

# A course on Quality of Service (QoS)

Jaume Barcelo

October 15, 2012



# Contents

<b>1</b>	<b>About the course</b>	<b>1</b>
1.1	Course Data . . . . .	1
1.2	Introduction . . . . .	1
1.3	Syllabus . . . . .	3
1.4	Bibliography . . . . .	5
1.5	Evaluation Criteria . . . . .	5
1.6	Survival guide . . . . .	6
1.6.1	How to pass the course . . . . .	6
1.6.2	Continuous Assessment . . . . .	6
1.6.3	Collaboration Policy . . . . .	7
1.6.4	Formula Sheet . . . . .	7
1.6.5	Questions and doubts . . . . .	8
1.6.6	Continuous feedback . . . . .	8
1.6.7	How to make you teacher happy . . . . .	9
<b>2</b>	<b>QoS metrics</b>	<b>11</b>
2.1	Delay . . . . .	12
2.2	Jitter . . . . .	13
2.3	Round Trip Delay . . . . .	13



# Chapter 1

## About the course

### 1.1 Course Data

Code: 21738

Course name: “Protocols de qualitat de servei en xarxes”

Teacher: Jaume Barcelo

Credits: 4

Year: 3rd year

Trimester: Spring

### 1.2 Introduction

This is a course on Quality of Service in data networks, which is usually abbreviated as QoS. QoS is about discriminating traffic. It is about favouring some data packets at the expense of others. The name can be misleading, as one might think that all packets are benefited from the implementation of QoS. This is not the case.

A good parallelism to understand QoS is road traffic. There

are some vehicles (typically police, ambulance and firefighters) that receive priority over the others. This is not perceived as something negative, as this vehicles incur in tasks that are more important or urgent than the average vehicle.

This parallelism is very illustrative to convey the idea, that QoS does not make the road wider, it simply prioritizes some traffic.

At some point, networks engineers may face the dilemma of investing their efforts and money in either implementing QoS (prioritizing traffic) or increasing the available bandwidth (making the road wider). The latter option has the advantage that it benefits all the traffic of the network. Ideally, if the bandwidth is sufficiently over-provisioned, the packets never have to wait in the routers queues. In the road analogy, if the roads are wide enough, there are never traffic jams.

Unfortunately, making the roads wider or the networks faster does not always solve the problem. As the users perceive that there is plenty of bandwidth available, they might decide to put it a better use by downloading collections of movies that they will never have time to see. There is nothing wrong with downloading collections of movies, but the sheer volumes of data may fill up the queues and introduce unacceptable delay and jitter in VoIP calls.

And why is this a problem? Well, at some point network engineers that deploying separate networks for each service represented too much work (and money). From an engineering and economic point of view, it is much more advantageous to offer the different services on a single network. It is common nowadays that telephony, video-conference, web, remote backup and file sharing services share the same network. The term to refer to these networks that support various services is “converged networks”. The only problem is that the different services have totally different requirements with regards to required bandwidth and delay.

The service that consume a large amount of bandwidth and do not have strict delay requirements can easily create a “traffic jam” that prevents the offering of services with low delay requirements that consume very little amount of bandwidth. Obviously, it is still possible to offer the two kinds of service simultaneously if we implement QoS mechanisms that prioritizes the low delay traffic.

QoS is a controversial topic. Net neutrality, which is related to QoS is even more controversial [1]. Nevertheless, QoS (or “bandwidth management”) has been used by ISPs in practice [2]. We will try to cover the subject from a neutral point of view and you, equipped with the knowledge of the course, will take your own decision about the usefulness of QoS.

The course is divided in three conceptually different parts: lectures, seminars and lab assignments. In lectures I will introduce you to the nuts and bolts of QoS. In the seminars, we will review the concepts of queueing theory covered in previous courses, and extend them to consider different traffic classes. In the lab assignments you will implement some QoS tools and, when possible, validate them with the methods studied in the seminars.

## 1.3 Syllabus

- Lectures

1. Introduction to QoS
2. QoS Requirements
3. QoS Service Level Agreements
4. QoS Mechanics
5. QoS Architectures
6. Deploying Diffserv

7. Capacity Admission Control
8. SLA and Network Monitoring
9. Core Capacity and Traffic Engineering

- Seminars

1. Review of basic concepts. Exponential distribution. Poisson Traffic. Little's Theorem. PASTA theorem.
2. Delay in a network interface with Poisson arrivals, a single (finite) buffer and exponential transmission time.
3. Delay in a network interface with Poisson arrivals, two traffic classes and exponential transmission time. Preemptive priority and non-preemptive priority.
4. Delay in a network interface with a general transmission time. Priority queueing in a network with a general transmission time.

