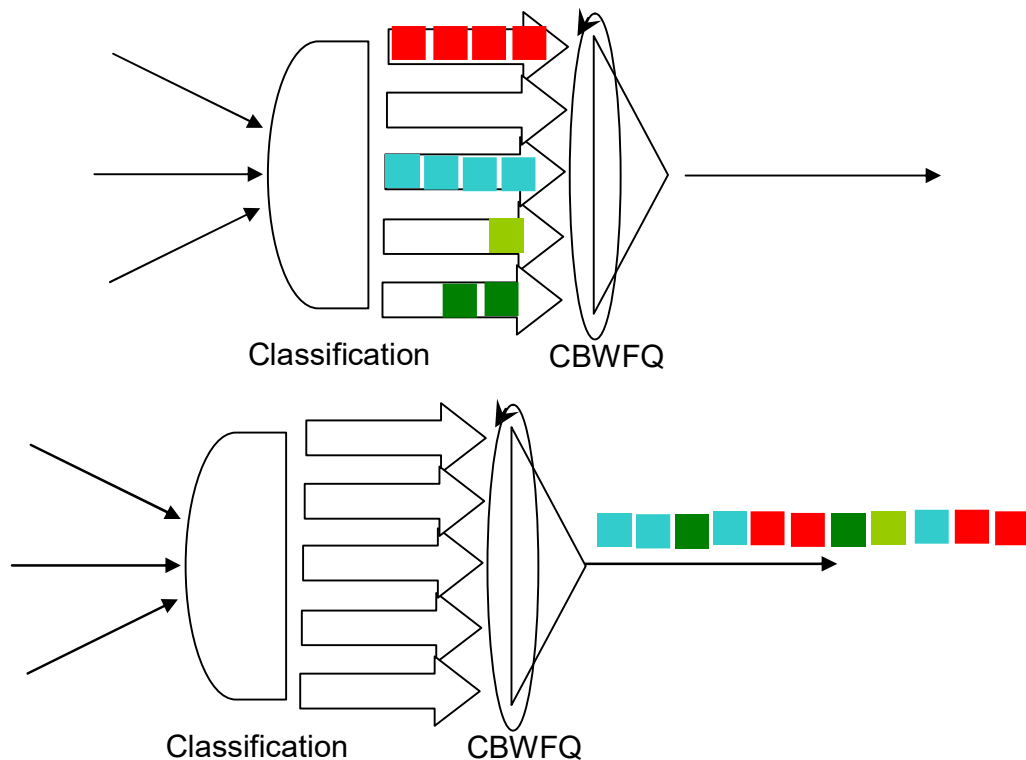
5.4 Même chose avec Class Based Weighted Fair Queing (CBWFQ). On fait l'hypothèse qu'on donne un poids de 2 à la file qui gère la Voix sur IP



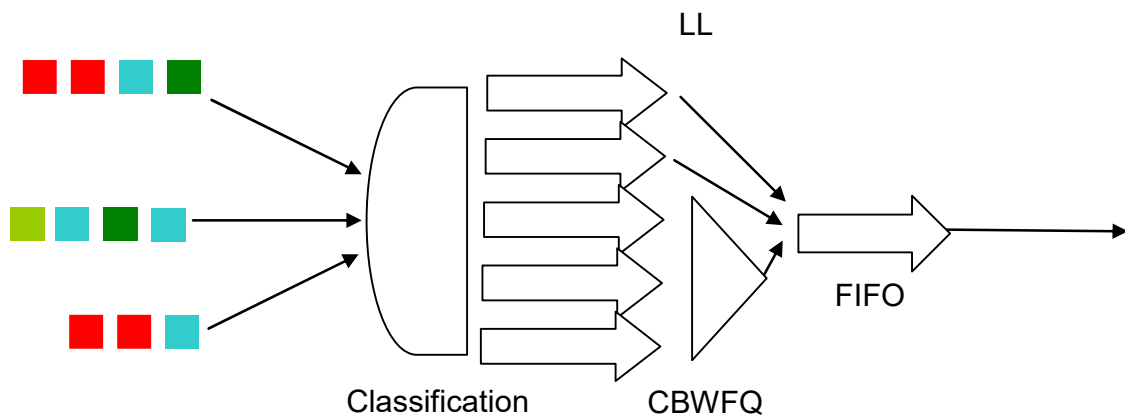
Correction :

CBWFQ, c'est en gros un Round Robin qui s'applique par classe de trafic. Une file correspond à une classe Diffserv.

