Distributed Multimedia systems:

INF 5040 autumn 2006

Lecturer: Frank Eliassen

Literature

- CDK4: Chapter 17
- TvS2: Chapter 4.4

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 processely 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, Simula & Ifi/UiO 3

Varying requirements

- Distributed continuous media applications have highly varying requirements to facilities from the underlying distributed systems platform
  - video on demand
  - video telephony
  - remote teaching
  - medical applications (telemedicine)
  - command and control systems

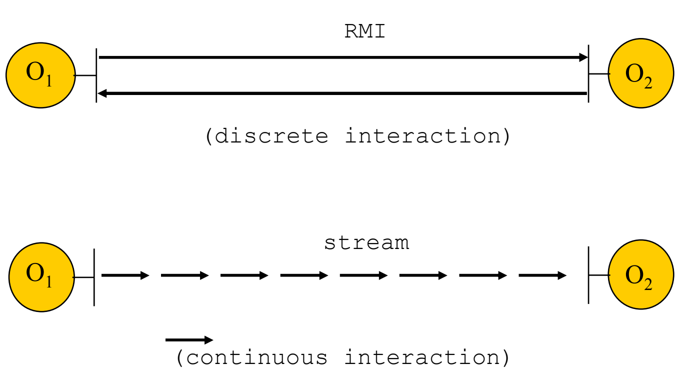
Frank Eliassen, Simula & Ifi/UiO 4

Key requirements from multimedia

- the need to represent continuous media in distributed systems
  - programming models
  - representation
- the need for real time synchronization mechanisms
- the need to specify and dynamically change the Quality of Service (QoS) of the transmission of continuous media
  - e.g., balance cost and quality

Frank Eliassen, Simula & Ifi/UiO 5

Support for continuous media: Programming models



Frank Eliassen, Simula & Ifi/UiO 6

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

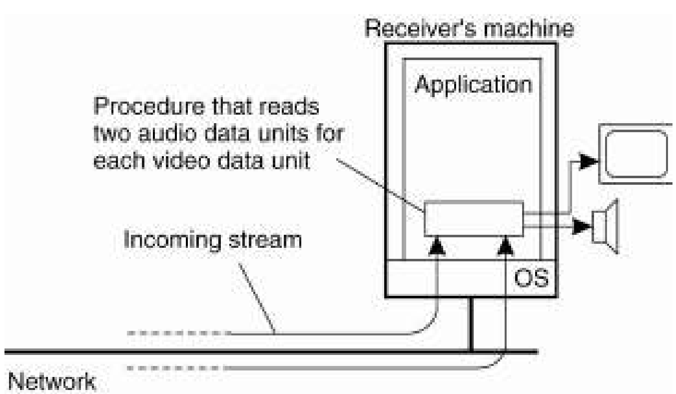
- Commitments
  - Continuous media requires a commitment to provide a given level of service
    - e.g.,. 25 frames per second of video
  - This commitment must last for the whole life time of the interaction

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)
  - 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, white-boards, shared documents)
- Consequences of distribution
  - must support synchronization of arbitrary configurations of media sources and sinks (distributed orchestra: synchronization within 50 ms)

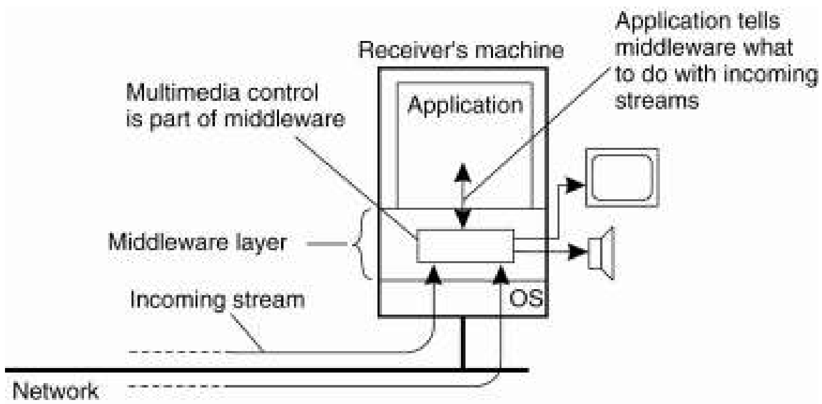
Synchronization mechanisms (1/2)

