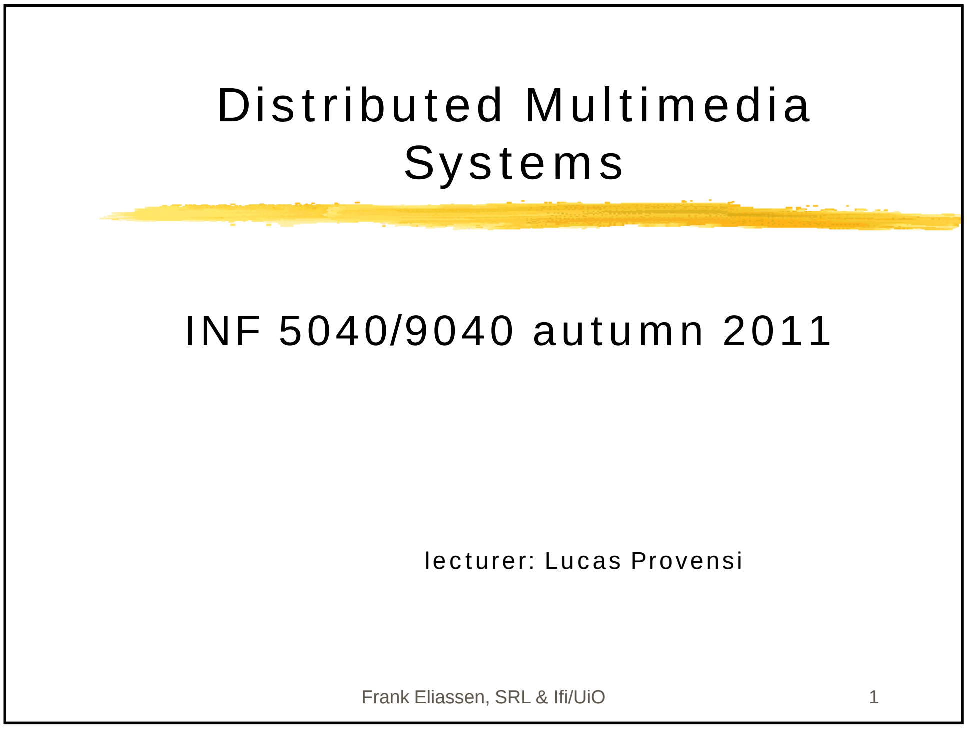
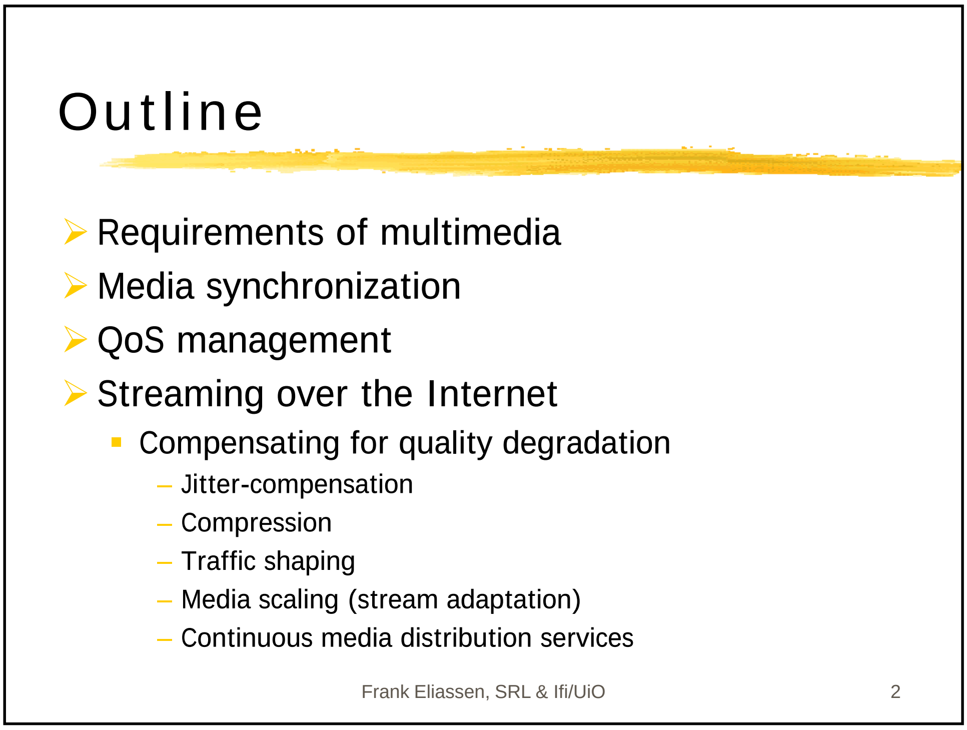


Distributed Multimedia Systems

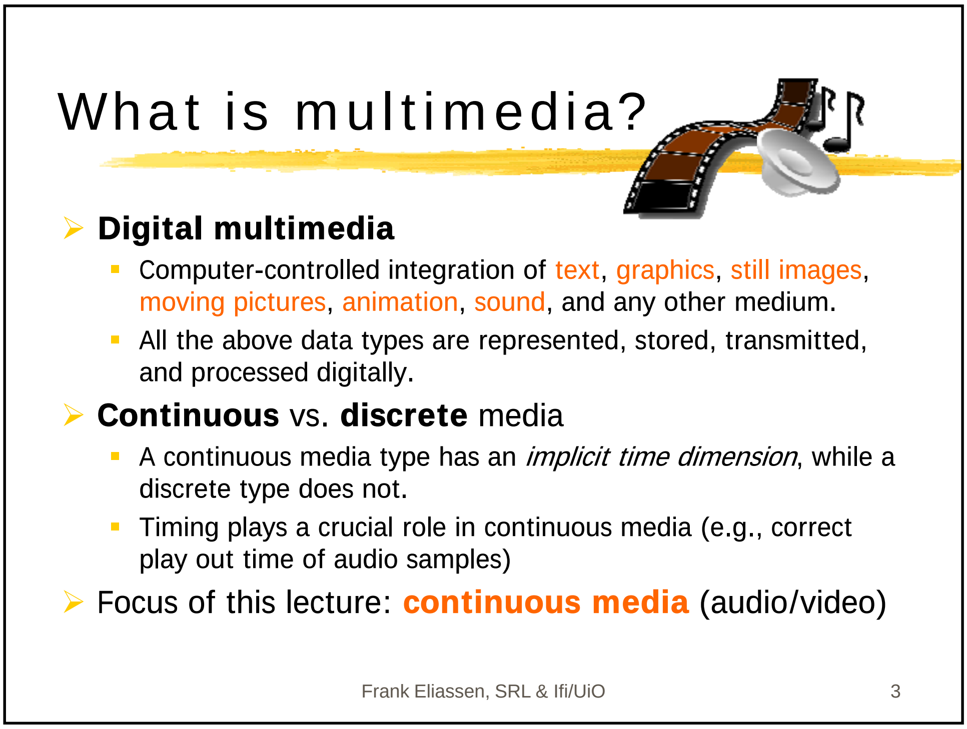
INF 5040/9040 autumn 2011

lecturer: Lucas Provensi



Outline

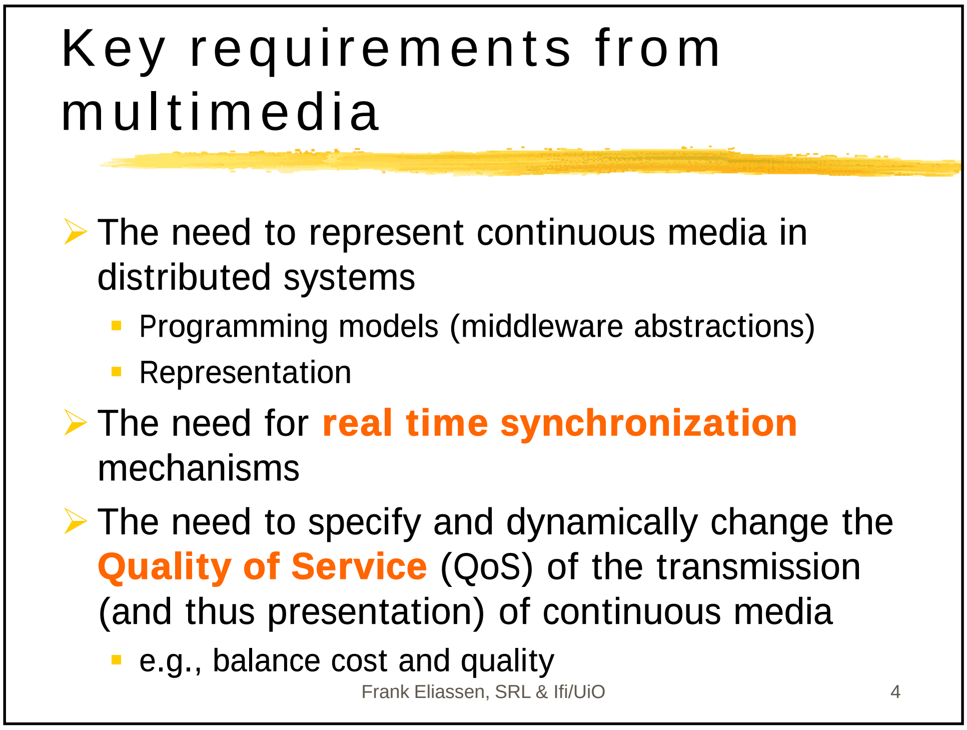
- Requirements of multimedia
- Media synchronization
- QoS management
- Streaming over the Internet
  - Compensating for quality degradation
    - Jitter-compensation
    - Compression
    - Traffic shaping
    - Media scaling (stream adaptation)
    - Continuous media distribution services



What is multimedia?