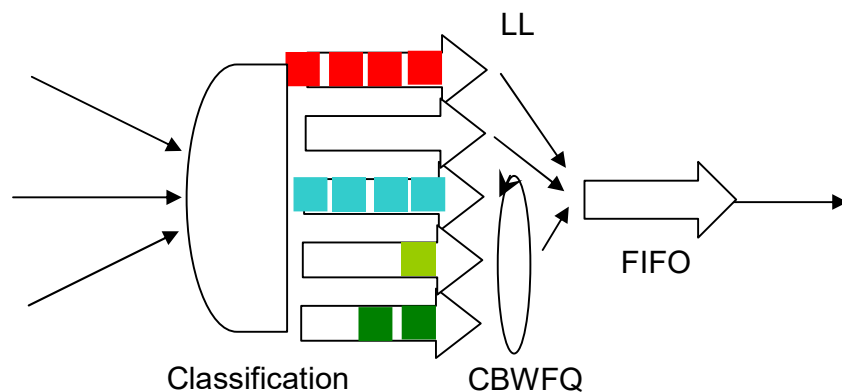
Dans l'exercice on a 5 classes. On a donc une figure qui ressemble à celle de la gestion par priorité en tous cas pour la classification du trafic.

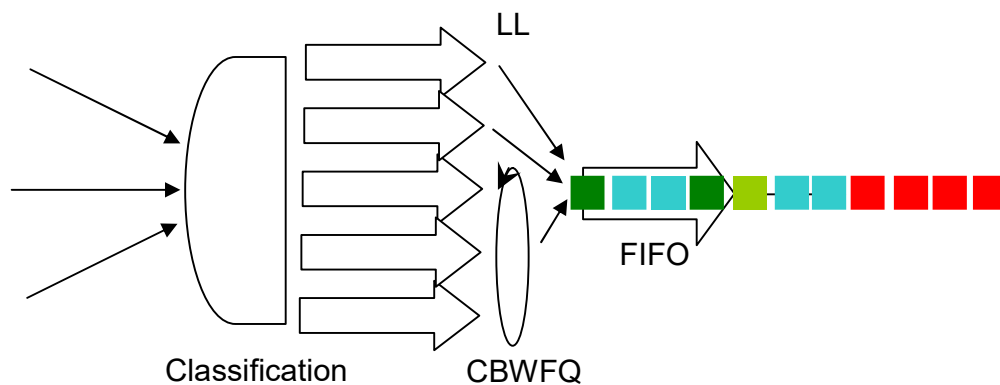


5.5 Gestion de QoS à la CISCO, LLQ (Low Latency Queuing), qui est une combinaison de PQ et de CBWFQ. Les trafics VoIP et Vidéo Conférence sont gérés en PQ à deux files respectant les priorités de ces deux trafics respectivement l'un p/r à l'autre. Le reste est géré en CBWFQ avec un poids de 2 sur le trafic de données ERP.



Correction :