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



Synchronization mechanisms (2/2)

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



Frank Eliassen, Simula & Ifi/UiO 11

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 (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 between different channels

Frank Eliassen, Simula & Ifi/UiO 12

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

Frank Eliassen, Simula & Ifi/UiO 13

QoS: question of resource management

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

Frank Eliassen, Simula & Ifi/UiO 14

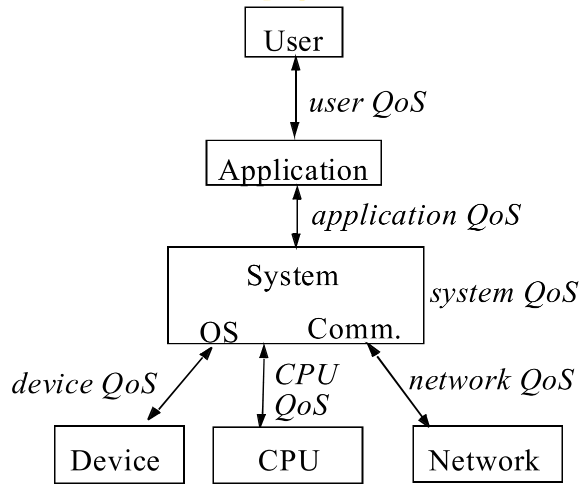
QoS-driven resource management

- 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, …

Tasks in QoS management

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

QoS specification: Layer/component specific QoS-model



QoS dimensions (1/3)

- User QoS dimensions
  - subjective
- Application QOS dimensions
  - media characteristics
    - resolution, depth, frame rate, …
  - transmission characteristics
    - end-to-end delay, …
  - media relations
    - skew, conversions (PAL ↔ NTSC)

QoS dimensions (2/3)

- System QoS dimensions (derived from application QoS)
  - bandwidth
  - burstiness
  - packet size
  - packet rate
  - delay (end-to-end, local)
  - jitter (variation in delay)
  - loss rate
  - ordered packet delivery
  - costs
  - ...

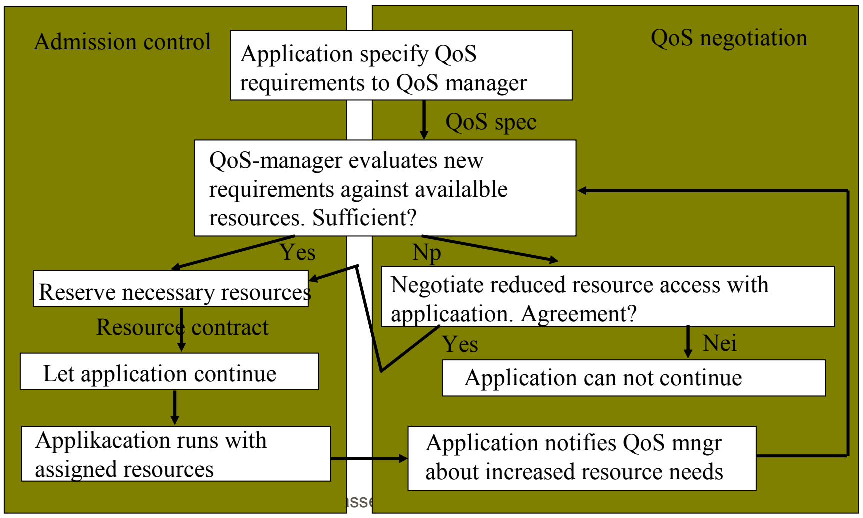
Frank Eliassen, Simula & Ifi/UiO 19

QoS dimensions (3/3)

- Network QoS dimensions (derived from system QoS)
  - network load (ave/min arrival time interval)
  - packet/cell size
  - latency connection establishment
  - network delay
  - Generally depends on type of network technology and service model (complicates QoS management)
- Device QoS dimension (derived from system QoS)
  - timeliness
  - throughput
- CPU QoS dimensions (derived from system QoS)
  - parameters depend on scheduling algorithm
  - period, deadline, priority, processing time per period, ...

