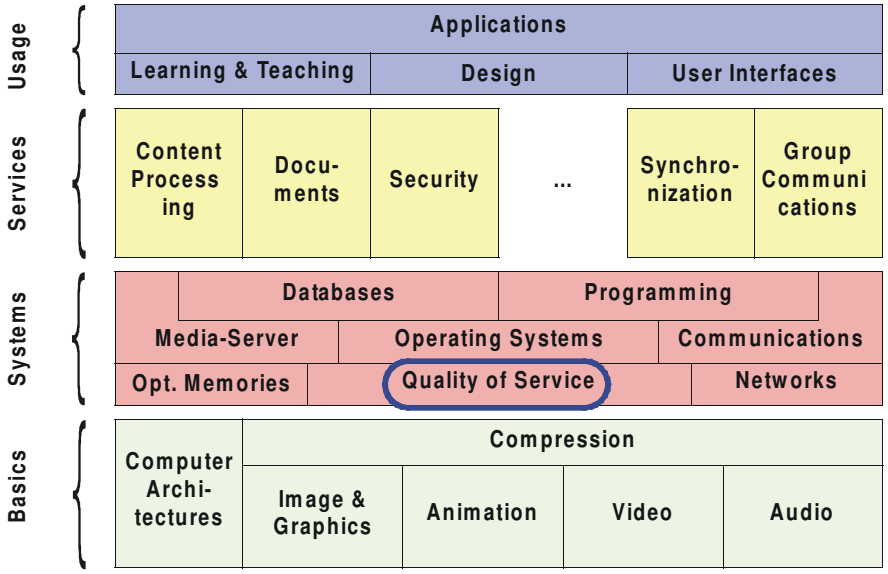
# 3. Quality of Service






## Content


3.1 Motivation

3.2 Characteristics of Real-Time / Multimedia Systems

3.3 QoS – Definition

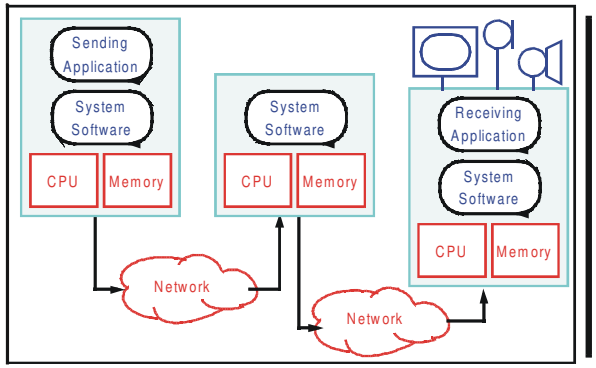
3.4 Resources

3.5 Providing QoS

3.6 QoS Architectures

# 3.1 Motivation

Kinds of systems we are dealing with are




Local:
- Harddisk recording
- Interactive DVD
- Computer based training

Distributed
- Conferencing
- Video on demand
- IP-Telephony

## Basic terminology

- Resources
- Realtime
- Quality of Service

What and how much of it do we need, and how do we describe that?



## Motivation for QoS

**A QoS model and its implications**

- QoS specification
- QoS calculation
- QoS enforcement

**QoS has different implications in different fields:**

- Operating system / Resource scheduling
- File system organization
- Compression
- Communication system support
- Media synchronization
- User Interface
- and more ...

# 3.2 Characteristics of Real-Time / Multimedia Systems

**Real-time System:**

*"A system in which the correctness of a computation depends not only on obtaining the right result, but also upon providing the result on time."*

**Real-time Process:**

*"A process which delivers the results of the processing in a given time-span."*

**Real-time applications - examples**

- **Control of temperature in a chemical plant**
  - driven by interrupts from external devices
  - these interrupts occur at irregular and unpredictable intervals
- **Example: Control of a flight simulator**
  - execution at periodic intervals
  - scheduled by a timer service which the application requests from the OS

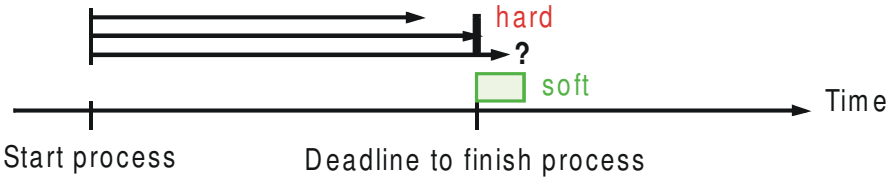
**Common characteristics:**

- internal and external events that occur periodically or spontaneous
- correctness also depends on meeting time constraints !



