



Multimedia Systems

WS 2009/2010

Quality of Service

Prof. Dr. Paul Müller

University of Kaiserslautern, Germany
Integrated Communication Systems Lab

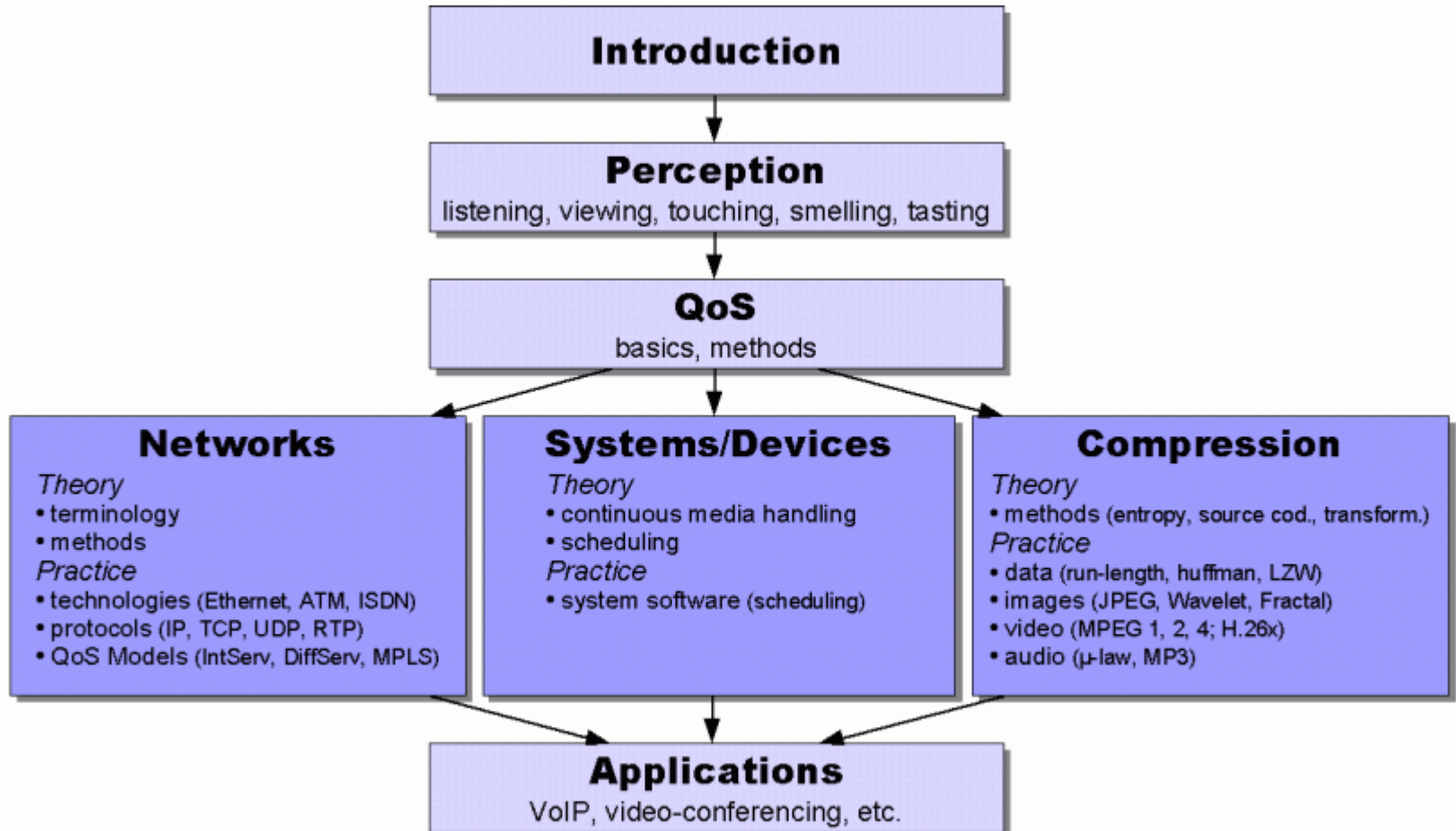
Email: pmueller@informatik.uni-kl.de



Outline

- Scaling and Adaption
- Resource Reservation
- Throughput and Delay
- Error Detection and Correction
- Real-time Systems
- Quality of Service
 - Service Objects
 - Transmission
 - End-to-End QoS
 - Negotiation
 - Provision, Control & Management

Sitemap



Quality of Service Overview

- Challenge: media processing is (often) related to "real-time".
- Problem: classical computer architectures, system software, and data networks were designed for batch processing
- Enabling appropriate media processing requires that sufficient resources are available at specific points in time within all steps of media processing

Definition:

Quality of Service (QoS) denotes the well-defined and manageable behaviour of a system according to measurable parameters. *

- **The implementation of QoS requires the availability of resource management!**
- There are two major approaches for enabling QoS:
 - Scaling and Adaptation of media
 - Resource Reservation
- Enabling appropriate media processing requires that sufficient resources are available at specific points in time within all steps of media processing