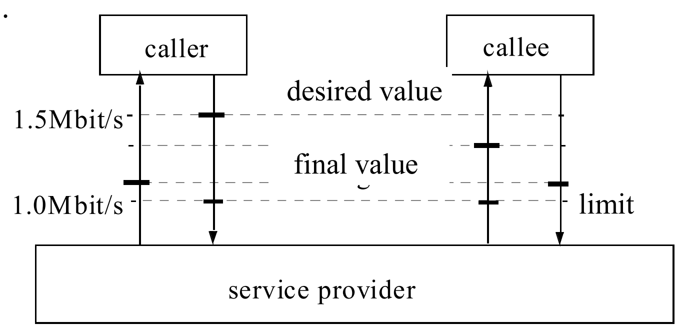
Frank Eliassen, Simula & Ifi/UiO 20

QoS negotiation



Example QoS negotation

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



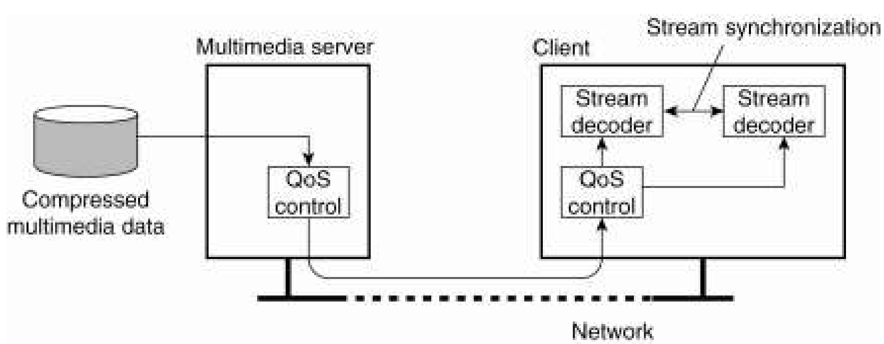
Streams and QoS

- Stream (sometimes called "flow")
  - QoS is fundamentally an end-to-end issue
  - A stream is the production, transmission, and the final consumption of a single continuous media type that is subject to a single end-to-end QoS statement

Frank Eliassen, Simula & Ifi/UiO 23

General architecture for QoS-aware streaming

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



Frank Eliassen, Simula & Ifi/UiO 24

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 dimentions
  - Represents a type of user or application requirements
- Example (QML)

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

Frank Eliassen, Simula & Ifi/UiO 25

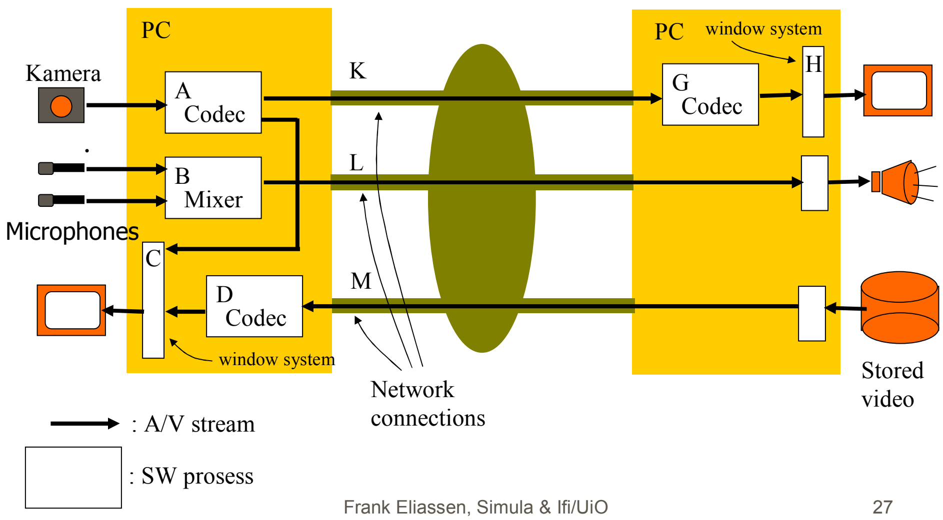
QoS categories for streaming

QoS categories	Ex. QoS-dimentions for stream interaction	Ex. QoS-dimentiones 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 committment levels: ”best effort” vs guaranteed

Frank Eliassen, Simula & Ifi/UiO 26

Example: resource needs A/V streaming app.



