



CS 414 – Multimedia Systems Design  
Lecture 12 – Introduction to  
Quality of Service

Klara Nahrstedt  
Spring 2008



# Administrative

- MP2 posted on 2/08
- Help session for MP2 on Wednesday, 2/13, in 3405 SC at 6pm



# Outline

- Requirements on Multimedia  
Communication and Operating Systems
- Real-time and multimedia
- Resource management
  - Resources
  - Quality of Service (QoS) Concept
  - Operations



# Requirements

- Resource management – resource reservation/allocation/adaptation
- Transport system – guaranteed delivery with respect to metrics such as delay, reliability, bandwidth requirements
- Process management – real-time processing of continuous data, communication and synchronization between processes/threads



# Requirements (2)

- Memory/Buffer management – guaranteed timing delay and efficient data manipulation
- File system/Media Servers – transparent and guaranteed continuous retrieval of audio/video
- Device management – integration of audio and video



# Resources Classification

- Resource is a system entity required by tasks for manipulating data
- Active versus Passive Resources (CPU – active; main memory – passive)
- Shared versus Exclusive Resources (CPU – shared; audio device – exclusive)
- Single Versus Multiple Resources (single CPU in PC, multiple CPU in multi-core system)




# Real-Time and Deadlines

- Real-time system – system in which correctness of computation depends not only on obtaining the right results, but also providing them on time
  - Examples: control of temperature in a chemical plant; control of a flight simulator
- Deadline – represents the latest acceptable time for the result delivery
  - Soft deadlines versus hard deadlines



# Real-Time and Multimedia

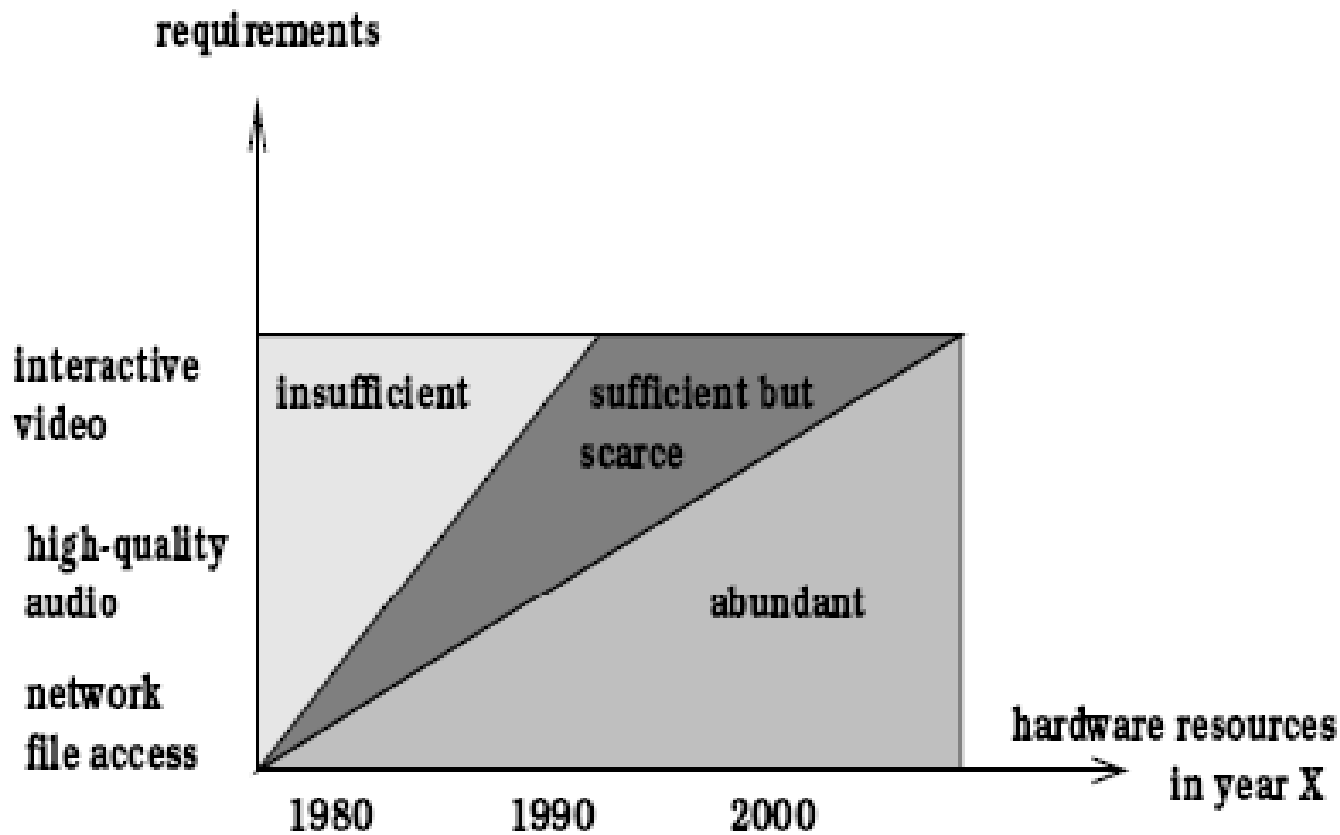
- Difference between RT requirements for traditional RT systems and Multimedia systems
  - Fault tolerance and security
  - Soft deadlines versus hard deadlines
  - Periodic behavior versus random behavior
  - Bandwidth requirements



