



University of Pisa  
MSc in Computer Engineering  
Advanced Network Architectures and Wireless Systems

## QoS project

Daniela Comola  
Sara Lotano  
Eugenia Petrangeli

Academic Year 2019/2020

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Requirements . . . . .	2
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	Address planning . . . . .	4
<b>3</b>	<b>Configuration</b>	<b>5</b>
3.1	Classification . . . . .	5
3.2	Marking . . . . .	5
3.2.1	Class-Based WFQ . . . . .	6
3.3	Policing . . . . .	7
3.4	MPLS - TE . . . . .	8
3.4.1	Test 1 . . . . .	9
3.4.2	Test 2 . . . . .	9
<b>4</b>	<b>Running the project</b>	<b>10</b>
4.1	Prerequisites . . . . .	10
4.2	Starting the network . . . . .	10

# 1 Introduction

This project considers a multimedia network in which Quality of Service requirements must be met using a Differentiated Services Architecture and MPLS TE.

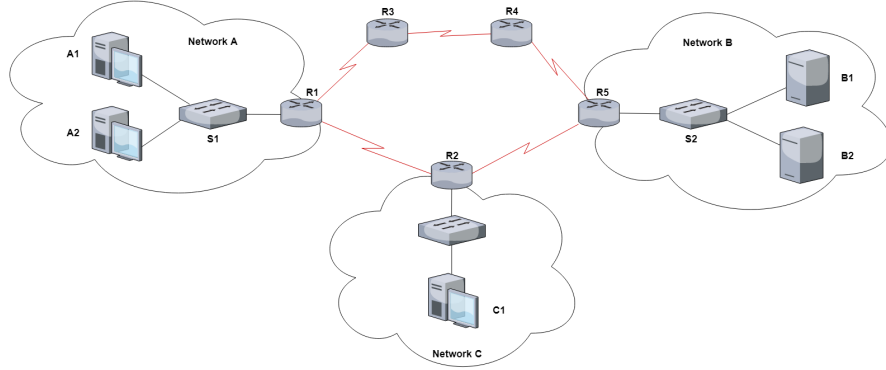


Figure 1: Network Architecture

The network consists of three different access networks (Network A, B and C) interconnected through a core network of five routers (R1 - R5). The communications happens from clients to servers according to the data flow characteristics shown below:

Source	Destination	Priority	Bandwidth
A1	B1	High	0.9 Mbps
A2	B2	Low	1.2 Mbps
C1	B1	High	0.9 Mbps

Table 1: Data flow characteristics

## 1.1 Requirements

- Each flow has to be conformed with the characteristics shown in the table. Moreover:
  - for high priority flows, the excess traffic must be downgraded to low priority.
  - for low priority traffic, the excess traffic must be dropped.
- For each high priority flow must be granted at worst 60% of the bandwidth for the link it traverses in the core network.

## 2 Design

The network was built using the network software emulator **GNS3**. The resulting topology is shown below:

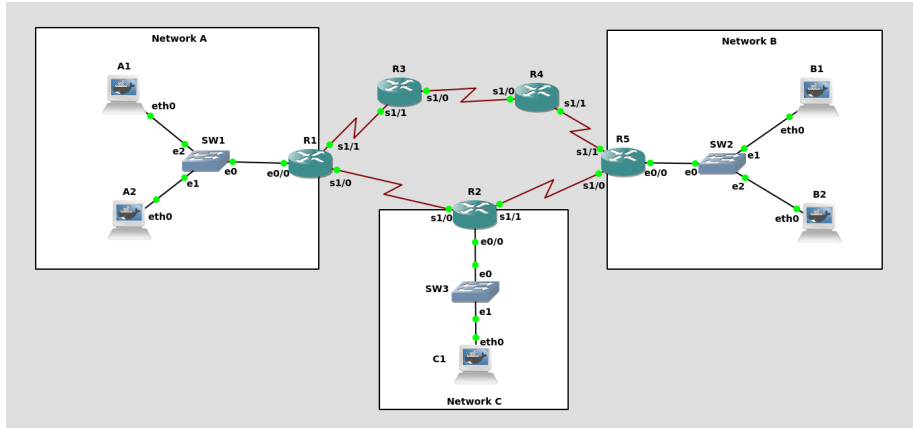


Figure 2: GNS3 network topology

To ensure basic communication between all hosts belonging to the network, all network interfaces have been configured using the static addressing plan shown in the table below and, on all routers, OSPF has been configured as the routing protocol.