- Lab Assignments

1. Program a UDP Poisson traffic generator and a traffic sink capable of computing delay (min/avg/-max). Packet drop should also be measured.
2. Program a packet buffer. It should support both exponential and deterministic transmission time. The buffer size is taken as a parameter and it may be infinite.
3. Program a buffer that implements priority queueing. It should support both exponential and deterministic transmission time. The buffer size is taken as a parameter and it may be infinite.
4. Implement a QoS tool of your choice: policer, token bucket, leaky bucket.



5. Combine the different QoS elements that you and your classmates have programmed in a QoS enabled network. Invent an scenario, describe the requirements and explain how your solution addresses such requirements.

## 1.4 Bibliography

The lecture closely follow the book:

John Evans, Clarence Filstis “Deploying IP and MPLS QoS for Multiservice Networks”.

## 1.5 Evaluation Criteria

The grading is distributed as follows:

- Lectures continuous assessment, 10%
- Seminars continuous assessment, 10%
- Blackboard problem solving, 10%
- Lab assignments, 10%
- Individual continuous assessment quiz, 10%
- Final exam, 50% (Possibility of re-take exam)

It is necessary to obtain a decent mark (4 out of 10 or 20 out of 50) in all the different evaluation aspects. To pass the course, 50 out of the total 100 points need to be obtained.

## 1.6 Survival guide

### 1.6.1 How to pass the course

Statistically speaking, you will pass the course if you do all the following:

- Attend lectures, participate and ask questions.
- Attend seminars, try to solve the problems on your own and discuss them in small groups.
- Volunteer to solve problems on the blackboard.
- Attend labs and use lab time to solve the lab assignments.
- Participate in the planning and coding of the labs assignments. Read carefully the code of other team members and make sure you understand it and can explain it to others.
- Study for the continuous assessment quiz, as it is a warming up exercise to face the final exams with success guarantees.

### 1.6.2 Continuous Assessment

In this course we implement continuous assessment. This means that if you work hard from day zero, the course will be pain-free.

Continuous assessment includes multiple-choice quizzes in lectures and seminars. You also have to write reports in labs (one for each group). The source code and the report for the labs is submitted via moodle. Remember to write always your name and NIA in all the material you hand in or upload to

moodle. You also have to include your name and NIA in all the source code files you send me.

There is not a template for the report. I recommend describing key design and implementation aspects, clarifying possible deviations or improvements on the initial assignment, and including examples. The example should include the input commands as well as the output results. The report and the code will be submitted at the end of the class via moodle. At the beginning of the next class, you will have to demonstrate that your programs actually work.

### 1.6.3 Collaboration Policy

You are encouraged to collaborate with other students in the resolution of problems and assignments. However, you should first try to solve it on your own. Then, you can discuss your solution with others and work together to find a better solution. Finally, you must ensure that you can solve the problem or assignment alone.

In the labs assignments you will work in teams of three people. Unless when explicit permission is given to re-use code, each team has to write their own code.

### 1.6.4 Formula Sheet

This course is not about memorizing equations. It is about understanding them. For this reason, you are allowed to use a *formula sheet* in individual tests as long as it fulfills the following requirements:

- It is a single page (one side).
- It is handwritten. Your own handwriting.

- It is delivered together with your answers when the test is finished.

### 1.6.5 Questions and doubts

I like to receive questions and comments. Normally, the best moment to express a doubt is during the class, as it is likely that many people in the class share the same doubt. If you feel that you have a question that needs to be discussed privately, we can discuss it right after the class.

### 1.6.6 Continuous feedback

At the end of the class, I will ask you to anonymously provide some feedback on the course. In particular, I always want to know:

- What is the most interesting thing we have seen in class.
- What is the most confusing thing in the class.
- Any other comment you may want to add.

In my previous experience, this information has proven to be invaluable in improving the course, detecting problems at an early stage, and adapting the course to the expectations of the students.

In labs, I will ask each group to hand in a short (few paragraphs) description of the work carried out in class, and the members of the group that have attended the class. Note that this is different from the lab report, which is the one that it is actually graded.

### 1.6.7 How to make you teacher happy

Avoid speaking while I am talking. It is not that you cannot talk in class. You can talk as much as you want when I am silent. I will make plenty of breaks in which I will ask you to discuss a question with your classmates. You can also take advantage of the moments in which I erase the blackboard or just scratch my head while staring. As long as I am not talking, you can talk with your classmates as much as you want. Obviously, questions are welcome at any time.



# Chapter 2

## QoS metrics

If you go to the postal office to submit a packet, you will be offered different options. In addition to the regular service, it is possible that an urgent service exists. Probably there is also the possibility sending the packet as certified mail, and some option for delivery notification. It is likely that there are also special services for packets that are voluminous or heavy.