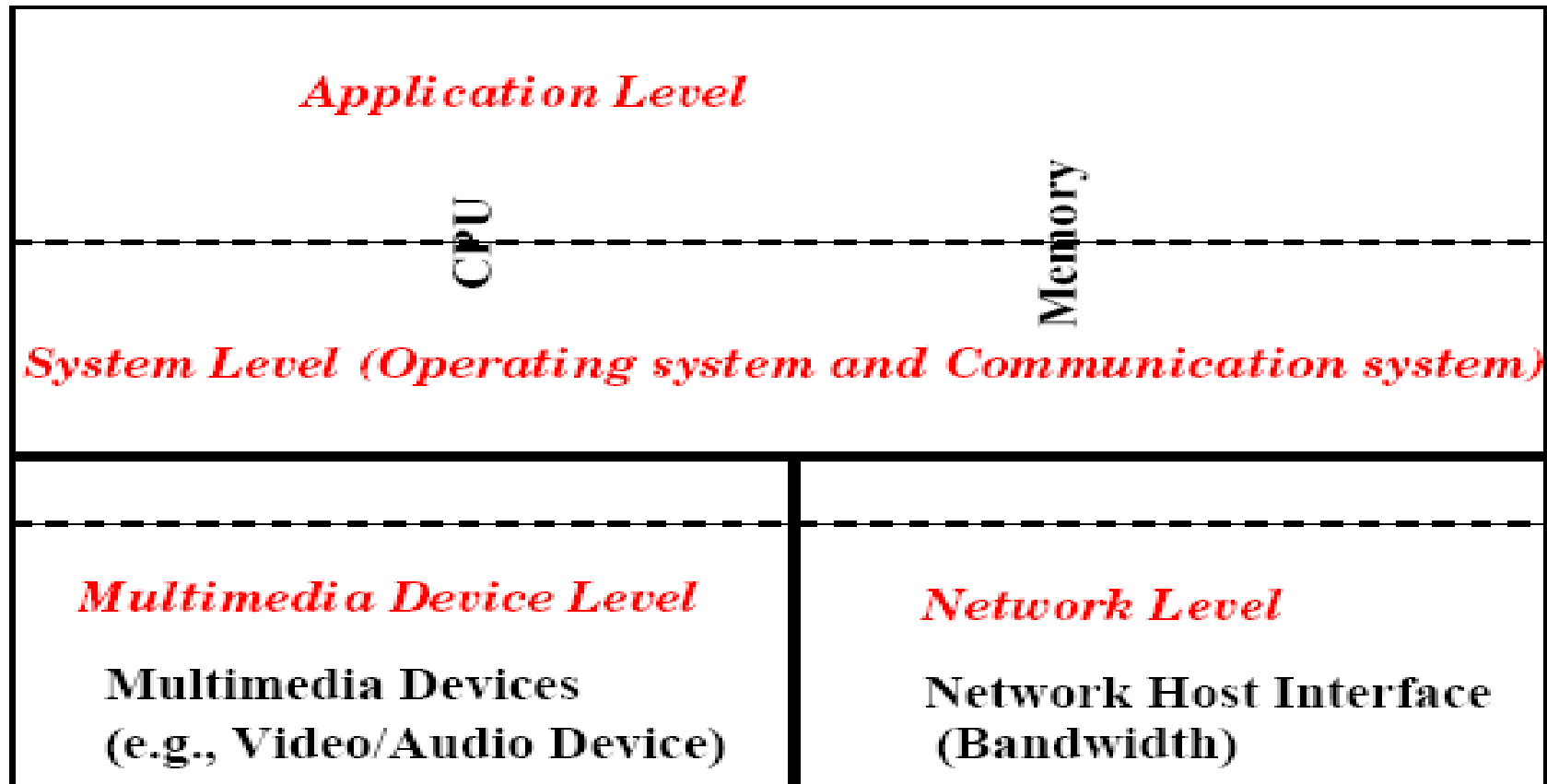
# Resource Management (Why do we need resource management?)

- Limited capacity in digital distributed systems despite data compression and usage of new technologies
- Need adherence for processing of continuous data by every hardware and software component along the data path
- Competition for resources exist in an integrated multimedia system

