UNIVERSITAT POLITÈCNICA DE CATALUNYA

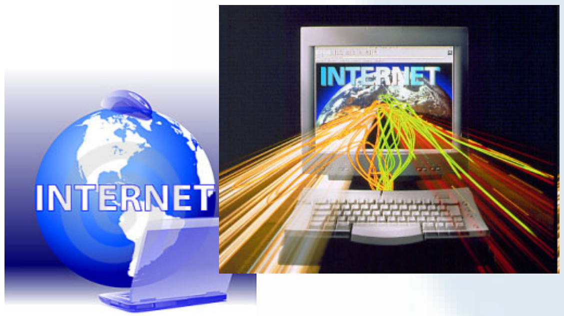
Departamento de Ingeniería Telemática

Ingeniería de Calidad de Servicio en redes IP

Mónica Aguilar Igartua
monica.aguilar@upc.edu **http://www-entel.upc.edu/monica.aguilar/**

QoS parameters for multimedia services

- Multimedia applications have different Quality of Service (QoS) requirements:
    - **Video-on-Demand (VoD) or video-streaming**: moderate end-to-end delay, **high throughput**, low error rate.
    - **Internet Telephony**: **very low end-to-end delay**, moderate throughput, moderate error rate.



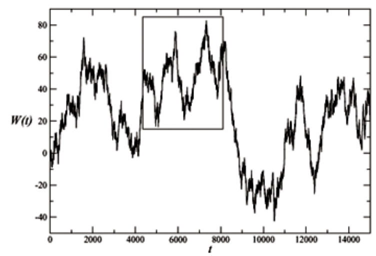
Departamento de Ingeniería Telemática

How a network can guarantee different levels of QoS?

1. Admission control → Are there enough resources?
2. Resource reservation → Guarantee my QoS.
3. Packet classification → Store each packet in the proper queue.
4. Schedulers → Define a policy to visit the queues and serve packets.
5. Congestion control → How do we drop packets from full queues?
6. Flow control → How sources can react and slow down their data transmission rate upon congestion?
7. QoS → Performance evaluation: losses, delay, bandwidth, QoE (Quality of Experience)

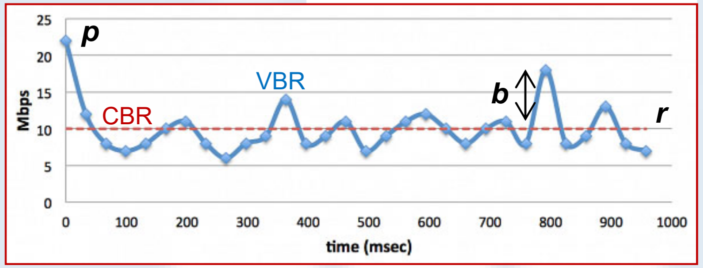
1. Admission control

We need to describe the traffic features→ bandwidth needed along time?



Basic types of traffic sources used in multimedia applications:

- **CBR** (Constant Bit Rate). E.g. PCM (Pulse Code Modulation) coded voice generates traffic at 64 Kbps.
- **VBR** (Variable Bit Rate). E.g. MPEG-VBR coded video, with either high scene changes (high bit rate) as well as speaker scenes (low bit rate).



Basic traffic parameters:

- **Peak rate (*p*)**. Maximum data rate in any time interval. CBR is completely defined with it.
- **Average rate (*r*)**. Mean of the traffic rate. VBR uses it.
- **Burst size (*b*)**. Number of packets that can be delivered under the peak rate. VBR is *bursty* (it generates traffic in bursts).

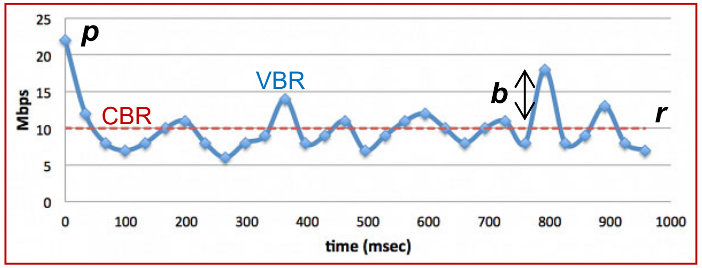
Departamento de Ingeniería Telemática

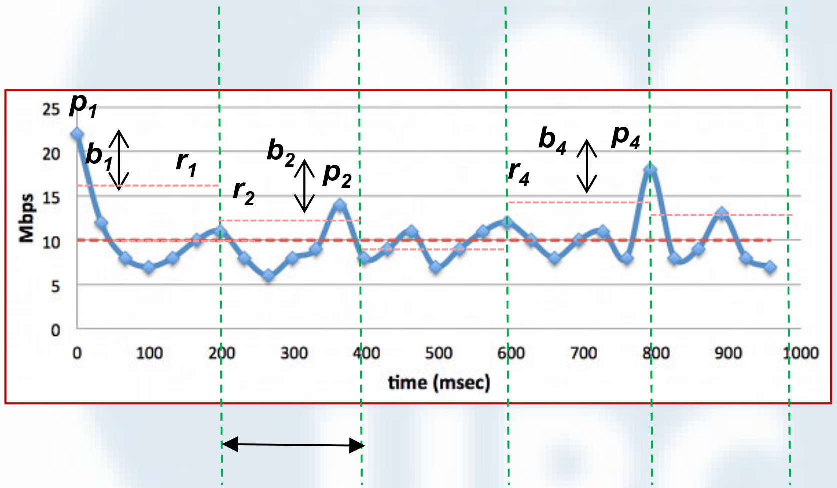
2. Resource reservation

How can we reserve **resources**? Bandwidth and memory space

Using a protocol, e.g. RSVP (Resource Reservation Protocol) → (*p*, *r*, *b*)

Instead of making a single reservation, it could be more suitable to arrange a dynamic and flexible scheme:

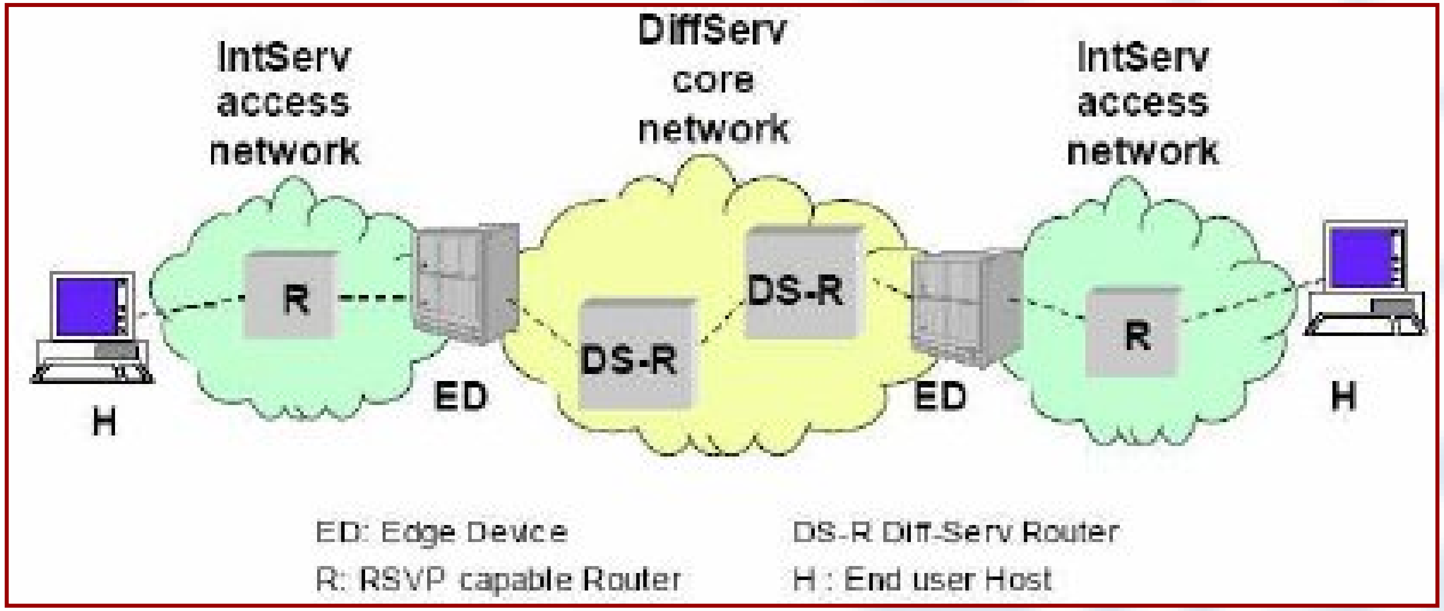




n·GoP (Group of Pictures)=12·250ms=3s, for instance

3. Packet classification

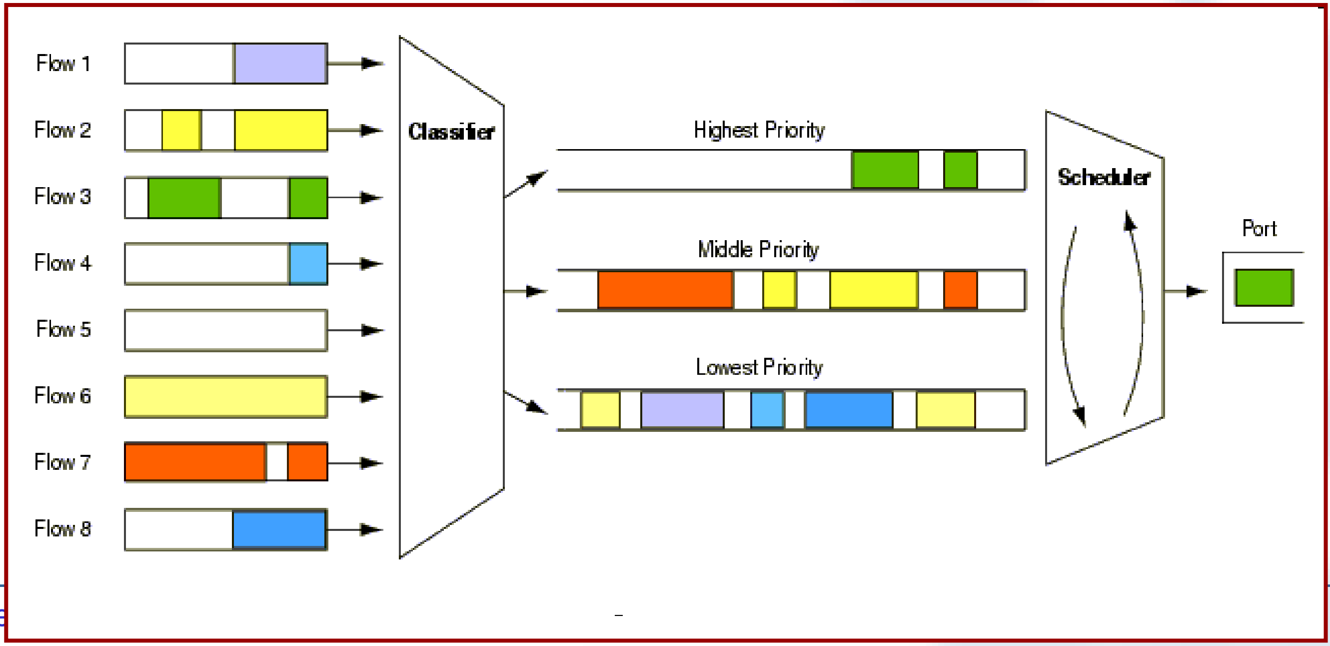
- Packets must be classified at the entrance of each hop of the network
  - **IntServ**: @IP_source, @IP_destination, #port_source, #port_destination, #protocol → Stream of packets of same flow
  - **Diffserv**: short label with local validity at each hop



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

3. Packet classification

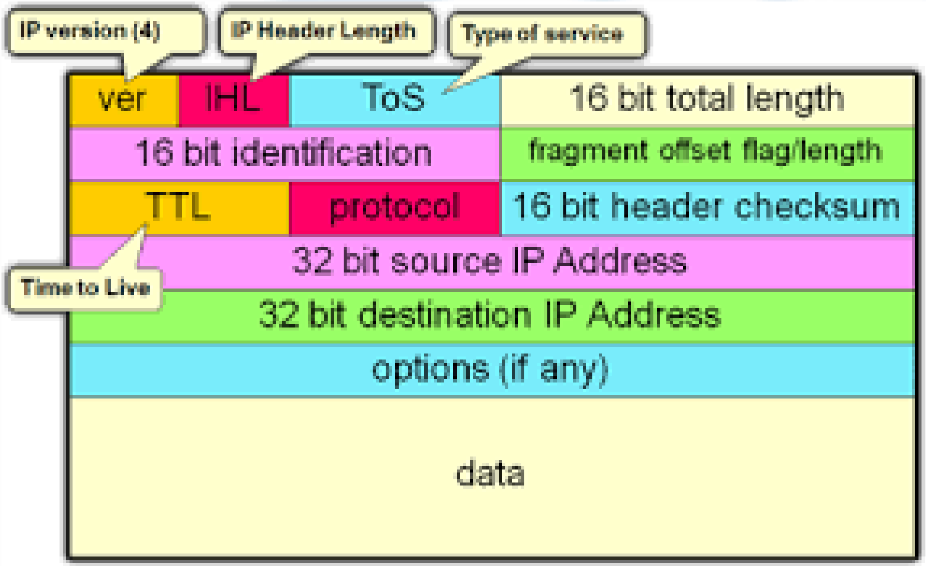
- DiffServ scales well in the Internet backbone
- Short Labels are appended in the packets at the entrance of each DiffServ hop (e.g., **Gold, Silver, Bronze**)
- Packets are stored in the proper queue → priority
- Packets are treated with a specific QoS in the hop → priority



3. Packet classification

- To treat differently the packets, we need to use **multiple queues** with different **priorities**.
  - The ToS (Type of Service) field of IPv4 supports 5 different services.
  - Sources may use this field to request a specific treatment from the network.