- **Digital multimedia**
  - Computer-controlled integration of text, graphics, still images, moving pictures, animation, sound, and any other medium.
  - All the above data types are represented, stored, transmitted, and processed digitally.
- **Continuous** vs. **discrete** media
  - A continuous media type has an *implicit time dimension*, while a discrete type does not.
  - Timing plays a crucial role in continuous media (e.g., correct play out time of audio samples)
- Focus of this lecture: **continuous media** (audio/video)

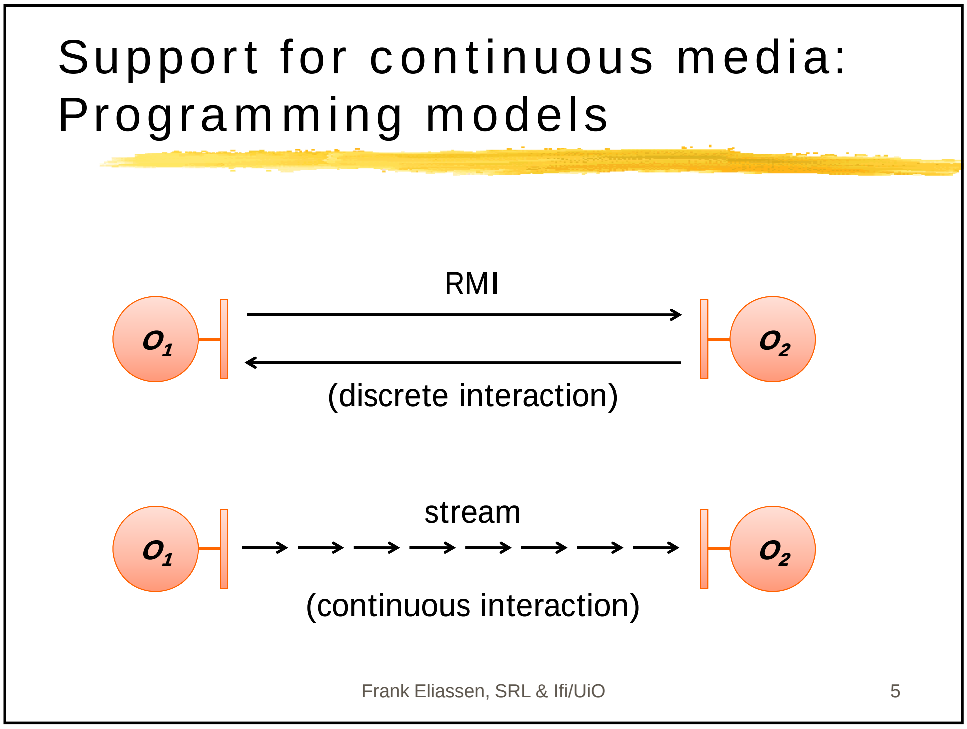
Frank Eliassen, SRL & Ifi/UiO 3



Key requirements from multimedia

- The need to represent continuous media in distributed systems
  - Programming models (middleware abstractions)
  - Representation
- The need for **real time synchronization** mechanisms
- The need to specify and dynamically change the **Quality of Service** (QoS) of the transmission (and thus presentation) of continuous media
  - e.g., balance cost and quality

Frank Eliassen, SRL & Ifi/UiO 4



Support for continuous media: Programming models

RMI

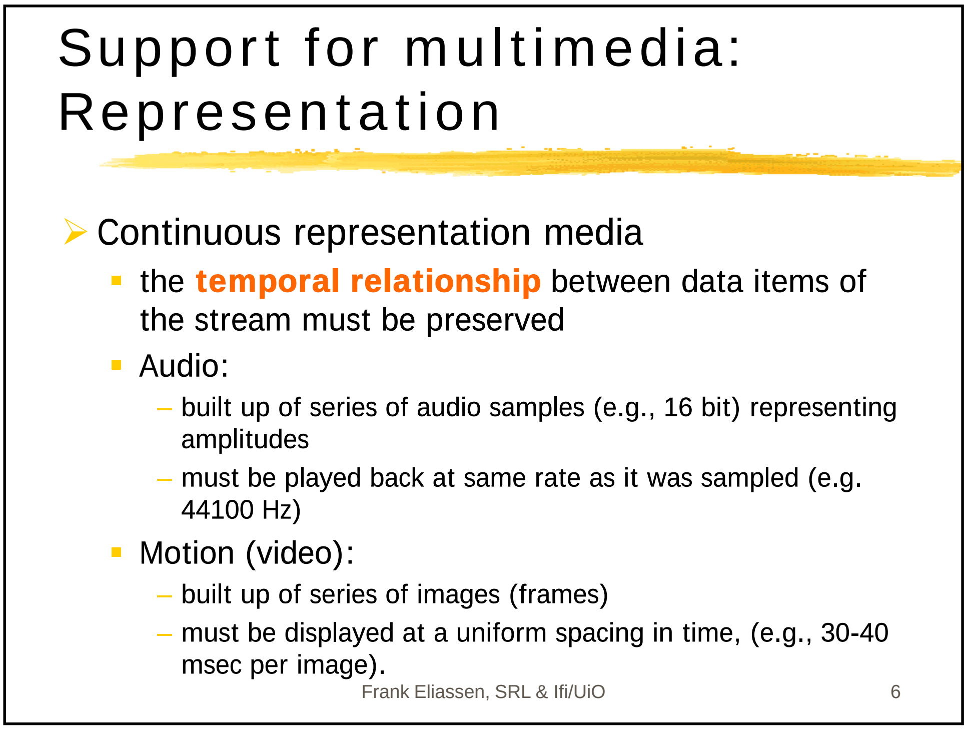
O_1 O_2

(discrete interaction)

stream

O_1 O_2

(continuous interaction)



Support for multimedia: Representation

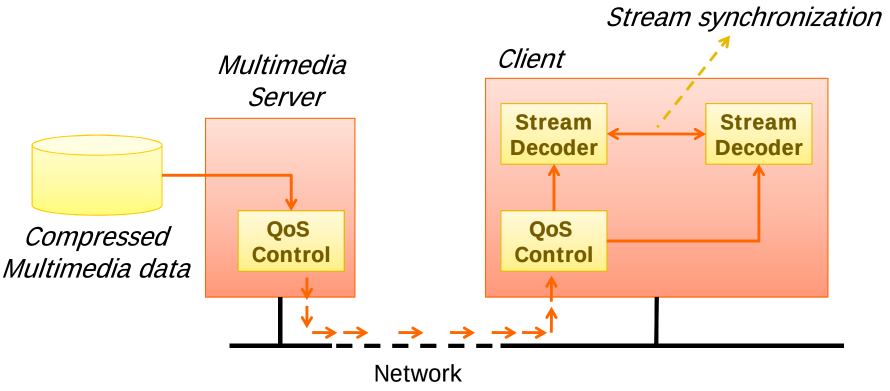
- Continuous representation media
  - the **temporal relationship** between data items of the stream must be preserved
  - Audio:
    - built up of series of audio samples (e.g., 16 bit) representing amplitudes
    - must be played back at same rate as it was sampled (e.g. 44100 Hz)
  - Motion (video):
    - built up of series of images (frames)
    - must be displayed at a uniform spacing in time, (e.g., 30-40 msec per image).

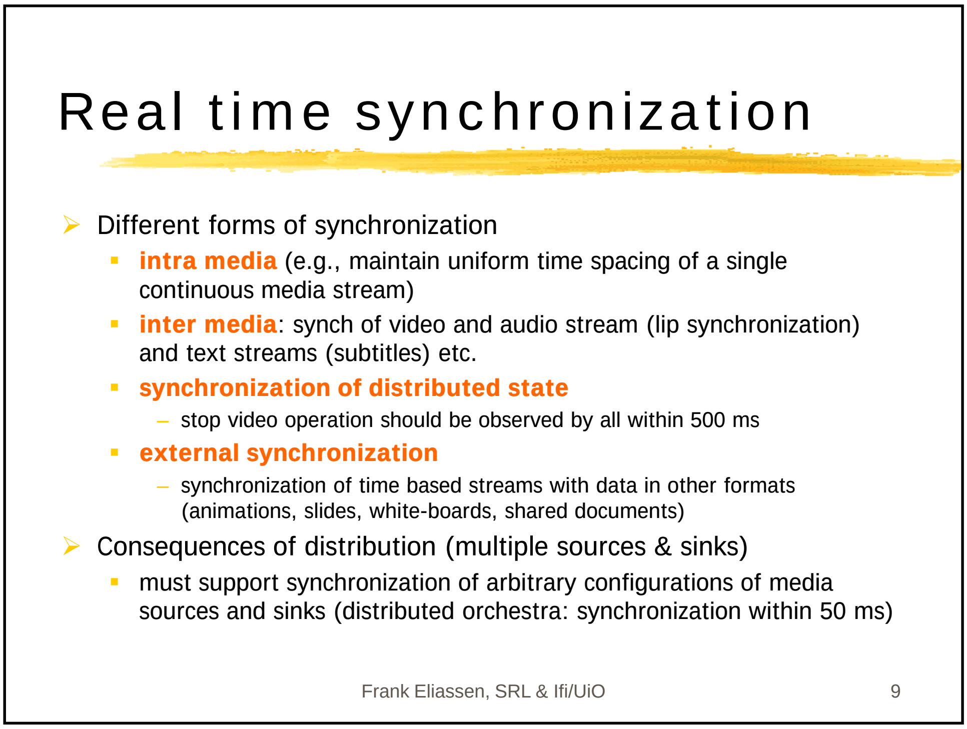
Support for multimedia: Systems support

- Commitments
  - Continuous media requires a commitment to provide a given level of service
    - e.g.,. 25 frames per second of video
  - This commitment must last for the whole life time of the interaction

