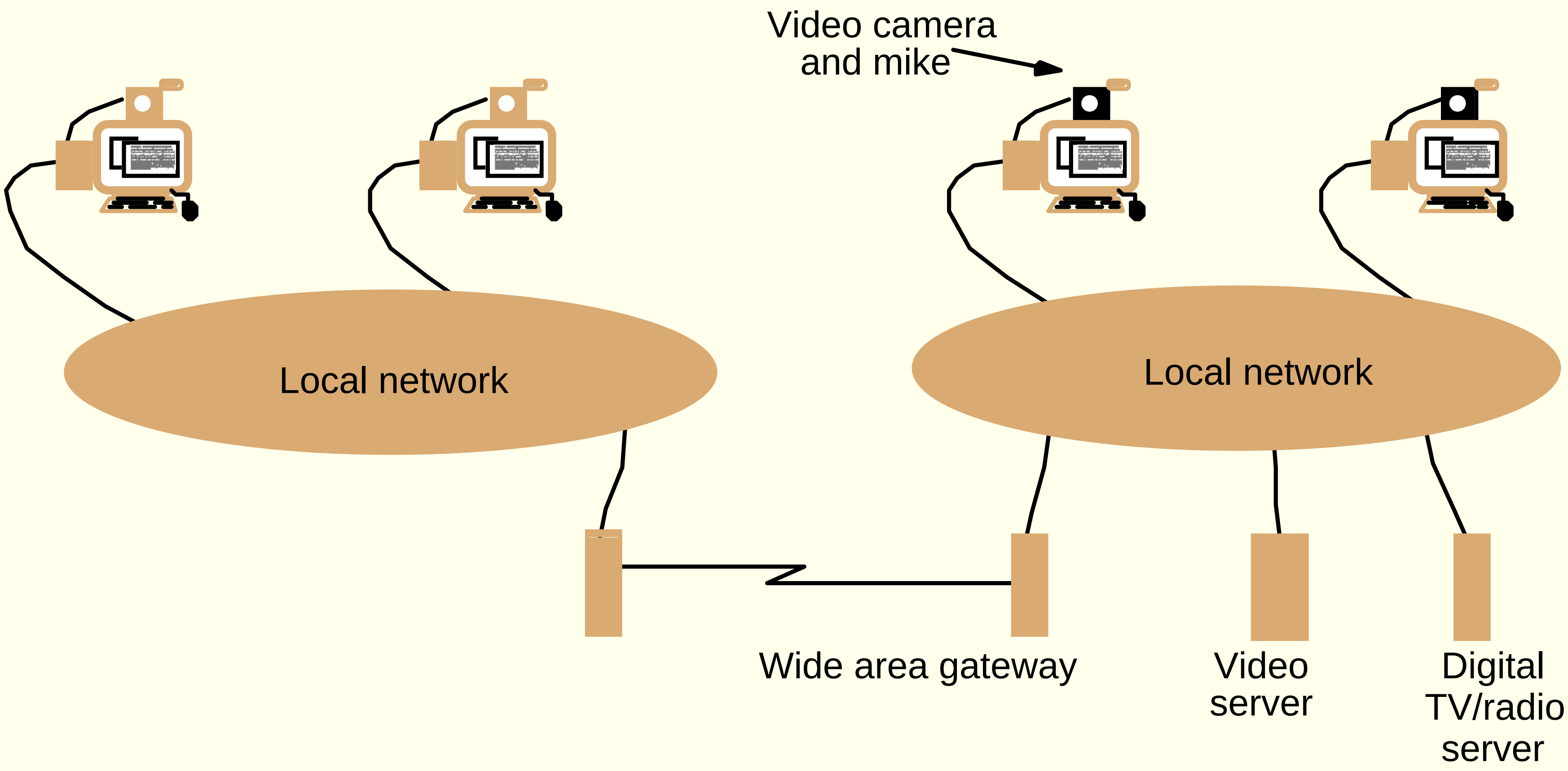
Chapter 17: Distributed Multimedia Systems

- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server

Introduction

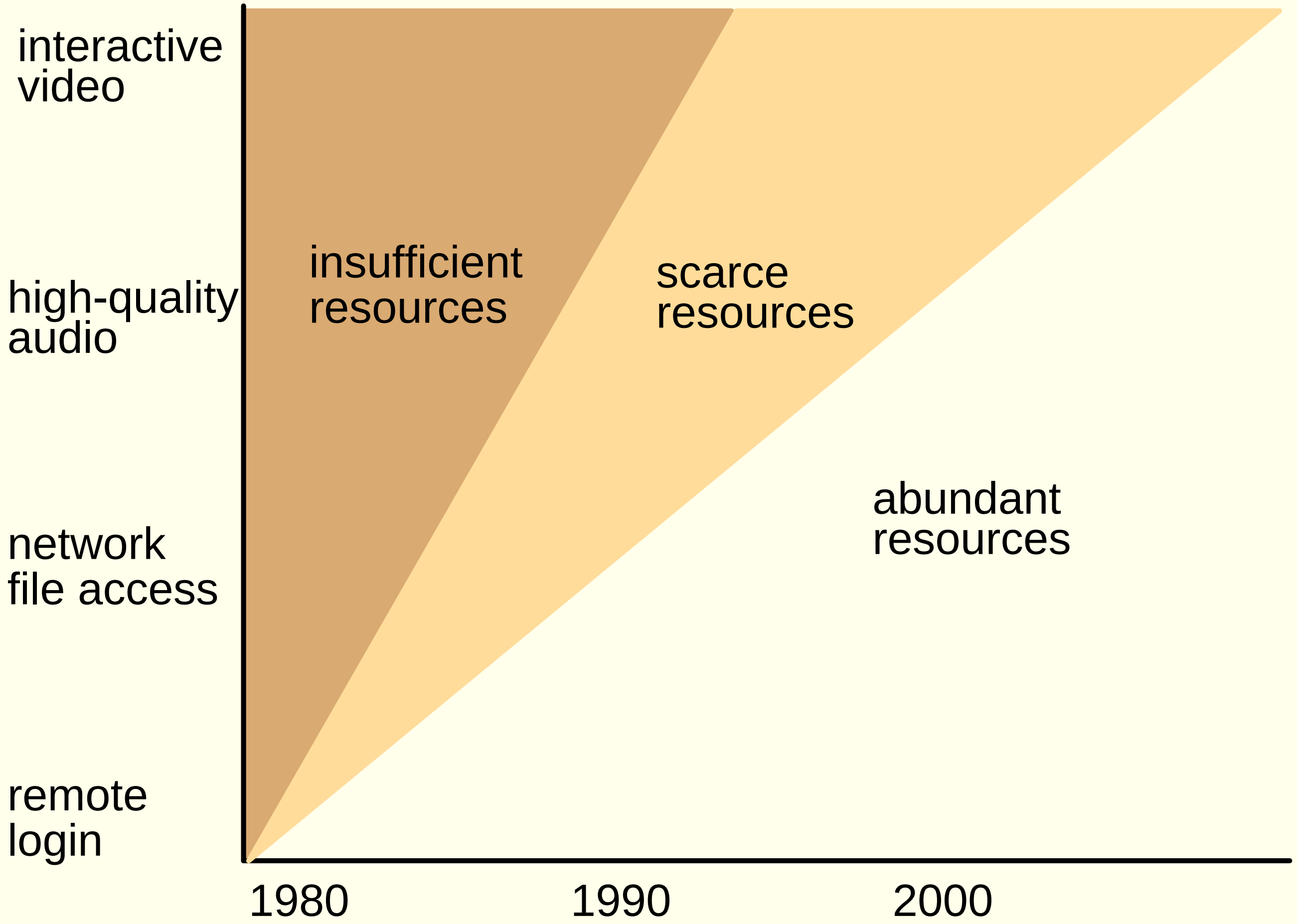
- **Distributed multimedia applications**
  - Networked video library, Internet telephony, videoconference
  - Generate and consume continuous streams of data in real time
- **Characteristics of multimedia applications**
  - Timely delivery of streams of multimedia data to end-users
    - Audio sample, video frame
- **To meet the timing requirements**
  - QoS(quality of service)

A typical distributed multimedia system



The window of scarcity

- **A history of computer systems that support distributed data access**



Chapter 17: Distributed Multimedia Systems

- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server

Characteristics of multimedia data

- **Continuous**
  - Refer to the user's view of the data
  - Video: a image array is replaced 25 times per second
  - Audio: the amplitude value is replaced 8000 times per second
- **Time-based**
  - The times at which the values are played or recorded affect the validity of the data
  - Hence, the timing should be preserved