* Source: R. Steinmetz and K. Nahrstedt, Multimedia Computing, Communications

Scaling and Adaptation

- Requires determination of available resources
 - automatic or manual
 - a priori measurements
 - monitor error rates during processing (within control loops)
 - manual selection of media quality by user or configuration
 - ability to adapt to changing amount of resources
- Ability of Adaptation
 - range (high to medium or very high to very low quality?)
 - granularity (few classes of quality or fine grained adjustment?)
 - static (a priori) or dynamic (at runtime)
in case of dynamic adaptation: spontaneous or negotiation required
 - type of resources considered by adaptation

Resource Reservation

End-to-End reservation of resources:

- amount of resources must be known for all types of critical resources
 - a priori reservation
 - adaptation of reserved resources
- resource reservation steps
 - describe amount of required resources
 - signal/negotiate resource reservation
 - policy control (who is allowed?)
 - admission control (how much is available?)
 - acknowledge availability of resources
 - perform reservation and adjust/initialise "resource scheduler"
 - during resource usage
 - user: monitor provision of resources
 - provider: monitor / adjust resource usage

Throughput

The bit rate between two communication endpoints is the number of binary digits that the network is capable of delivering and accepting per time unit.

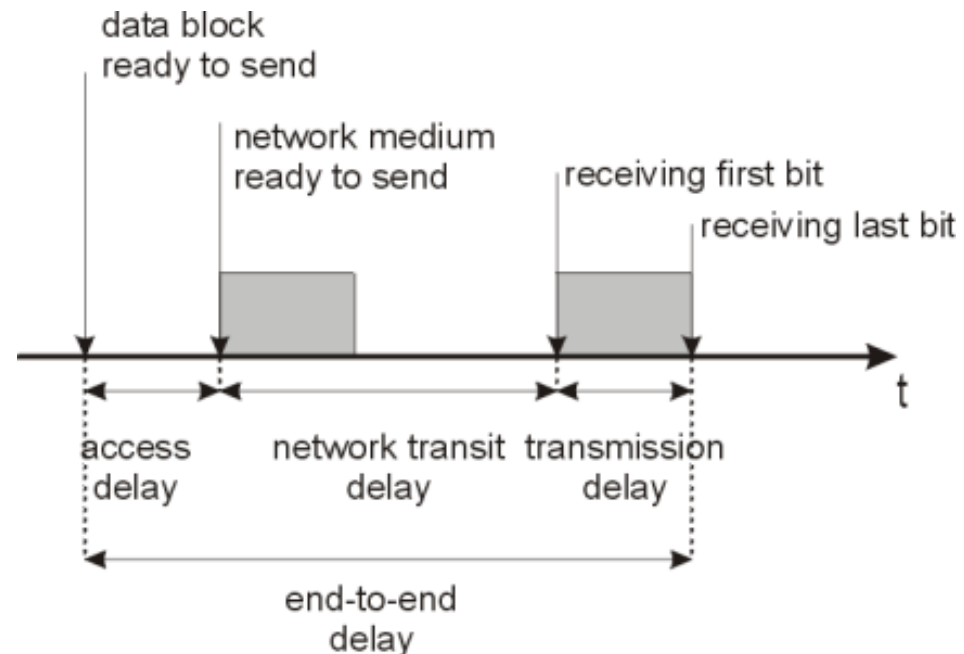
- The commonly used time unit is one second, but often the actually observed time unit is another one
 - ISDN offers a throughput of 8 bit per 125 μ s
- distinguish individual and aggregate bit rates
- access speed versus bit rate
 - access speed refers to the frequency at which bits may be sent or received
- bit rate could be constant or variable
- burstiness
 - peak bit rate (PBR): maximum bit rate during a short time interval
 - mean bit rate (MBR): averaged bit rate over a longer time interval
 - burstiness is the relation between PBR and MBR