General architecture for QoS-aware streaming

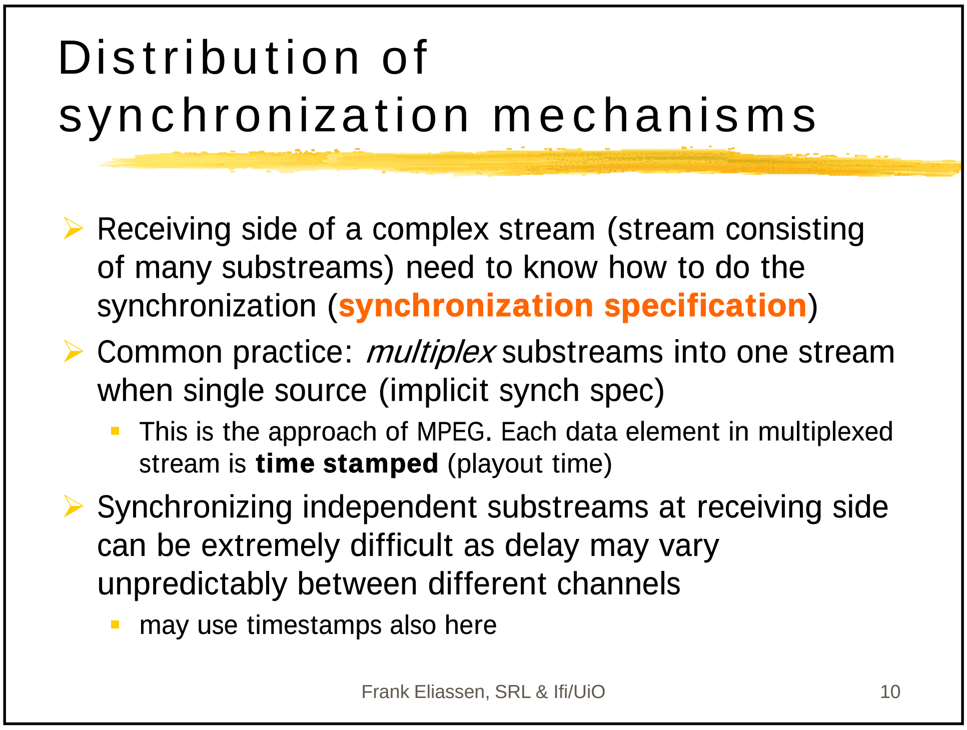
- QoS-aware streaming of stored multimedia data over a network





Real time synchronization

- Different forms of synchronization
  - **intra media** (e.g., maintain uniform time spacing of a single continuous media stream)
  - **inter media**: synch of video and audio stream (lip synchronization) and text streams (subtitles) etc.
  - **synchronization of distributed state**
    - stop video operation should be observed by all within 500 ms
  - **external synchronization**
    - synchronization of time based streams with data in other formats (animations, slides, white-boards, shared documents)
- Consequences of distribution (multiple sources & sinks)
  - must support synchronization of arbitrary configurations of media sources and sinks (distributed orchestra: synchronization within 50 ms)

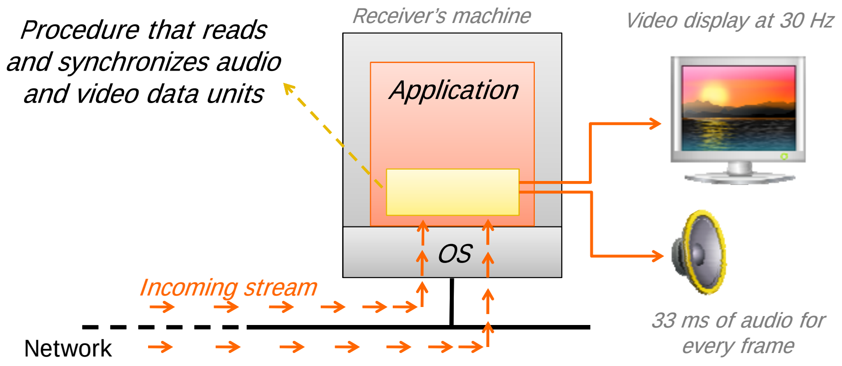


Distribution of synchronization mechanisms

- Receiving side of a complex stream (stream consisting of many substreams) need to know how to do the synchronization (**synchronization specification**)
- Common practice: *multiplex* substreams into one stream when single source (implicit synch spec)
  - This is the approach of MPEG. Each data element in multiplexed stream is **time stamped** (playout time)
- Synchronizing independent substreams at receiving side can be extremely difficult as delay may vary unpredictably between different channels
  - may use timestamps also here

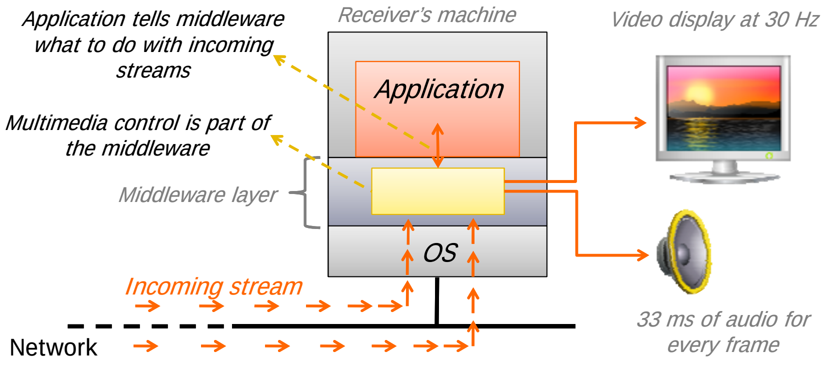
Synchronization mechanisms (1/2)

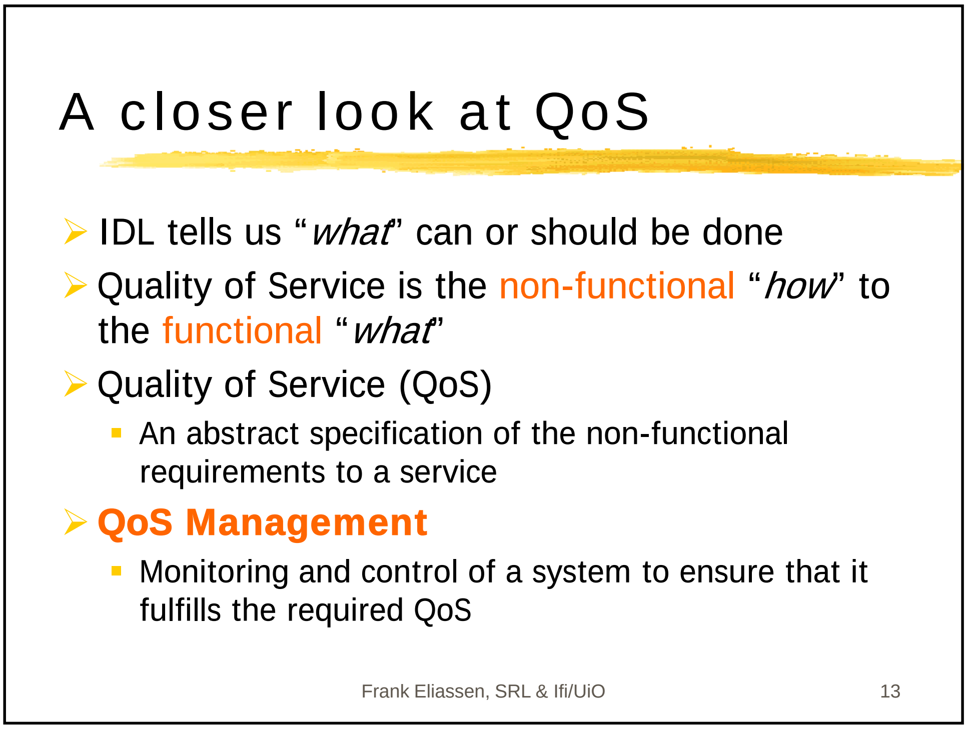
- The principle of explicit synchronization on the level of data units.


