



TP QoS

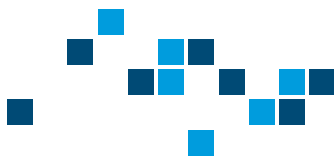
Antoine Laguette, Guillaume Tisserand, Juliette Bluem

4 octobre 2022



**UNIVERSITÉ
DE LORRAINE**

LORRAINE INP
les talents se lèvent à l'Est



1 Introduction

Lors de ce TP, nous allons découvrir la Qualité de Service (QoS). Ce terme regroupe les techniques et technologies qui permettent de gérer un trafic en termes de débit. Nous allons découvrir deux des principales formes de la QoS : Le marquage de trafic et sa mise en forme (shaping).

Pour cela, nous mettons en place l'architecture suivante, composée de deux routeurs et deux hôtes.

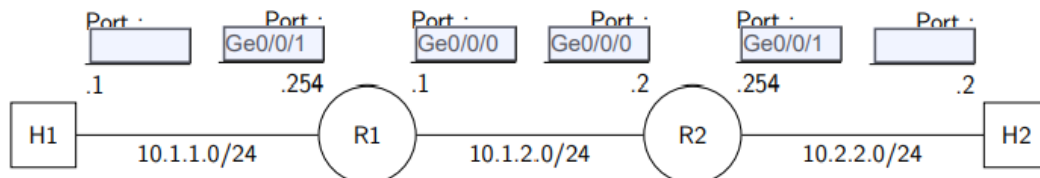


FIGURE 1 – Topologie du réseau

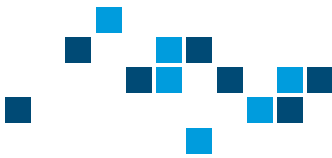
2 Marquage de trafic

Le marquage de trafic consiste à attribuer des préférences de priorité pour le trafic important afin qu'il atteigne sa destination de manière rapide et fiable.

Dans notre réseau, sur le routeur R1, nous créons deux ACL. L'une d'elles servira pour le trafic UDP, et l'autre pour le trafic TCP. Nous leur faisons correspondre deux class-map distinctes qui serviront à trier le trafic selon différents critères :

```
R1(config-cmap)#match ?
access-group      Access group
any               Any packets
atm               Match on ATM info
atm-vci           Match on atm vci
class-map         Class map
cos               IEEE 802.1Q/ISL class of service/user priority values
destination-address Destination address
discard-class     Discard behavior identifier
dscp              Match DSCP in IPv4 and IPv6 packets
fr-de             Match on Frame-relay DE bit
fr-dlci           Match on fr-dlci
input-interface   Select an input interface to match
ip                IP specific values
mpls              Multi Protocol Label Switching specific values
not               Negate this match result
packet            Layer 3 Packet length
precedence        Match Precedence in IPv4 and IPv6 packets
protocol          Protocol
qos-group         Qos-group
security-group    Security group
source-address    Source address
traffic-category  Match on traffic-category
vlan              VLANs to match
```

FIGURE 2 – Critères possible avec le class-map



Voici un bilan de nos class-map :

```
R1#show class-map
Class Map match-any class-default (id
Match any

Class Map match-all UDP (id 1)
Match access-group 100

Class Map match-all TCP (id 2)
Match access-group 101
```

FIGURE 3 – Caption

Nous observons que [class-map match-all] correspond à un ET logique, tandis que [class-map match-any] correspond à un OU logique.

Nous créons maintenant une policy-map qui marquera le champ DSCP du trafic UDP à 0, et celui de TCP à EF. Nous plaçons cette dernière en input sur l'interface de R1 vers H1.

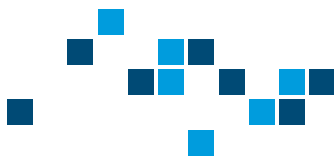
Notons que le DSCP (Differentiated Services Code Point) est un champ dans l'entête d'un paquet IP. Le but de ce champ est de permettre la différenciation de services ou DiffServ.

Nous créons maintenant du trafic UDP et TCP simultanément sur H1 et H2 à l'aide d'iperf2. Et nous observons sur H2 les paquets qui arrivent grâce à Wireshark.

