4.2 Quality of Service in the Network

4.2.1 General Considerations for QoS in Networks

A fundamental rule

Continuous media require Quality-of-Service support in the network.

Idea: Service Level Agreement

A contract between the application and the network.

- The **source** specifies the traffic it will generate and promises to conform to that specification.
- The **network** promises the transmission of this traffic with guaranteed QoS.

QoS Parameters

Traffic description of the source

- Type of traffic: CBR, VBR, UBR, ...
- For constant traffic: bit rate
- For bursty traffic: average bit rate, peak bit rate, maximum duration of peaks

QoS parameters of the network

- Delay
- Delay jitter (variance of the delay)
- Maximum loss rate

Real-World Examples

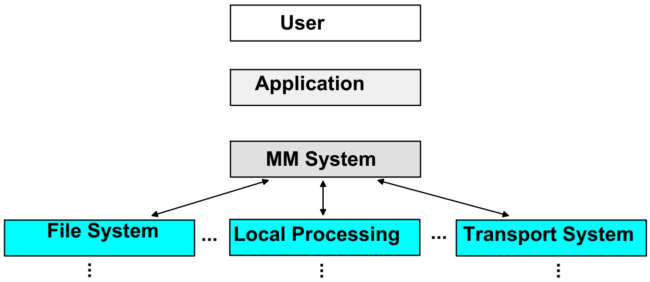
How resources in the network influence QoS

Three examples:

- A large playout buffer at the receiver (e.g., for video) allows to compensate more delay jitter, but it will increase the end-to-end delay.
- With increasing load of a router, the average waiting time of packets in the queues will increase. New connections through that router should only be accepted when delay guarantees given to the existing connections can still be maintained (Connection Acceptance Control).
- The CPU power of a router determines the maximum number of packets that can be handled per time interval.

The QoS Mapping Problem

How is QoS mapped from level to level?

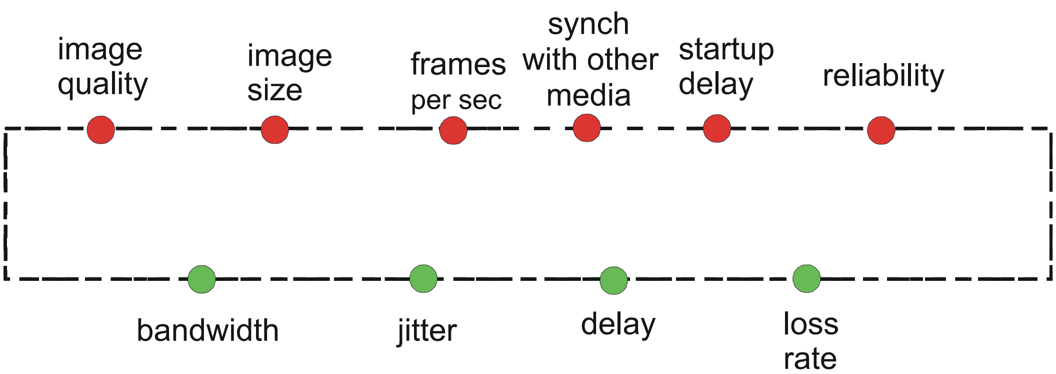


Example

User level:	Play a video at intermediate quality
Transport level:	Bit rate of the video, CBR or VBR, maximum startup delay
Network level:	Maximum packet size, maximum packet rate, maximum end-to-end packet delay, maximum delay variance

QoS mapping in networks is a current research problem, not yet well understood.

A QoS Mapping Example



Deterministic QoS Guarantees

The QoS negotiated between application and network will be guaranteed at all times (**hard QoS bounds**).

QoS calculation is based on:

- Hard bounds of the traffic generated by the source
- Worst-case assumptions concerning concurrent streams and available network resources

Advantage

- QoS guarantees will always be fulfilled, even under the worst load conditions

Disadvantages

- No statistical multiplexing gain for VBR traffic
- More frequent rejection of new connections

For multimedia applications deterministic guarantees are often not required (unlike for the control of real-time processes). Example: video quality in a video conference.