## 2.1 Address planning

Network	Device	Interface	IP address	Network mask
<b>Core</b>	<b>R1</b>	Serial 1/0	10.1.1.1	/30
		Serial 1/1	10.2.2.1	/30
		Loopback 0	192.168.4.1	/32
	<b>R2</b>	Serial 1/0	10.1.1.2	/30
		Serial 1/1	10.5.5.1	/30
		Loopback 0	192.168.4.2	/32
	<b>R3</b>	Serial 1/0	10.2.2.2	/30
		Serial 1/1	10.3.3.1	/30
		Loopback 0	192.168.4.3	/32
	<b>R4</b>	Serial 1/0	10.3.3.2	/30
		Serial 1/1	10.4.4.1	/30
		Loopback 0	192.168.4.4	/32
	<b>R5</b>	Serial 1/0	10.5.5.2	/30
		Serial 1/1	10.4.4.2	/30
		Loopback 0	192.168.4.5	/32
<b>A</b>	<b>R1</b>	Ethernet 0/0	192.168.1.1	/24
	<b>A1</b>	Ethernet 0	192.168.1.2	/24
	<b>A2</b>	Ethernet 0	192.168.1.3	/24
<b>B</b>	<b>R2</b>	Ethernet 0/0	192.168.2.1	/24
	<b>B1</b>	Ethernet 0	192.168.2.2	/24
	<b>B2</b>	Ethernet 0	192.168.2.3	/24
<b>C</b>	<b>R5</b>	Ethernet 0/0	192.168.3.1	/24
	<b>C1</b>	Ethernet 0	192.168.3.2	/24

## 3 Configuration

The configuration of routers has been made considering the following steps:

### 3.1 Classification

Inside the R1 router the following access-lists have been defined:

```
1 access-list 100 permit ip host 192.168.1.2 host 192.168.2.2
2 access-list 101 permit ip host 192.168.1.3 host 192.168.2.3
```

Both activated on the Ethernet0/0 input interface.

Each access-list has been associated with a class-map:

- access-list 100 with class-map **A1\_TO\_B1**
- access-list 101 with class-map **A2\_TO\_B2**

Inside the R2 router, on Ethernet0/0 interface, the following access-list has been defined:

```
1 access-list 100 permit ip host 192.168.3.2 host 192.168.2.2
```

and it has been associated with class-map **C1\_TO\_B1**

### 3.2 Marking

The following DSCP values were used to mark the traffic:

- **EF** (Expedited Forwarding): used to mark high-priority flows.
- **AF11** (Assured Forwarding): used to mark low-priority flows.

The marking is done only at the ingress of the core network, within the routers R1 and R2.

To check the correct marking of high-priority packets, we can use Wireshark to inspect one of the packets belonging to the data flow generated from A1 to B1:

No.	Time	Source	Destination	Protocol	Length
55	142.681073	192.168.1.2	192.168.2.2	UDP	
56	142.690207	192.168.1.2	192.168.2.2	UDP	
57	142.703832	192.168.1.2	192.168.2.2	UDP	
58	142.715957	192.168.1.2	192.168.2.2	UDP	
59	142.725567	192.168.1.2	192.168.2.2	UDP	
60	142.737280	192.168.1.2	192.168.2.2	UDP	
61	142.749121	192.168.1.2	192.168.2.2	UDP	
62	142.760790	192.168.1.2	192.168.2.2	UDP	