Procedure that reads and synchronizes audio and video data units
Receiver's machine
Application
OS
Video display at 30 Hz
33 ms of audio for every frame
Incoming stream
Network

Synchronization mechanisms (2/2)

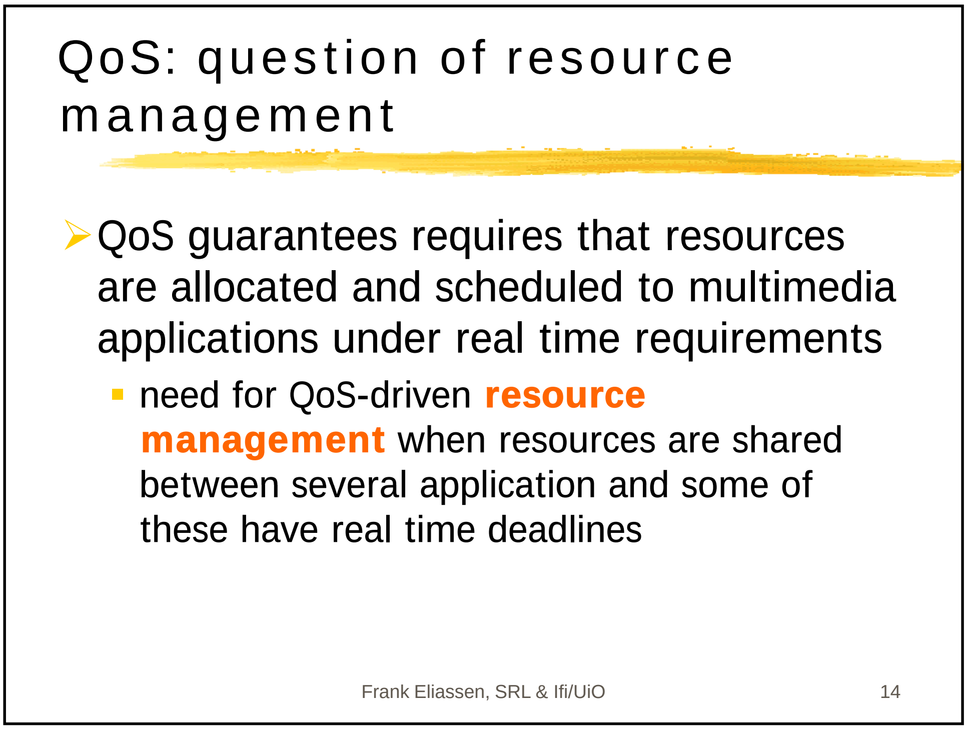
- The principle of synchronization as supported by high level interfaces.


Application tells middleware what to do with incoming streams
Multimedia control is part of the middleware
Middleware layer
Receiver's machine
Application
OS
Video display at 30 Hz
33 ms of audio for every frame
Incoming stream
Network



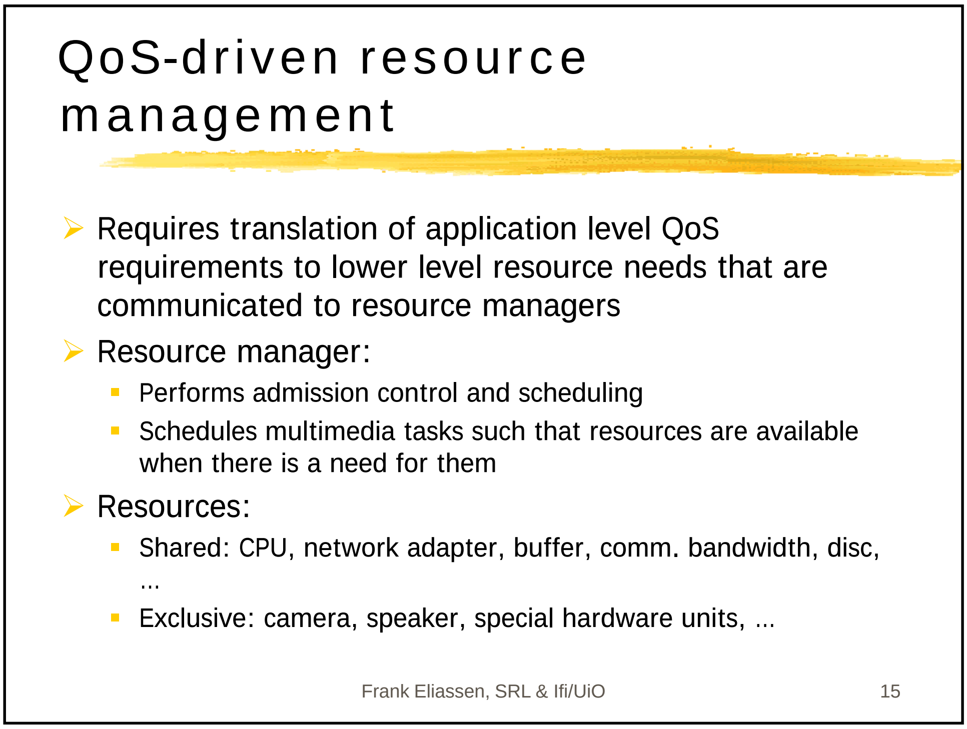
A closer look at QoS

- IDL tells us "*what*" can or should be done
- Quality of Service is the non-functional "*how*" to the functional "*what*"
- Quality of Service (QoS)
  - An abstract specification of the non-functional requirements to a service
- **QoS Management**
  - Monitoring and control of a system to ensure that it fulfills the required QoS



QoS: question of resource management

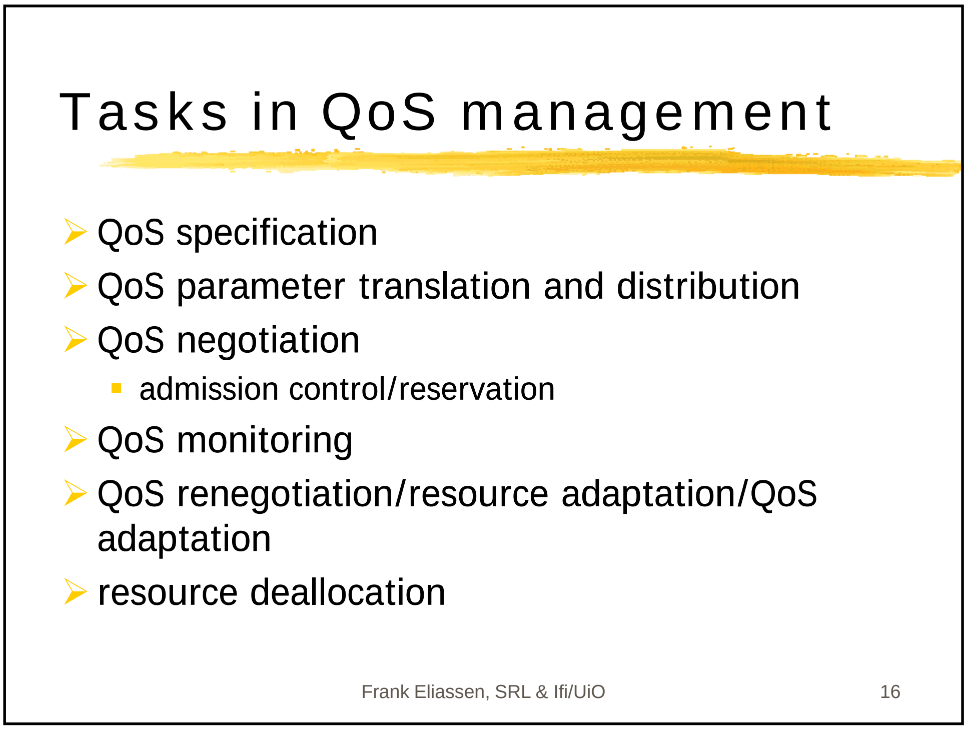
- QoS guarantees requires that resources are allocated and scheduled to multimedia applications under real time requirements
  - need for QoS-driven **resource management** when resources are shared between several application and some of these have real time deadlines



QoS-driven resource management

- Requires translation of application level QoS requirements to lower level resource needs that are communicated to resource managers
- Resource manager:
  - Performs admission control and scheduling
  - Schedules multimedia tasks such that resources are available when there is a need for them
- Resources:
  - Shared: CPU, network adapter, buffer, comm. bandwidth, disc, ...
  - Exclusive: camera, speaker, special hardware units, ...