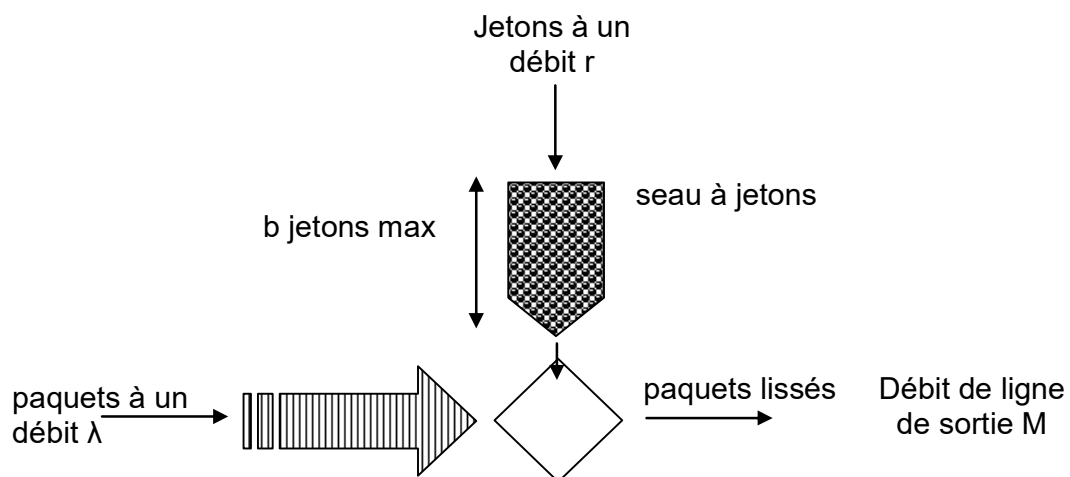




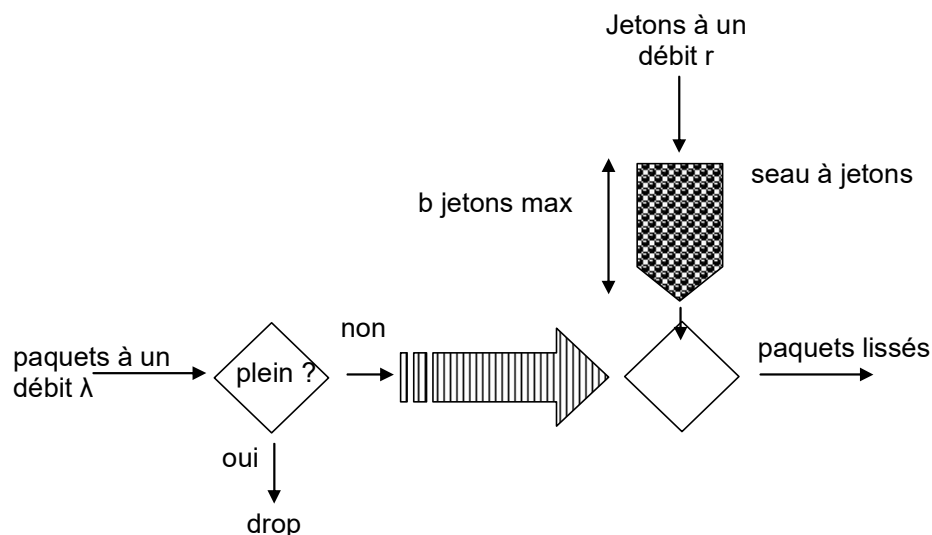
Remarque : Pour une synthèse présentée de façon différente et intéressante, consulter le lien "Scheduling and Policing Mechanisms": https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/06.06.htm (consulté en octobre 2019)

Exercice 3 : Gestion du lissage à l'aide du mécanisme de seau à jetons (Token Bucket)

On symbolise le mécanisme de lissage (traffic shaping), appelé "token bucket", associé à un flot par le dessin ci-dessous :



Pour faire évoluer le mécanisme ci-dessus vers du contrôle d'admission, on peut ajouter un test d'élimination avant la file des paquets traités avec le seau à jetons :



Sur un routeur que le flot applicatif traverse, à l'aide d'une sonde, on observe la suite de datagrammes suivante :

- La file reçoit :
 1. à t_0 un paquet de 200 octets,
 2. à $t_0+25\text{ms}$ un paquet de 400 octets,
 3. à $t_0+75\text{ms}$ un paquet de 450 octets, puis
 4. à $t_0+105\text{ms}$ un paquet de 500 octets,
 5. à $t_0+150\text{ms}$ un paquet de 900 octets,
 6. et enfin à $t_0+190\text{ms}$ un paquet de 150 octets.
- Le seau est rempli avec 1000 jetons initialement, et en reçoit 1000 toutes les 100ms. Donc à t_0 , il contient déjà 1000 jetons quand le premier paquet arrive.

Question 1

Que se passe-t-il lors de la traversée du mécanisme de lissage (on prend le dessin sans élimination) ? Expliquer le comportement de l'ensemble du dispositif.

Correction :

N° événement	Taille en octets	Date	Nombre de jetons dans le seau	Etat p/r à la sortie
		Avant t_0	1000	
1	200	t_0	$1000-200=800$	sort
2	400	t_0+25	$800-400=400$	sort
3	450	t_0+75	$400-450 < 0$	attend
		t_0+100	1000 (on ne dépasse jamais la profondeur du seau, 1000 ici)	
	450	t_0+100	$1000-450=550$	sort
4	500	t_0+105	$550-500=50$	sort
5	900	t_0+150	Pas assez de jetons	attend
		t_0+200	1000	
	900	t_0+200	$1000-900=100$	sort
6	150	t_0+190	Pas assez de jetons	attend
		t_0+300	1000	
	150	t_0+300	$1000-150=850$	sort

Question 2

En déduire l'algorithme de gestion d'un lissage par seau à jetons.