# Window of Insufficient Resources




# Layered Partition of Multimedia Systems with respect to Required Resources and Individual Services





# Quality of Service

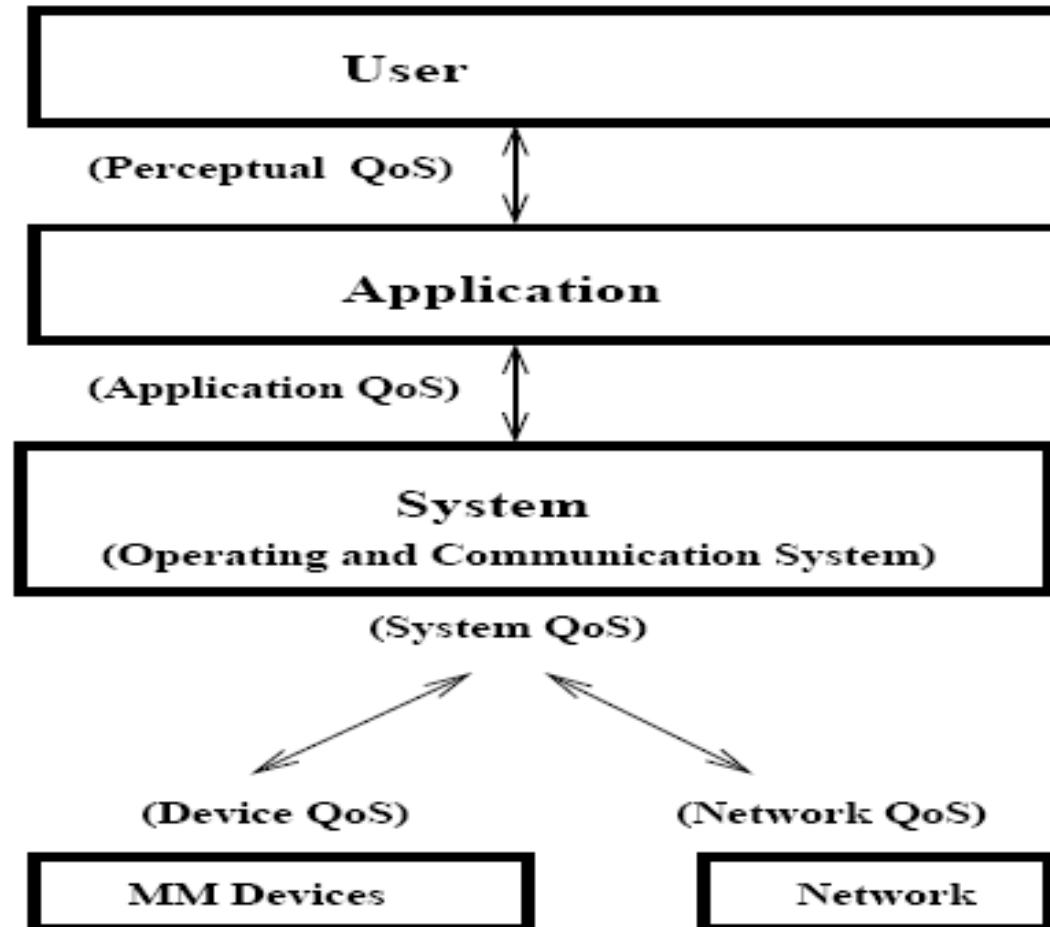
- MM systems consist of set of services
- To provide generic MM services, services get parameterized with so called **Quality of Service**
- Examples of QoS parameters:
  - QoS for Audio service:
    - Sample rate – 8000 samples/second
    - Sample resolution – 8 bits per sample
  - QoS for network service:
    - Throughput – 100 Mbps
    - Connection setup time – 50 ms



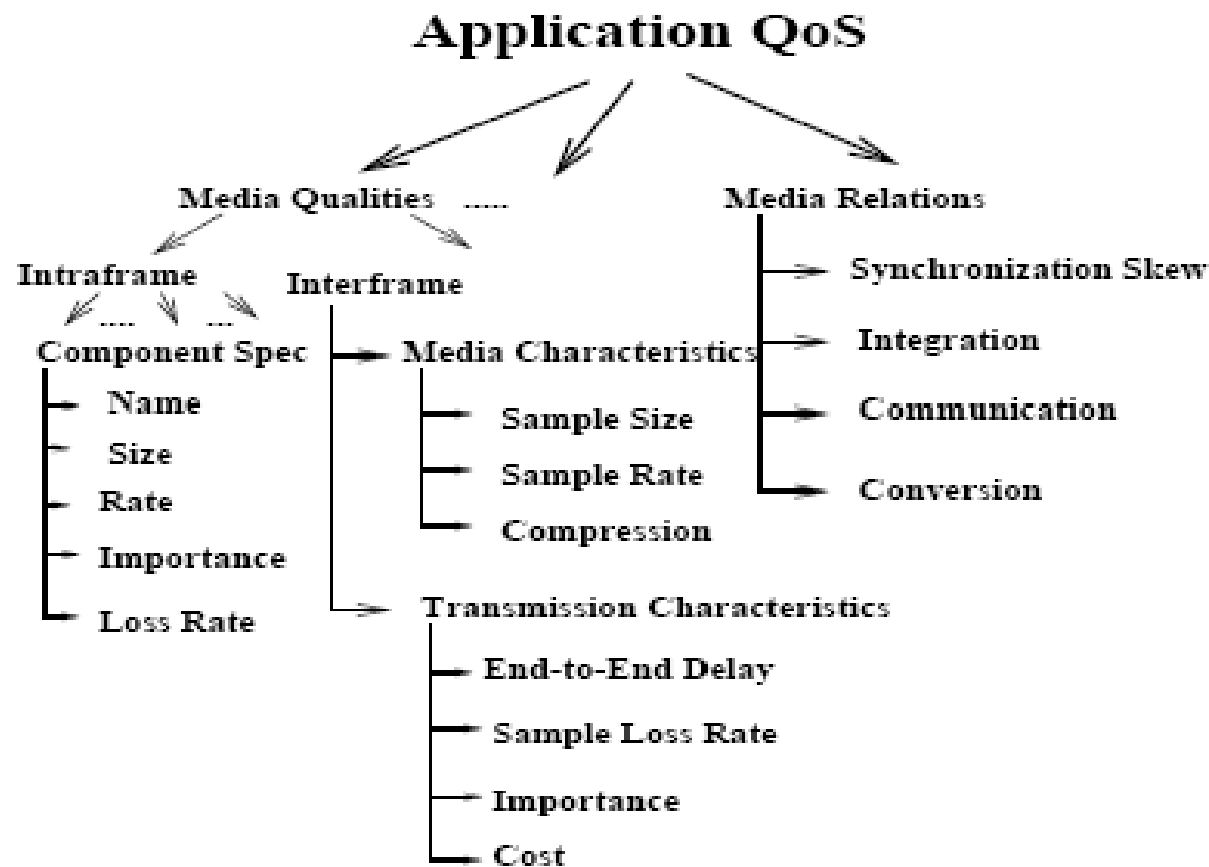
## QoS (cont.)