QoS-enabled packet switched networks also offer different kinds of services for data packet delivery. In this chapter, we will review the different metrics that are relevant for data networks. These metrics can be used to establish *service level agreements* (SLAs) which are contracts specifying the QoS expected from a network. These contracts should also specify how the metrics are actually measured.

As an example, if a network guarantees a delay below 100 ms, it should be specified whether this makes reference to the maximum, the delay or the 95% percentile. The measures will also differ depending whether 5 minutes averages or 1 hour averages are considered. This makes the specification of SLAs tricky.

## 2.1 Delay

Delay is the time that it is required to traverse the network from the entry point to the exit point. Delay is normally considered for real-time services such as voice over IP (VoIP). The total end-to-end delay is simply the sum of the delay suffered in each of the hops in the data network. As an example, 2.1 shows a network with four hops.

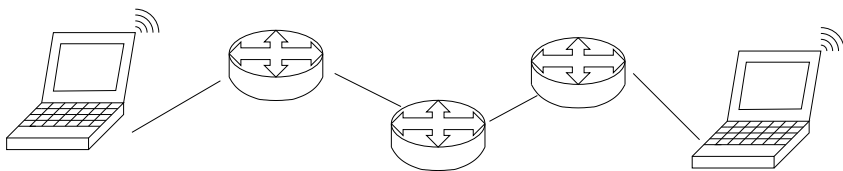


Figure 2.1: A network with two terminals, three routers and four hops.

In each hop, there are four different contributions to delay:

1. Processing: The time required for the router or switching device to put the packet on the outgoing interface queue. Very short.
2. Queueing: Waiting time on the outgoing interface queue. Very short if the queue is empty.
3. Transmission: The time required to put the packet on the transmission medium. It is a function of the packet length and transmission rate. Short in high-speed transmission media.
4. Propagation: The time that it takes for the packet to travel the distance from the hop source to the hop destination. Short time over short distances. And very long when it involves a trip to a geostationary satellite.



## 2.2 Jitter

Jitter is the variation of delay. This aspect is specially relevant for real-time and streaming applications. These applications expect the packets to arrive regularly in time. As an example, if the encoder application takes a voice stream and splits it into 20 ms chunks that are encoded and sent as packets, the receiver application will expect to receive one packet every 20 ms to reconstruct the voice stream.

If packets are sent every 20 ms but each of them requires a different time to traverse the network, the separation between packets will no longer be 20 ms at the receiving end. Applications sensitive to jitter use a de-jittering buffer that holds some packets and feeds the decoder at regular intervals. The packets that suffered a short delay in the network will wait for a longer time in the de-jitter buffer and the packets that suffered a longer delay in the network stay in the buffer for a shorter time. With this technique, jitter is effectively suppressed at the expense of increasing delay, as shown in figure 2.2.



Figure 2.2: The de-jitter buffer removes jitter at the expense of adding delay.

## 2.3 Round Trip Delay

The round trip delay is the measure of delay used for elastic and interactive applications. It is the time required for a packet from source to destination and then back to the source again. You can easily find the round trip delay from your host using the ping command.

```
$ ping www.happyforecast.com
PING happyforecast.com (184.107.100.65)
  56(84) bytes of data.
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=1 ttl=48 time
    =122 ms
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=2 ttl=48 time
    =121 ms
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=3 ttl=48 time
    =122 ms
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=4 ttl=48 time
    =122 ms
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=5 ttl=48 time
    =122 ms
64 bytes from s106.panelboxmanager.com
  (184.107.100.65): icmp_req=6 ttl=48 time
    =121 ms
^C
— happyforecast.com ping statistics —
6 packets transmitted, 6 received, 0% packet
  loss, time 5004ms
rtt min/avg/max/mdev =
  121.864/122.066/122.181/0.230 ms
```

An interactive application (such as web browsing) requires a few round trip delays to complete the dialogue between the client and the server. Furthermore, the round trip delay also limits how fast the TCP congestion window can grow. The congestion window grows upon the reception of TCP acknowledgements. If it takes long for the TCP acks to arrive, the congestion window grows slowly.

---

In fact, it becomes difficult to take fully advantage of connections with a high bandwidth x delay product. These are known as “long fat pipes” or “long fat networks” (which is shortened as LFN and sometimes pronounced elephant), as explained in RFC 1072 [3].



# Bibliography

- [1] Gari R. Bachula. Testimony of Gary R. Bachula, Vice President, Internet2.
- [2] A. Cooper, A. Jacquet, and A. Soppera. Bandwidth Usage and Management: A UK Case Study. In *Research Conference on Communications, Information and Internet Policy (TPRC)*, 2011.
- [3] V. Jacobson and R.T. Braden. TCP extensions for long-delay paths. RFC 1072 (Historic), October 1988. Obsoleted by RFCs 1323, 2018, 6247.