Correction :

Le seau contient N jetons au maximum, soit n le nombre de jetons courant. On peut formaliser le comportement décrit ci-dessus de la façon suivante :

1. Tous les ΔT un ensemble de jetons (fixé à l'avance, généralement N) est mis dans le seau
2. Si le seau est plein, le jeton est supprimé
3. Si un paquet de m octets arrive, si $m > n$, alors le paquet reste bloqué dans la file, il est dit non conforme
4. Sinon, il poursuit sa route sur le réseau et on retire m jetons du seau

On retrouve à peu près cette formulation sur [wikipedia](https://en.wikipedia.org/wiki/Token_bucket) https://en.wikipedia.org/wiki/Token_bucket, consulté le 10/11/2019.

Un trafic entre deux applications est décrit dans un SLA (Service Level Agreement). Il y a un SLA par sens. Les octets, quand ils sont à spécifier, sont des octets au niveau datagramme IP, donc il faut prendre en compte les entêtes IP dans les calculs. La spécification utilise la structure `_flowspec` suivante :

```
typedef struct _flowspec {
    ULONG      TokenRate;
    ULONG      TokenBucketSize;
    ULONG      PeakBandwidth;
    ULONG      Latency;
    ULONG      DelayVariation;
    SVCETYPE   ServiceType;
    ULONG      MaxSduSize;
    ULONG      MinimumPolicedSize;
} // ULONG = entier 32 bits non signé
```

- **TokenRate**, *octets par seconde*, le débit de transmission du flot entre le client et le serveur
- **TokenBucketSize**, *octets*, taille du seau à jeton, crédit maximum associé à un seau
- **PeakBandwidth**, *octets par seconde*, débit max possible du flot, supérieur ou égal à TokenRate
- **Latency**, *microseconds*, latence
- **DelayVariation**, *microseconds*, gigue, difference entre la latence min et la latence max
- **ServiceType**,

SVCETYPE_GUARANTEED	Trafic de type Expedited Forwarding
SVCETYPE_CONTROLLEDLOAD	Trafic de type Assured Forwarding
SVCETYPE_QUALITATIVE	Trafic Best Effort++, l'application ne sait pas quantifier la QoS requise bien qu'elle sache que Best Effort est insuffisant. Au niveau IP, le trafic est traité en Best Effort, au niveau liaison il bénéficie, si mis en œuvre, d'une priorité supérieure
SVCETYPE_BESTEFFORT	Trafic de type Best Effort

- **MaxSduSize**, *octets*, taille maximum d'un datagramme

- **MinimumPolicedSize**, *octets*, taille minimum d'un datagramme

Question 3

Quel type de paramètre, du point de vue de la gestion de la QoS performance, manque-t-il dans cette spécification ?

Correction :

Il manque la sensibilité du flux à la perte des datagrammes qui se caractérise par un taux de perte.

Question 4

Proposer un token rate et une profondeur de seau pour que le mécanisme de lissage puisse supporter les rafales de trafic et remplir la structure `_flowspec` pour un flot de données qui a les caractéristiques suivantes :

- Taille des données applicatives : constantes de 1460 octets,
- Protocoles de transport : UDP et RTP (Real-time Transport Protocol)
- Trafic non temps réel, sujet à des rafales de trafic, mais important au bon fonctionnement de l'entreprise quand il est activé, contraint par la latence qui doit être inférieure ou égale à 200 ms, la gigue doit rester faible inférieure à 2ms
- Débit moyen applicatif au niveau IP : 1,6 Mb/s
- Débit crête applicatif au niveau IP : 3,2 Mb/s, les rafales de trafic ne durent pas plus de 500 ms soit une demi-seconde⁶.

On précise que le lien de sortie du routeur qu'on étudie est de 4 Mb/s.