ToS	Service
1000	Minimum delay
0100	Maximum throughput
0010	Maximum reliability
0001	Minimum cost
0000	Normal service



4. Queue schedulers

- **Objectives** of the queue scheduling algorithms to support QoS in packet switched networks (e.g., the Internet):
  - Bandwidth Sharing (BW)
  - Provide fairness to the competitor streams
  - Provide different QoS levels if required
  - Guarantee minimum and maximum BW
  - Guarantee loss bounds
  - Guarantee delay bounds
  - Reduce jitter delay

4.1. Queue scheduling objectives

- If packets arrive at the exit-queue of the server faster than they can be served, they will wait in a queue.
- If the queue has a limited capacity, packets may be dropped.
- The order in which the queue scheduler serves the packets, has an influence in the packet loss rate and the delay.
- The queue scheduler should guarantee a certain QoS (max delay, max losses, min bandwidth…).

4.2. MAX-MIN algorithm

- MAX-MIN Algorithm to distribute resources fairly:
  - Distributes the resources in a fair way between streams that compete for the same resources.
  - Firstly, this scheme allocates the **smallest demand** for all the flows. **Remaining** resources are distributed **equally** among competing flows. And this is iterated successively.
  - Let's K be the number of competing flows. Each flow demands x_1, x_2, …, x_k resources (assuming that $x_1 \leq x_2 \leq \ldots \leq x_k$) and the total available resources is R.
  - Initially, the flow with lowest demand *(i.e.* x_1) gets R/K resources.
  - If $R/K>x_1$, the remaining $(R/K)-x_1$ is distributed between the K-1 remaining streams.

4.2. MAX-MIN algorithm

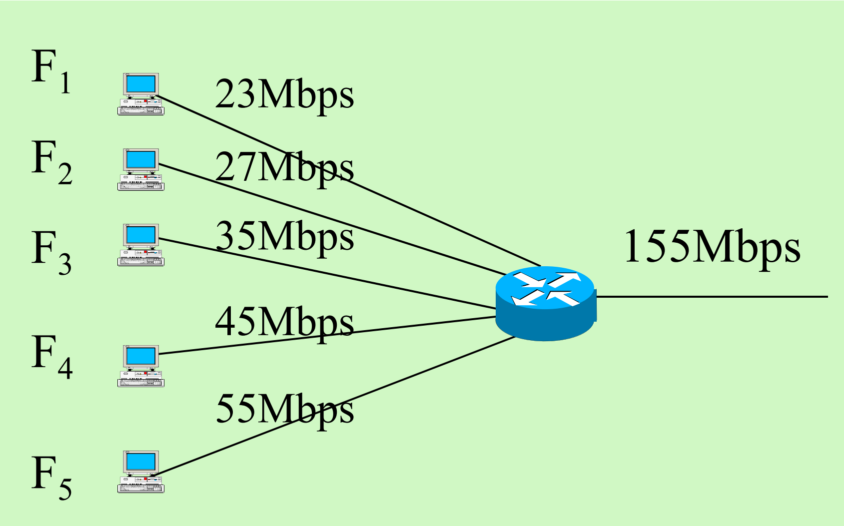
- The *K-1* remaining streams receive these amount of resources:

$$\frac{R}{K}+\frac{\left(R/K\right)-x_1}{K-1}$$

- The process iterates until either all resource requests have been met or until all available resources are finished.
- This algorithm guarantees that a flow either gets **what it wants** or **at least it doesn't get a worse amount** than any other competing flow.
- Let's see an example.

Departamento de Ingeniería Telemática

4.2. MAX-MIN algorithm



Final assignation:
F_1 = 23 Mbps
F_2 = 27 Mbps
F_3 = 35 Mbps
F_4= 35 Mbps
F_5= 35 Mbps

Distribute the resources fairly:

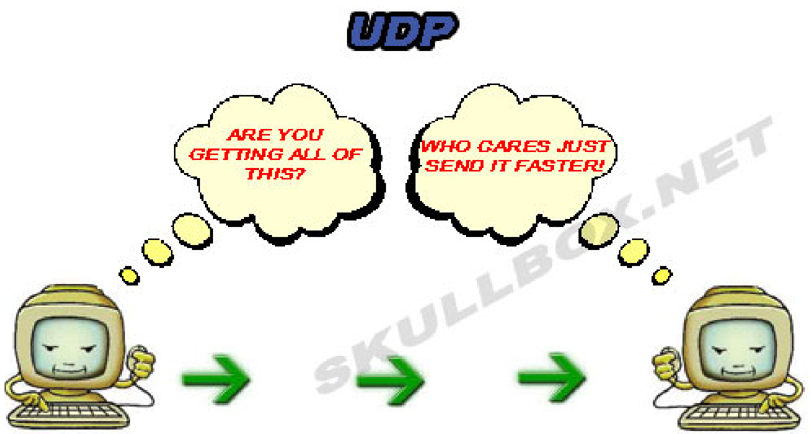
- 155/5=31 Mbps for F_1, but there remain more available resources:
- 31-23=8 → 31+8/4=33 Mbps for F_2, but there still are remaining resources:
- 33-27=6 → 33+6/3=35 Mbps for F_3, and there are no more available resources. So that,
- 35 Mbps for F_4 and F_5.

