universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA
MESTRADO ENG. COMPUTADORES E INFORMÁTICA

ARQUITECTURA DE COMUNICAÇÕES

LAB GUIDE

Objectives

- Quality of Service in IP networks.
- Mechanisms for the management of router queues.
- DiffServ-based end-to-end quality of service.

Duration

1 week.

1. Experiments with queuing mechanisms

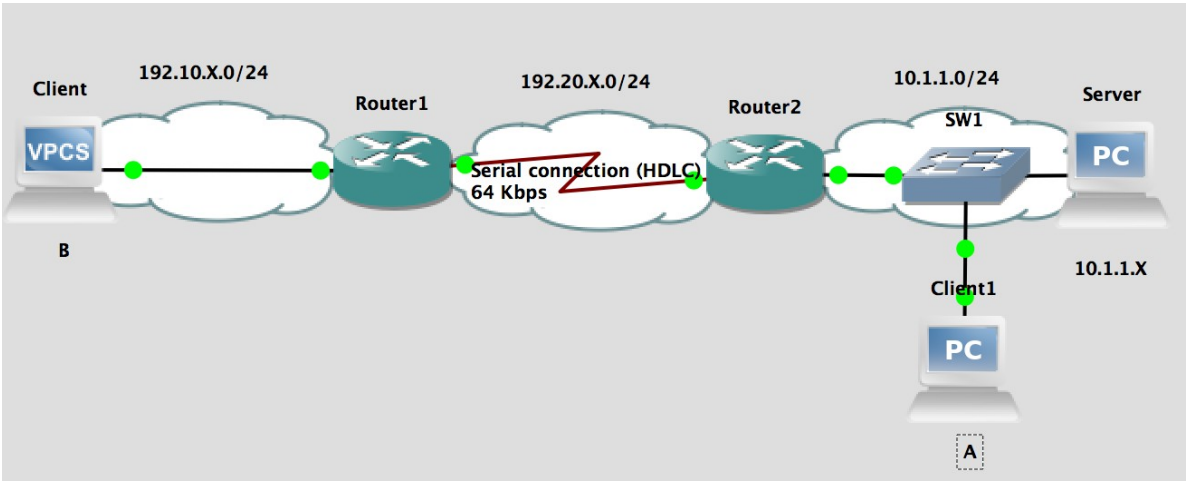
1. Set up a network composed by 2 routers interconnected through their serial ports using V.35 cables, as specified in the next figure. If required, share routers with the neighboring group.

Start your PC in Linux operating system.

Configure a bandwidth of 64Kbits/s in the serial connection between both routers and use the HDLC protocol to encapsulate data. **Pay attention to the difference between DTE and DCE side of the cable.**

Configure the IP addresses at the interfaces of the routers as depicted in the figure. Configure the OSPF protocol in both routers in order to guarantee global connectivity in the network.

Note: Use X as your group number.



Console of routers Cisco in Linux

To configure the *router* it is required to use the *Minicom* program. To run this program, open a Linux console and type `minicom -D /dev/ttyUSB0`. You can use *putty* in Windows, selecting the "serial" option in the connection type.

You will need a serial port in your computer – or then use the USB adapters available in the laboratory. Please check the serial port number before connecting (Linux: /dev/ttyUSBx; Windows: COMx) with the program. In alternative, you can use the PCs in the laboratory.

The configuration settings by default are 9600bps, no parity, disable hardware flow control.

Configuration of the Cisco routers serial interfaces:

When connecting 2 routers through synchronous serial interfaces, these interfaces become configured by default with the level 2 HDLC protocol. In order to activate the serial interfaces, it is necessary to configure the clock rate of the connection. This can be done by executing the following command at the interface of the router that was connected as DCE (this functionality is indicated in the label located at the end of the V.35 cable):

```
Router#configure terminal
Router(config)#interface Serial 0/0
Router(config-if)#clock rate 64000
```

Note: the DTE will adjust its rate automatically to the one imposed by the DCE.