## Deadlines in Realtime Systems

**A deadline represents the latest acceptable time to finish an operation, e.g., for the presentation of a processing result**




- **Hard deadlines:**
  - should never be violated
  - result presented too late (after deadline) has no value for the user
  - violation means severe (potentially catastrophic) system failure
  - Example: Nuclear power plant

- **Soft deadlines:**
  - deadlines are not missed by much
  - in some cases the deadline may be missed, but not too many deadlines are missed
  - presented result still has some value for the user
  - Example: train/airplane arrival / departure

## Realtime System - Requirements

**Primary goal:**

- deterministic behaviour according to specification
- results in a variety of requirements

**Mandatory requirements:**

- Predictable (fast) handling of time-critical events
- Adequate schedulability
- Stability under overload conditions

**Desirable requirements:**

- Multi-tasking capabilities
- Short interrupt latency
- Fast context switching
- Control of memory management
- Proper scheduling
- Fine-granularity of timer services
- Rich set of interprocess communication and synchronisation mechanisms



## Multimedia Systems

A new application area for real-time systems with special characteristics:

- Typically soft real-time and not (that) critical
- Requirements may often be adapted to ensure proper handling, e.g., scaling of data streams to available bit rates

**Characteristics**

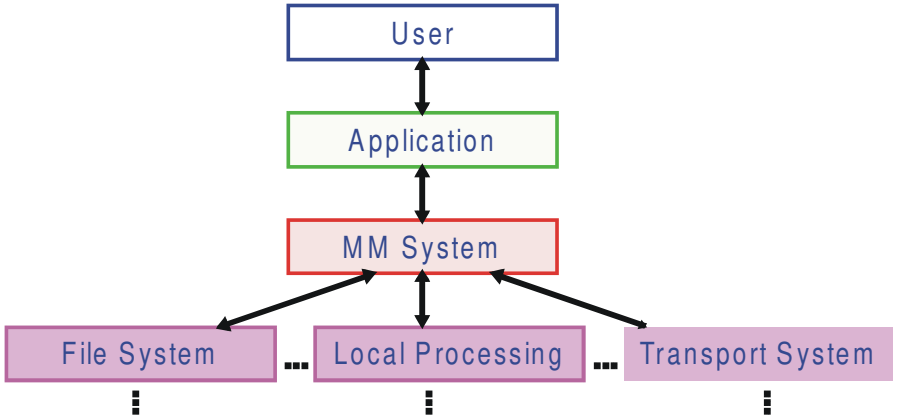
- Periodic processing
- Large bandwidth
- End-to-end guarantees
- Fault-tolerance
- Fairness
- Standardization

# 3.3 QoS - Definition

**Quality of Service** =

„well-defined and controllable behavior of a system according to quantitatively measurable parameters“

**Layer model**




Different service objects:

- Media / Streams
- Tasks
- Memory areas



## QoS - Layer Model

**Examples: both qualitative / quantitative description**

**Perception QoS**

- Tolerable Synchronisation Drift
- Visual Perceptability

**Application QoS**

- Media Parameters
- Media (Transmission) Characteristics

**System QoS**

- CPU Rate / Usage
- Available Memory

**Communication QoS**

- Packet Size / Rate
- Bandwidth
- End-to-End Delay

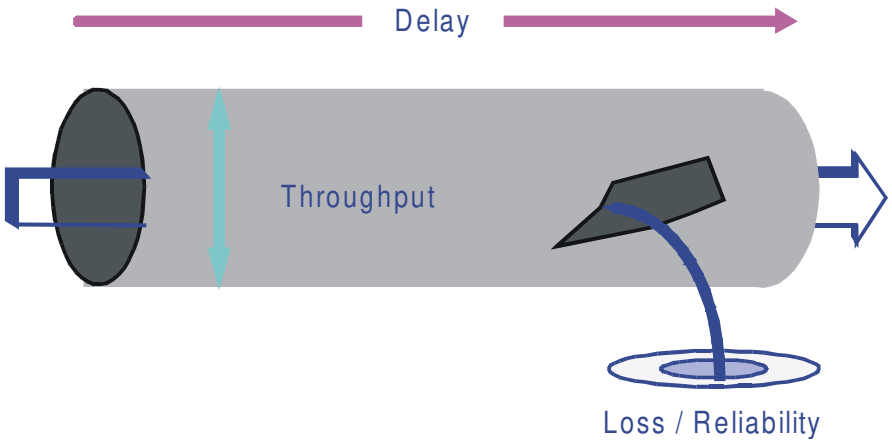
**Device QoS**

- Seek / Data Transfer Rate
- Sample Rate / Resolution

## QoS Parameters – Example: Transport System

Common parameters concerning the Transport System are:

- Throughput
- Delay / Jitter
- Loss / Reliability




But also:

- Security
- Cost
- Stability (Resilience)



## QoS Parameter Example

**Delay**

- Maximum end-to-end delay for transmission of one packet
- Delay jitter = maximum variance of transmission times

**Throughput**

- Maximum long-term rate = maximum amount of data units transmitted per time interval (e.g. ,packets or bytes per second)
- Maximum burst size
- Maximum packet size

**Loss**

- Sensitivity class: ignore / indicate / correct losses
- Loss rate = maximum number of losses per time interval
- Loss size = maximum number of consecutively lost packets

## Service Classes

**Guaranteed Service**

- Values or intervals of QoS parameters
  - deterministic (at any time)
  - statistical (consider a time interval or a certain propability)

$$QoS_{min} <= P <= QoS_{max}$$

**Predictable Service**

- consider history
  - from the very beginning of calculation
  - in a shifting time window
- "if it was like that in the last ..., you can rely on ..."

