Distributed Multimedia systems:

INF 5040/9040 autumn 2009

Lecturer: Frank Eliassen

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
Literature

- CDK4: Chapter 17
- TvS2: Chapter 4.4

- Recommended (not examinable):

Frank Eliassen, Simula & III/UIO

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, Simula & III/UIO
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

Support for continuous media: Programming models

(discrete interaction)

(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

![Diagram of QoS-aware streaming architecture]

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.
Synchronization mechanisms (2/2)

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

Lip sync using reactive handlers

Adapted from G. Blair, J.B. Stefani, Open Distributed Processing and Multimedia, Addison-Wesley, 1997

1 Adapted from G. Blair, J.B. Stefani, Open Distributed Processing and Multimedia, Addison-Wesley, 1997

Frank Eliassen, Simula & III/UiO
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

Frank Eliassen, Simula & II/UiO
**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/QoS adaptation
- resource deallocation
QoS specification and translation: 
Layer/component specific QoS-model

```
User

Application

System

OS

Comm.

Device

CPU

QoS

Network

user QoS (subjective)

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

system QoS (bandwidth, burstiness, ...)

device QoS (throughput, ...)

network QoS (network load, network delay)
```

Frank Eliassen, Simula & II/UiO

QoS negotiation

```
Admission control

Application specify QoS requirements to QoS manager

QoS-specification

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

Yes

Resource contract

Let application continue

Application runs with assigned resources

No

Negotiate reduced resource access with application. Agreement?

Yes

Application can not continue

No

Application notifies QoS manger about increased resource needs
```

INF 5040/9040 autumn 2009
Example QoS negotiation

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

```qml
type Performance = contract {
    delay: decreasing numeric msec;
    throughput: increasing numeric mb/sec;
};
```
# QoS categories for streaming

<table>
<thead>
<tr>
<th>QoS categories</th>
<th>Ex. QoS-dimensions for stream interaction</th>
<th>Ex. QoS-dimensions for discrete interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeliness</td>
<td>End-to-end delay, max allowed jitter</td>
<td>End-to-end delay per interaction</td>
</tr>
<tr>
<td>Volume</td>
<td>Observed throughput as frames per second</td>
<td>Observed throughput as bytes per second</td>
</tr>
<tr>
<td>Reliability</td>
<td>% frame loss, bit error rate per frame</td>
<td>bit error rate in individual interactions</td>
</tr>
</tbody>
</table>

## Varying commitment levels: "best effort" vs guaranteed

Frank Eliassen, Simula & UiO

---

## Example:
resource needs A/V streaming app.

![Diagram](image)

- **PC**: Computer
- **Codec**: Audio/Video codec
- **Mixer**: Audio mixer
- **G, H**: Other components
- **Network connections**: Data transmission paths
- **Stored video**: Resulting video content
- **A/V stream**: Audio/video content flow
- **SW process**: Software processes

Frank Eliassen, Simula & UiO
Example (cont’d):
Resource needs

<table>
<thead>
<tr>
<th>Component</th>
<th>Bandwidth</th>
<th>Latency</th>
<th>Loss rate</th>
<th>Resource needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>Out: 10 frames sec/raw video 640x480x16bits</td>
<td>Null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Codec</td>
<td>In: 10 frames sec/raw video Out: MPEG-1 stream</td>
<td>Interactive</td>
<td>Low</td>
<td>10 ms CPU every 100 ms 10 Mbyte RAM</td>
</tr>
<tr>
<td>B Mixer</td>
<td>In: 2x44 Kbits/sec audio Out: 1x44 Kbits/sec audio</td>
<td>Interactive</td>
<td>Very low</td>
<td>1 ms CPU every 100 ms 1 Mbyte RAM</td>
</tr>
<tr>
<td>H Window-system</td>
<td>In: variable Out: 50 frames/sec framebuf.</td>
<td>Interactive</td>
<td>Low</td>
<td>5 ms CPU every 20 ms 5 Mbyte RAM</td>
</tr>
<tr>
<td>K Network connection</td>
<td>In/out: MPEG-1 stream ca. 1.5 Mbits/sec</td>
<td>Interactive</td>
<td>Low</td>
<td>1.5 Mbits/sec, stream protocol w/low loss rate</td>
</tr>
<tr>
<td>L Network connection</td>
<td>In/out: Audio 44Kbits/sec</td>
<td>Interactive</td>
<td>Very low</td>
<td>44 Kbits/sec, stream protocol w/very low loss rate</td>
</tr>
</tbody>
</table>

Frank Eliassen, Simula & III/UiO

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 receive 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, Simula & III/UiO
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?)

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
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 (retransmission)
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 guarantees 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

Quality degradation in networks

Multimedia server (or “live” source)

Multimedia client

Video frames

Network

Data flow

Network router

• frame loss
• jitter

Frank Eliassen, Simula & UiO
Compensating for jitter

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

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 reached in 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 frames 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: typical 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:
  - 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
  - NEW: 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

(a) Leaky bucket (eliminates bursts)
(b) Token bucket (gives max burst size)

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 interval
- Quality of guarantee

Bandwidth including degree of burstiness
Minimum delay and max acceptable jitter
Total no of acceptable losses over given time interval, plus max no of consecutive message losses
Compensating for variation in bandwidth: Stream adaptation

- When QoS cannot 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
  - 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
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 (student presentation next week)
- 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

- Network filtering can also be used in P2P streaming

---

Media distribution service with in-network filtering

Frank Eliassen, Simula & III/UIO
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