Probabilistic QoS Guarantees

QoS values have **soft bounds**.

QoS calculation is based on:

- A stochastic description of the traffic load
- A probabilistic specification of the behavior of the network ("in 95% of all cases the delay will be < 100 ms")

Advantages

- Statistical multiplexing gain
- More parallel connections can be permitted

Disadvantages

- QoS will not be optimal at all times
- Quite difficult to implement

QoS Definition: The ATM Example

The standard for ATM (Asynchronous Transfer Mode) defines precisely what traffic types a source can send:

- **CBR, VBR, ABR, UBR**

The ATM standard also defines precisely the parameters that characterize VBR traffic:

- **average cell rate, peak cell rate, maximum peak size** (see ITU-T standard Q.93b)

QoS and Reservation

No QoS guarantee without reservation of resources in the network!

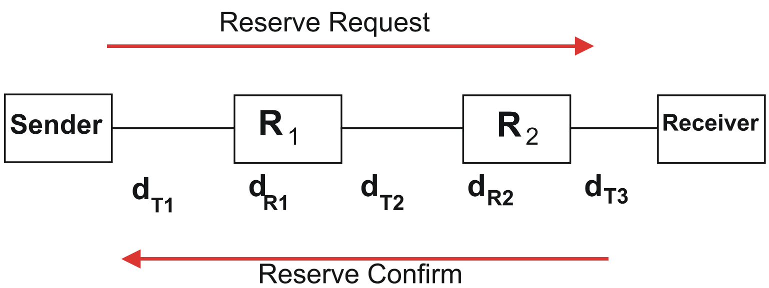
- We need local resource management within the network nodes (inner nodes and end nodes).
- We need reservation protocols.
- We need a surveillance of the sources („source policing“) to ensure that they conform to their traffic specifications.

Example: A Simple Protocol For Resource Reservation

The Principle

- The QoS parameter we are interested in is **delay**.
- Connection-oriented communication
- Reserve all the resources you can get on the path from the source to the destination
- When arriving at the destination, compute the left-over resources.
- Relax the left-over resources on the way back, re-distributing them, with the confirmation message

Protocol Flow (1)



QoS requested by the application: D_{max} = 140 ms

d_{T1} = 5 ms

d_{T2} = 15 ms

d_{T3} = 5 ms

d_{R1} = 10 ms

d_{R2} = 25 ms

Protocol Flow (2)

Reserve Request message

$\sum d_i =$ 60 ms

Left over: 80 ms

Relaxation strategy: equally over all routers

Reserve Confirm message

d_{maxR1} = 50 ms

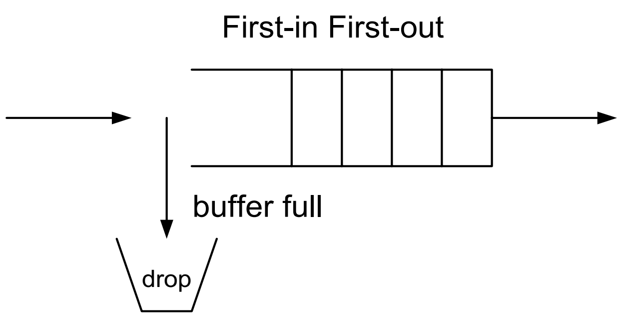
d_{maxR2} = 65 ms

4.2.2 QoS in the Internet

Current Internet: Best Effort model

The current Internet is built on the **Best Effort** model. In this model, all packets are treated equally. Packets are serviced by the routers in FIFO sequence with "Tail-Drop" in the case of overload.

The advantage of this model is that it is simple enough to run at very high speeds.



Multimedia Applications Require QoS

The problem of the Best Effort model is that it does not suffice any more to satisfy the requirements of new applications, in particular audio and video and other real-time applications, such as games.