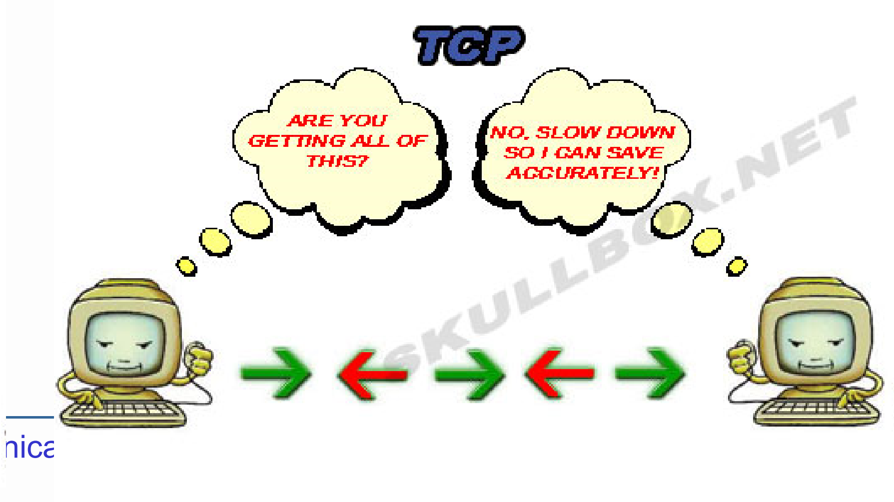
4.3. Queue scheduling techniques

- The most used algorithms are:
  - First In First Out (FIFO)
  - Priority Queuing (PQ)
  - Round Robin (RR)
  - Weighted Round Robin (WRR)

First In First Out (FIFO)

- Packets of all the streams arrive to one common queue, served **from the head** of the queue.
- If a packet arrives when the queue is full, it is **dropped**.
- It is **not a fair** scheduler scheme.
  - A **greedy source** (e.g. **UDP**) can occupy most of the queue and cause delay to other flows that use the same queue.
  - Congestion sensitive flows (e.g. **TCP**) that reduce their sending rates in case of congestion, will be **penalized** in front of **UDP** flows (with no congestion control).





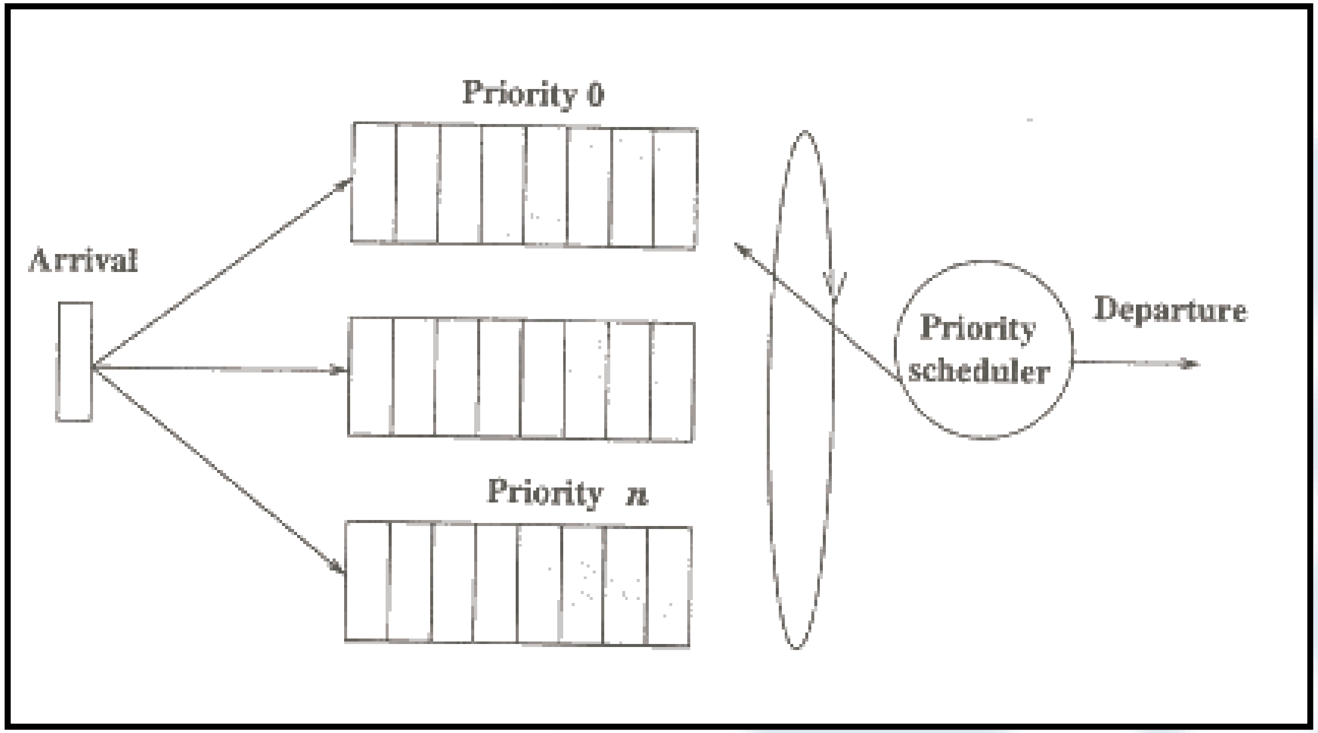
First In First Out (FIFO)

- **Simple** technique usually used in packet switching networks.
- Using a **single queue** with an FIFO scheduler, there are limitations to support QoS:
  - The streams cannot be isolated using a single queue.
  - It is very difficult to guarantee delay bounds or bandwidth guarantees to a specific flow.
  - If **different services** are required, **we need multiple queues** to separate the flows.

Priority Queuing (PQ)

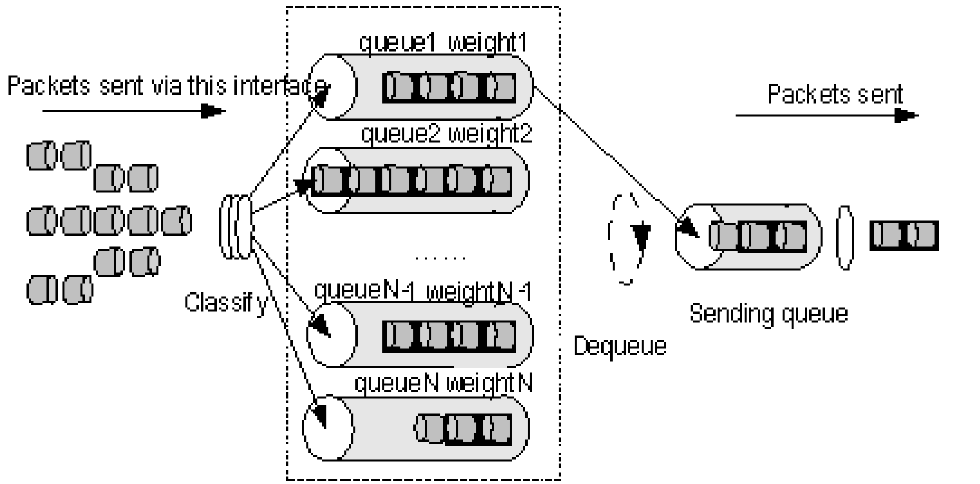
- The scheduler serves packets from the **highest priority queue** before it attends to a lower priority queue.
- Packets of the **lowest priority queue** are served only when the rest of the higher priority queues are empty.
- The **highest priority queue** has the **lowest delay**, **highest throughput**, and **lowest loss**.
- PQ is a simple algorithm that takes into account priorities, **but it only guarantees the QoS of the most priority queue.**

Priority Queuing (PQ)



Round Robin (RR)

- It maintains 1 queue per flow or class, so that every packet is stored in the proper queue.
- Queues are served in a *round-robin* way, **serving one packet** from each nonempty queue **in every turn**.
- This scheme is "**fair**" because each nonempty flow sends exactly one packet per cycle.