Chapter 17: Distributed Multimedia Systems

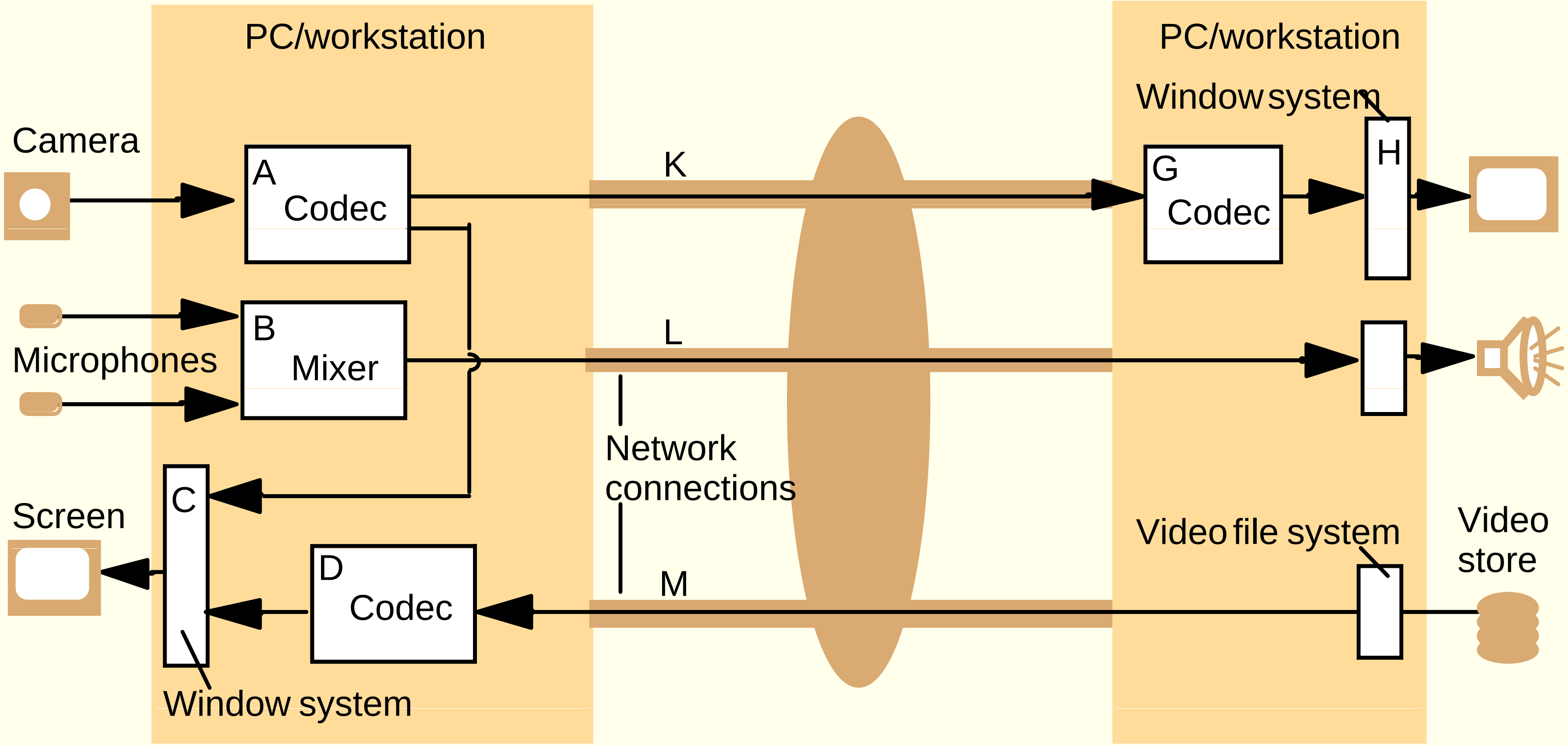
- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server

Traditional computer systems

- **Multimedia App. compete with other App. for**
  - Processor cycles, bus cycles, buffer capacity
  - Physical transmission links, switches, gateways
- **Best-effort policies**
  - Multi-task OS: round-robin scheduling or other scheduling
    - shares the processing resource on a best-effort basis among all of the task currently competing
  - Ethernet
    - manages a shared transmission medium in best-efforts manner
- **Best-effort policies are not fit to multimedia apps.**
- **In order to achieve timely delivery,**
  - applications need guarantees that the necessary resources will be allocated and scheduled at the required times

    → QoS management

Typical infrastructure components for multimedia applications



: multimedia stream
White boxes represent media processing components, many of which are implemented in software, including:
codec: coding/decoding filter
mixer: sound-mixing component