Frank Eliassen, Simula & Ifi/UiO 27

Example (cont'd): Resource needs

Component	Bandwidth	Latency	Loss rate	Resource needs
Camera	Out: 10 frames sec/raw video 640x480x16bits		Null	
A Codec	In: 10 rammer sec/raw video Out: MPEG-1 stream	Interactive	Low	10 ms CPU every 100 ms 10 Mbyte RAM
B Mixer	In: 2x44 Kbits/sec audio Out: 1x44 Kbits/sec audio	Interactive	Very low	1 ms CPU every 100 ms 1 Mbyte RAM
H Vindow-system	In: variabelt Out: 50 frames/sek framebuf.	Interactive	Low	5 ms CPU every 20 ms 5 Mbyte RAM
K Network connection	Inn/ut: MPEG-1 stream ca. 1.5 Mbits/sec	Interactive	Low	1.5 Mbits/sec, stream prococol w/low loss rate
L Network connection	Inn/ut: Audio 44Kbits/sek	Interactive	Very low	44 Kbits/sec, stream protocol w/ very low loss rate

Frank Eliassen, Simula & Ifi/UiO 28

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)

Frank Eliassen, Simula & Ifi/UiO 29

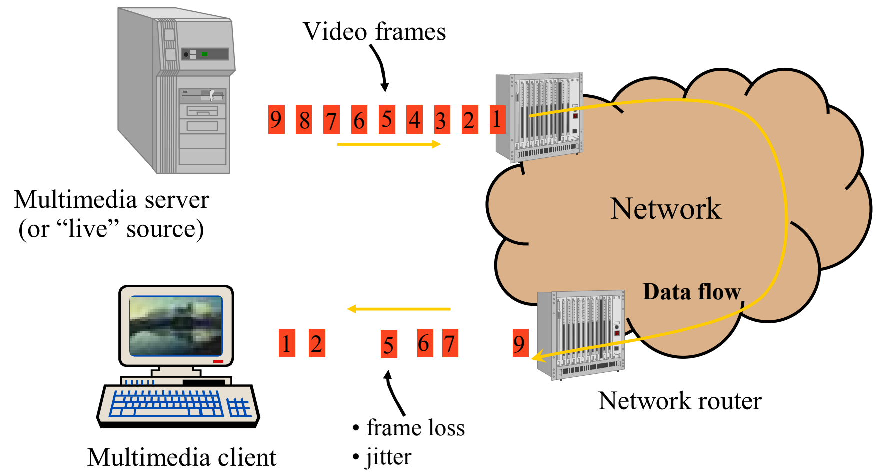
Unfortunately ...

… 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

Frank Eliassen, Simula & Ifi/UiO 30

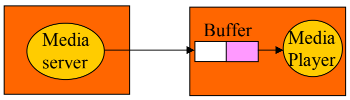
Quality degradation in networks



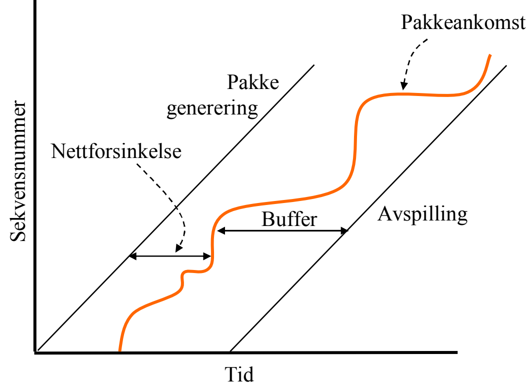
Frank Eliassen, Simula & Ifi/UiO 31

Compensating for jitter

Playout buffer:

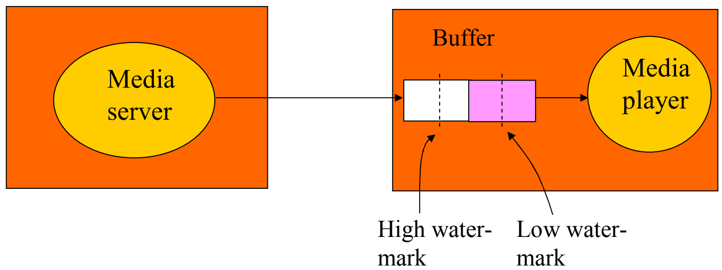


Typical method for Internet streaming (RealNetworks, Windows Media Player, QuickTime)