On rappelle aussi le calcul suivant (issu de https://en.wikipedia.org/wiki/Token_bucket) :

Durée max d'une rafale $T_{max} = b/(M-r)$ si $r < M$, sinon il y a un problème de dimensionnement puisqu'on autorise avec le seau plus de trafic que ce qu'on peut écouler sur la ligne de sortie

Volume max de la rafale : $T_{max} * M$

Où

- b est le nombre max de jetons qu'on peut mettre dans le seau (profondeur du seau)
- M est le débit de transmission en sortie de la file
- r est le débit de remplissage du seau en jetons

Correction :

On reprend la structure `_flowspec` :

```
typedef struct _flowspec {
```

⁶ Une rafale de 3,2Mb/s pendant une demi seconde, crée un volume de 1,6Mb supplémentaires à écouler. Le temps pour l'écouler va dépendre de la profondeur du seau, de sa vitesse de remplissage en jetons et du débit de la ligne de sortie. Le débit moyen applicatif étant de 1,6Mb/s et la ligne de sortie à 4Mb/s, on peut écouler les 1,6 Mb du trafic accumulé en une seconde.

- `ULONG TokenRate;`
`ULONG TokenBucketSize;`

Par rapport à la formule ci-dessus, on connaît M (4Mb/s ou 0.5Mo/s), et T_{max} (0,5s). Deux façons de voir le calcul :

- On considère le `TokenRate`, r , en régime permanent égal au débit moyen applicatif au niveau IP. Du coup $r = 1.6\text{Mb/s}$, soit 0.2 Mo/s . Avec la formule $T_{max} = b/(M-r)$, on a une équation à résoudre en b .

$$b = (M-r) \text{ (en octets ou en bits par seconde)} * T_{max} \text{ (en secondes)}$$

$$b = (0.5 - 0.2) * 0.5 = 0.3 * 0.5 = 0.15\text{Mo} \text{ soit } 1.2\text{Mb}$$

Donc `TokenRate` vaut 1.6 Mb/s et `TokenBucketSize` vaut 1.2Mb.

- On considère qu'on ne connaît ni b ni r a priori. Cela fait une équation à 2 inconnues. Il faut fixer une des deux valeurs pour obtenir une réponse.

On peut tenter de proposer de fixer une valeur pour b et voir ce qu'il advient ? Dans un premier temps, on peut imaginer qu'un tampon b de 0,6Mb, soit, 0,075Mo/s ou 75K octets, pourrait suffire. Cela représente environ 50 datagrammes de 1500 octets, ce n'est pas déraisonnable.

D'après la formule ci-dessus : $T_{max} = b/(M-r)$, on obtient $M-r = b/T_{max}$, et donc $r = M - (b/T_{max})$ si $r < M$.

Numériquement, $r = 4 - 0,6/0,5$ (toutes les unités sont en Mb et en secondes), finalement $r = 4 - 1.2 = 2.8\text{Mb/s}$, et on a bien $r < M$.

Donc `TokenRate` vaut 2.8Mb/s et `TokenBucketSize` vaut 0,6Mb. Par rapport au premier calcul, la taille du saut est divisée par 2, et le son taux de remplissage est double. Ça ne semble pas surprenant.

- `ULONG PeakBandwidth;`
C'est donné dans l'énoncé : 3.2Mb/s ou **0.4 Mo/s**
- `ULONG Latency;`
Elle doit être inférieure ou égale à 200ms, soit **200 000 microsecondes**.
- `ULONG DelayVariation;`
Le trafic est sensible à la gigue, 2ms, soit **2000 microsecondes**.
- `SERVICETYPE ServiceType;`
Ce n'est pas un trafic temps réel. Ce n'est pas un trafic quelconque, ça élimine Best Effort. On va donc prendre le service **`SERVICETYPE_CONTROLLEDLOAD`**.
- `ULONG MaxSduSize;`
`ULONG MinimumPolicedSize;`
Les données applicatives ont toute la même taille, 1460 octets. On y ajoute l'entête IP (20 octets), l'entête UDP (8 octets), l'entête RTP (12 octets), soit 40 octets. Du coup `MaxSduSize` et `MinimumPolicedSize` valent **1500 octets**.

On voit apparaître dans certains documents une autre formule : $C + r.S = M.S$

- C = Token bucket capacity in bytes.
- S = Burst length in seconds. (T_{max} pour notre exercice)
- r = Rate of token arrival in bytes/sec.
- M = maximum output rate in bytes/sec.

Proposée en particulier dans CS 4/54201 Computer Communication Network Kent State University Dept. of Computer Science



Cette formule sert à calculer S (Maximum Burst Length for Token Bucket) quand on connaît C, r et M. On a alors $S = C/(M-r)$