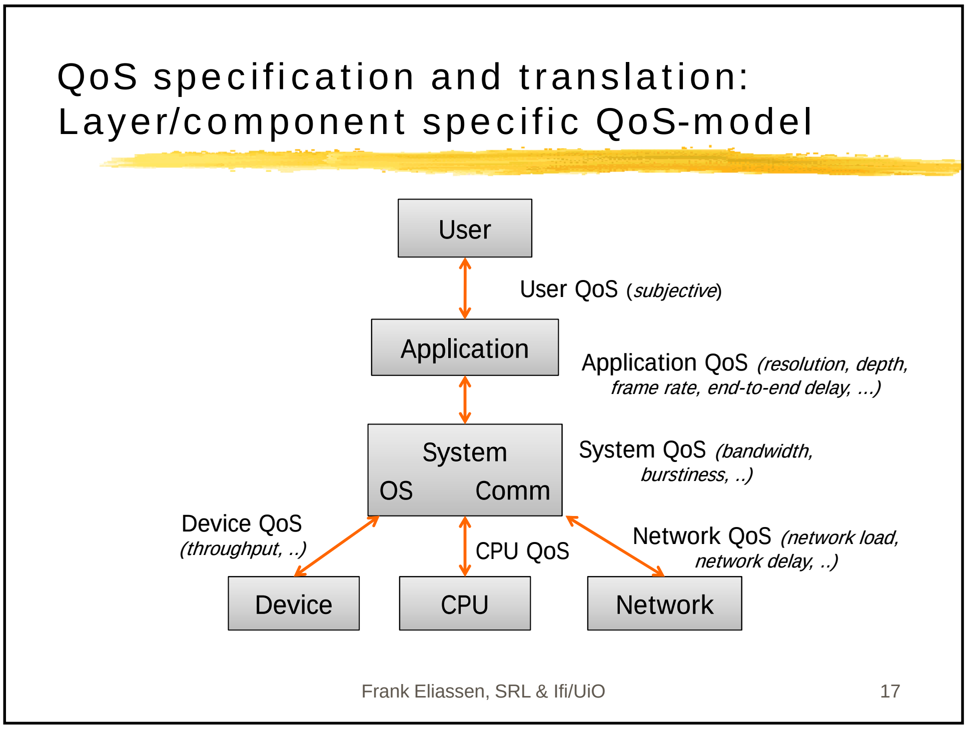
Frank Eliassen, SRL & Ifi/UiO 15



Tasks in QoS management

- QoS specification
- QoS parameter translation and distribution
- QoS negotiation
  - admission control/reservation
- QoS monitoring
- QoS renegotiation/resource adaptation/QoS adaptation
- resource deallocation

Frank Eliassen, SRL & Ifi/UiO 16



QoS specification and translation: Layer/component specific QoS-model

User

User QoS (*subjective*)

Application

Application QoS *(resolution, depth, frame rate, end-to-end delay, ...)*

System

OS

Comm

System QoS *(bandwidth, burstiness, ..)*

Device QoS *(throughput, ..)*

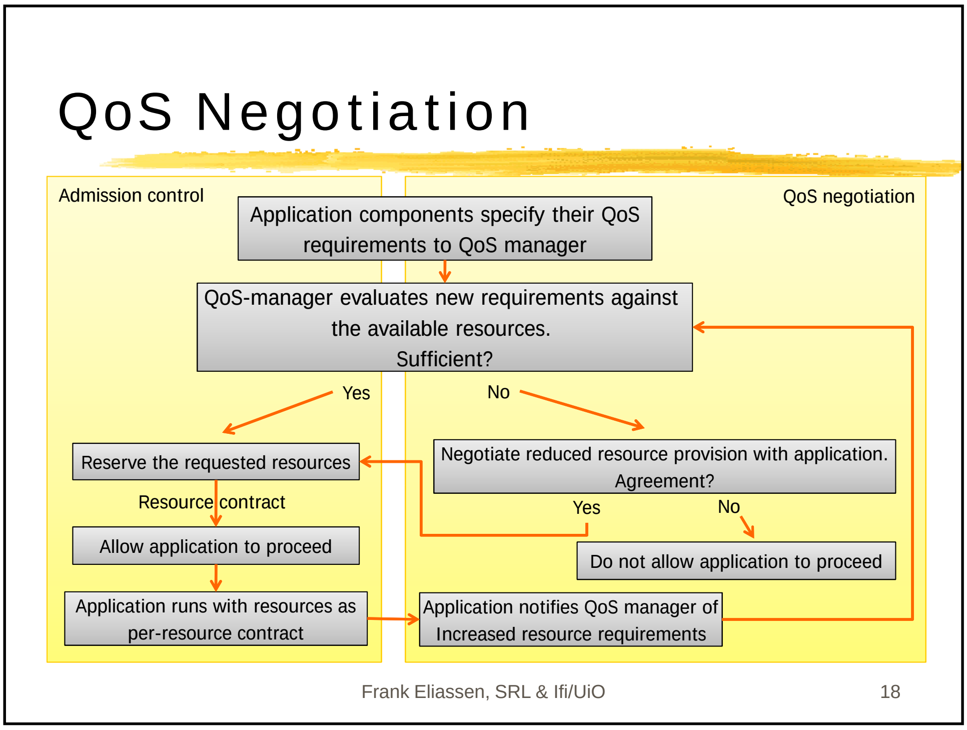
CPU QoS

Network QoS *(network load, network delay, ..)*

Device

CPU

Network



QoS Negotiation

Admission control

QoS negotiation

Application components specify their QoS requirements to QoS manager

QoS-manager evaluates new requirements against the available resources. Sufficient?

Yes

No

Reserve the requested resources

Resource contract

Allow application to proceed

Application runs with resources as per-resource contract

Negotiate reduced resource provision with application. Agreement?

Yes

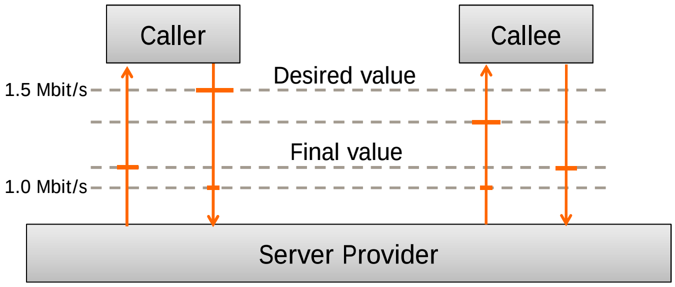
No

Do not allow application to proceed

Application notifies QoS manager of Increased resource requirements

Example QoS negotation

- For each parameter, specify
  - desired value and lowest acceptable value
- Ex.: Bandwidth : {1.5Mbit/s,1.0Mbit/s}



QoS models for streaming

- Usually expressed as a set of QoS categories and dimensions
- **QoS dimension** – an aspect of QoS that can be measured on a stream
  - delay, throughput, ...
- **QoS category** – a grouping of QoS dimensions
  - Represents a type of user or application requirements
- Example (**QML**)

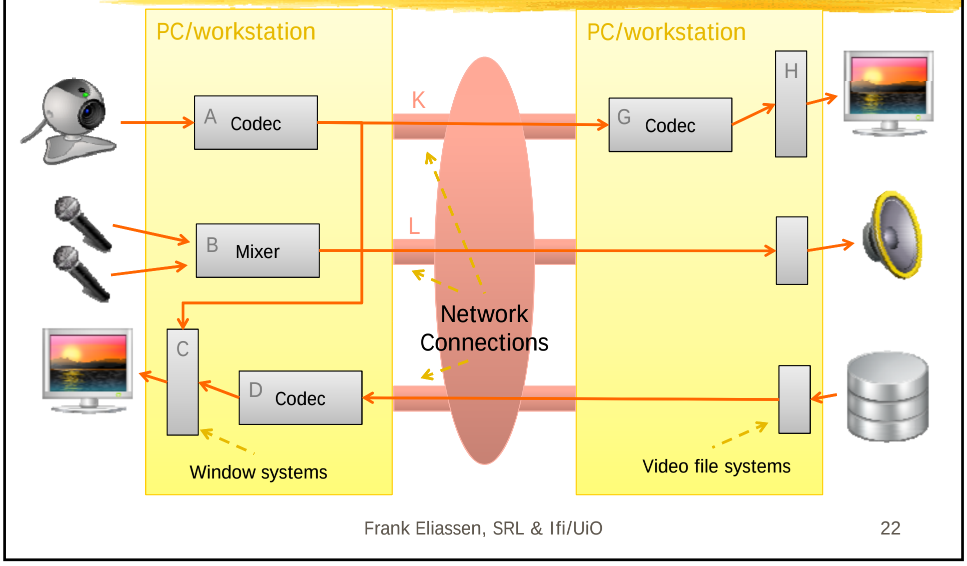
```
type Performance = contract {
    delay: decreasing numeric msec;
    throughput: increasing numeric mb/sec;
};
```