- QoS concept comes from networking service and was introduced for specification how **good** the offered network services are
- Services are performed on different objects
  - Media sources
  - Media sinks
  - Connections
- QoS specification characterizes the service objects

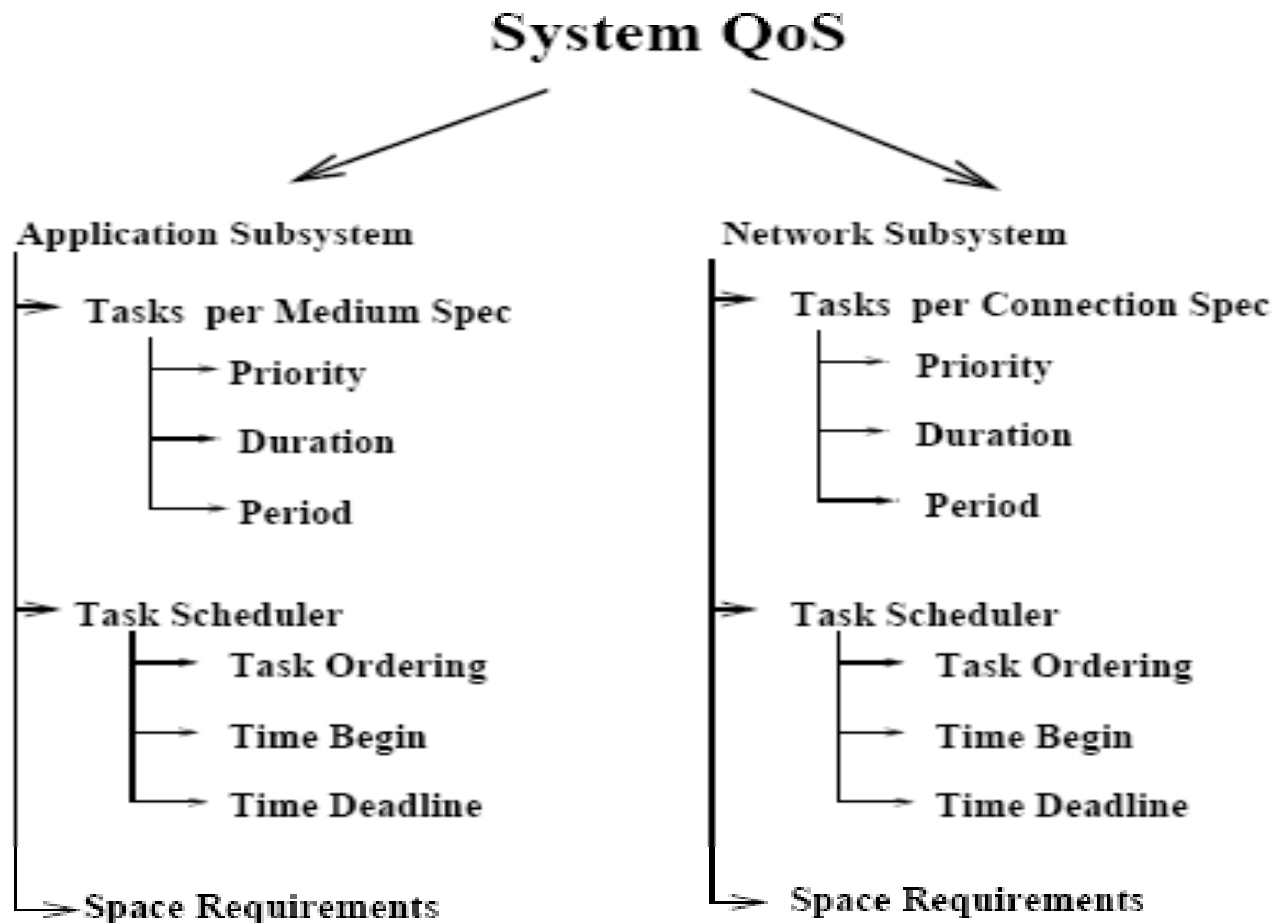
# Layered Model for QoS



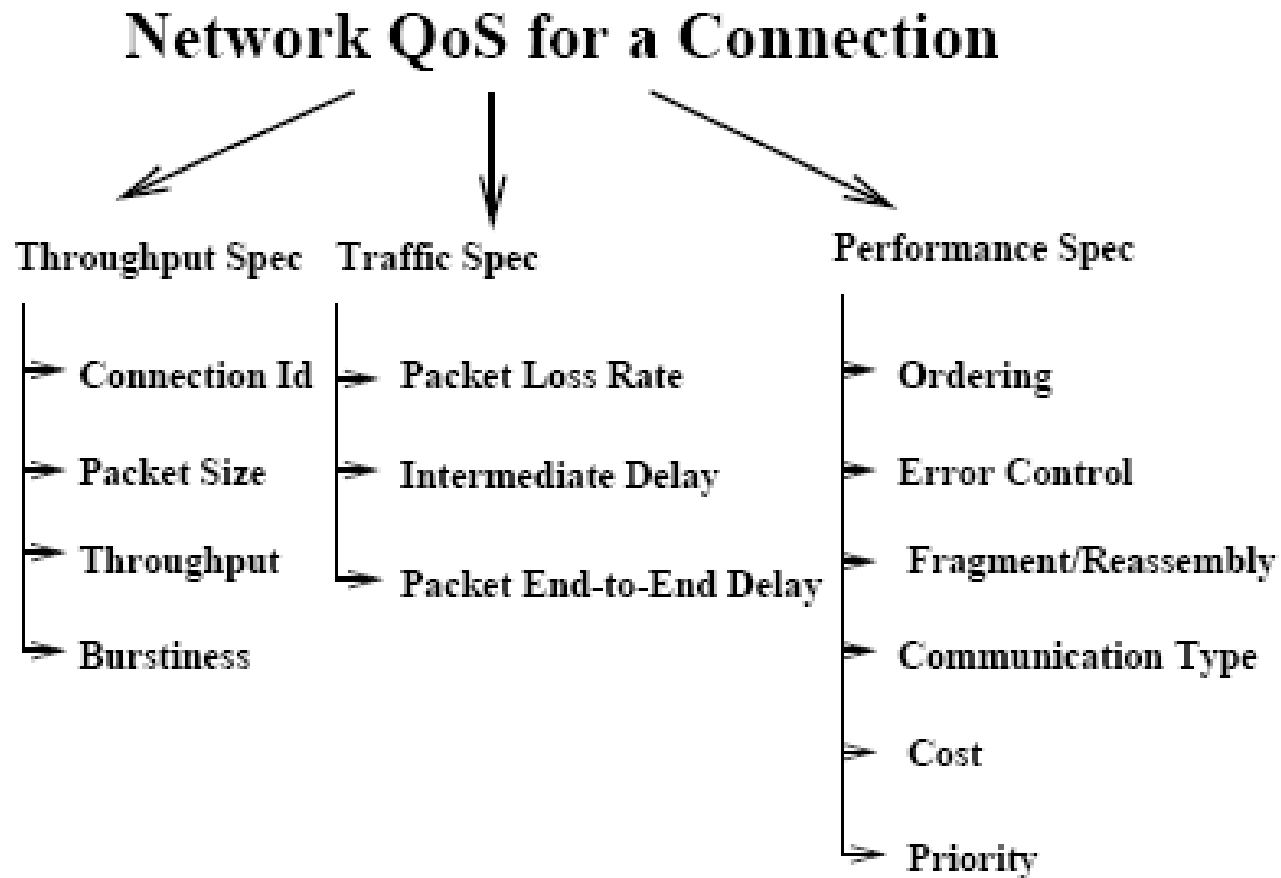
# Application QoS Parameters



# System QoS Parameters



# Network QoS Parameters



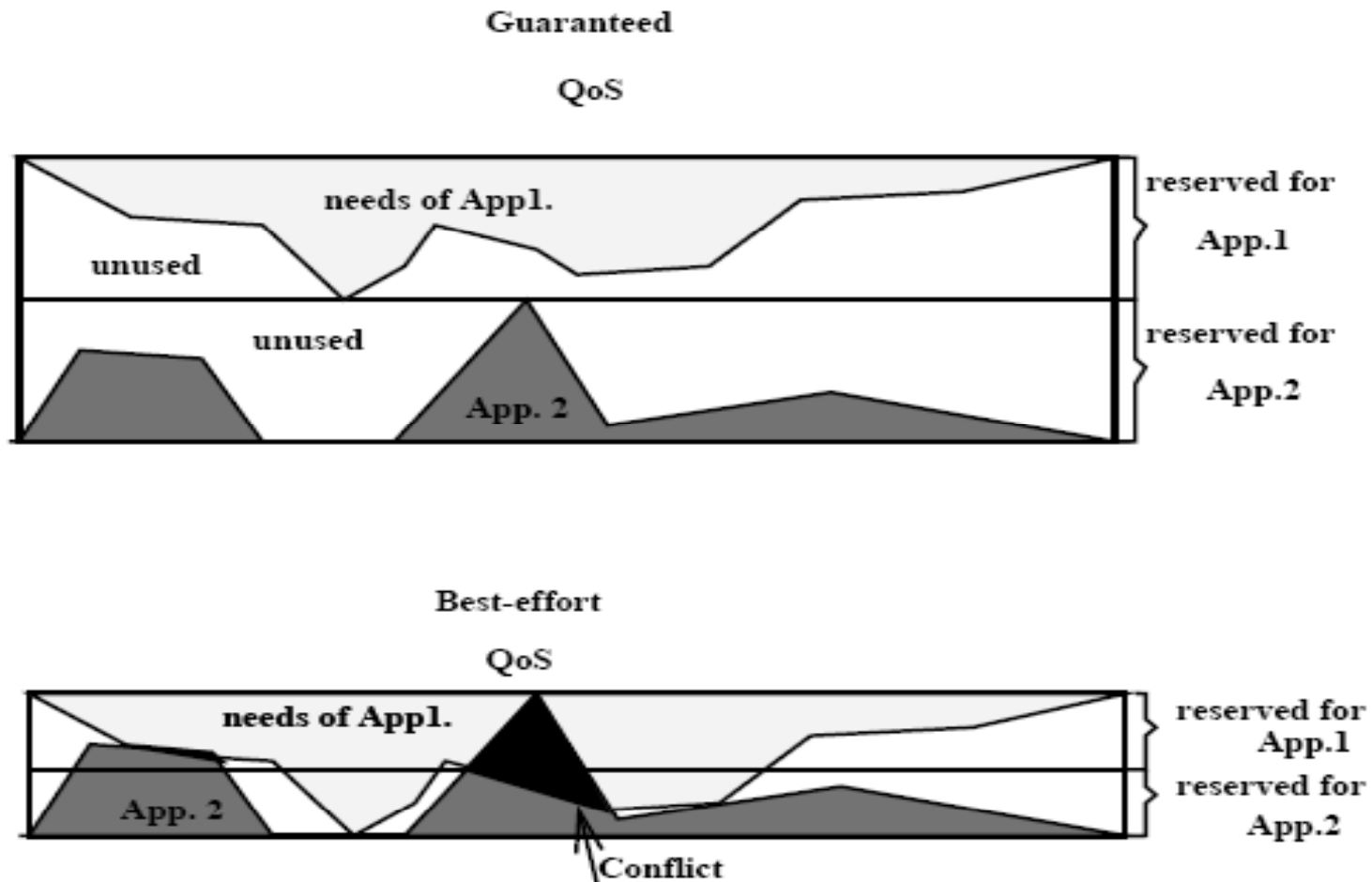


# QoS Classes

- **Guaranteed Service Class**
  - QoS guarantees are provided based on deterministic and statistical QoS parameters
- **Predictive Service Class**