Frank Eliassen, Simula & Ifi/UiO 32

Streaming based on watermarks



- The server sends data as fast as possible until high watermark is reached in the buffer. The player then asks the server to pause
- When low watermark is reachedin the buffer, the player asks the server to send more data as above

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
    ADSL: typisk 2-6Mbit/s downstream,
    256-625 Kbit/s upstream
    UMTS: up to 2Mbit/s

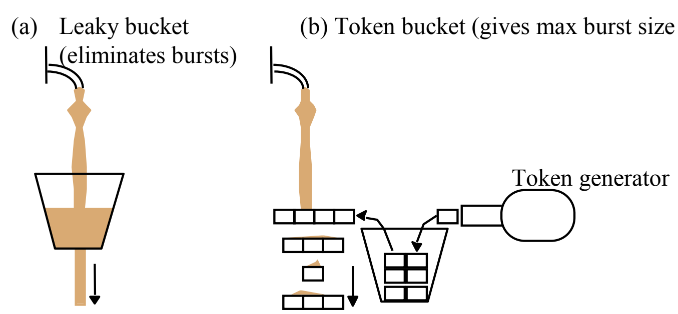
Use of compression

- Compress prior to storing/transmission
- Decompress prior to presentation
- Typical compression rates for modern open image and video compression standards:
  - JPEG: < 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.261 px64: 100:1 - 2000 : 1 (video telephony ISDN 64Kbits - 2Mbps)
- Compression algorithms can be *lossless* or *lossy* and are typically *asymmetric*

Frank Eliassen, Simula & Ifi/UiO 35

Handling "flow burstiness" by traffic shaping

- Regulating the degree of variation in bandwidth of a stream (burst: #media packet with to early arrival)
- Regulating by "smoothing" buffer at sender side



Frank Eliassen, Simula & Ifi/UiO 36

QoS parameters for streams: RFC 1363 flow spec

- Protocol version
- Max transmission unit
- Token bucket rate
- Token bucket size
- Max transmission rate
- Min delay noticed
- Max delay variation
- Loss sensitivity
- Burst loss sensitivity
- Loss intervall
- Quality of guarantee

Max transmission unit, Token bucket rate, Token bucket size, Max transmission rate: Båndbredde inklusive grad av "burstiness"

Min delay noticed, Max delay variation: Minimum forsinkelse og maks akseptabel jitter

Loss sensitivity, Burst loss sensitivity, Loss intervall: Totalt antall akseptable tap over gitt intervall, pluss maks antall etterfølgende meldingstap

Frank Eliassen, Simula & Ifi/UiO 37

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)

Frank Eliassen, Simula & Ifi/UiO 38

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
    - subsamling of given signal

Frank Eliassen, Simula & Ifi/UiO 39

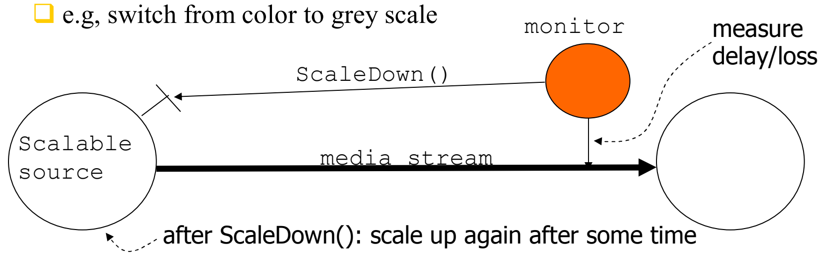
Video scaling (1/2)

- Temporal scaling
  - reduce frame rate
  - simplest for streams based on intra frame coding (e.g., Motion JPEG)
  - more complex for streams based på 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 coding (e.g., JPEG and MPEG-2)
- Quality/SNR scaling
  - filtering higher frequencies in video signal
  - implies loss of quality (i.e. loss of details)