Round Robin (RR)

- There is **no advantage** for a **greedy source**:
  - Only its queue becomes longer, increasing its own delay.
  - Other flows are not affected by its behaviour.
- However, a **fair** resource allocation would be obtained only if all packets had the **same length**.
- But **if packet sizes are variable**, (like in the Internet) there is a fairness problem:
  - **If a queue has larger packets** than the others, the *round-robin* scheduler will be most of the time serving the large-packets queue!!
- RR does **not** distinguish between **different QoS**, because it tries to allocate **resources fairly** among all the queues.

Weighted Round Robin (WRR)

- It is a simple modification of the RR scheduler.
- Instead of serving 1 packet per queue in every turn, **it serves n_i packets** → the n_i value is chosen to allocate a specific fraction of the link bandwidth to the *i-th* queue.
- Every queue has assigned a **weight** = the fraction of bandwidth it will receive.
- The **number of packets** to be served in one turn is **calculated** from the assigned **weight**.
- To **avoid the unfairness** problem due to different packet lengths, the **mean packet sizes** are also considered.
- Let's see one example:

Weighted Round Robin (WRR)

- Assume 3 links that share the same outgoing link. These links have weights of 0.33, 0.66 and 1.0, respectively:
  - a link with MTU (*Maximum Transfer Unit*) of 500 bytes
  - an *Ethernet* network (MTU=1500 bytes)
  - an FDDI (*Fiber Distributed Data Interface*) network (MTU=4500 bytes)
  - Calculate the weights normalized to their respective packet sizes.

Weighted Round Robin (WRR)

- The weights of the three links keep these relations:

*2 *3

0.33 ---- 0.66 ---- 1.0

- Number of packets per cycle that are served from each link, keeping the same above proportion:

MTU $MTU_{ETHERNET}$ MTU_{FDDI}

?

bytes/packet

packets/cycle

bytes/cycle

Solution here…

- ? packets from the first link, ? from the second and ? from the third will be served, in every cycle of the WRR algorithm.

Weighted Round Robin (WRR)

- The weights of the three links keep these relations:

$$0.33 \text{ ---- } \underbrace{0.66}_{\cdot 2} \text{ ---- } \underbrace{1.0}_{\cdot 3}$$

- Number of packets per cycle that are served from each link, keeping the same above proportion:

MTU	$MTU_{ETHERNET}$	MTU_{FDDI}	
500	1500	4500	bytes/packet
$\cdot a \downarrow$	$\cdot b \downarrow$	$\cdot c \downarrow$	packets/cycle
$500 \cdot a$	$1500 \cdot b$ (*2)	$4500 \cdot c$ (*3)	**bytes/cycle**

$$500 \cdot a \cdot 2 = 1500 \cdot b$$
$$500 \cdot a \cdot 3 = 4500 \cdot c$$

$$a \cdot 2 = 3 \cdot b$$
$$a \cdot 1 = 3 \cdot c$$

$$a = 3$$
$$b = 2$$
$$c = 1$$

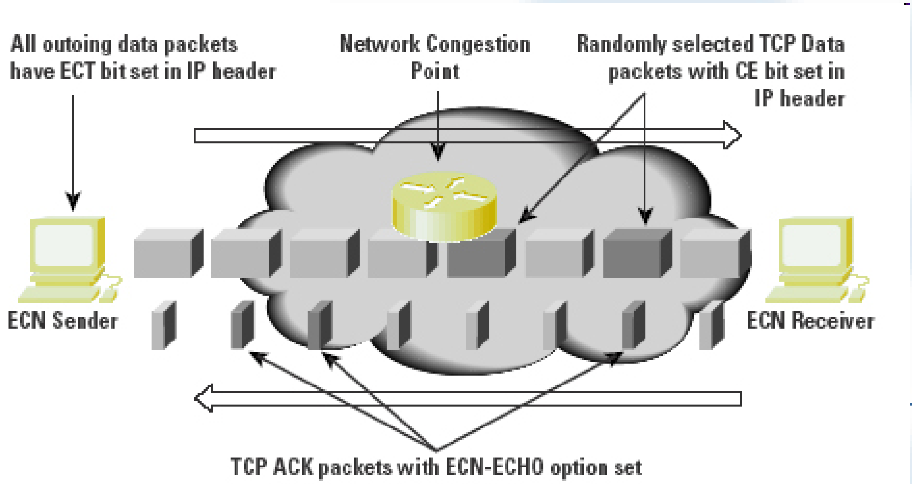
- 3 packets from the first link, 2 from the second and 1 from the third will be served, in every cycle of the WRR algorithm.

5. Congestion Control

- Basic algorithms and queue schemes:
  - Explicit Congestion Notification (ECN)
  - Packet drop tail scheme
  - Random Early Detection (RED)
  - Weighted Random Early Detection (WRED)

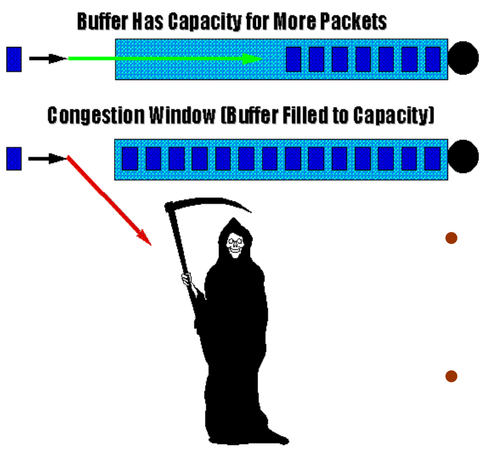
Explicit Congestion Notification (ECN)

- RFC-2481 proposed this scheme in 1999 about signaling congestion to senders.
- It uses 2 bits from the IPv4 ToS field.
  - **ECT** (ECN Capable Transport). End systems indicate if they are capable of ECN or not.
  - **CE** (Congestion Experienced). Network elements (e.g. routers) set this flag (i.e. 1) when local congestion is experienced.
  - A congested network element could set the CE flag only if the ECT bit is set. Otherwise, it may choose to drop the packet.



Packet drop tail scheme

- Packet dropping helps to avoid or to reduce congestion.
- **Drop-tail scheme**: It is easy to implement. If a queue is full upon the arrival of a packet, the **new** arrived packet **is dropped**.
- **PROBLEMS**:
  - **Lockout**: One or more flows could monopolize the queue space on a router.
  - **Full queues**: Queues could be maintained nearly full. The tail drop signals congestion only when the queue is full and packets start getting dropped → congestion could continue for a long period of time.
  - It seems more fair to **distribute packet losses randomly** in the queue → losses distributed among several flows.
  - However, it is more complex to implement...