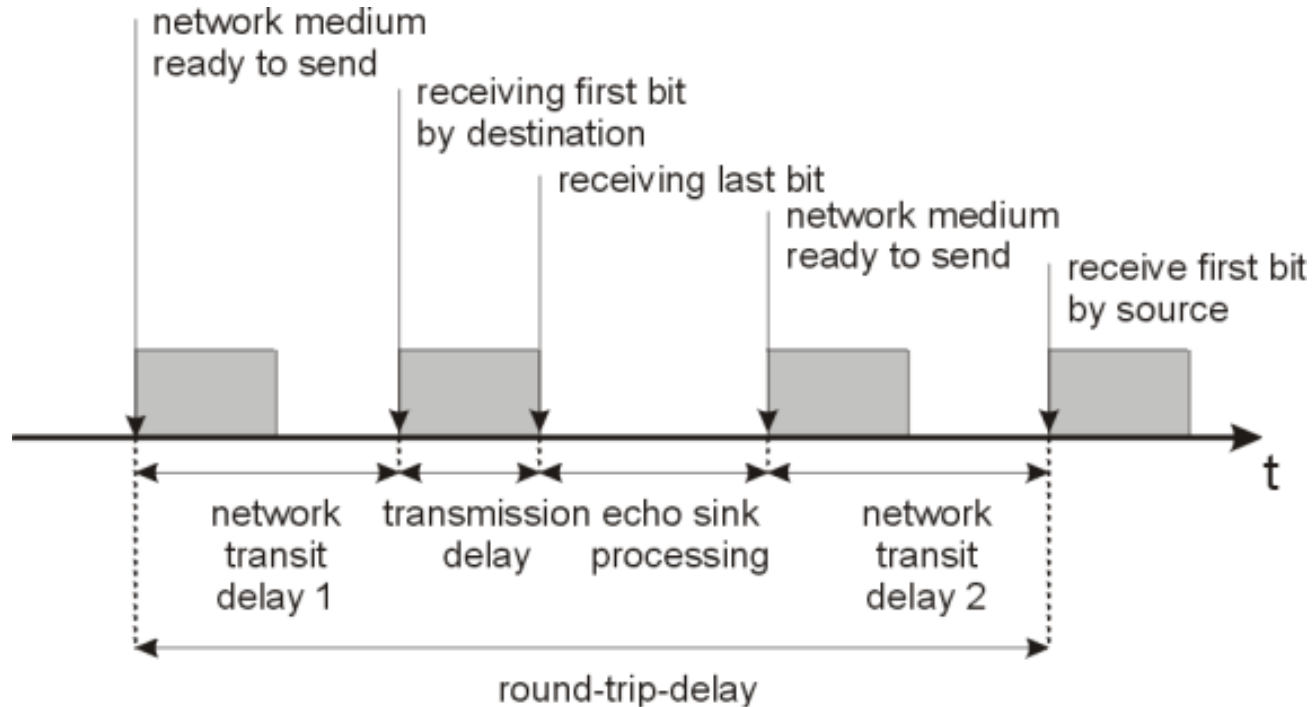
Delay

Delay of a service object is the sum of the following delays:

- access delay**
 the time between the request to send a data block and the emission of the first bit
- (network) transit delay**
 the time between emission of the first bit of a data block and its reception at the destination endsystem (network transit delay is also called latency)
- transmission delay**
 the time between emission of the first and the last bit of a data block



Round-Trip-Delay (RTT)



- The RTT is also called response time
- RTT is a good metric for interactive applications



Delay Variation

The Delay variation relates to variation of end-to-end delay

- also called "jitter"
- distinguish average and maximum delay variation

Originator of delay variation

- the physical environment causes jitter in magnitude of nano seconds
- intermediate switches/routers may cause jitter by
 - variation of medium access time, e.g. CSMA/CD (usually micro-seconds)
 - store and forward switching delays, e.g. by internal node congestion (micro-seconds)
 - waiting time caused by flow control mechanisms (micro- or even milli-seconds)
- In ISDN there is only a physical jitter, whereby in IP networks all types of jitter occur.

Delay vs Loss





Reliability

- Mechanisms
 - Error detection
 - Error handling
 - requires error detection
 - posteriori, e.g. retransmission
 - a priori, e.g. forward error correction
- Error types
 - Data loss
 - Data alternation
 - Data duplication, miss insertion or wrong delivery
 - Failure of components usually not considered for multimedia systems

▶▶▶ Error Detection and Error Correction (1)

Motivation:

Many applications require that data is transmitted correctly, i.e. that data reaching its destination is the same than the data transmitted

Problem:

- Channels are not an ideal medium
 - noise/interferences on the channel
 - losses on purpose (e.g. dropping packets in case of congestion)

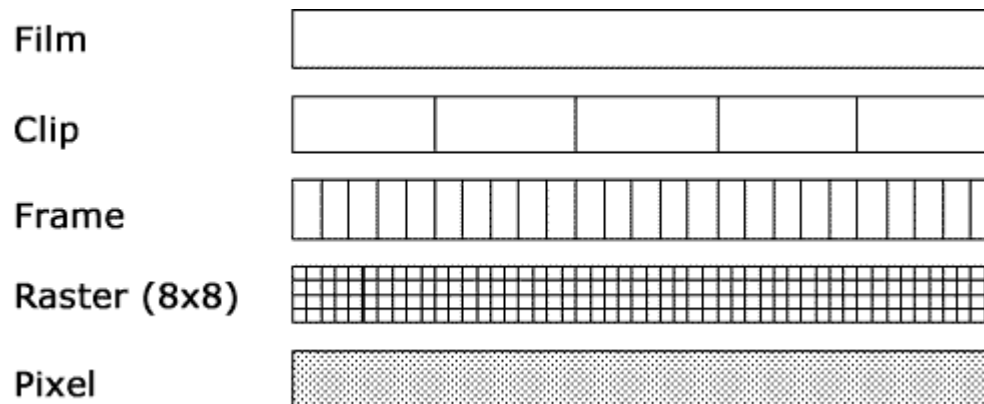
▶▶▶ Error Detection and Error Correction (2)