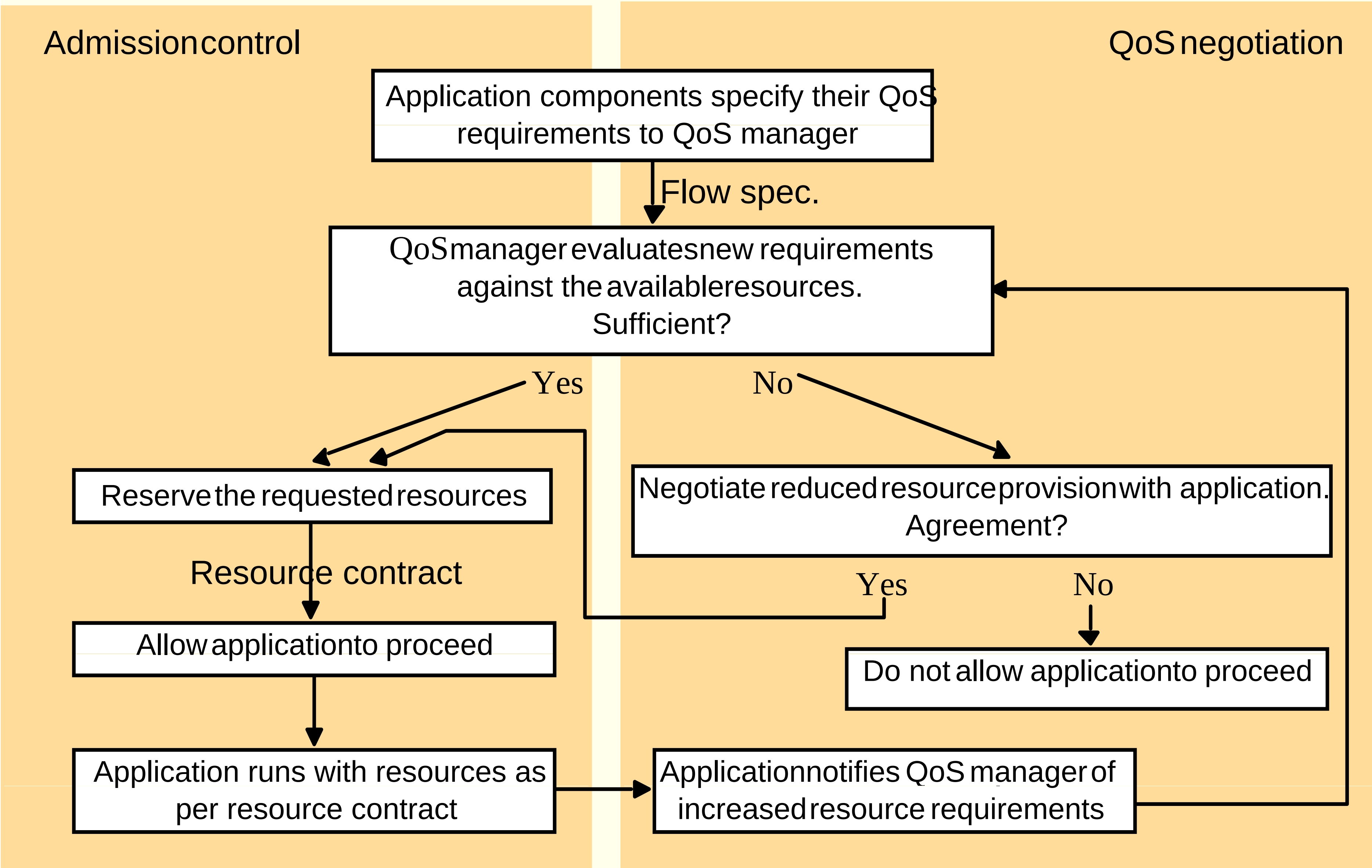
QoS specifications for components of the application shown in Figure 15.4

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;
		Out:	MPEG-1 stream			10 Mbytes RAM
B	Mixer	In:	2 × 44 kbps audio	Interactive	Very low	1 ms CPU each 100 ms;
		Out:	1 × 44 kbps audio			1 Mbytes RAM
H	Window system	In:	various	Interactive	Low	5 ms CPU each 100 ms;
		Out:	50 frame/sec framebuffer			5 Mbytes RAM
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

The QoS manager’s task

- The OoS manager’s two main subtasks are:
  - Quality of service negotiation
  - Admission control