Currently the most popular QoS technology for IP is the **Differentiated Services architecture (DiffServ).**

With DiffServ, traffic is separated into **classes** according to QoS requirements. DiffServ then treats each class in a **differentiated** manner according to its QoS needs. Assignment of the class is done by the application.

Two Traffic Classes: Elastic and Real-Time

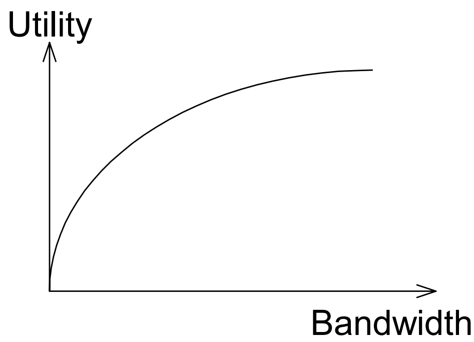
According to the **DiffServ** model, traffic is classified into two classes: **elastic traffic** and **real-time** traffic.

Examples of elastic traffic are file transfer, electronic mail and remote terminal access. Elastic traffic is tolerant of delays.

Examples of real-time traffic are audio and video. Real-time traffic is very sensitive to delay.

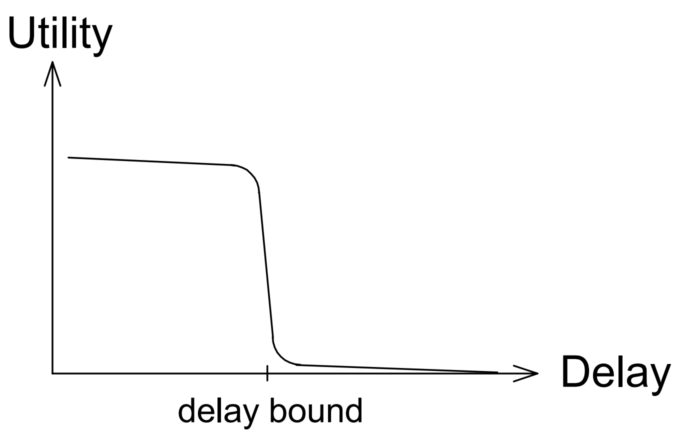
Elastic Traffic

The performance of elastic traffic depends mainly on the bandwidth. It experiences a diminishing marginal rate of performance enhancement as bandwidth increases.



Real-Time Traffic

Real-time traffic needs its data to arrive within a given delay bound. Data has no value if it arrives later than this bound.



Goals of DiffServv

DiffServ is designed to satisfy the different requirements of elastic and real-time traffic as they are identified by their utility functions.

DiffServ aims at providing QoS with simple mechanisms so that it scales well and can be deployed easily. Some previous proposals for QoS in the Internet have failed due to their complexity.

The fundamental DiffServ principle

Push the complexity to the “edges” of the network (i.e., to the end systems or edge routers of sub-networks). Keep processing in the core routers as simple as possible.

Pushing the Complexity to the Edge (1)