▶ Frame 57: 1512 bytes on wire (12096 bits), 1512 bytes captured (12096 bits) on interface 0  
 ▶ Ethernet II, Src: aa:bb:cc:00:05:00 (aa:bb:cc:00:05:00), Dst: c2:59:21:94:1d:58 (c2:59:21:94:1d:58)  
 ▼ Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.2.2  
     0100 .... = Version: 4  
     .... 0101 = Header Length: 20 bytes (5)  
     ▼ Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)  
         1011 10.. = Differentiated Services Codepoint: Expedited Forwarding (46)  
         .....000 = Explicit Congestion Notification: Not ECN-Capable Transport (0)  
         Total Length: 1498  
         Identification: 0x584c (22604)

Figure 3: Example of packet marked with DSCP value: ef

The same type of control can be performed for low-priority packets. For example, analyzing those generated from A2 to B2:

No.	Time	Source	Destination	Protocol	Length
964	53.918278	192.168.1.3	192.168.2.3	UDP	
965	53.927170	192.168.1.3	192.168.2.3	UDP	
966	53.936000	192.168.1.3	192.168.2.3	UDP	
967	53.944070	192.168.1.3	192.168.2.3	UDP	
968	53.952377	192.168.1.3	192.168.2.3	UDP	
969	53.960503	192.168.1.3	192.168.2.3	UDP	
970	53.970700	192.168.1.3	192.168.2.3	UDP	
971	53.978919	192.168.1.3	192.168.2.3	UDP	

▶ Frame 967: 1512 bytes on wire (12096 bits), 1512 bytes captured (12096 bits) on interface 0  
 ▶ Ethernet II, Src: aa:bb:cc:00:05:00 (aa:bb:cc:00:05:00), Dst: 5a:9a:5a:ed:8d:36 (5a:9a:5a:ed:8d:36)  
 ▼ Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.2.3  
     0100 .... = Version: 4  
     .... 0101 = Header Length: 20 bytes (5)  
     ▼ Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)  
         0010 10.. = Differentiated Services Codepoint: Assured Forwarding 11 (10)  
         .....000 = Explicit Congestion Notification: Not ECN-Capable Transport (0)  
         Total Length: 1498  
         Identification: 0x083b (2107)

Figure 4: Example of packet marked with DSCP value: af11

### 3.2.1 Class-Based WFQ

In order to guarantee the requirement that each high-priority flow must be guaranteed in the worst case for 60% of the bandwidth on each link it crosses within the core network, the first thing to do is to configure the following access-list on all routers:

```
1 access-list 102 permit ip any any dscp ef
```

which is associated with the class-map **HIGH\_PRIO\_FLOW**.

In order to set the bandwidth share for the high-priority traffic, we define:

```
1 policy-map OUT
```

```

2      class HIGH_PRIO_FLOW
3      bandwidth percent 60

```

and we apply this configuration as output service-policy.

### 3.3 Policing

To meet the bandwidth characteristic of the data flow, we have used **CAR** (Committed Access Rate) that is adopted at the ingress routers to limit ingress traffic of a flow.

We define a policing function, applied on the R1 and R2 Ethernet0/0 interfaces, whose purpose is to limit high-priority traffic to 0.9 Mbps and to remark the exceeded traffic. This behavior is obtained with the option exceed-action set-dscp-transmit 10 (the value 10 has been chosen because it corresponds to the value AF11 used for low-priority traffic).

For example, the configuration on the R1 router is as follows:

```

1      rate-limit input access-group 100 896000 5000 5000 conform-
      action transmit exceed-action set-dscp-transmit 10

```

To check the correct policy configuration we can, for example, generate 1M traffic from host A1 to host B1 and check the correct application of the rules set on the R1 router. Looking at the R1 router console, we can see that part of the packets have been transmitted while the remaining part has been remarked with the new DSCP value of 10, as we expected.

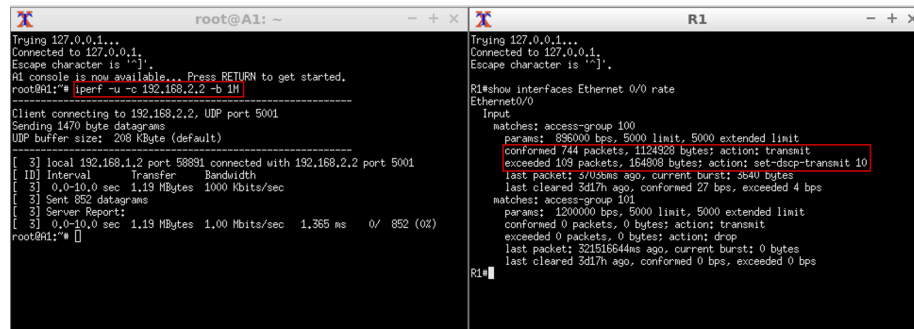


Figure 5: Example of packet remarking

The same behavior is obtained by generating a data flow from host C1 to host B1 and analyzing the status of the Ethernet 0/0 interface of the R2 router.