The QoS manager's task



Quality of service negotiation

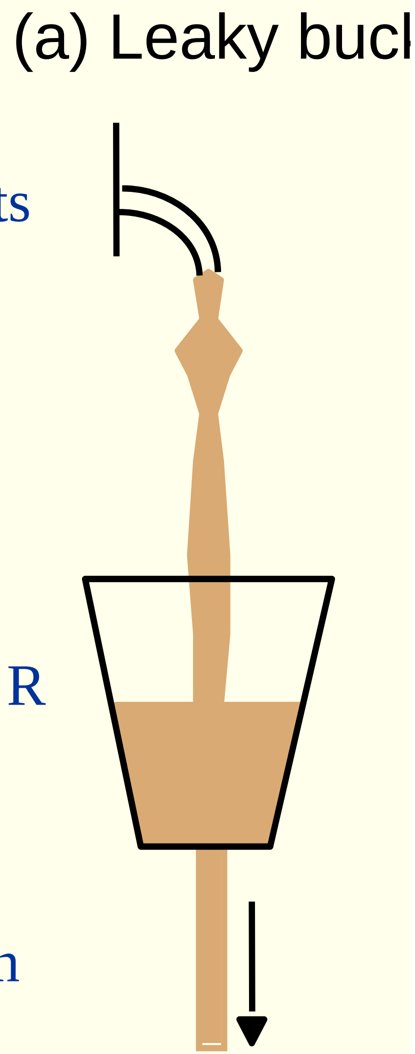
- Resource requirements specification
  - bandwidth
    - The rate at which a multimedia stream flows
  - Latency
    - The time required for an individual data element to move through a stream from the source to the destination
    - jitter
  - Loss rate
    - Data loss due to unmet resource requirements
    - A rate of data loss that can be accepted. E.g., 1%

The usage of resource requirements spec.

- Describe a multimedia stream
  - Describe the characteristics of a multimedia stream in a particular environment
  - E.g. a video conference
    - Bandwidth: 1.5Mbps; delay: 150ms, loss rate: 1%
- Describe the resources
  - Describe the capabilities of resources to transport a stream
  - E.g. a network may provide
    - Bandwidth: 64kbps; delay: 10ms; loss rate: 1/1000

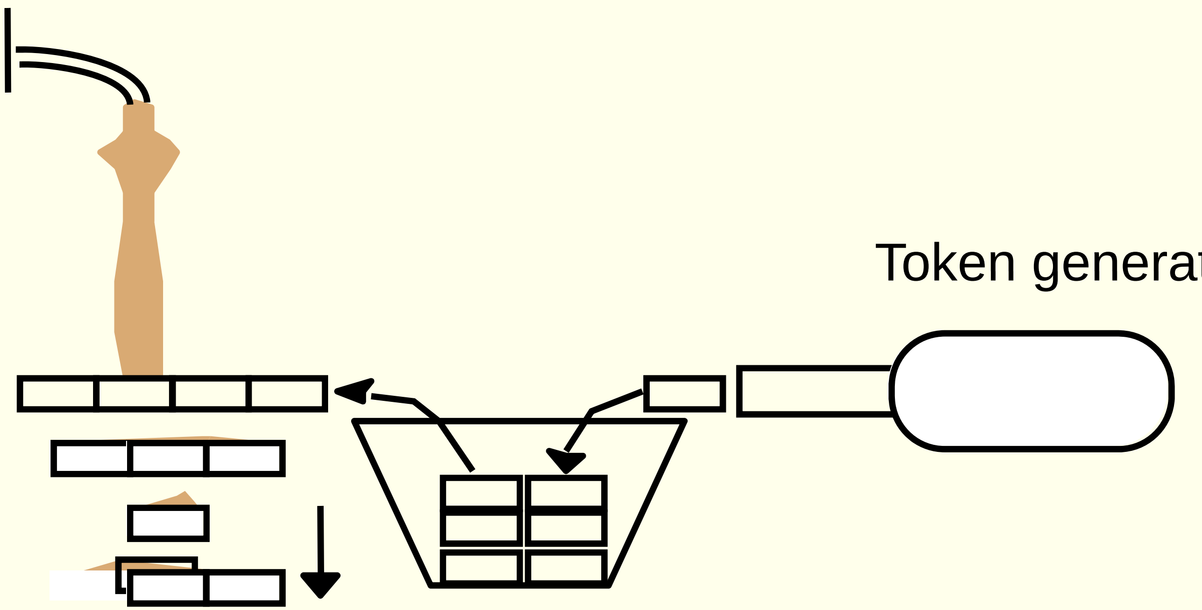
Traffic shaping

- Traffic shaping
  - Output buffering to smooth the flow of data elements
  - Leaky bucket, Token bucket
- The **leaky bucket** algorithm
  - completely eliminate burst
  - R
    - A stream will never flow with a rate higher than R
  - B
    - Size of the buffer
    - Bound the time for which an element will remain in the buffer



Traffic shaping (2)

- The **token bucket** algorithm
  - Allow larger burst
  - Token is generated at a fixed rate of *R*
  - the tokens are collected in a bucket of size *B*
  - Data of size *S* can be sent only if at least *S* tokens are in the bucket
  - Ensure: over any interval *t*, the amount of data sent is not larger than *Rt+B*,