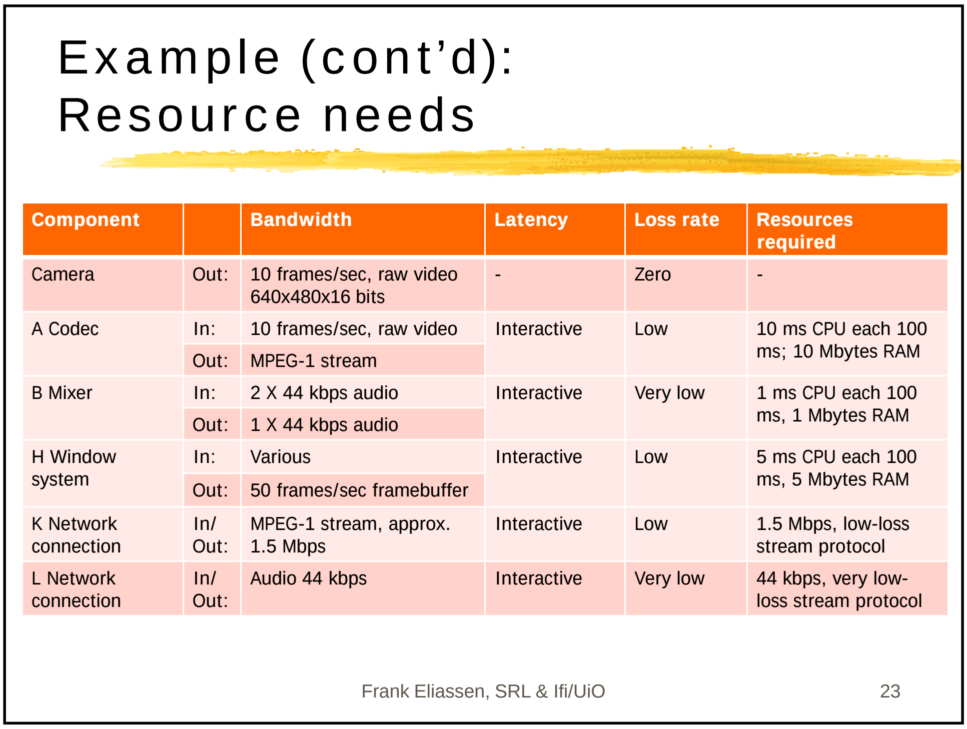
QoS categories for streaming

QoS categories	Ex. QoS-dimentions for stream interaction	Ex. QoS-dimensions for discrete interaction
Timeliness	End-to-end delay, max allowed jitter	End-to-end delay per interaction
Volume	Observed throughput as frames per second	Observed throughput as bytes per second
Reliability	% frame loss, bit error rate per frame	bit error rate in individual interactions

Varying commitment levels: “best effort” vs. guaranteed

Example: Resource needs A/V streaming app.

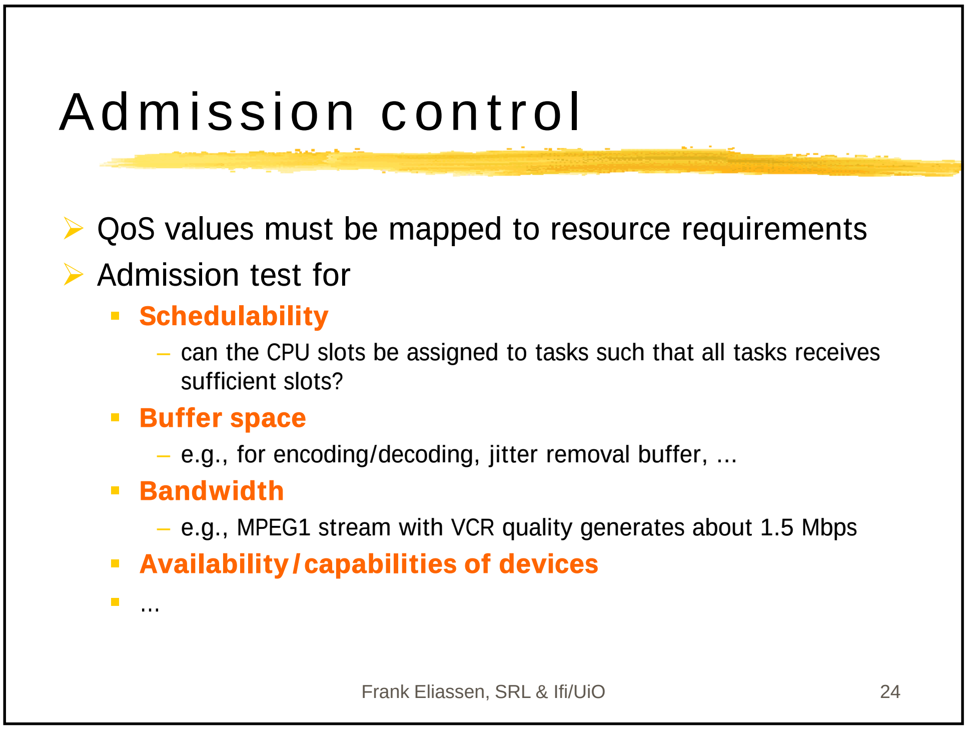




Example (cont’d): Resource needs

Component		Bandwidth	Latency	Loss rate	Resources required
Camera	Out:	10 frames/sec, raw video 640x480x16 bits	-	Zero	-
A Codec	In:	10 frames/sec, raw video	Interactive	Low	10 ms CPU each 100 ms; 10 Mbytes RAM
	Out:	MPEG-1 stream			
B Mixer	In:	2 X 44 kbps audio	Interactive	Very low	1 ms CPU each 100 ms, 1 Mbytes RAM
	Out:	1 X 44 kbps audio			
H Window system	In:	Various	Interactive	Low	5 ms CPU each 100 ms, 5 Mbytes RAM
	Out:	50 frames/sec framebuffer			
K Network connection	In/ Out:	MPEG-1 stream, approx. 1.5 Mbps	Interactive	Low	1.5 Mbps, low-loss stream protocol
L Network connection	In/ Out:	Audio 44 kbps	Interactive	Very low	44 kbps, very low-loss stream protocol

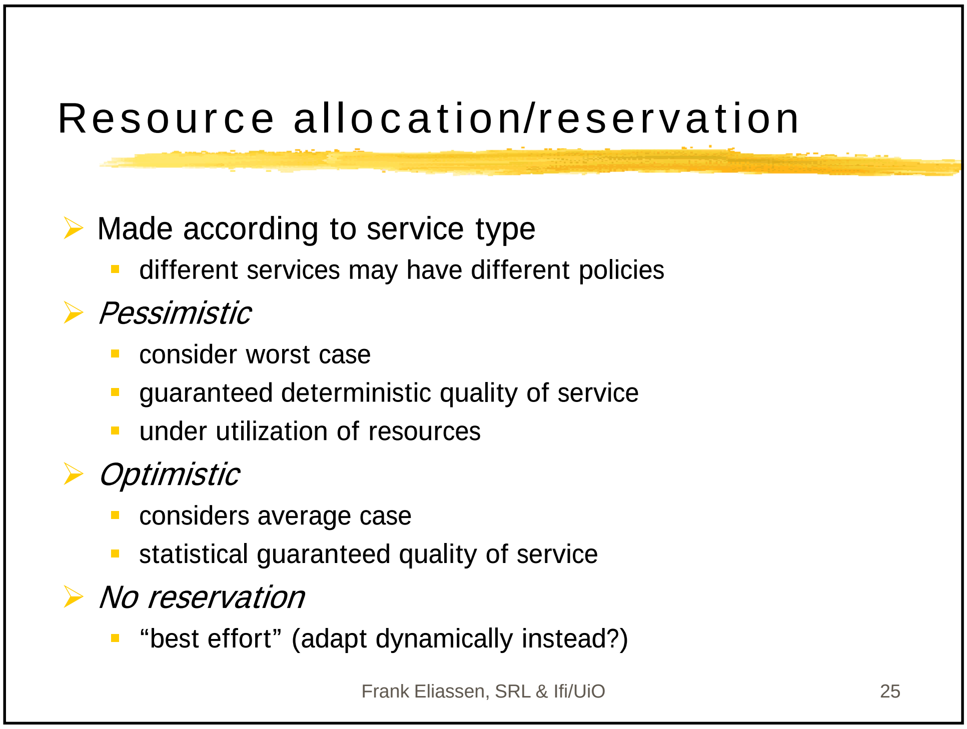
Frank Eliassen, SRL & Ifi/UiO 23



Admission control

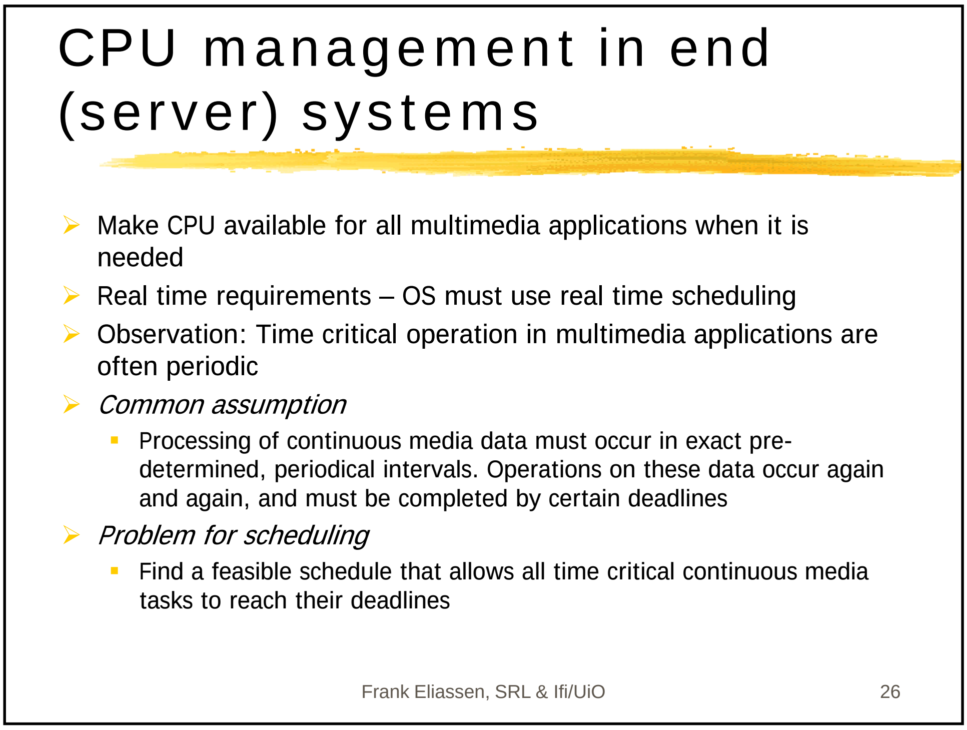
- QoS values must be mapped to resource requirements
- Admission test for
  - **Schedulability**
    - can the CPU slots be assigned to tasks such that all tasks receives sufficient slots?
  - **Buffer space**
    - e.g., for encoding/decoding, jitter removal buffer, …
  - **Bandwidth**
    - e.g., MPEG1 stream with VCR quality generates about 1.5 Mbps
  - **Availability/capabilities of devices**
  - …

Frank Eliassen, SRL & Ifi/UiO 24



Resource allocation/reservation

- Made according to service type
  - different services may have different policies
- *Pessimistic*
  - consider worst case
  - guaranteed deterministic quality of service
  - under utilization of resources
- *Optimistic*
  - considers average case
  - statistical guaranteed quality of service
- *No reservation*
  - “best effort” (adapt dynamically instead?)