  - QoS parameter values are estimated and based on the past behavior of the service
- **Best Effort Service Class**
  - There are no guarantees or only partial guarantees are provided

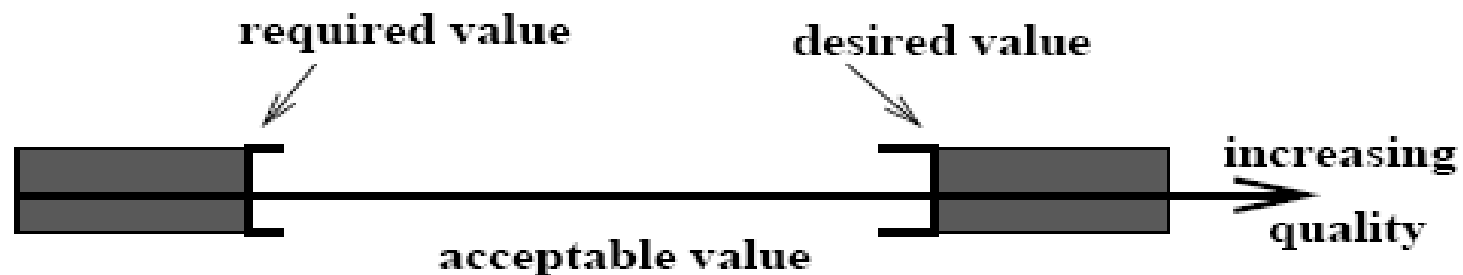
# QoS Classes (cont.)


QoS Class determines: (a) reliability of offered QoS, (b) utilization of resources



# Deterministic QoS Parameters

- **Single Value:**  $QoS_1$  – average ( $QoS_{ave}$ ), contractual value, threshold value, target value
- **Pair Value:**  $\langle QoS_1, QoS_2 \rangle$  with  
     $QoS_1$  – required value;  $QoS_2$  – desired value  
    Example:  $\langle QoS_{avg}, QoS_{peak} \rangle$ ;  $\langle QoS_{min}, QoS_{max} \rangle$





# Deterministic QoS Parameter Values

- Triple of Values  $\langle QoS_1, QoS_2, QoS_3 \rangle$ 
  - $QoS_1$  – best value
  - $QoS_2$  – average value
  - $QoS_3$  – worst value
- Example:
  - $\langle QoS_{peak}, QoS_{avg}, QoS_{min} \rangle$ , where QoS is network bandwidth



# Guaranteed QoS

- We need to provide 100% guarantees for QoS values (hard guarantees) or very close to 100% (soft guarantees)
- Current QoS calculation and resource allocation are based on:
  1. Hard upper bounds for imposed workloads
  2. Worst case assumptions about system behavior
- 1. **Advantages:** QoS guarantees are satisfied even in the worst case case (high reliability in guarantees)
- 2. **Disadvantage:** Over-reservation of resources, hence needless rejection of requests