Flow specification – RFC 1363

- Bandwidth
  - The maximum transmission unit and maximum transmission rate
  - The burstiness of the stream
    - The token bucket size and rate
- Delay
  - The minimum delay that an application can notice, the maximum jitter it can accept
- Loss rate
  - The total acceptable number of losses over a certain interval
  - The maximum number of consecutive losses

Admission control

- Avoid resource overload
- Protect resource from requests that they cannot fulfill
- **Bandwidth reservation**
  - Reserve some portion of resource bandwidth exclusively
- **Statistical multiplexing**
  - Reserve minimum or average bandwidth
  - Handle burst that cause some service drop level occasionally
  - Hypothesis
    - a large number of streams the aggregate bandwidth required remains nearly constant regardless of the bandwidth of individual streams

Chapter 17: Distributed Multimedia Systems

- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server
- Summary

Resource Management

- To provide a certain QoS level to an application, a system needs to have sufficient resources, it also needs to make the resources available to an application when they are needed (scheduling).
- Resource Scheduling: A process needs to have resources assigned to them according to their priority. Following 2 methods are used:
  - **Fair Scheduling**
  - **Real-time scheduling**

Resource scheduling

- **Fair scheduling**
  - when several streams compete for the same resource
  - Round-robin
    - Packet-by-packet
    - Bit-by-bit
- **Real-time scheduling**
  - Earliest-deadline-first (EDF)
    - Each media element is assigned a *deadline* by which it must be sent out
    - The scheduler send media elements according to their *deadline*

Chapter 17: Distributed Multimedia Systems

- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server

Stream adaptation

- Stream adaptation
  - An application adapt to changing QoS levels when a certain QoS cannot be guaranteed
- Drop pieces of information
  - Audio stream
    - Drop can be noticed immediately by the listener
  - Video stream
    - Motion JPEG: easy since frames are independent
    - MPEG: difficult since frames are interdependent
- Increase delay
  - Acceptable for non-interactive applications
- Two methodologies are used:
  - Scaling
  - Filtering

Scaling

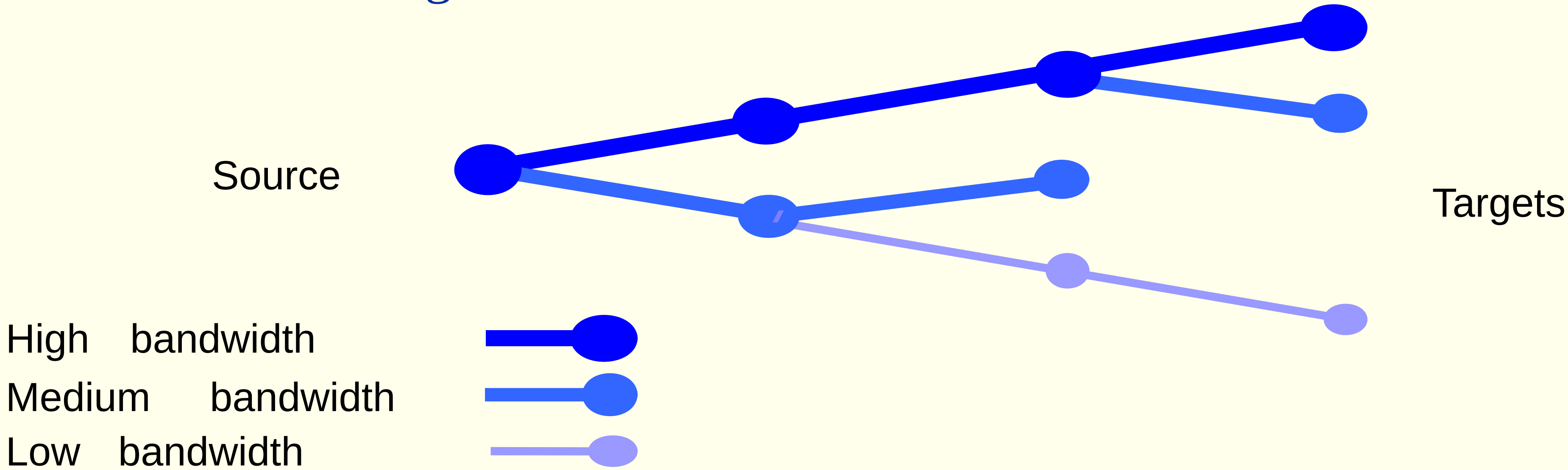
- When to perform scaling
  - Adapt a stream to the bandwidth available in the system before it enters a bottleneck resource
- Scaling approach
  - **Implementation**
    - A monitor process at the target
    - A scaler process at the source
    - Monitor keeps track of the arrival times of messages in a stream. Delayed messages are an indication of bottle neck in the system.
    - Monitor sends a scale-down message to the source that scales up again