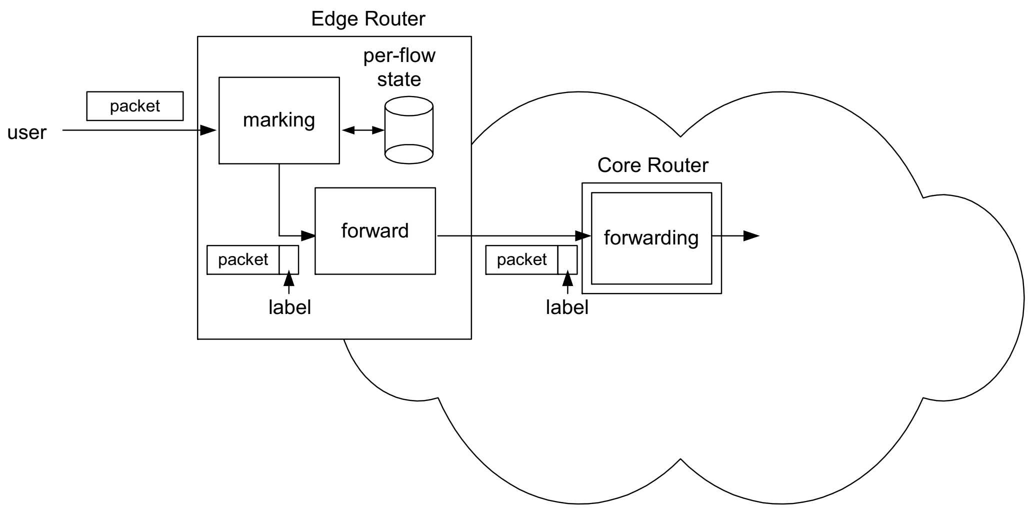
Edge (end system)

- Keeps per-flow state
- Marks the packets according to:
  - Sending rate of the user
  - Contract between the user and the network (Service Level Agreement, SLA)
- **The mark is inserted into a label field in the IP packet header.**

Core (router)

- Gives a differentiated treatment to each packet according to its label, in particular a different priority in the packet queue of the router
- **Does not keep per-flow state**

Pushing the Complexity to the Edge (2)



Two-Bit Differentiated Services Architecture

The so-called “Two-Bit Architecture“ is the basis of the current proposal in the IETF. It proposes three different levels of service:

- Premium Service
- Assured Service
- Best-Effort Service

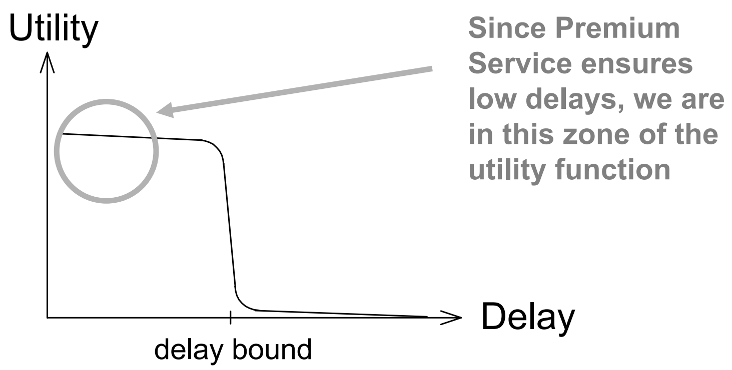
Packets get differentiated by two bits in their header:

- Premium bit (P-bit is on)
- Assured Service bit (A-bit is on)

Premium Service

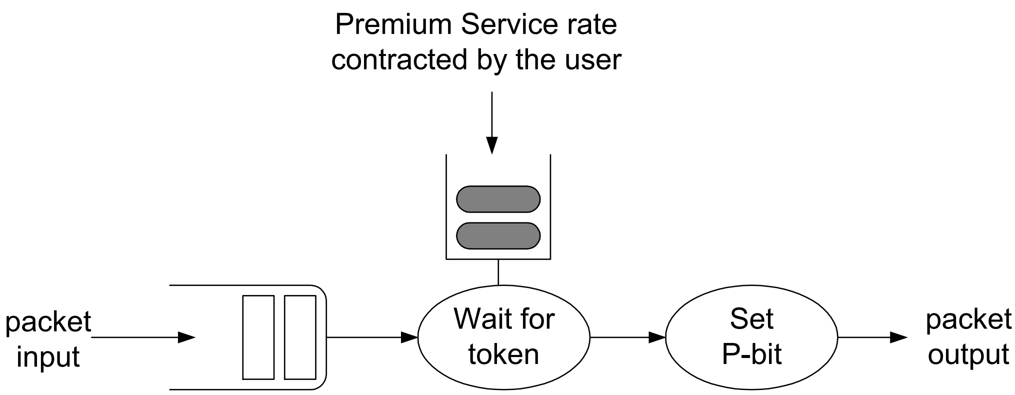
Premium Service provides very low delay and jitter to its packets. Therefore it is suited **for real-time traffic**.