- Error detection:
 - Means:
 - checksums (e.g. parity bits, CRC)
 - Not all conceivable errors can be detected!
- Error correction (if required):
 - Means:
 - retransmission of data (e.g. in TCP)
 - employing coding theory
 - Not all conceivable errors can be corrected!
- Terms:
 - EDC: Error Detection Code
 - ECC: Error Correction Code
 - FEC: Forward Error Correction

Logical Data Units (LDU)

- Today's systems were designed to handle discrete data only
 - Therefore continuous media is viewed as periodical and discrete data
 - Logical Data Units (LDUs) of different granularity

Example:

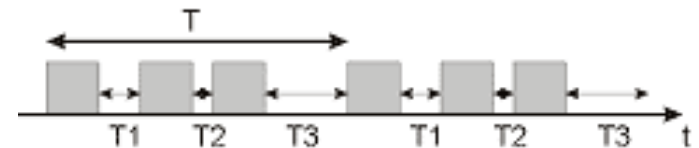


- closed LDUs: known length and number (e.g. film on a hard disk)
- open LDUs: unknown length or number (e.g. input of a camera)

Properties of Continuous Media

- **Periodicity**

- strongly periodic, fixed inter arrival time
- weakly periodic, variation of inter arrival time is fixed
- not periodic



- **Joining**

- joined packages, no gap between packages (strongly periodic)
- not joined



- **Size**

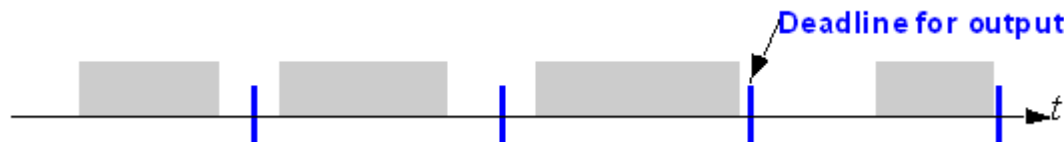
- strongly regular size, all packages have the same size
- weakly regular size, variation of package size is fixed
- irregular

Deadline driven Processing

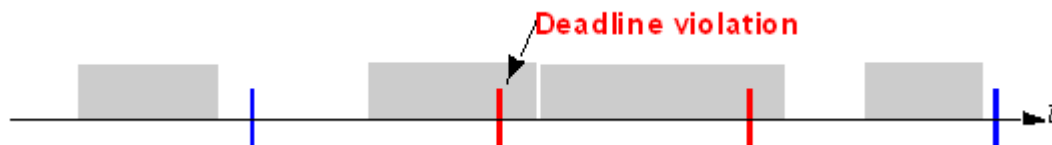
- Basic parameters for real-time processes: arrival time, processing time, deadline (arrival time of LDU ~ occurrence of an event)
- Ideally: regular arrival and constant processing time lead to regular output



- Acceptable: limited variation of arrival and processing time



- Not acceptable: increased delay or jitter lead to deadline violation





Real-Time Systems

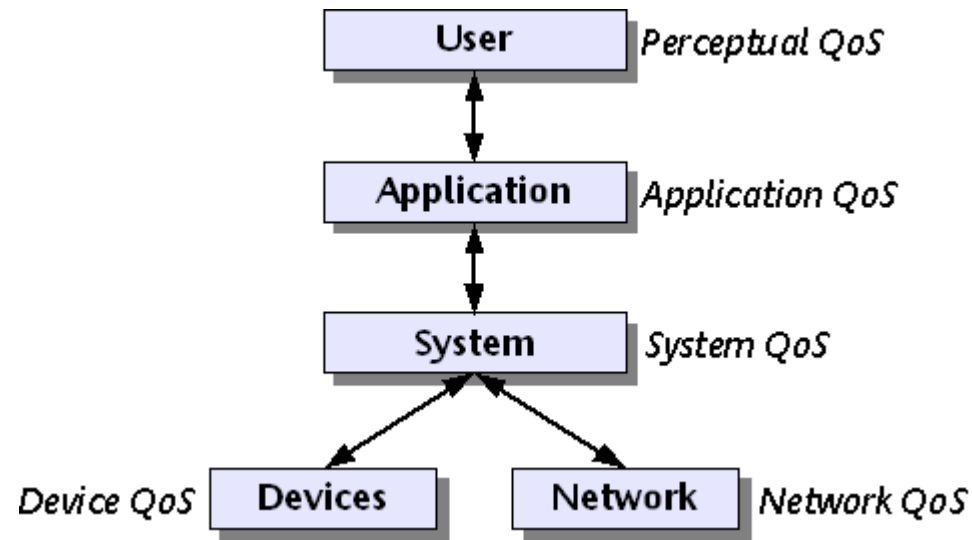
According to DIN 44300: A system is considered real-time capable when it can respond to an (exterior) event under all conditions with a defined (deterministic) response ... it is crucial that the response occurs within a predetermined time