Different scaling methods

- **Temporal scaling**
  - Decrease the number of video frames transmitted within an interval
- **Spatial scaling**
  - Reduce the number of pixels of each image in an video stream, e.g., JPEG and MPEG-2
- **Frequency scaling**
  - Modify the compression algorithm applied to an image
- **Amplitudinal scaling**
  - Reduce the color depths for each image pixel
- **Color space scaling**
  - Reduce the number of entries in the color space, e.g., from color to grey-scale presentation

Filtering

- Scaling is not suitable to a stream that involves several receivers
  - Since scaling is conducted at the source, a *scale-down* message will degrade the quality of all streams
- Filtering
  - A stream is partitioned into a set of hierarchical sub-streams
  - The capacity of nodes on a path determines the number of sub-streams a target receives



Chapter 17: Distributed Multimedia Systems

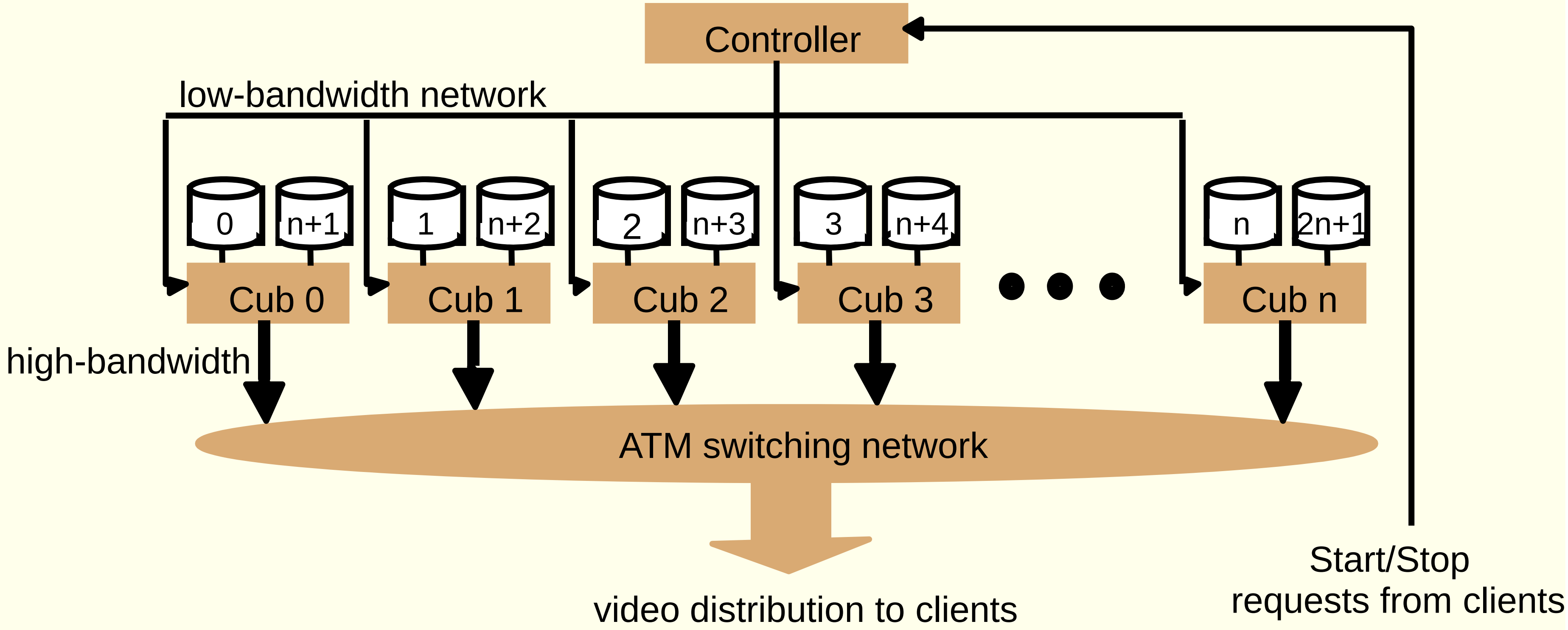
- Introduction
- Characteristics of multimedia data
- Quality of service management
- Resource management
- Stream adaptation
- Case study: the Tiger video file server

Design goals

- Video-on-demand for a large number of users
  - A large stored digital movie library
  - Users can perform pause, rewind, fast-forward
- Quality of service
  - Constant rate
  - a maximum jitter and low loss rate
- Scalable and distributed
  - Support up to 10000 clients simultaneously
- Low-cost hardware
  - Constructed by commodity PC
- Fault tolerant
  - Tolerant to the failure of any single server or disk