# Predictive QoS Parameters

- We utilize QoS values ( $QoS_1, .. QoS_i$ ) and compute average
  - $QoS_{bound}$  step at  $K > i$  is  $QoS_K = 1/i * \sum_j QoS_j$
- We utilize QoS values ( $QoS_1, , QoS_i$ ) and compute maximum value
  - $QoS_K = \max_{j=1, \dots, i} (QoS_j)$
- We utilize QoS values ( $QoS_1, , QoS_i$ ) and compute minimum value
  - $QoS_K = \min_{j=1, \dots, i} (QoS_j)$



# Best Effort QoS

- No QoS bounds or possible very weak QoS bounds
- Advantages: resource capacities can be statistically multiplexed, hence more processing requests can be granted
- Disadvantages: QoS may be temporally violated



# Quality-aware Service Model

## ■ Quality-aware Autonomous Single Service

- Consists of a set of functions
- Accepts input data with QoS level  $QoS^{in}$ 
  - $QoS^{in}=[q_1^{in}, \dots, q_n^{in}]$
- Generates output data with QoS level  $QoS^{out}$ 
  - $QoS^{out}=[q_1^{out}, \dots, q_n^{out}]$

## ■ Example: Video player service

- Input QoS: [Recorded Video Frame Rate, Recorded Frame Size, Recorded Pixel Precision]
- $QoS^{in}=[30\text{fps}, 640 \times 480 \text{ pixels}, 24 \text{ bits per pixel}]$
- Output QoS: [Playback Video Frame Rate, Playback Frame Size, Playback Pixel Precision]
- $QoS^{out}=[20\text{fps}, 320 \times 240 \text{ pixels}, 24 \text{ bits per pixel}]$



# Quality-aware Service Model

## ■ Quality-aware Composite Service

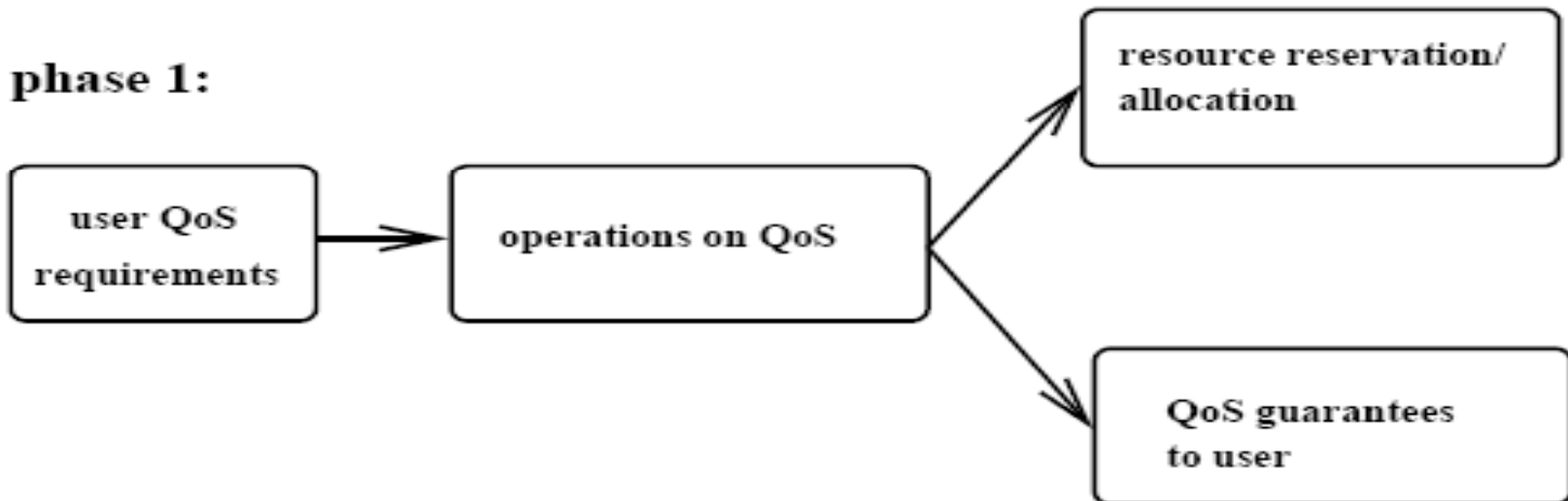
- Consists of set of autonomous services that are connected into a directed acyclic graph, called service graph
- Is correct if the inter-service satisfied the following relation:
  - $QoS^{out}$  of Service  $K$  'satisfies'  $QoS^{in}$  of Service  $M$  iff
  - $q_{Kj}^{out} = q_{MI}^{in}$  for  $q_{MI}^{in}$  being single QoS value
  - $q_{Kj}^{out}$  is in  $q_{MI}^{in}$  for  $q_{MI}^{in}$  being a range of QoS value