A typical user of this service could be a company willing to pay a premium price to run a high-quality video conference over the Internet.



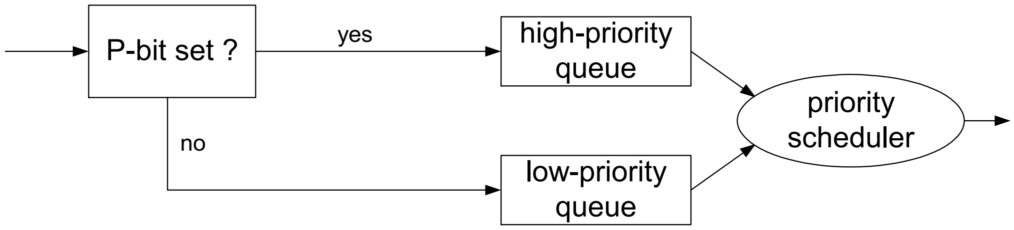
Premium Service: End System Action

Marking: The end system or edge router turns the P-bit of the packets on if the sender is in conformance to the SLA. The Premium traffic is shaped/ smoothed to prevent traffic bursts from being injected into the network.



Premium Service: Core Router Action

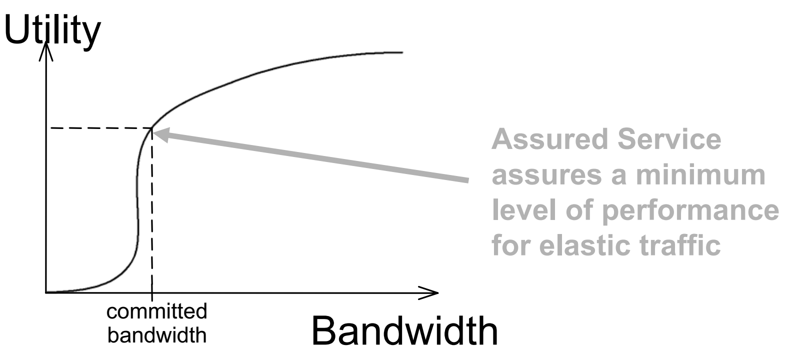
Forwarding: Core routers forward Premium packets first, other packets later.



Assured Service

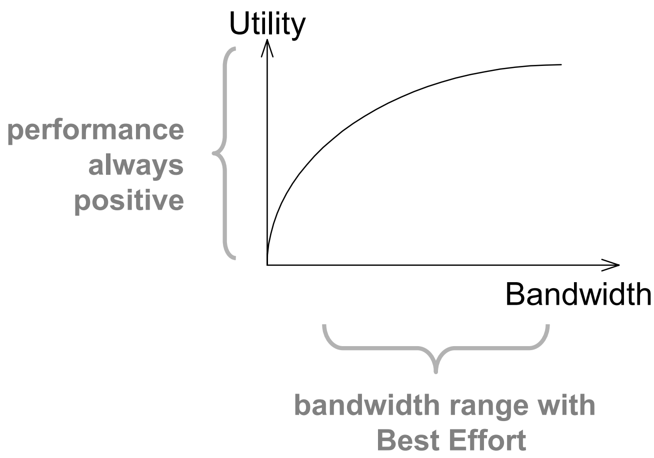
Assured Service does not provide delay guarantees, but it assures a committed bandwidth. Therefore, it is suited **for elastic traffic** that requires some performance guarantees.

A typical user of this service could be a company doing business on the Web willing to pay a certain price to make its service reliable and give its users a fast feel of its Web site.