Configuration of the PC in *Linux*

To configure the PC IP address, execute the following command:

```
ifconfig enp4s0 <IPaddress> netmask <IPmask>
or
ip addr add <IPaddress>/<IPmask> dev enp4s0
```

To configure the Gateway of the PC, execute the following command:

```
route add default gw <IPaddress>
or
ip route add default via <IPaddress>
```

The command `ip route` shows the routing table of the PC and the configured *Gateway*.

2. Using the show interfaces serial 0 command in both routers, verify which are the existing queues, which is the queuing management mechanism that is being used and to what queues is it being applied.

Configure the first-in-first-out (FIFO) mechanism in both serial interfaces. Using the show interfaces serial 0 command in both routers, verify which are the existing queues, which is the queuing management mechanism that is being used and to what queues is it being applied.

Configuration of the FIFO discipline

First, you have to examine which management mechanism if configured by default:

```
Router#show interfaces
```

If the default queuing management mechanism in the interface that you want to configure is FIFO, you don't have to take any action. If the configured mechanism is Weighted Fair Queuing, you have to deactivate it:

```
Router#configure terminal
Router(config)#interface Serial 0
Router(config-if)#no fair-queue
```

3. Place your PC in position A and configure it accordingly. Verify if *iperf* is installed on your PC: if it is not installed, install it with the following command:

```
apt-get install iperf
```

In your server, verify if iperf is installed on your server: if it is not installed, install it. Configure the respective server gateway (or specific routes). Move the PC to position B, configure it accordingly and test the connectivity with your server.

4. At the PC (in position B), open a terminal, kill any pendent *iperf* processes, start *iperf* in your PC as a UDP server on port 5000:

```
killall iperf
iperf -u -s -p 5000 &
```

Start a Wireshark capture. Open a SSH connection to the server, and use *iperf* to send IP/UDP packets to your PC, during 2 minutes, at a rate of 128Kbits/s and having a length of 1000 bytes (972 bytes of data + 28 bytes of headers):

```
iperf -u -c 192.10.X.Y -p 5000 -t120 -l 972 -b 128K
```

Wait 2 minutes and from the Wireshark's Conversations Statistics, estimate the network performance. Justify the obtained results.

Note: 192.10.X.Y should be your PC IP address.

5. Start 3 new *iperf* UDP servers in your PC on ports 5001 to 5003:

```
iperf -u -s -p 5001 &
iperf -u -s -p 5002 &
iperf -u -s -p 5003 &
```

Start a Wireshark capture. At the server, use *iperf* to generate 4 IP/UDP packet flows, during 3 minutes, with the following packet size distribution:

- 64 bytes packets sent at 25.6 kbits/s to port 5000;
- 128 bytes packets sent at 25.6 kbits/s to port 5001;
- 512 bytes packets sent at 38.4 kbits/s to port 5002;
- 1024 bytes packets sent at 38.4 kbits/s to port 5003.

To that purpose, create the following bash script (traffic.sh) to start simultaneously four instances of
`iperf`:

```
#/bin/bash
iperf -u -c 192.10.X.Y -p 5000 -t180 -l 36 -b 25.6K &
iperf -u -c 192.10.X.Y -p 5001 -t180 -l 100 -b 25.6K &
iperf -u -c 192.10.X.Y -p 5002 -t180 -l 484 -b 38.4K &
iperf -u -c 192.10.X.Y -p 5003 -t180 -l 992 -b 38.4K &
```

At the server run the bash script:

```
bash traffic.sh
```

Wait 3 minutes and from the Wireshark's Conversations Statistics, estimate the network performance.

Justify the obtained results, having in mind the way FIFO queuing systems work.

6. Change the queuing management mechanism of the router 2 serial interface to weighted fair queue (WFQ), by configuring a limit of 64 messages per queue, 16 queues and 0 queues that can be reserved for RSVP. Use the `show queuing fair` command to verify the correct configuration of the router. Using the `show interfaces serial 0` command in both routers, verify which are the existing queues, which is the queuing management mechanism that is being used and to what queues is it being applied.

Configuration of the WFQ discipline

In order to activate the WFQ mechanism at an interface (for example, at interface Serial 0) with the default parameters, you should execute:

```
Router#configure terminal
Router(config)#interface Serial 0
Router(config-if)#fair-queue
```

7. With the WFQ mechanism configured, at the server execute the bash script (four *iperf* instances) to generate the same flows that were specified in experiment 5. Wait 3 minutes and, from the Wireshark's Conversations Statistics, estimate the network performance. Compare these results with the ones obtained in experiment 5 and justify your conclusions by having in mind the way the WFQ management discipline works.