**Best Effort Service**

- no quarantees given

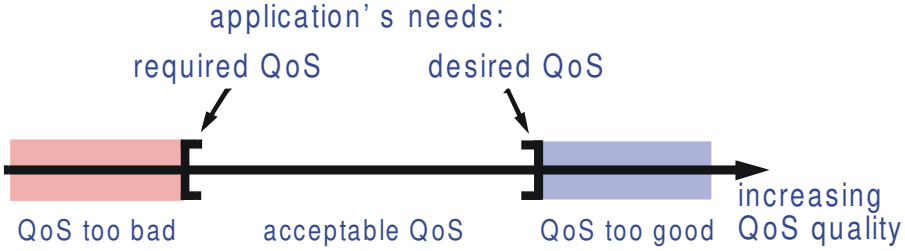


## QoS Intervals (1)

Parameter values result in

- acceptable regions
- inacceptable regions

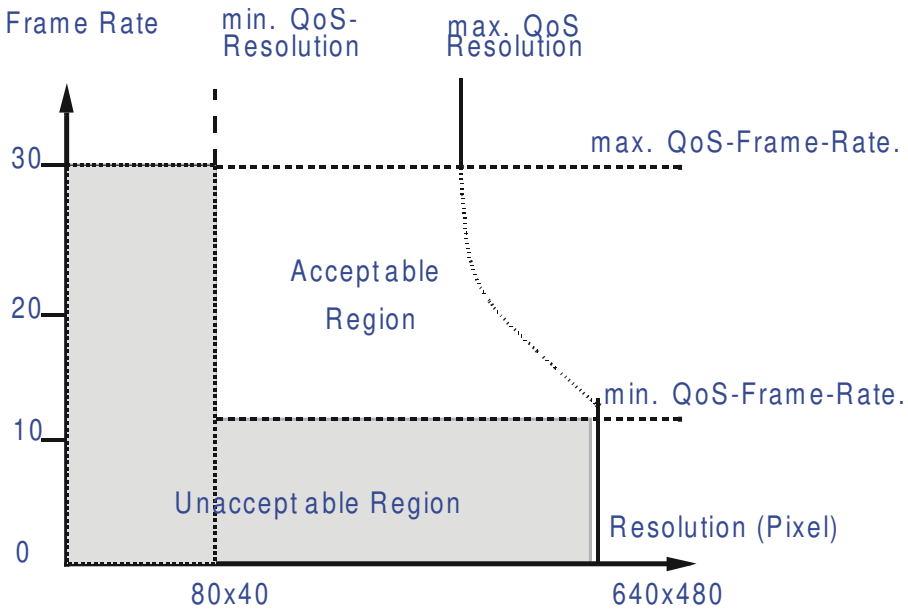
of QoS in one-dimensional intervals




- Below required QoS level - no useful service
- Above required QoS level - unnecessary (useless) resource consumption / cost

## QoS Intervals (2)

Also: multidimensional intervals






# 3.4 Resources

## Classification

**By functionality**

- active resources
  - actively fulfill a certain task
  - e.g., processor, network adapter
- passive resources
  - provide “space”
  - e.g., memory, frequency spectrum, file system

**By availability for concurrent usage**

- exclusive
- shared

**By occurence**

- single
- multiple

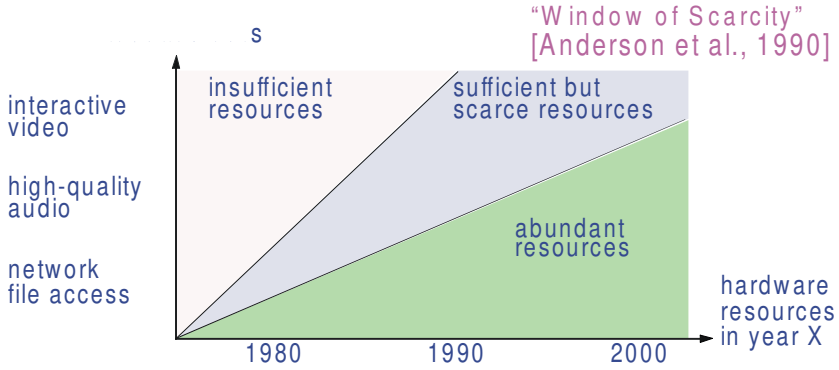
**Common parameter:**

- “Capacity“ - allows quantitative description

# Resources - Availability

**Starting point:**

- scarce, but sufficient resources




**Goal**

- Provide best service at the lowest possible cost

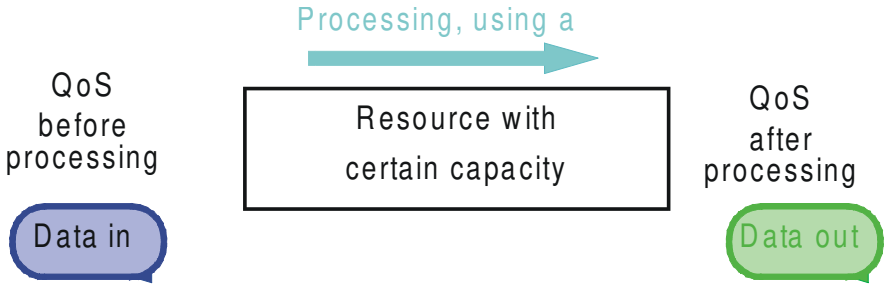
**Conclusion**

- We need resource management in all components of a multimedia system!