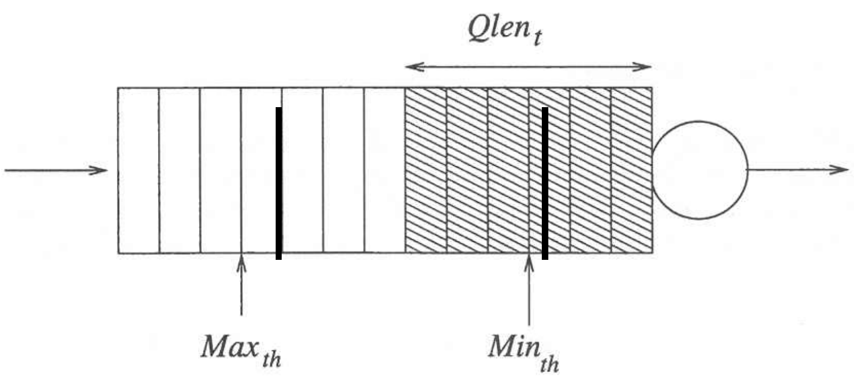


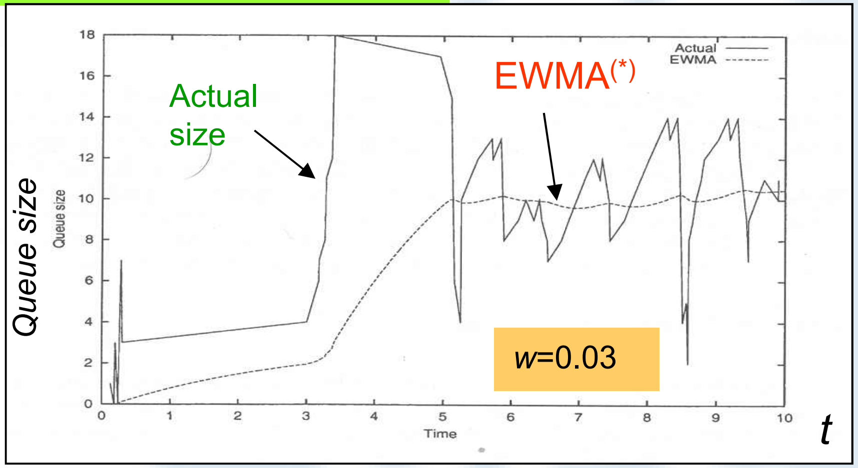
(*) **Exponentially Weighted Moving Average**

Random Early Detection (RED)

- S. Floyd, V. Jacobson (1993). It drops packets **randomly** from the active flows.
- RED monitors the queue length.
  - *Qlen*$_t$ is the queue length at instant *t*.
  - RED has two thresholds: Max_{th}, Min_{th}
  - Average value of queue length at instant *t*, $AvQlen_t$

$$AvQlen_t = (1-\varpi)\cdot AvQlen_{t-1} + \varpi \cdot Qlen_t$$

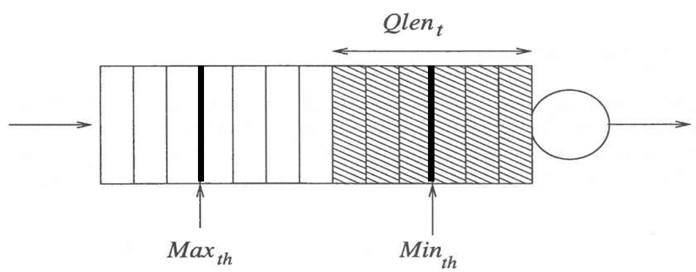




Random Early Detection (RED)

- Packet drop probability, *p*
  - If $AvQlen_t < Min_{th}$, packet is buffered. No congestion (p=0).
  - If $AvQlen_t > Max_{th}$, packet is dropped. High congestion (p=1).
  - If $Min_{th} \leq AvQlen_t < Max_{th}$, packet is dropped with probability p.

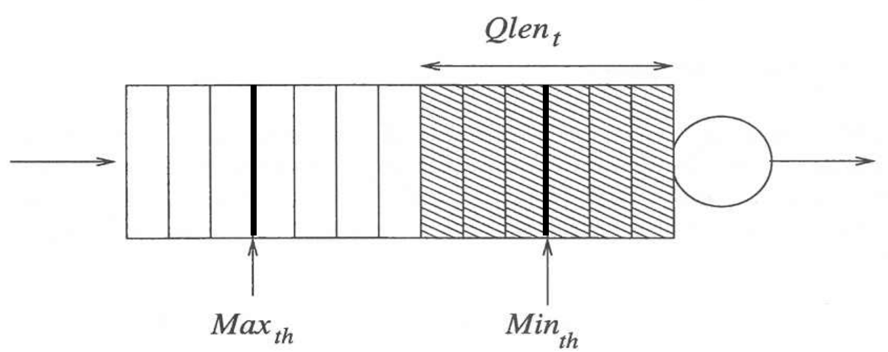
$$p = Max_p \cdot \frac{AvQlen_t - Min_{th}}{Max_{th} - Min_{th}}$$



Random Early Detection (RED)

- RED performs **Congestion Avoidance**.
- RED starts dropping packets before the queue gets full.
- **Adaptive flows** get the **congestion notification early**.
- But RED does **not support different services**, where packets should be discarded selectively: **WRED does!**

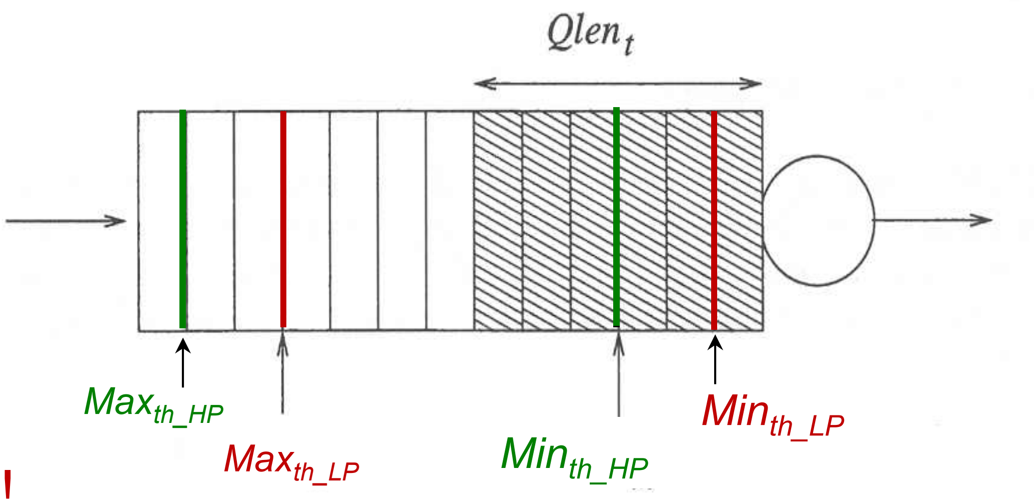
$$p = Max_p \cdot \frac{AvQlen_t - Min_{th}}{Max_{th} - Min_{th}}$$