8. Change the queuing management mechanism of the router 2 serial interface in order to consider a higher strict priority for all UDP traffic destined to port 5003 and a lower strict priority for all UDP traffic destined to port 5000. Using the `show interfaces serial 0` command in both routers, verify which are the existing queues, which is the queuing management mechanism that is being used and to what queues is it being applied. Reduce the size of each queue to a 10 times lower value.

Configuration of the priority queueing discipline

In order to configure an interface with strict priority, first you have to create priority lists in global configuration mode by using the priority-list command:

```
Router#configure terminal
Router(config)#priority-list 1 protocol ip low udp 5000
Router(config)#priority-list 1 queue-limit 2 4 6 8
```

Then, the priority-group command must be used to activate the previously configured priority lists. For example, at the Serial 0 interface:

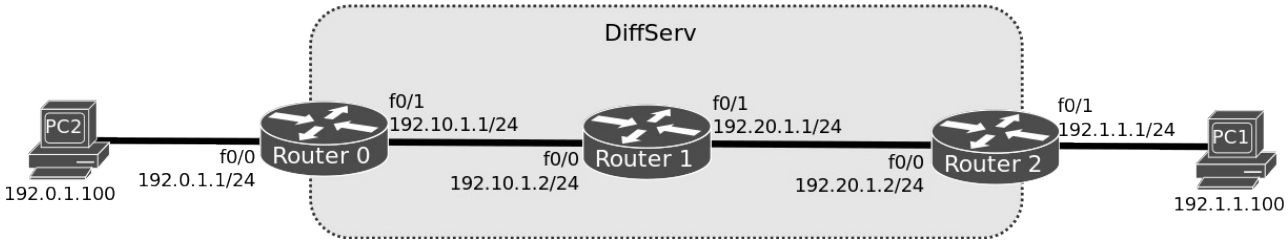
```
Router#configure terminal
Router(config)#interface Serial 0
Router(config-if)#priority-group 1
```

9. With the priority queueing mechanism still configured, at the server execute the bash script (four *iperf* instances) to generate the same flows that were specified in experiment 5. Wait 3 minutes and from the Wireshark's Conversations Statistics, estimate the network performance. Compare these results with the ones obtained in experiment 5 and justify your conclusions by having in mind the way queuing strict prioritization works.

2. Implementing DiffServ for end-to-end QoS

Set up the following network. Configure the IP addresses at the routers' interfaces and configure the necessary routing mechanisms (e.g., OSPF) in order to have full connectivity.

For simplicity reasons, traffic will be unidirectional, from PC1 to PC2.



1. Consider 4 traffic classes with the following characteristics and requirements:

- **Premium class** will be marked with a **DSCP value of 46 (EF)**.
- **Gold class** will be marked with a **DSCP value of 10 (AF11, low drop probability)**.
- **Silver class** will be marked with a **DSCP value of 22 (AF23, high drop probability)**.
- Everything else is considered as belonging to the "**best-effort**" traffic **class**.
- The premium class should be forwarded with the lowest delay possible up to a maximum of 10% of the link bandwidth during periods of congestion.
- The gold class should be treated preferentially over the silver class.
- The gold and silver classes should have 20 percent and 15 percent, respectively, of the interface bandwidth as the minimum bandwidth guarantees.
- Best-effort class should be policed to 512 kbps.