System architecture

- **One controller**
  - Connect with each server by low-bandwidth network
- **Cubs – the server group**
  - Each cub is attached by a number of disks (2-4)
  - Cubs are connected to clients by ATM



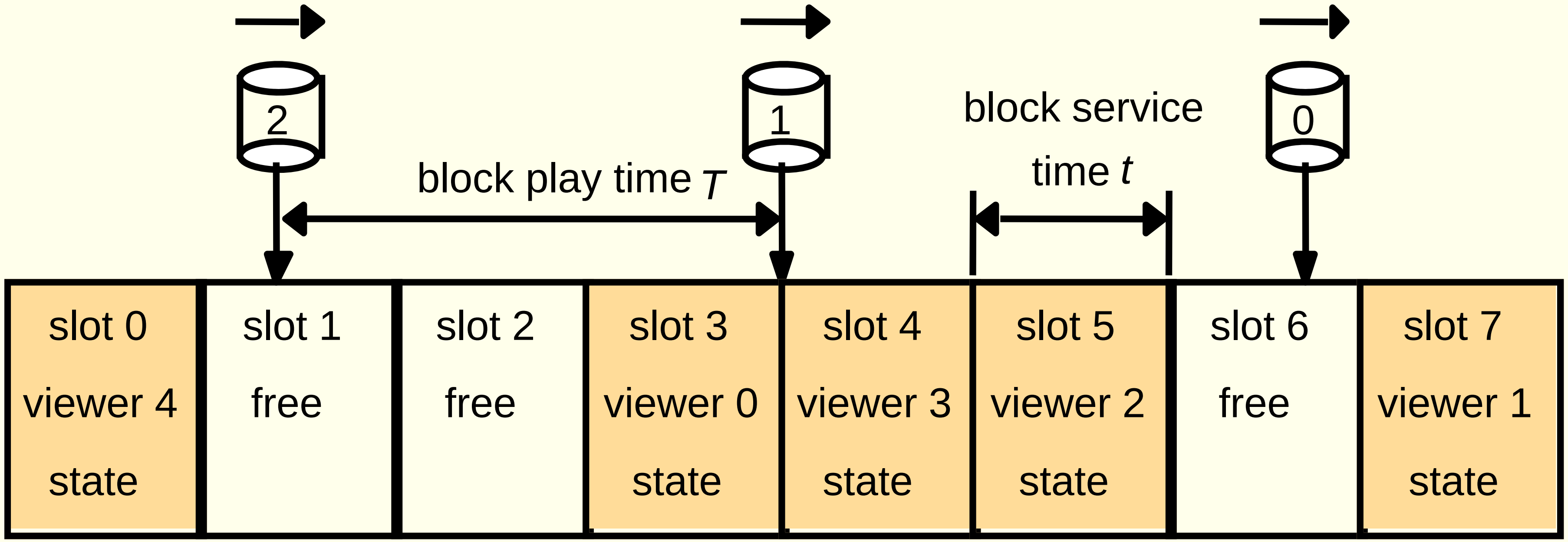
Storage organization

- Stripping
  - A movie is divided into blocks
  - The blocks of a movie are stored on disks attached to different cubs in a sequence of the disk number
  - Deliver a movie: deliver the blocks of the movie from different disks in the sequence number
  - Load-balance when delivering hotspot movies
- Mirroring
  - Each block is divided into several portions (*secondaries)*
  - The *secondaries* are stored in the successors
    - If a block is on a disk i, then the *secondaries* are stored on disks i+1 to $i+d$
  - Fault-tolerance for single cub or disk failure

Distributed schedule

- **Slot**
  - The work to be done to play one block of a movie
- **Deliver a stream**
  - Deliver the blocks of the stream disk by disk
  - Can be viewed as a slot moving along disks step by step
- **Deliver multiple streams**
  - Multiple slots moving along disks step by step
- **Viewer state**
  - Address of the client computer
  - Identity of the file being played
  - Viewer's position in the file
  - The viewer's play sequence number
  - Bookeeping information

Tiger schedule



Distributed schedule (2)

- *Block play time - T*
  - The time that will be required for a viewer to display a block on the client computer
  - Typically about 1 second for all streams
  - The next block of a stream must begin to be delivered *T* time after the current block begin to be delivered
- *Block service time – t* (a slot)
  - Read the next block into buffer
  - Deliver it to the client
  - Update viewer state in the schedule and pass the updated slot to the next cub
  - T / t typically result in a value > 4
- The maximum streams the Tiger system can support simultaneously
  - T/t * the number of disks