Weighted Random Early Detection (WRED)

- Packets are market with different drop levels.
- Packets with higher drop probability will be dropped first.
- WRED maintains a different set {Max_{th}, Min_{th}, Max_p} for each traffic class.

$$\cancel{p = Max_p \cdot \frac{AvQlen_t - Min_{th}}{Max_{th} - Min_{th}}}$$



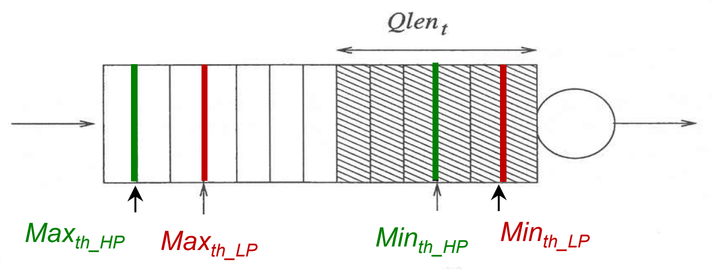
We need 2 new expressions!!

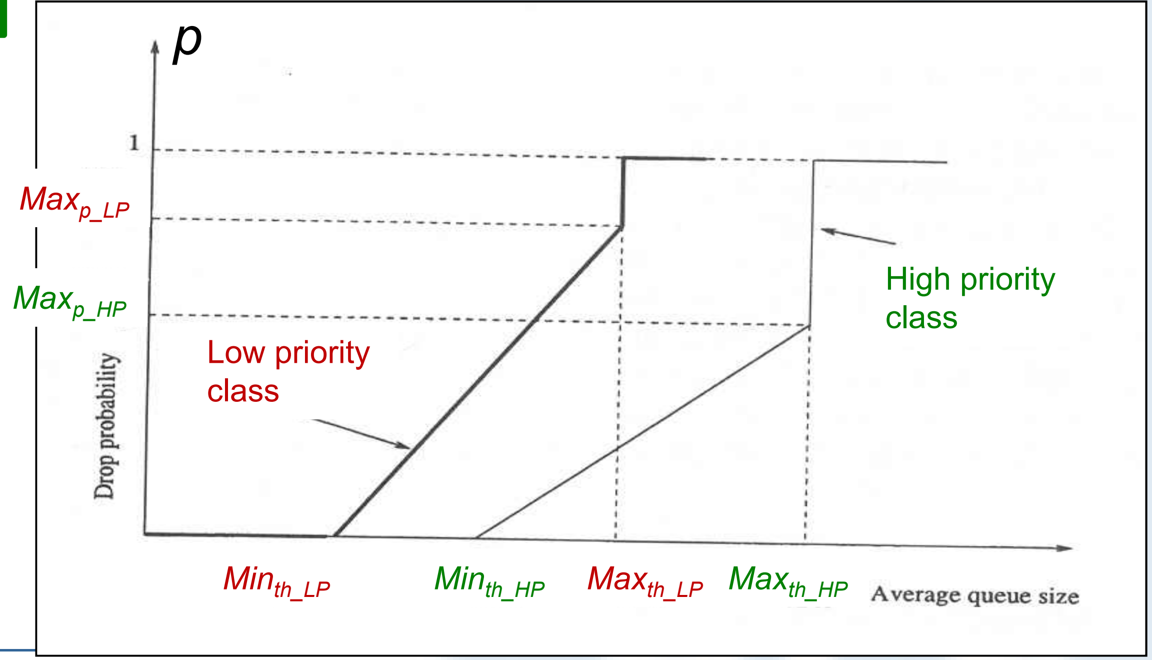
Weighted Random Early Detection (WRED)

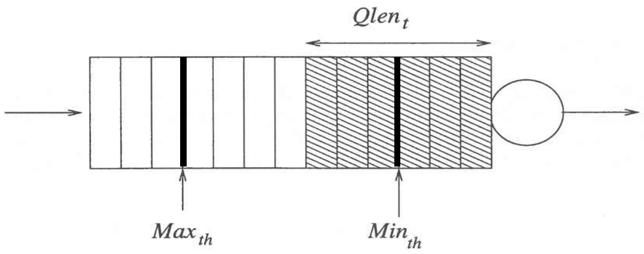
$$\cancel{p = Max_p \cdot \frac{AvQlen_t - Min_{th}}{Max_{th} - Min_{th}}}$$

$$p_{HP} = Max_{p\_HP} \cdot \frac{AvQlen_t - Min_{th\_HP}}{Max_{th\_HP} - Min_{th\_HP}}$$

$$p_{LP} = Max_{p\_LP} \cdot \frac{AvQlen_t - Min_{th\_LP}}{Max_{th\_LP} - Min_{th\_LP}}$$





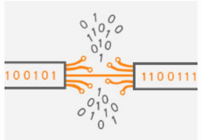


6. Flow Control

- Using a congestion control scheme (ECN, RED…), routers can warn sources of a congestion situation
  - Sources slow down their data transmission rate to alleviate the congested network situation.
  - **Reactive measures**: Sources react upon congestion notification
  - **Proactive measures**: Sources react once warned of network getting congested soon.
- Once the network recovers from congestion, sources can increase their data transmission rates.

7. Quality of Service (QoS)

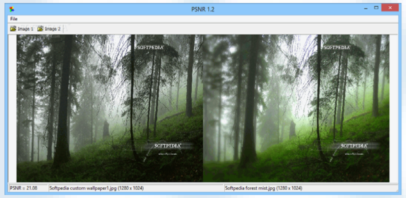
- Performance Evaluation. Parameters measured:
  - Objective measures:



    - % of packet losses
    - Average end-to-end packet delay (sec)

    - Throughput (Byes/sec)

    - Peak Signal to Noise Ratio (PSNR)



    - …
  - Subjective measures:

    - Mean Opinion Score (MOS)
    - Quality of Experience (QoE)

    - …

7. Quality of Service (QoS)

- **Peak-Signal-to-Noise Ratio (PSNR)**

$$PSNR(X,Y) = 10 \cdot \log_{10} \frac{255^2}{\frac{1}{M \cdot N} \sum_{j=1}^{N} \sum_{i=1}^{M} (x_{ij} - y_{ij})^2}$$

for 8 bits/pixel

- Mean Opinion Score (MOS)