2. **Defining Classes and classifying packets**

The different services will be emulated generating traffic from PC1 to PC2 using a specific port number. In order to classify traffic in the different classes, three extended access lists (where the port numbers for the different services vary from 3001 to 3003) must be defined at Router 2:

```
Router2(config)#access-list 101 permit udp any any eq 3001
Router2(config)#access-list 102 permit udp any any eq 3002
Router2(config)#access-list 103 permit udp any any eq 3003

Router2(config)#class-map match-all EF
Router2(config-cmap)#match access-group 101
Router2(config)#class-map match-all AF11
Router2(config-cmap)#match access-group 102
Router2(config)#class-map match-all AF23
Router2(config-cmap)#match access-group 103
```

3. Configure the SETDSCP policy map in order to **mark packets** according to the previously described scheme:

```
Router2(config)#policy-map SETDSCP
Router2(config-pmap)#class EF
```

```
Router2(config-pmap-c)#set ip dscp 46
Router2(config-pmap)#class AF11
Router2(config-pmap-c)#set ip dscp 10
Router2(config-pmap)#class AF23
Router2(config-pmap-c)#set ip dscp 22
```

Apply this policy map in the **input** direction of interface f0/1 of Router 2 by using the following command:

```
service-policy input SETDSCP
```

4. Start a capture at network 192.20.1.0/24. In order to emulate the different services,

execute the following commands at PC1:

```
ping 192.0.1.100 -2 -p 3001
ping 192.0.1.100 -2 -p 3002
ping 192.0.1.100 -2 -p 3003
ping 192.0.1.100 -2 -p 3004
```

By looking at the captured packets, confirm the DSCP values that were set at the edge of the DiffServ

network.

5. In order to configure the different **per hop behavior (PHB)** match the different DSCP values to the traffic classes and create the EDGE policy map:

```
Router2(config)#class-map match-all PREMIUM
Router2(config-cmap)#match ip dscp 46
Router2(config)#class-map match-all GOLD
Router2(config-cmap)#match ip dscp 10
Router2(config)#class-map match-all SILVER
Router2(config-cmap)#match ip dscp 22
Router2(config)#class-map match-all BEST-EFFORT
Router2(config-cmap)#match ip dscp 0
---
Router2(config)#policy-map EDGE
Router2(config-pmap)#class PREMIUM
Router2(config-pmap)#priority percent 10
Router2(config-pmap)#class GOLD
Router2(config-pmap)#bandwidth percent 20
Router2(config-pmap)#class SILVER
Router2(config-pmap)#bandwidth percent 15
Router2(config-pmap)#class BEST-EFFORT
Router2(config-pmap)#police 512000 100000 100000 conform-action set-dscp-transmit 0
                    (rate limit of 512Kbps; normal burst of 100Kbytes and maximum burst of 100Kbytes)
```

Apply this policy map in the **output** direction of interface f0/0 of Router 2 by using the following command:

```
service-policy output EDGE
```

6. At the core router (Router 1), traffic classes correspond to the service classes that were already defined at the edge router.

```
Router1(config)#class-map match-all PREMIUM
```

```
Router1(config-cmap)#match ip dscp 46
Router1(config)#class-map match-all GOLD
Router1(config-cmap)#match ip dscp 10
Router1(config)#class-map match-all SILVER
Router1(config-cmap)#match ip dscp 22
Router1(config)#class-map match-all BEST-EFFORT
Router1(config-cmap)#match ip dscp 0
```

At Router 1 create the CORE policy map for prioritization, bandwidth guarantee for each class and congestion control:

```
Router1(config)#policy-map EDGE
Router1(config-pmap)#class PREMIUM
Router1(config-pmap)#priority percent 10
Router1(config-pmap)#class GOLD
Router1(config-pmap)#bandwidth percent 20
Router1(config-pmap)#class SILVER
Router1(config-pmap)#bandwidth percent 15
Router1(config-pmap)#class BEST-EFFORT
Router1(config-pmap)#police 512000 100000 100000 conform-action set-dscp-transmit 0
                                (rate limit of 512Kbps; normal burst of 100Kbytes and maximum burst of 100Kbytes)
```

Apply this policy map in the **output** direction of interface f0/0 of Router1.

Note that this is only an illustrative example of how DiffServ could be configured on a core router.

7. By using the

```
show policy-map <map_name>
show policy-map interface <interface>
show interface <interface>
```

commands, monitor and troubleshoot the operation of the DiffServ network.

8. Consider that PC1 and PC2 will have a database synchronization service that must have the lowest delay possible (up to 15% of link bandwidth). The synchronization traffic uses TCP (port 6622) and only the traffic with source in PC1 and destination to PC2 must be allocated to this traffic class.