A second policing function has been defined only for the R1 router that limits low-priority traffic to a rate of 1.2 Mbps and drops the exceeded traffic.

```

1      rate-limit input access-group 101 1200000 5000 5000 conform-
      action transmit exceed-action drop

```



To check the correct policy configuration we generated 2M traffic from host A2 to host B2. We can see that part of the packets have been transmitted while the rest have been dropped, as we expected.

```

root@A2: ~
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
A2 console is now available... Press RETURN to get started.
root@A2:~# iperf -u -c 192.168.2.3 -b 2M

Client connecting to 192.168.2.3, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 192.168.1.3 port 36644 connected with 192.168.2.3 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  2.39 MBytes  2.00 Mbits/sec
[ 3] Sent 1702 datagrams
[ 3] Server Report:
[ 3] 0.0-10.5 sec  1.40 MBytes  1.11 Mbits/sec  2.551 ms 706/ 1702 (41%)
root@A2:~#

R1
*Mar 6 08:38:30.458: ZLINK-5-CHANGED: Interface Ethernet0/2, changed state to administratively down
*Mar 6 08:38:30.468: ZLINK-5-CHANGED: Interface Ethernet0/3, changed state to administratively down
*Mar 6 08:38:30.478: ZLINK-5-CHANGED: Interface Serial1/2, changed state to administratively down
*Mar 6 08:38:30.478: ZLINK-5-CHANGED: Interface Serial1/3, changed state to administratively down
R1#show interfaces Ethernet 0/0 rate
Ethernet0/0
  Input
    matches: access-group 100
    params: 888000 bps, 5000 limit, 5000 extended limit
    conformed 0 packets, 0 bytes; action: transmit
    exceeded 0 packets, 0 bytes; action: set-dscp-transmit 10
    last packet: 12133ms ago, current burst: 0 bytes
    last cleared 00:01:53 ago, conformed 0 bps, exceeded 0 bps
    matches: access-group 101
    params: 1200000 bps, 5000 limit, 5000 extended limit
    conformed 999 packets, 1509566 bytes; action: transmit
    exceeded 707 packets, 1068384 bytes; action: drop
    last packet: 2530ms ago, current burst: 0 bytes
    last cleared 00:01:53 ago, conformed 106811 bps, exceeded 75495 bps
  
```

Figure 6: Example of packet dropping

### 3.4 MPLS - TE

To correctly configure MPLS we have enabled the layer 3 switching technology **CEF** (Cisco Express Forwarding) on all routers.

To guarantee 60% bandwidth for the high priority traffic, a tunnel has been created from router R1 to router R5, through router R3 and R4. Without any tunnel, the OSPF protocol would always forward high priority traffic to the R2 router as it is on the shortest path to reach R5.

**Tunnel 0** has been created by specifying the path, called *longpath*:

```

1 ip explicit-path name longpath enable
2 next-address 10.2.2.2
3 next-address 10.3.3.2
4 next-address 10.4.4.2

```

In order to differentiate routing we can set the following static routes on R1:

```

1 ip route 192.168.2.2 255.255.255.255 Tunnel 0
2 ip route 192.168.2.3 255.255.255.255 Serial 1/0