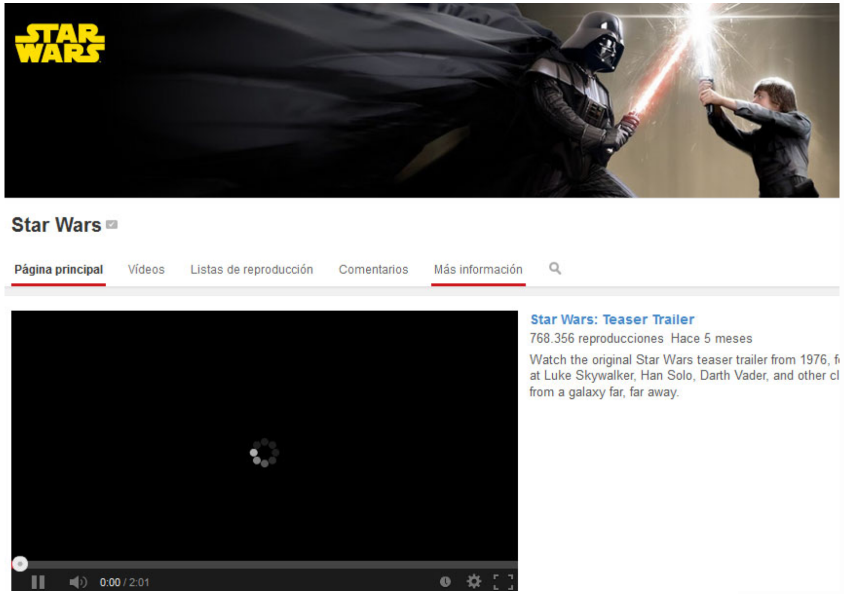
  - The MOS is an **average quality score** over a **large set of subjects**.
  - The MOS is the arithmetic mean of all the individual scores, and can range from **1 (worst) to 5 (best).**

Mean opinion score (MOS)

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

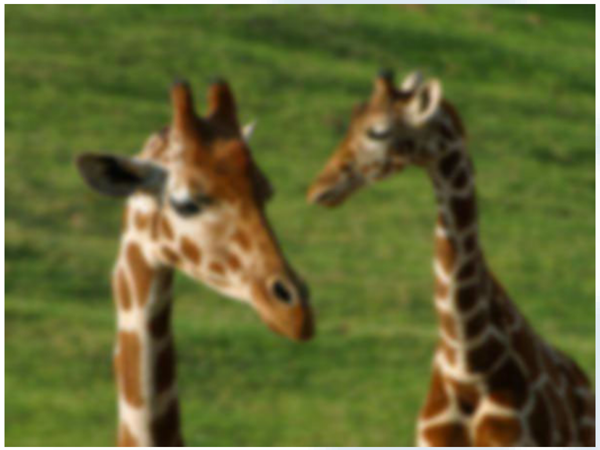
Quality of Experience (QoE)

- **Response time**: It is the time elapsed from the moment when a user sends a request and the moment of **receiving the content**. For instance, the time it takes for a video to be reproduced since it was requested.



Quality of Experience (QoE)

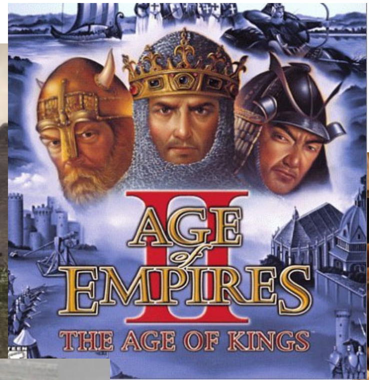
- Usually, the **QoE is a referenceless measure**.
  - The **user does not have** access to **the original video** to compare the received video. Customers do not know the original content.
  - However, one can detect effects such as **blockiness, blur or jerkiness** directly in the received video.



Perception of individual still images in a motion picture. In conventional cinematography, the images are filmed and displayed at 24 frames per second, at which speed **jerkiness** (***entrecortado***) is not normally discernible.

Quality of Experience (QoE)

- **Delivery Synchronization**: When there are many users that interact with a common service (e.g., **Online Games**), content must be received by all the users **at the same time**.



China's online gaming industry generated record high revenues of $5 billion in 2010!!!

megajocs.com

http://www.megajocs.com/

Quality of Experience (QoE)

- **Freshness**: Metric that accounts for the **time elapsed since a content was generated till it is received** by the user. It is important in **live video streaming**, since users want to receive the content as soon as possible.

Many platforms let anyone with an Internet connection and a camera to broadcast live their events to their own viewers. One can stream ***live video events*** *such as weddings, corporate events, concerts..*

Online video live streaming platforms:

USTREAM

http://www.ustream.tv/

twitcam by livestream

http://twitcam.livestream.com/

StreamingVideoProvider

streamhoster

Quality of Experience (QoE)

- **Blocking**: It specifies how much fluid the video stream is reproduced.
  - Due to the occupation of the buffers at the receiver.
  - If the **buffer** becomes **empty** and the required packet is not available in the buffer at the receiver, then the reproduction of the **video stops**.
  - It is very **annoying** for the user…

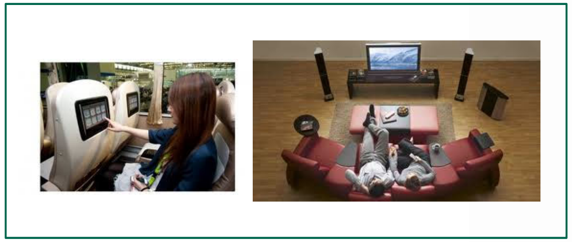
Learn the Alphabet with Peppa Pig!

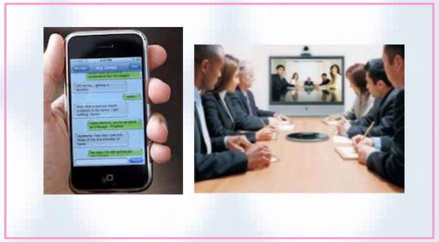
Star Wars Episode I: The Phantom Menace - Trailer

Quality of Experience (QoE)

• QoE measures **human expectations**, **feelings**, perceptions and **satisfaction** with respect to a particular product, service or application.

• QoE is influenced by the user's terminal device (for example Low Definition TV *vs.* High Definition TV), the environment (in the car *vs.* at home), his/her kind of telephone (cellular *vs.* corded), the nature of the content and its importance (casual texting *vs.* critical videoconference communications).





Although QoE is subjective, it is **the only measure that really counts for the customer** of a service

UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Departamento de Ingeniería Telemática

Ingeniería de Calidad de Servicio en redes IP

Muchas gracias por vuestra atención

Mónica Aguilar Igartua
monica.aguilar@upc.edu **http://www-entel.upc.edu/monica.aguilar/**