- The correctness of a real-time system includes
 - correct computing results
 - delivered within a given time interval
- note: results must not be delivered too early either
- The behaviour of a real-time system must be deterministic
- Internal processing must adjust to (external) events, which may occur
 - deterministically, i.e. occurrence know a priori, e.g. periodic events
 - stochastically, i.e. randomly, e.g. triggered by user interaction

Hard and Soft-Deadlines

- Hard deadlines:
 - deadline violation equals missing data
 - typical related to "real" processes (e.g. within control engineering)
- Soft deadlines:
 - are like a reference point in time
 - missing a soft deadline may still produce acceptable results, if
 - not too many deadlines are missed
 - the deadlines are not missed by much
- Typically, multimedia systems use "soft deadlines" often in conjunction with periodic events (for continuous media).

QoS Layering



The Quality of Service will be defined by different parameters on each layer.

Service Objects

- Originally QoS were defined for networks only
- QoS specifications should be defined for all involved service objects, examples:
 - Application: media streams
 - Middleware: distributed objects, software components, services, invocation methods
 - System: tasks, system services
 - Device: media buffer, functions
 - Network: channels (connections, flows, virtual circuits,...), sessions
- Each service object requires specific sets of QoS parameters.
- Trivial: achieving a specified output QoS requires a certain input QoS

QoS Parameter (1)

Perceptual QoS

- perceptive quality:
 - media quality (very good, good, medium,...)
 - resolution (window size, audio frequency range,...)
 - color or sampling accuracy (high or low)
 - response time (interactive, batch)
 - security (high, low)
- pricing: user should be able to define pricing limits

QoS Parameter (2)

Application QoS

- media quality
 - media characteristics: frame rate, frame resolution, color and sampling accuracy,...
 - transmission characteristics: end-to-end delay, jitter,...
- media relations: e.g. synchronization media streams, max. skew
- adaptation rules: e.g. actions if network bandwidth is scarce

Transmission Characteristics

- **Human-to-human interactive applications:**
 - Delay sensitivity
 - Synchronisation drifts (skew)
 - Lip synchronisation (continuous media synchronisation)
 - Pointer synchronisation (continuous/ discrete media synchronisation)
- **Man-machine interactive applications:**
 - Synchronisation
 - Lip synchronisation (continuous media synchronisation)
 - Pointer synchronisation (continuous/ discrete media synchronisation)
 - Cross media/ multiple format synchronisation
 - Media quality
 - Media validity
- **Non-interactive applications:**
 - Synchronisation
 - Continuous/ discrete media synchronisation
 - Cross media/ multiple format synchronisation

Synchronization

Strong synchronisation:

- Multiplexing:
 - One data channel
 - Media and synchronisation information are jointly delivered



- Appropriate for:
 - Lip synchronisation

Weak synchronisation:

- Timecode/ clock synchronisation:
 - Media linked via time references
- Appropriate for:
 - Pointer synchronisation (continuous/ discrete media synchronisation)
 - Cross media/ multiple format synchronisation

QoS Parameter 3

System QoS

- Quantitative criteria (measurable)
 - bit per second, throughput (per second), delay, response time, max. error rates...
 - task period, tasks cpu load or processing time
 - buffer size
- Qualitative criteria (functions needed)
 - inter stream synchronization
 - ordered data delivery, error recovery
- qualitative criteria may be related to quantitative criteria

Processing Requirements

- **Goal:**
 - Processing of continuous media with high data rates
- **Periodic and aperiodic processing:**
 - Deadlines for continuous-media processing
 - No hard deadlines such as in traditional real-time systems
 - No starvation of aperiodic requests
 - No priority inversion
- **Existing workstation operating systems:**
 - Mostly no support for real-time processing
 - Some offer 'real-time' priority schemes
 - Based on fairness aspects
 - No provision of resource management for real-time requests



QoS Parameter 4

Device QoS

- timing
- buffer
- media type (audio codec incl. parameters)

Network QoS

- network load
 - average or minimal inter arrival time, burstiness
 - average or max. packet size
- network performance
 - max. or average delay for a packet, jitter (= delay variation)
 - loss or error rates

Service Classes

1. Guaranteed Service

- deterministic QoS, specified by exact boundaries
e.g. $Delay_{min} < Delay(Service) < Delay_{max}$
- statistical QoS, specified by probabilities
e.g. $P(Delay(Service) < Delay_{max}) = p$

2. Predictable Service

- estimate QoS based on past behavior use average or weighted average of past service behavior