Frank Eliassen, Simula & Ifi/UiO 40

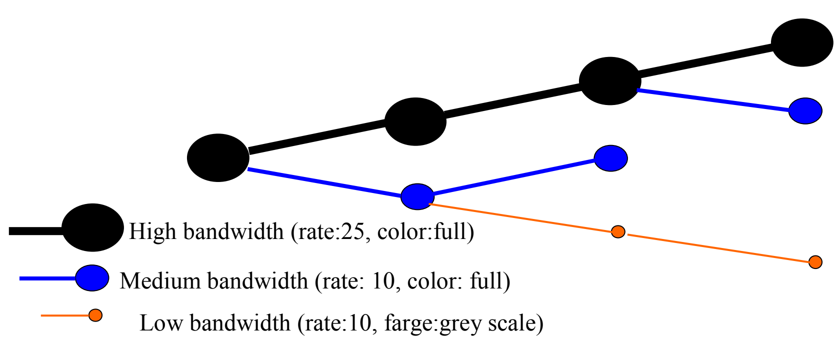
Video - scaling (2/2)

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



Media distribution (one-to-many) with heterogeneous receivers

- Requires 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



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?
    - EDF and RM CPU-scheduling vs "round-robin"
  - 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
  - …

Resource allocation/reservation

- Made acording to service type
  - different services may have different policies
- *Pessimistic*
  - consider worst case
  - guaranteed deterministic quality of service
  - under utilization of resources
- *Optimistic*
  - considers avaerage case
  - statistical guaranteed quality of service
- *no reservation*
  - "best effort"

Resource management in end systems

- Make CPU available for all multimedia applications when ut is needed
- Real time requirements ⇒ OS must use real time schedulering
  - "fair scheduling" ⇒ "best effort"
- 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

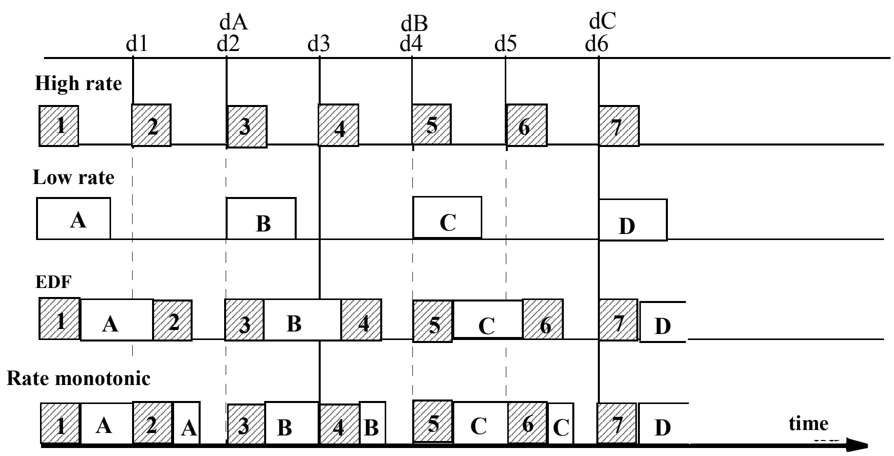
Resource allocation in Internet?

- IntServ: new service model for Internet
  - 3 classes of service, differently priced
  - Best effort service (as today's Internet)
  - Controlled-load service
    - network will appear lightly loaded
  - Guaranteed service
    - gives guaranteed bandwidth and max delay
  - Based on new protocols (RSVP and IPv6)
    - many open issues, including scalability issues and payment model
- Alternative model: DiffServ
  - All flows/packets aggregated into three different QoS classes

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)

EDF vs RM



Requirement to operating system

- Must be able to satisfy requirements both from (soft) real time applications and "elastic" applications
  - operatiing system must be able to apply several different scheduling algorithms concurrently
- Example: Two-level scheduling
  - Different scheduling algorithm are given a certain share of the CPU time
  - Each scheduling algorithm has responsibility for a group of tasks
  - Lowest level scheduling determines which scheduling algorithm that will run (in accordance with its CPU share)

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 is the (only) alternative

Adaptive video streaming demonstration

- If time allows ... scalable video demonstrator