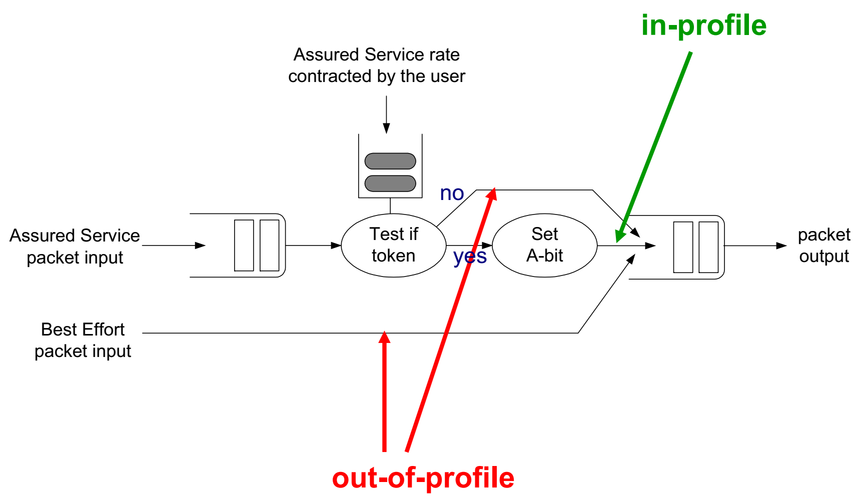
Best Effort Service

Best Effort Service does not provide any kind of delay or bandwidth guarantees. It is meant to be used **for non-QoS traffic**. It should be used for elastic traffic only; with real-time traffic there is the danger of experiencing a null performance.



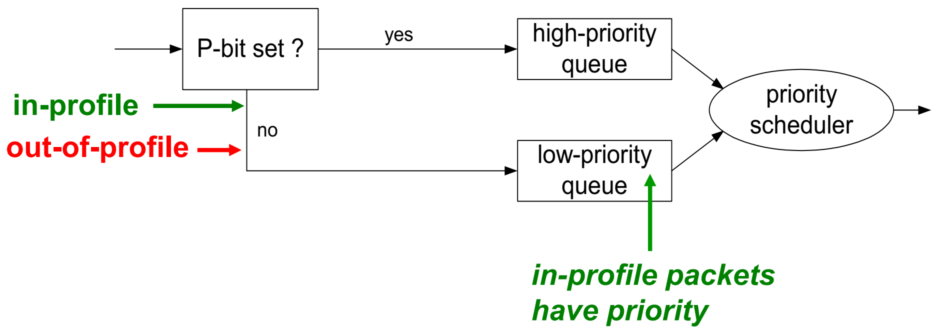
Assured and Best Effort Services: End Systems

Marking: Packets are classified **in-profile** if they belong to a user that has contracted a specific capacity with the Assured Service and stays within the expected capacity profile, and are classified **out-of-profile** otherwise.



Assured and Best Effort Services: Core Routers

Forwarding: All packets, in-profile and out-of-profile, are put into the same queue: the low-priority queue. The low-priority queue is managed in such a way that congestion leads to dropping the out-of-profile packets first, while in-profile packets are very unlikely to be dropped. This will be true as long as the admission control is such that in-profile packets alone do not congest the network.



Admission Control

Admission Control is necessary in order to guarantee that:

- Premium Service traffic is limited to some small amount (say, 20 %) of the bandwidth of input links.
- In-profile packets of Assured Service alone do not congest the network.

This is performed with the Bandwidth Broker (BB):

- The BB keeps track of all Premium/Assured Service contracted in the network.
- Based on this information it decides whether new requests can be granted.

Design and implementation of bandwidth brokers for the Internet is still a research issue.

DiffServ Summary

- Edge routers keep per-flow state for the flows traversing them (a small number).
- Core routers do not need to keep per-flow state.
- How to assign marks to IP packets in the end systems is an application matter.
- The main difficulty with the deployment of the IETF's DiffServ is admission control to be provided by a *bandwidth broker* which is not yet well defined.