CPU management in end (server) systems

- Make CPU available for all multimedia applications when it is needed
- Real time requirements – OS must use real time scheduling
- Observation: Time critical operation in multimedia applications are often periodic
- *Common assumption*
  - Processing of continuous media data must occur in exact pre-determined, periodical intervals. Operations on these data occur again and again, and must be completed by certain deadlines
- *Problem for scheduling*
  - Find a feasible schedule that allows all time critical continuous media tasks to reach their deadlines

EDF and RM

- Two algorithms for scheduling of periodic tasks
- *Earliest Deadline First (EDF)*
  - Tasks with the earliest deadline have highest priority
  - Dynamic and optimal algorithm
    - by arrival of new task, must calculate a new priority order
- *Rate Monotonic (RM)*
  - Tasks with shortest period have highest priority
  - Optimal for periodic tasks
- *Deadline violations*
  - aborts task that can not reach their deadlines
  - application specific handling by suitable language mechanisms (e.g., callbacks)

Streaming over the Internet

- Characteristics of the Internet
  - Internet is based on TCP/IP (Transmission Control Protocol / Internet Protocol)
  - TCP/IP
    - is robust
    - is implemented over most network types
    - enable a wide spectrum of applications (file transfer, email, distributed computing, etc.)
    - preserves content (retransmisson)

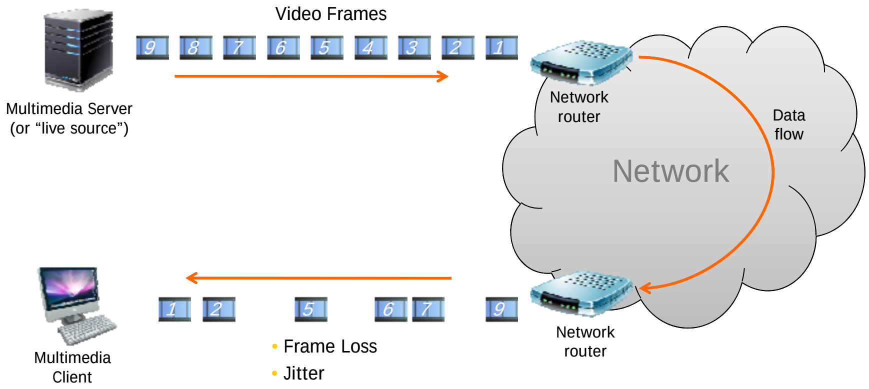
Unfortunately ...

- ... time based continuous media and Internet as we know it, is not a perfect match:
  - Internet is based on the principle of “best effort”
    - provides no guaranties wrt bandwidth and delay!!
  - No assumptions is made regarding underlying hardware
  - In contrast, satisfaction of requirements to streaming of continuous media depends on knowledge about available resources

Protocols for streaming media

- *Transport protocols*
  - Provides end-to-end network transport functions for streaming applications
  - E.g. **RTP** and **RTCP**
    - Implemented on top of **UDP** / **TCP**
- *Session control protocols*
  - Defines the messages and procedures to control the delivery of the multimedia data during an established session
    - Define control sequences, such as SETUP, PLAY and PAUSE
  - E.g. **RTSP** (uses RTP) and **SIP** (Independent of transport protocol)

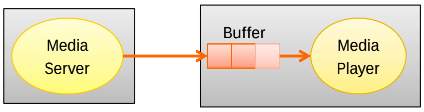
Quality degradation in networks

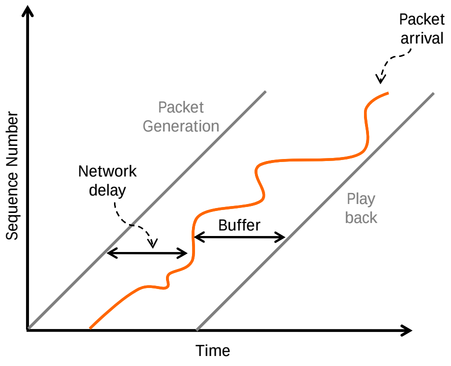


- Frame Loss
- Jitter

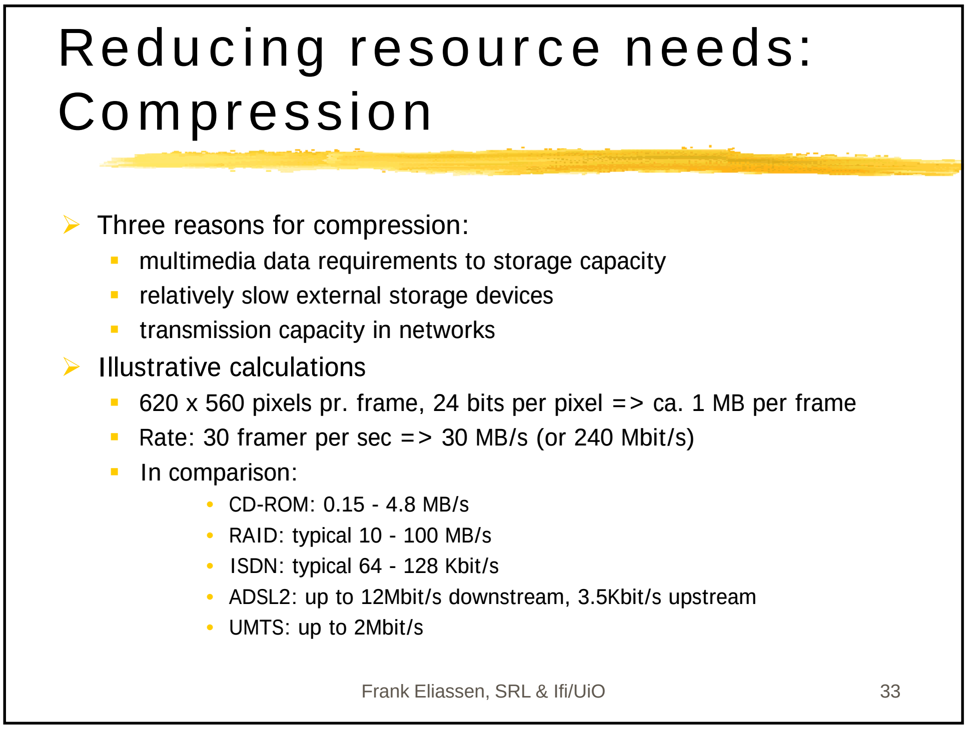
Compensating for jitter

- Playout buffer:



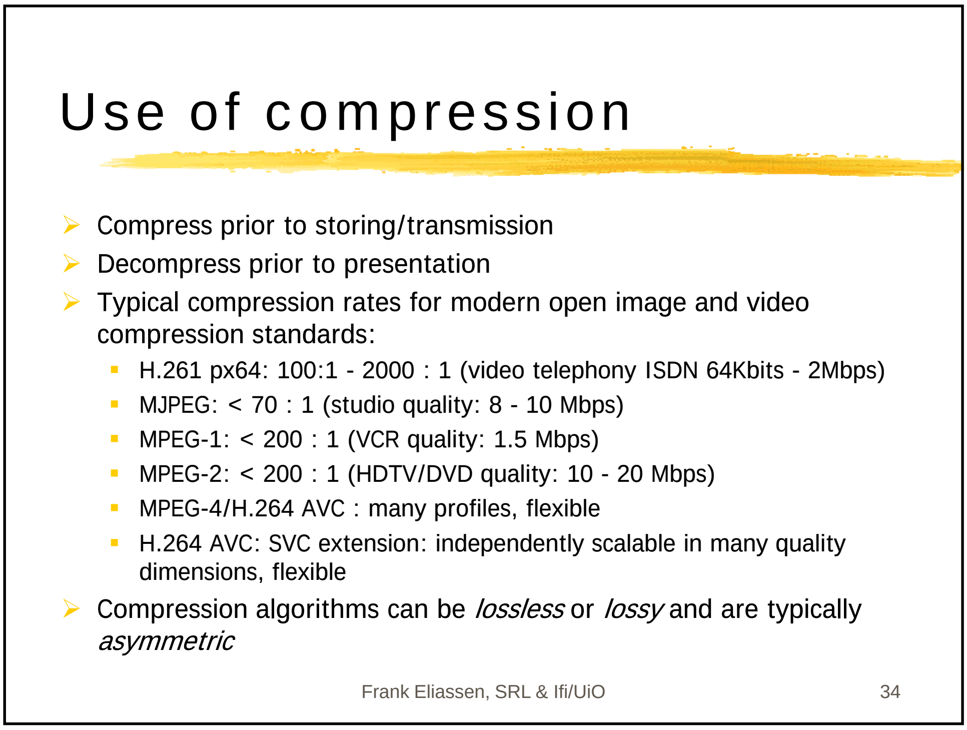


- Converts variable network delay (“jitter”) into fixed delay
- Typical method for Internet streaming
- Real Networks, Windows Media Player, QuickTime