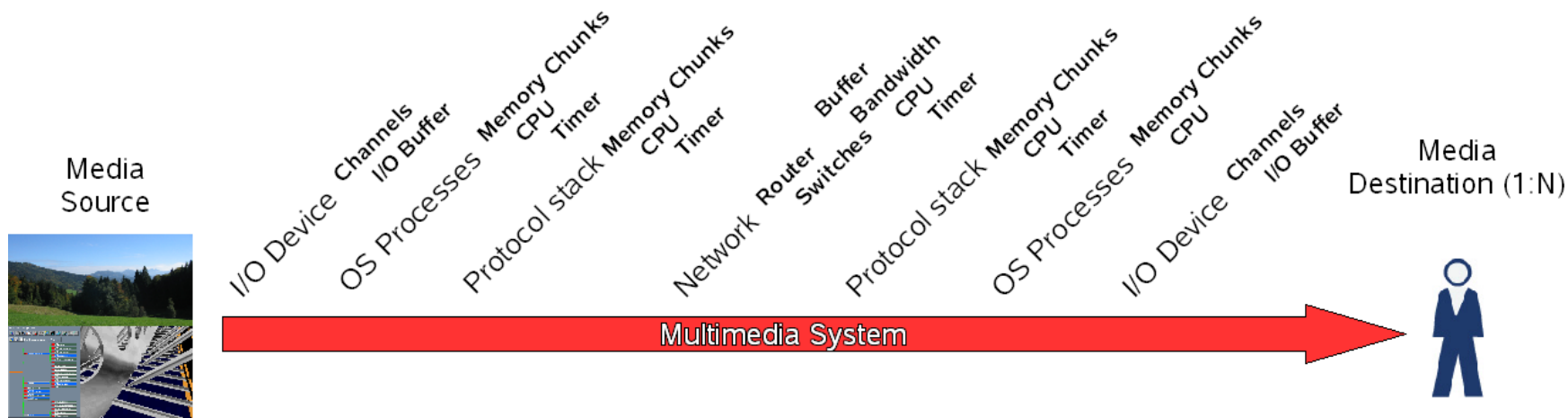
3. Best Effort Service

- no QoS specification at all

These service classes differ according to:

- reliability of the offered QoS (which class provides a QoS according to our definition of QoS?)
- utilization of resources

End-to-End QoS



- End-to-End QoS requires adequate QoS of each service object
- Each service object must have sufficient resource

Service Concatenation

- Providing end-to-end Quality of Service requires the concatenation of services
- Model of a single service S
 - $QoS(S)_{in} = [q_1^{in}, \dots, q_n^{in}]$
 - $QoS(S)_{out} = [q_1^{out}, \dots, q_m^{out}]$
 - $Resources(S) = [r_1, \dots, r_k]$
- Model of a service concatenation
 - a concatenation of services build a directed acyclic graph
 - a quality aware concatenation of services requires that the inter-service "satisfaction" is valid
i.e. i, j , where S_i is the predecessor of S_j : $QoS(S_i)_{out}$ must match $QoS(S_j)_{in}$
 - $QoS(S)_{out}$ match $QoS(S')_{in}$ if:
 $q_i^{in} QoS(S')_{in} \quad q_j^{out} QoS(S)_{out}$ with $q_j^{out} q_i^{in}$

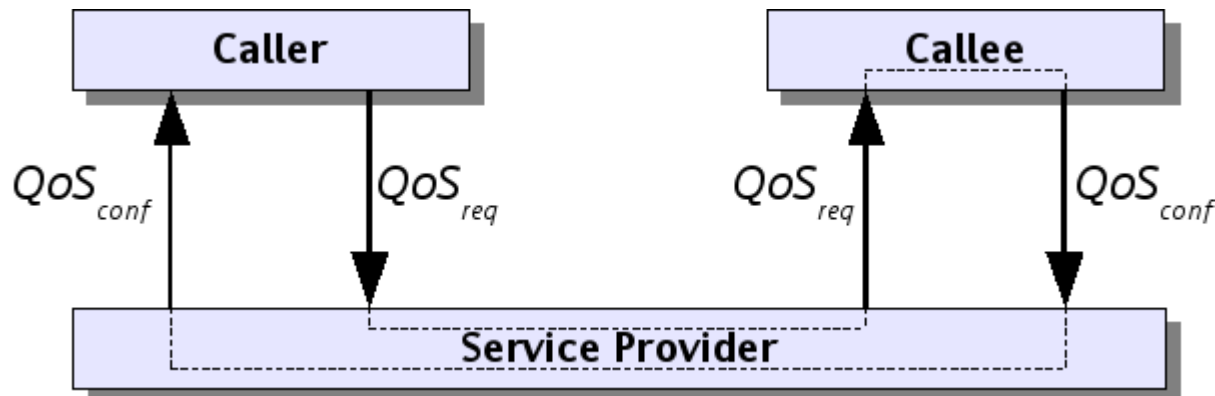


Resources

- Identify scarce resources → at least these must be managed
- Different management required for:
 - exclusive resources: (early) reservation
 - (virtual) memory
 - virtual audio / video (e.g. window area)
 - shared resources: time scheduled access
 - CPU (task scheduler)
 - network link (packet scheduler)
 - buffer (queueing mechanism)
 - native audio / video
 - (shared) memory (semaphores)
 - parallel access resources: read only
 - video, audio input
 - mouse position