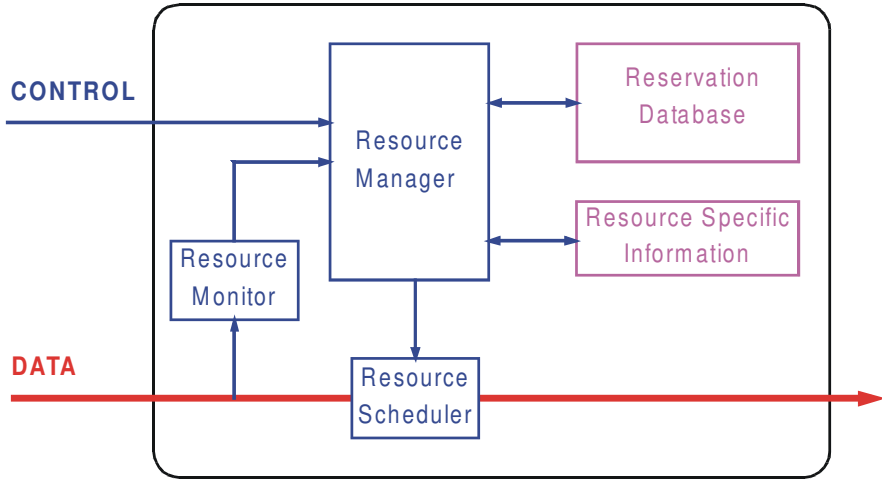


# Relationship Between QoS and Resources

**Model**



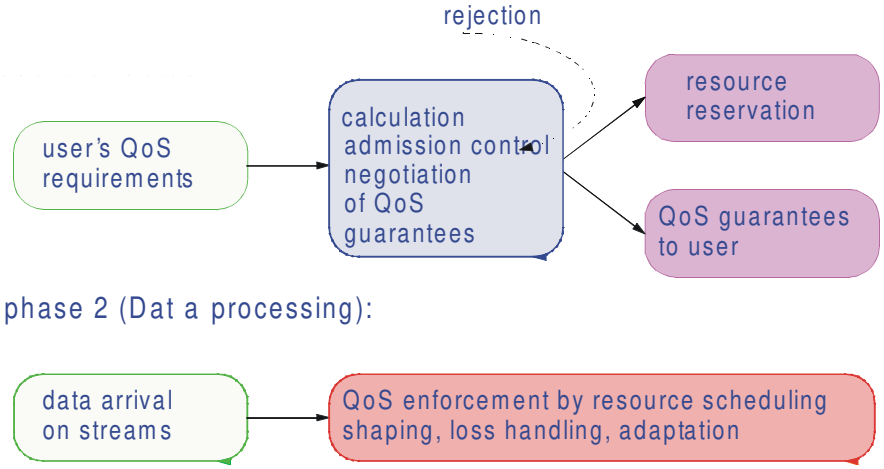
## Architecture



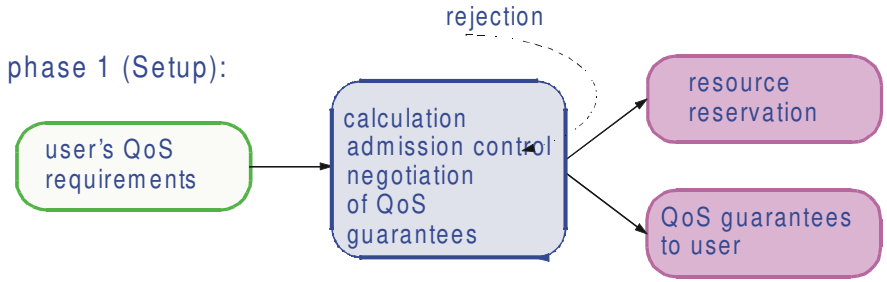



# 3.5 Providing QoS

## Resource Management Phases



# 3.5.1 QoS Provisioning – Setup Phase




Definition of required parameters
- implicitly or explicitly by application or user

Distribution and Negotiation

Translation between different layers
- especially if they use different semantics / notations