## ■ Example:

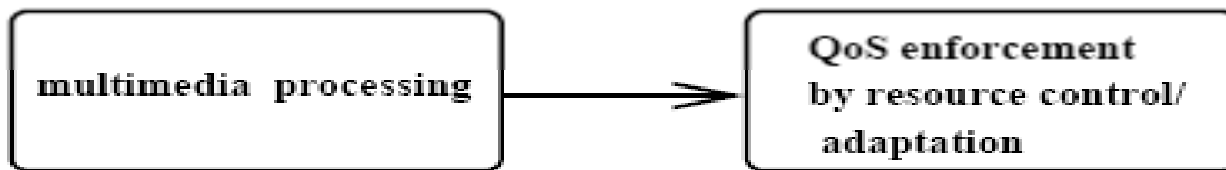
- Video-on-demand service, consists of two services: retrieval service and playback service
  - Output quality of the retrieval service needs to correspond to input quality of playback service, or at least falls into the range of input quality of playback service


# Relation between QoS and Resources

**phase 1:**



**phase 2:**

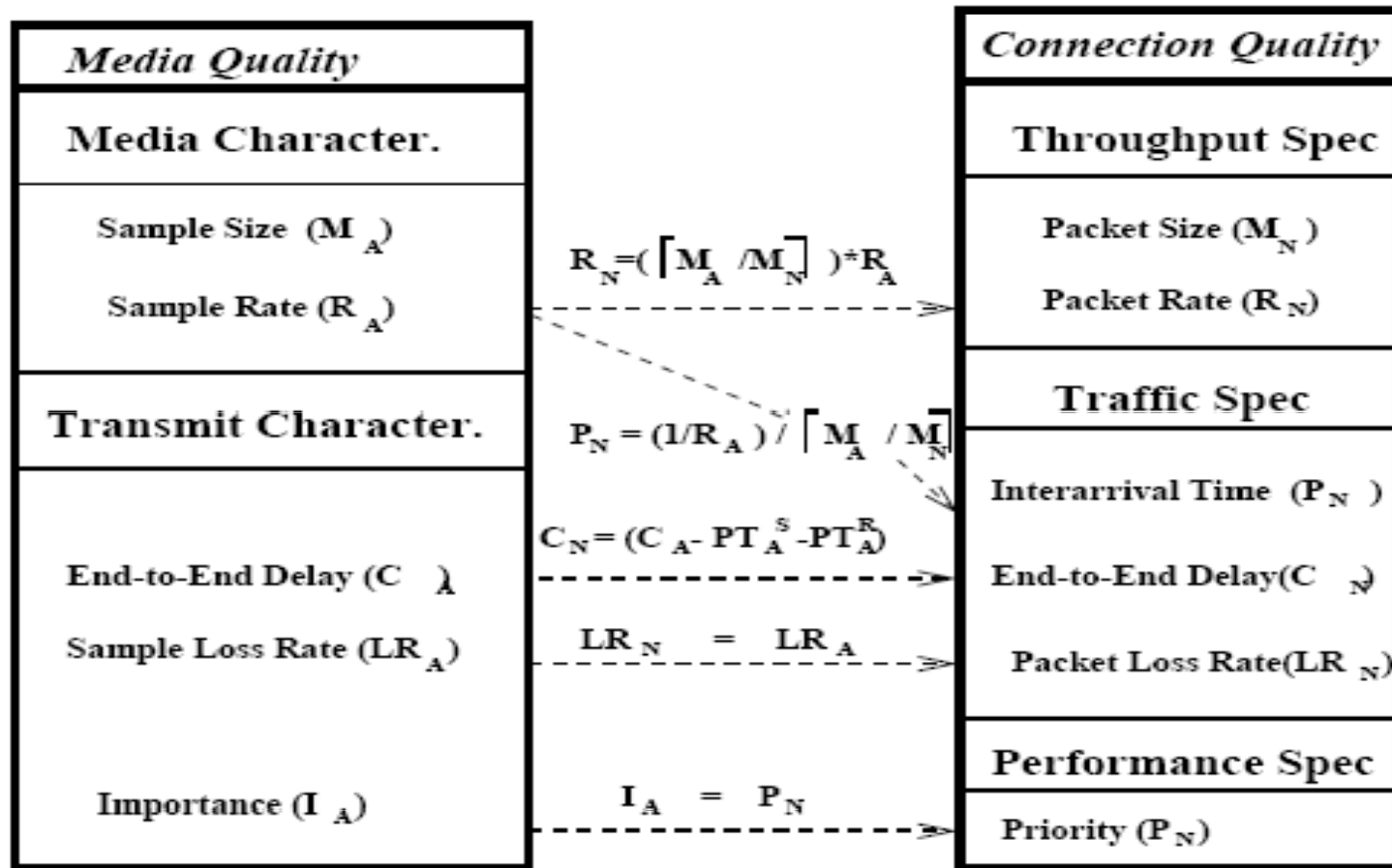




# Operations on QoS in Phase 1 (Translations)

- Layered Translation of QoS parameters  
(must be bidirectional)
  - Human (user QoS) – application QoS
  - Application QoS – system QoS
  - System QoS – network QoS
- Media Scaling
  - Transparent scaling
  - Non-transparent scaling

# Layered Translation (Example)





# Media Scaling (Examples)

## ■ Audio

- Transparent scaling difficult (one hears the quantization noise)
- Non-transparent scaling should be used

## ■ Video

- Temporal scaling
- Spatial scaling
- Color space scaling (reduction of number of entries in color space)



# Conclusion

- QoS – an important concept in multimedia systems
- Very different types of QoS parameters and values
- Important relation between QoS and Resources
- Need to understand operations on QoS and their impact on resource management