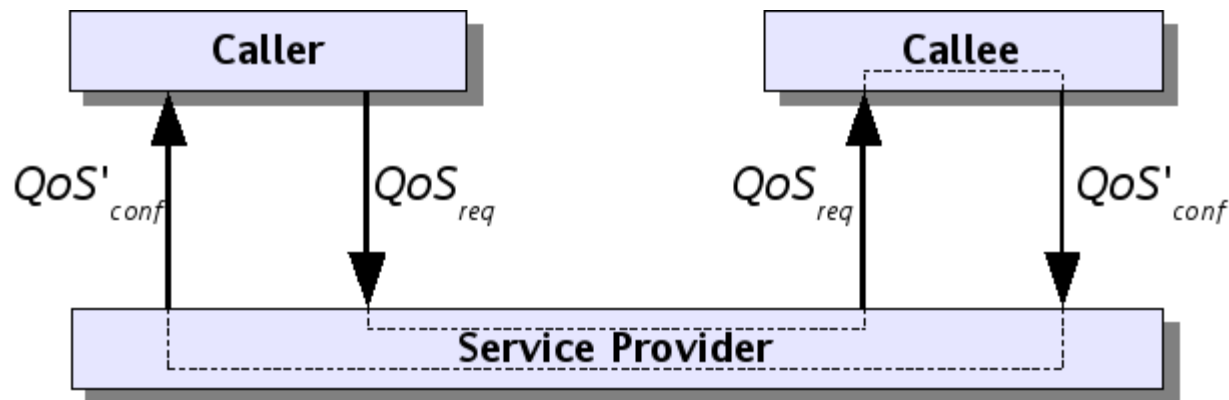
QoS Negotiation 1

- The availability of resources is an essential prerequisite for QoS
- Guarantee availability by resource reservation, this requires
 - a protocol for signaling resource reservation
 - a negotiating procedure
- Unilateral Negotiation:
 - QoS level given by caller, no modification allowed
 - callees may still reduce quality of received data, e.g. TV broadcasts



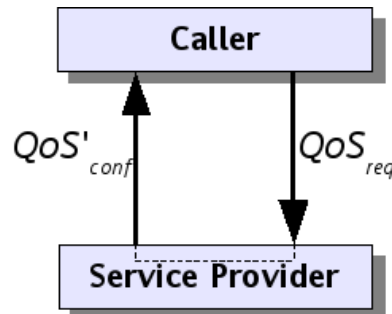
QoS Negotiation 2

- Bilateral Peer-to-Peer Negotiation:
 - only the callee may modify QoS specification
 - Note: caller may specify/suggest a lower bound or a QoS interval

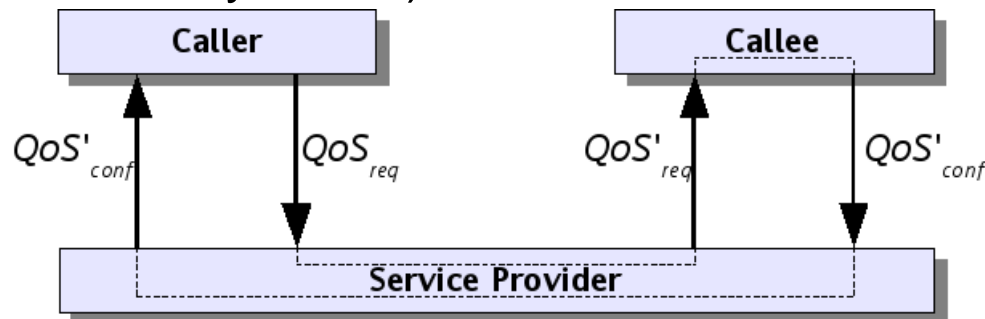


QoS Negotiation 3

- Bilateral Layer-to-Layer Negotiation:
 - may require mapping of QoS specification
 - Negotiation with provider only

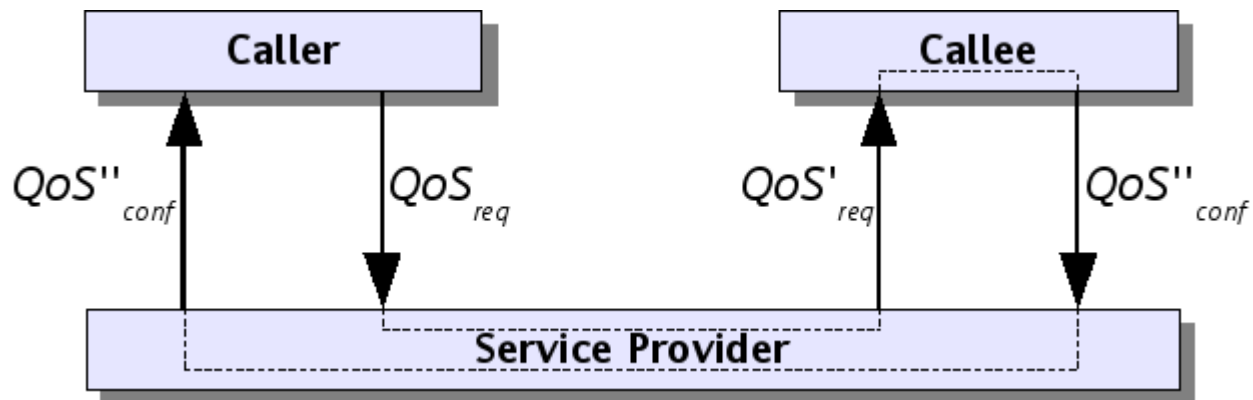


- Negotiation with provider and confirmation by callee (no modification by callee)