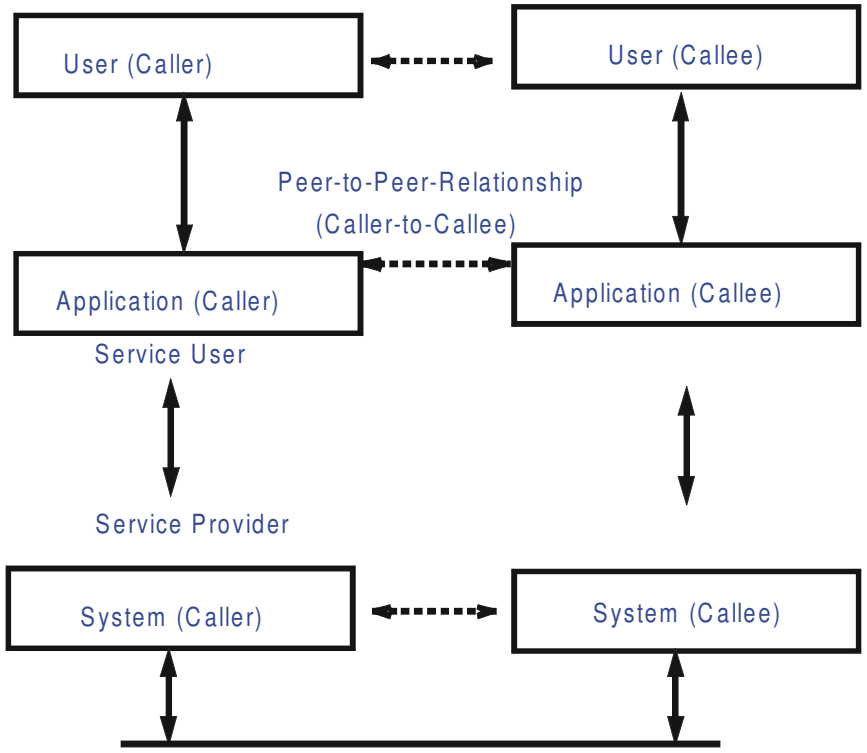
Transformation
- QoS parameter => Resource requirements

Allocation and coordination of resources
- along path(s) from source(s) to sink(s)



# QoS Calculation and Negotiation

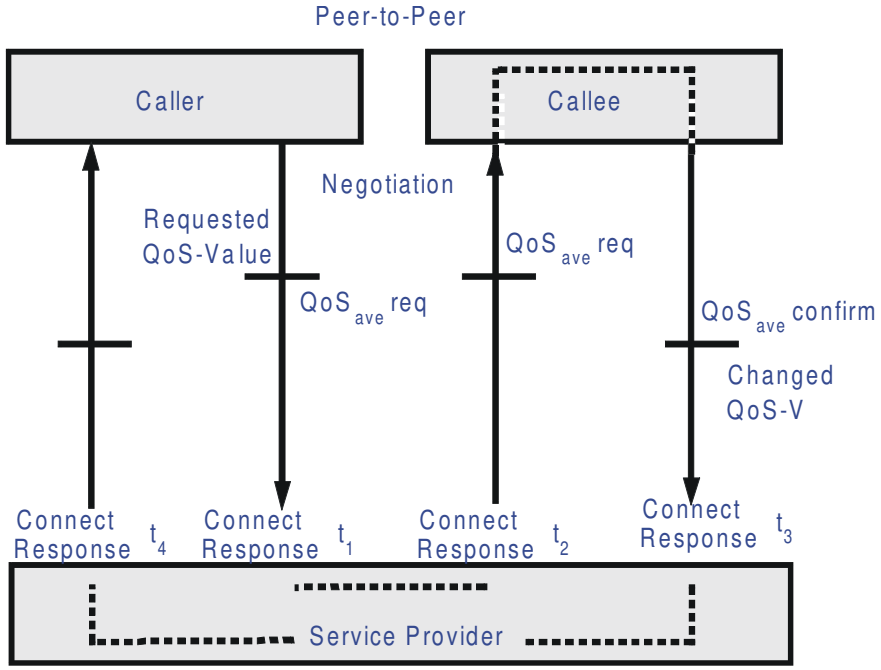
**Model**



# QoS Negotiation (1)

**Bilateral peer-to-peer**

- service provider may not modify requested QoS parameters
- only service user at receiver side may modify (lower) value(s) in the confirmation message






# QoS Negotiation (2)

**Bilateral layer-to-layer**

- **only between adjacent layers**
  - between local service users and providers
  - between sender and network

**Unilateral**

- **no modification of requested QoS parameters allowed, but just accept or reject**
- receiver may accept QoS parameter although he cannot meet them
  - example: color TV broadcast

**Hybrid**

- **uses unilateral mode for a certain bilateral layer-to-layer negotiation**
  - example: broadcast/multicast communication

===> heterogeneity of receivers

**Further:**

- **trilateral for information exchange**
- **trilateral for a limited target value**

## Admission Control

The system checks whether requested resources are and will be available. Especially important for shared resources:

- CPU
- network paths
- buffer space.