```

We can use **traceroute** command to print the route that packets, from A1 and A2, take to reach the hosts B1 and B2 respectively.

```

root@A1:~# traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1) 2.069 ms 2.433 ms 2.418 ms
 2 10.2.2.2 (10.2.2.2) 24.573 ms 25.288 ms 26.140 ms
 3 10.5.5.2 (10.5.5.2) 30.708 ms 31.133 ms 33.888 ms
 4 10.4.4.2 (10.4.4.2) 34.344 ms 35.065 ms 35.446 ms
 5 192.168.2.2 (192.168.2.2) 35.150 ms 36.644 ms 36.640 ms
root@A1:~#

root@A2:~# traceroute 192.168.2.3
traceroute to 192.168.2.3 (192.168.2.3), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1) 2.241 ms 3.077 ms 3.072 ms
 2 10.1.1.2 (10.1.1.2) 3.500 ms 4.291 ms 4.344 ms
 3 10.5.5.2 (10.5.5.2) 6.809 ms 7.728 ms 8.454 ms
 4 192.168.2.3 (192.168.2.3) 14.580 ms 14.630 ms 19.957 ms
root@A2:~#

```

Figure 7: Traceroute used on A1 and A2 hosts

To test the correctness of the guaranteed bandwidth in the worst case at 60% we can perform some tests using **iperf**. In both cases, we must remember that the bandwidth of a serial link is equal to 1544 Kbit/s.

### 3.4.1 Test 1

In this first test, we generate 1 Mbps traffic from the hosts A1 and C1.

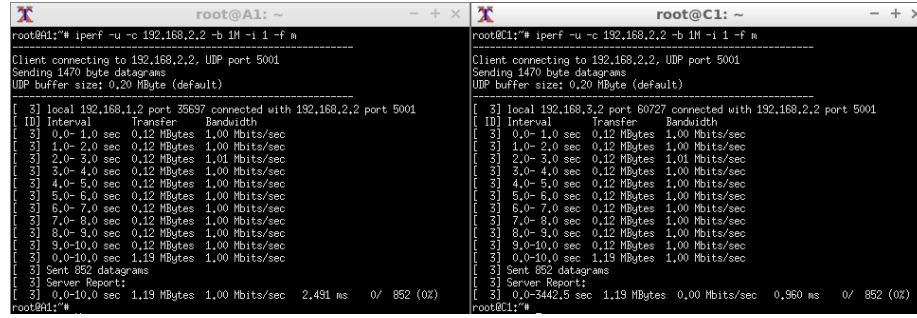


Figure 8: Data flows generated from A1 and C1

As can be seen, the percentage of lost packages is equal to 0% in both cases. This is because data flows follow different paths as shown above.

### 3.4.2 Test 2

In the second test, we generate 1 Mbps traffic from hosts A2 and C1. In this case, the traffic will follow the same path through the R2 and R5.

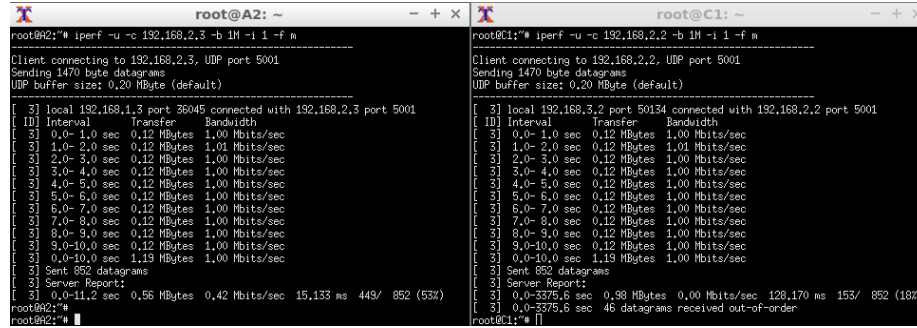


Figure 9: Data flows generated from A2 and C1

As shown by the C1 console, the percentage of lost packets is only 18%, so the bandwidth constraint is satisfied.

## 4 Running the project

### 4.1 Prerequisites

To develop the multimedia network, we used **GNS3** (Graphical Newtork Simulator) which allows the emulation of real hosts and real Cisco Routers.

### 4.2 Starting the network

In order to open the network, just run the following command:

```
1 $gns3 multimedia_network.gns3
```

Once the network is opened, all devices must be started. When the LEDs of all interfaces are green, the network is ready to be used.