10.1.1.1	10.2.2.2	TCP	Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
10.2.2.2	10.1.1.1	TCP	0100 = Version: 4
10.1.1.1	10.2.2.2	TCP 0101 = Header Length: 20 bytes (5)
10.1.1.1	10.2.2.2	TCP	✓ Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
10.2.2.2	10.1.1.1	TCP	1011 10.. = Differentiated Services Codepoint: Expedited Forwarding (46)
10.1.1.1	10.2.2.2	TCP00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
10.1.1.1	10.2.2.2	TCP	Total Length: 1500
10.2.2.2	10.1.1.1	TCP	Identification: 0xebcf (60367)
			✓ Flags: 0x4000. Don't fragment
65842 13:08:00,499708 10.1.1.1 10.2.2.2 UDP 151			✓ Frame 65858: 1512 bytes on wire (12096 bits), 1512 bytes captured (12096 bits) on interface V
65846 13:08:00,499708 10.1.1.1 10.2.2.2 UDP 151			> Ethernet II, Src: 48:2e:72:5d:d2:81 (48:2e:72:5d:d2:81), Dst: Dell_a5:d7:f8 (b8:85:84:a5:d7:f8)
65847 13:08:00,522217 10.1.1.1 10.2.2.2 UDP 151			✓ Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
65848 13:08:00,522219 10.1.1.1 10.2.2.2 UDP 151			0100 = Version: 4
65849 13:08:00,522239 10.2.2.2 10.1.1.1 ICMP 99		 0101 = Header Length: 20 bytes (5)
65850 13:08:00,537363 10.1.1.1 10.2.2.2 UDP 151			✓ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
65851 13:08:00,553068 10.1.1.1 10.2.2.2 UDP 151			0000 00.. = Differentiated Services Codepoint: Default (0)
65852 13:08:00,553087 10.1.1.1 10.2.2.2 UDP 151		00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
65853 13:08:00,568899 10.1.1.1 10.2.2.2 UDP 151			Total Length: 1408
65854 13:08:00,584476 10.1.1.1 10.2.2.2 UDP 151			

FIGURE 4 – Paquets TCP et UDP sur H2

La commande [show policy-map interface Ge0/0/01] permet d'observer le nombre de paquets ayant été traités par la policy-map.



3 Shaping de trafic

Le shaping trafic est le contrôle du volume des échanges sur un réseau. Son but est d'optimiser ou de garantir les performances en retardant les paquets qui correspondent à certains critères.

Afin de le mettre en application, nous commençons par limiter la liaison entre R1 et R2 à 100 Mb/s. Puis grâce à un iperf entre nos hotes, nous voyons que nous sommes finalement limités à 52Mb/s. Nous continuons ainsi, ce ne sera pas handicapant.

Nous identifions que la méthode actuelle de gestion de de file d'attente de l'interface Ge0/0/0 de R1 est FIFO (first in first out).

Nous créons encore deux class-map comme précédemment, mais cette fois-ci entre R1 et R2. Elles filtreront les packets EF et BE.

```
R1#show class-map
Class Map match-all EF (id 3)
  Match dscp ef (46)

Class Map match-all BE (id 4)
  Match dscp default (0)

Class Map match-any class-default (id 0)
  Match any

Class Map match-all UDP (id 1)
  Match access-group 100

Class Map match-all TCP (id 2)
  Match access-group 101

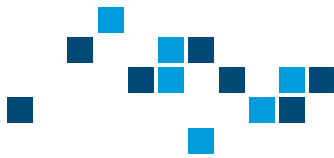
R1#show policy-map
Policy Map MARK
  Class UDP
    set dscp default
  Class TCP
    set dscp ef

Policy Map SHAPE
  Class BE
    bandwidth 10 (%)
  Class EF
    bandwidth 89 (%)
```

FIGURE 5 – class-map mises en place

Puis une policy-map qui reserve 10% de bande-passante à la classe BE, et 89% à la classe EF. Nous observons que la méthode de file d'attente de l'interface est basée sur les classes.

Pour finir notre travail sur la mise en forme de trafic, nous créons 100 Mb/s de trafic UDP et 100 Mb/s de trafic TCP simultanément de H1 vers H2 à l'aide d'iperf. Nous remarquons que la répartition n'est pas tout à fait identique à ce que nous avons paramétré : le trafic UDP représente 70% de la bande passante au lieu des 89% attendus



4 Conclusion

Grâce à ce TP, nous avons pu découvrir la qualité de services en travaillant sur deux points distinct.

Dans une première partie : le marquage de trafic. Via la mise en place d'ACL que nous avons fait matché à des class-map UDP et TCP. Puis par le marquage des champs DSCP de chacune d'elles.

Dans une seconde partie, nous avons travaillé la mise en forme de trafic. Les class-map et la policy-map nous ont permis de répartir la charge de la bande passante de notre réseau selon nos souhaits.