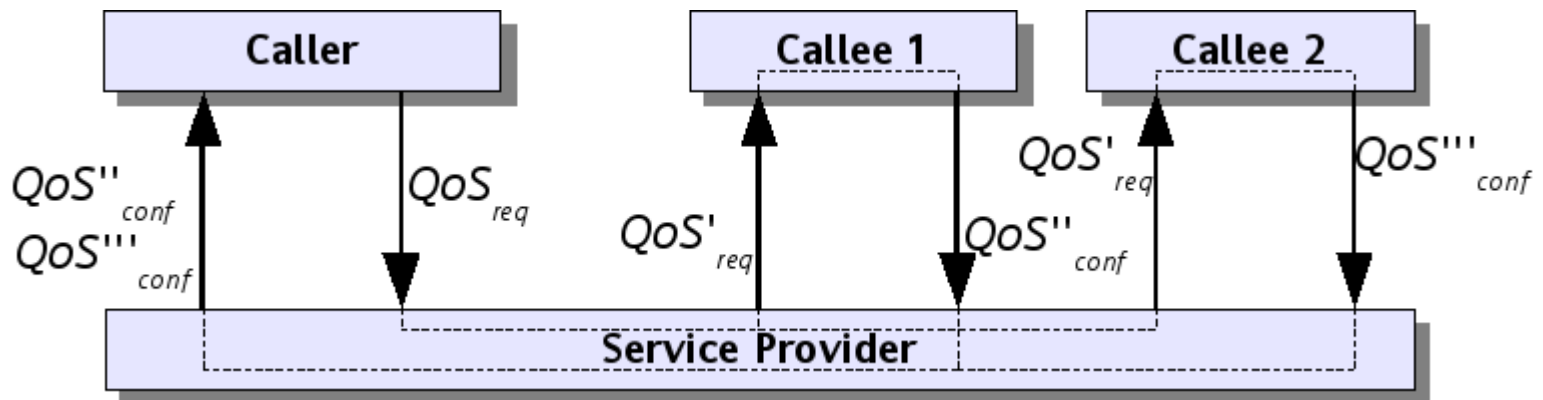
QoS Negotiation 4

- Trilateral Negotiation:
 - first the provider may modify the QoS
 - then the callee may also modify the QoS
 - Negotiation with provider and callee



QoS Negotiation 5

- Hybrid Negotiation:
 - negotiate with one provider
 - and several callees, usefull within multicast trees
 - Negotiation with provider and several callees





QoS Mechanisms Overview

- Classification of QoS Mechanisms:
 - Provisioning
 - Control
 - Management
- Management and control are similar but differ according to time scale

QoS Provisioning

Mechanisms for QoS provisioning:

- mapping between service levels
 - automatic mapping relieves user/application developer from dealing with technical details of lower levels
 - "good video quality" -> bandwidth requirement
- admission control
 - a local test for resource
 - results are prebooked and used for negotiation processes
 - confirmation required for reservation
- resource negotiation / reservation
 - requires a resource reservation protocol, that implements one or more negotiation procedure(s)
 - must interact with routing



QoS Control 1

Mechanisms for QoS control:

- flow scheduling
 - within endsystems, several flows are independent of each other
 - within the network flows may be aggregated
- flow shaping
 - fine grained regulation of flows based on user supplied parameters
 - aims is to keep negotiated traffic characteristics
 - reduction of peaks/bursts improves network performance
lower lossrates (for all) but higher delay (for shaped flow)
- flow policing
 - control whether a flow fulfills negotiated parameters
 - typically used between administration domains / charging boundaries



QoS Control 2

Mechanisms for QoS control:

- flow control
 - coarse grained regulation of flows, e.g. transport or application level
 - open loop, resource reservation in advance, i.e. no control at run-time
 - closed loop, control based on feedback, i.e. adaptation at run-time
- flow synchronization
 - event synchronization
 - playout of media, e.g. for lip synchronization

QoS Management

Mechanisms for QoS management:

- QoS monitoring
 - like flow policing, but monitor the QoS provided
 - end-to-end or at reference points
 - statistics also used for network planning
- QoS availability
 - capability of QoS monitoring to interact with applications, e.g. "which parameters to monitor?", "signal changes of provided QoS"
- QoS degradation
 - capability of signaling and describing lowered QoS to an application
- QoS maintenance
 - adjust resources based on monitored QoS
- QoS scalability
 - QoS adaptation, by endsystems
 - QoS filtering, by the communication systems, e.g. transcoding within gateways