**A simple rule**

Check whether the sum of the resources already in use and new request(s) is less or equal to the available resource capacity.

More specific: check for

- schedulability
- availability of buffers (space)
- bandwidth

**Note:**

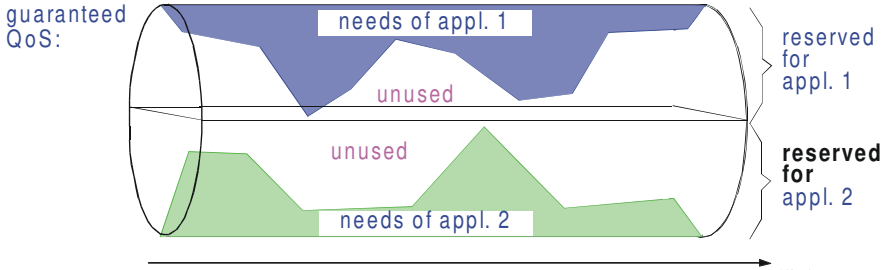
- strong relationship with Pricing / Billing
- efficient mechanisms will use “economic feedback” to prevent users from always requesting the maximum



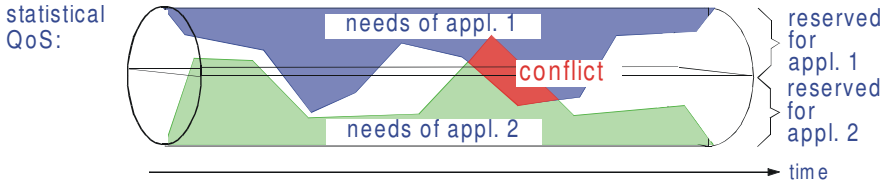
## Resource Reservation

**Fundamental concept for the reliable provision of QoS guarantees!**

- pessimistic - results in **Guaranteed QoS**




- optimistic - results in **Statistical QoS**




- May use monitoring and react on overload conditions (e.g., CPU load

## Resource Reservation Aspects - Example

**Example**

Communication System ===> variety of aspects

**Reservation Models**

- Sender-initiated
- Receiver-initiated
- Explicit vs. Implicit
- Out-of-Band vs. In-Band

**Reservation Style**

- Semantics and Notation
- Heterogeneity and multicast support

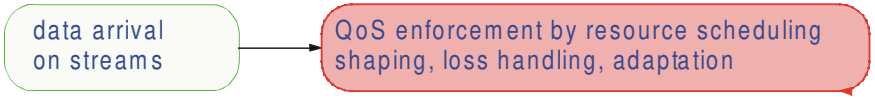
**Reservation Protocols**

- IP V.5: ST-II
- RSVP (Resource reSerVation Protocol)



# 3.5.2 QoS Provisioning - Data Processing Phase

phase 2 (Dat a processing):




**Maintain resource reservations**

**Use**:

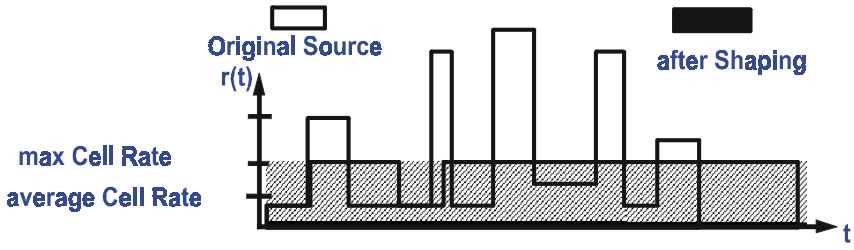
- adequate traffic shaping (to ensure characteristics of processed data)
- Scheduling algorithms
- feedback and adaption of the streams

# Shaping

Characteristics of Multimedia Traffic:

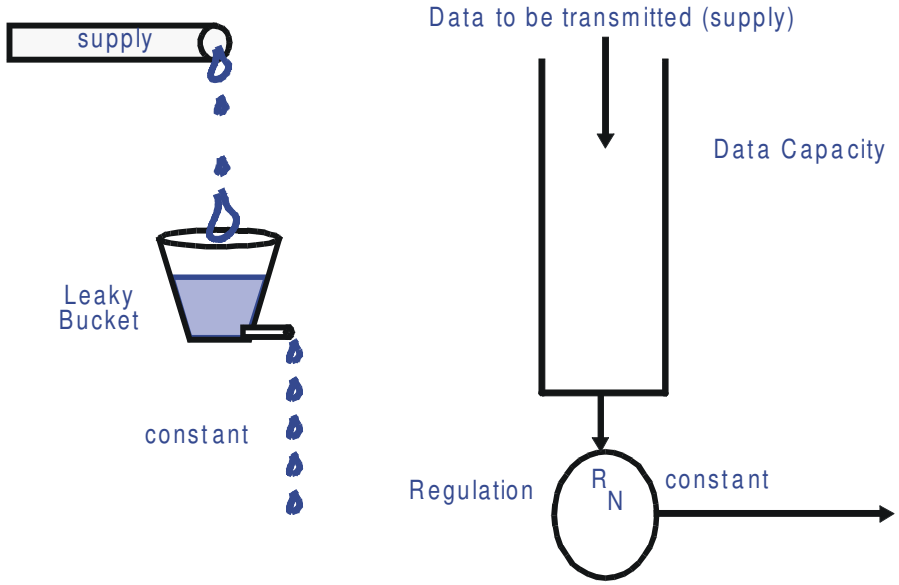
- bursty
- concurrent requests may cause problems though quarantees could be met (e.g., buffer overflow)

## Basic principle






# Shaping – Leaky Bucket Algorithm




## Bucket Size

- determines maximum capacity till overflow (drop) and possible delay

## Other Algorithms

- Token Bucket Algorithm
- Token Bucket Algorithm with Leaky Bucket Rate Control

## Loss Handling

**Error Detection**

- **by means of redundancy / checks / analysis**

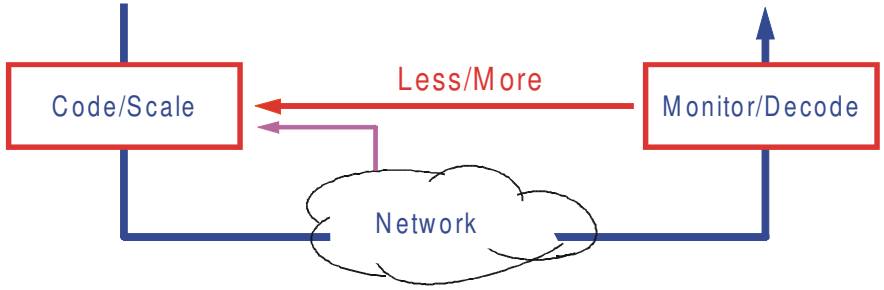
**Loss handling algorithms fall into two basic categories:**

- **Retransmission**
  - Go-back-N retransmission
  - Selective retransmission
  - Using partially error-free streams
- **Prevention**
  - Forward Error Correction (FEC)
  - Priority Coding
  - Slack Automatic Repeat Request



## Adaption - Feedback Control

- Monitor the load of network and local end-system resources
- If significant changes occur, take appropriate action to reduce generated load:
  - Explicit communication – receiver tells sender to slow down
  - Completely in network on a hop-by-hop basis
  - By feedback from congested network nodes to the sender.




Variety of possible reactions

- e.g., layered transmission
- adaptive degradation of the stream quality
- ...

# 3.6 QoS Architectures

**Examples (communication layer)**

- **Heidelberg Transport System (HeiTS)**
  - uses ST-II (IPv5)
- **Internet Integrated Services**
  - use existing infrastructure, but deploy dedicated handling of flows (streams) in the transfer system
  - Resource Reservation Protocol RSVP to support heterogenous needs
- **Differentiated Service**
  - Granularity based on the TOS (Type Of Service) IP Header Field
  - Define service classes, negotiate service level agreements and ensure dedicated treatment of flows that behave as described
- **IPv6**
  - QoS support was an important design criterion from the beginning
  - Dedicated header fields to allow classification / dedicated treatment of flows