Reducing resource needs: Compression

- Three reasons for compression:
  - multimedia data requirements to storage capacity
  - relatively slow external storage devices
  - transmission capacity in networks
- Illustrative calculations
  - 620 x 560 pixels pr. frame, 24 bits per pixel => ca. 1 MB per frame
  - Rate: 30 framer per sec => 30 MB/s (or 240 Mbit/s)
  - In comparison:
    - CD-ROM: 0.15 - 4.8 MB/s
    - RAID: typical 10 - 100 MB/s
    - ISDN: typical 64 - 128 Kbit/s
    - ADSL2: up to 12Mbit/s downstream, 3.5Kbit/s upstream
    - UMTS: up to 2Mbit/s

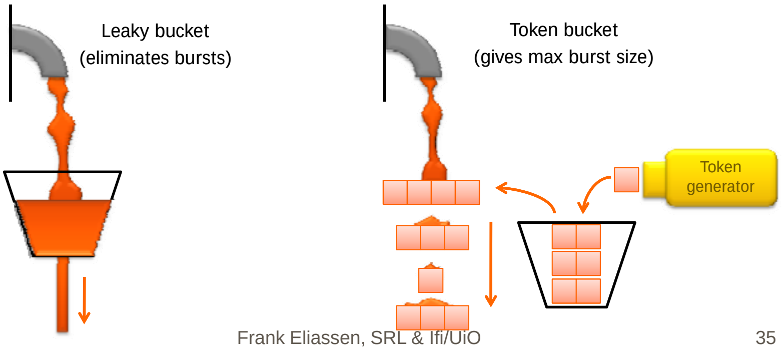


Use of compression

- Compress prior to storing/transmission
- Decompress prior to presentation
- Typical compression rates for modern open image and video compression standards:
  - H.261 px64: 100:1 - 2000 : 1 (video telephony ISDN 64Kbits - 2Mbps)
  - MJPEG: < 70 : 1 (studio quality: 8 - 10 Mbps)
  - MPEG-1: < 200 : 1 (VCR quality: 1.5 Mbps)
  - MPEG-2: < 200 : 1 (HDTV/DVD quality: 10 - 20 Mbps)
  - MPEG-4/H.264 AVC : many profiles, flexible
  - H.264 AVC: SVC extension: independently scalable in many quality dimensions, flexible
- Compression algorithms can be *lossless* or *lossy* and are typically *asymmetric*

Handling "stream burstiness" by traffic shaping

- Compression leads to temporal variation in bandwidth consumption
- Regulating the degree of variation in bandwidth consumption of a stream (burst: #media packets with too early arrival)
- Regulating by "smoothing" buffer at sender side



QoS parameters for streams: RFC 1363 flow spec

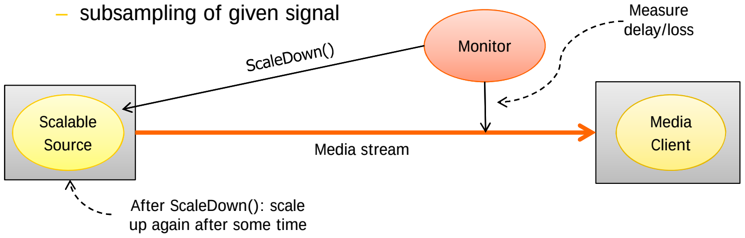
	Protocol version
Bandwidth including degree of burstiness	Max transmission unit
	Token bucket rate
	Token bucket size
	Max transmission rate
Minimum delay and and max acceptable jitter	Min delay noticed
	Max delay variation
Total no of acceptable losses over given time interval, plus max no of consecutive message losses	Loss sensitivity
	Burst loss sensitivity
	Loss interval
	Quality of service guarantee

Compensating for variation in bandwidth: Stream adaptation

- When QoS can not be guaranteed
  - applications must adapt to changes in resource availability
  - for continuous media streams: adjust presentation quality
- Basis for adaptation
  - drop some of the data
- Insufficient bandwidth and no video data is dropped
  - => arbitrary data is lost (=> visual noise in video)

Media scaling

- Adapt a stream to available bandwidth
  - simplest for “live” streaming
    - can dynamically choose encoding
  - for stored streams
    - depends on encoding method what forms of scaling that are possible
  - Approach
    - subsampling of given signal



    - Issue: who decides how to scale – Sender or Receiver?

Video scaling (1/2)

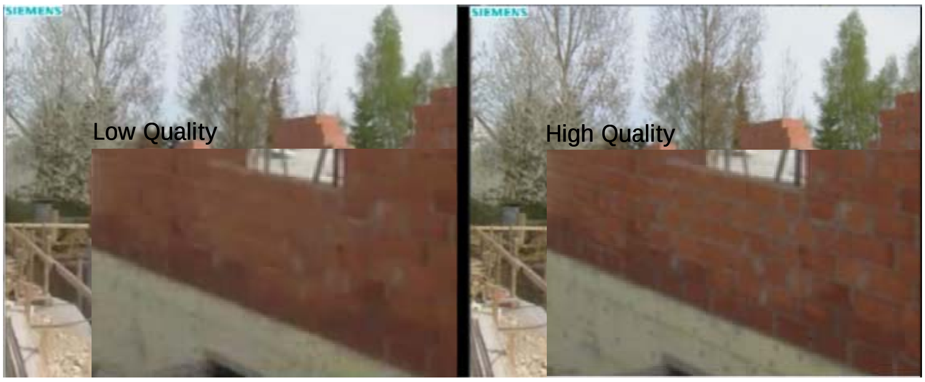
- Temporal scaling
  - reduce frame rate
  - simplest for streams based on intra frame coding (e.g., Motion JPEG)
  - more complex for streams based on inter frame coding (delta compression) which are most modern encoding schemes
- Spatial scaling
  - reduce no of pixels in each frame in video-stream
  - (often) based on hierarchical encoding (e.g., JPEG and MPEG-2)
- Quality/SNR scaling (Frequency scaling)
  - filtering higher frequencies in video signal
  - implies loss of quality (i.e. loss of details)
- SVC extension of H264 AVC combines all of the above

Video scaling (2/2)

- Amplitude scaling
  - reduce color depth for every pixel
  - e.g., used in H.261 to achieve constant bandwidth
- Color space scaling
  - reduce resolution of color space (reduce pixmap)
  - e.g, switch from color to grey scale

Video scaling example

- Example:
  - H.264 Quality scaling

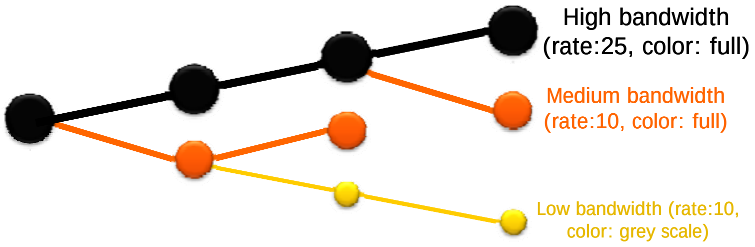


Continuous media distribution services

- Usually overlay networks (on top of IP) designed with the aim providing QoS and delivering continuous media to many receivers in a cost effective way
- Different content to many receivers (like in VOD)
  - **Content replication**: caching, mirroring (e.g., Akamai)
- Same content to many receivers (like in broadcast)
  - **Application level multicast** (IP multicast not ubiquitous)
  - **P2P streaming** (CoolStreaming)
- Same content to heterogeneous receivers
  - Adjust to resource poorest receiver, or
  - (Overlay) **Network filtering** (based on media scaling)

Media distribution service using network filtering

- Filtering in network (e.g., using overlay). Example:
  - Distribution-tree with filtering, adapting QoS to each receiver
  - applies scaling in every relevant node in path from sender to receiver



High bandwidth (rate:25, color: full)

Medium bandwidth (rate:10, color: full)

Low bandwidth (rate:10, color: grey scale)

  - Network filtering can also be used in P2P streaming

Summary

- Multimedia applications require mechanisms that enable them to handle large amounts of time dependent data
- Most important mechanism: QoS management
- QoS is a question of resource management
- Resource management implies
  - admission control
  - scheduling function
- When resources can not be reserved, adaptation (media scaling & rate control) is the (only) alternative