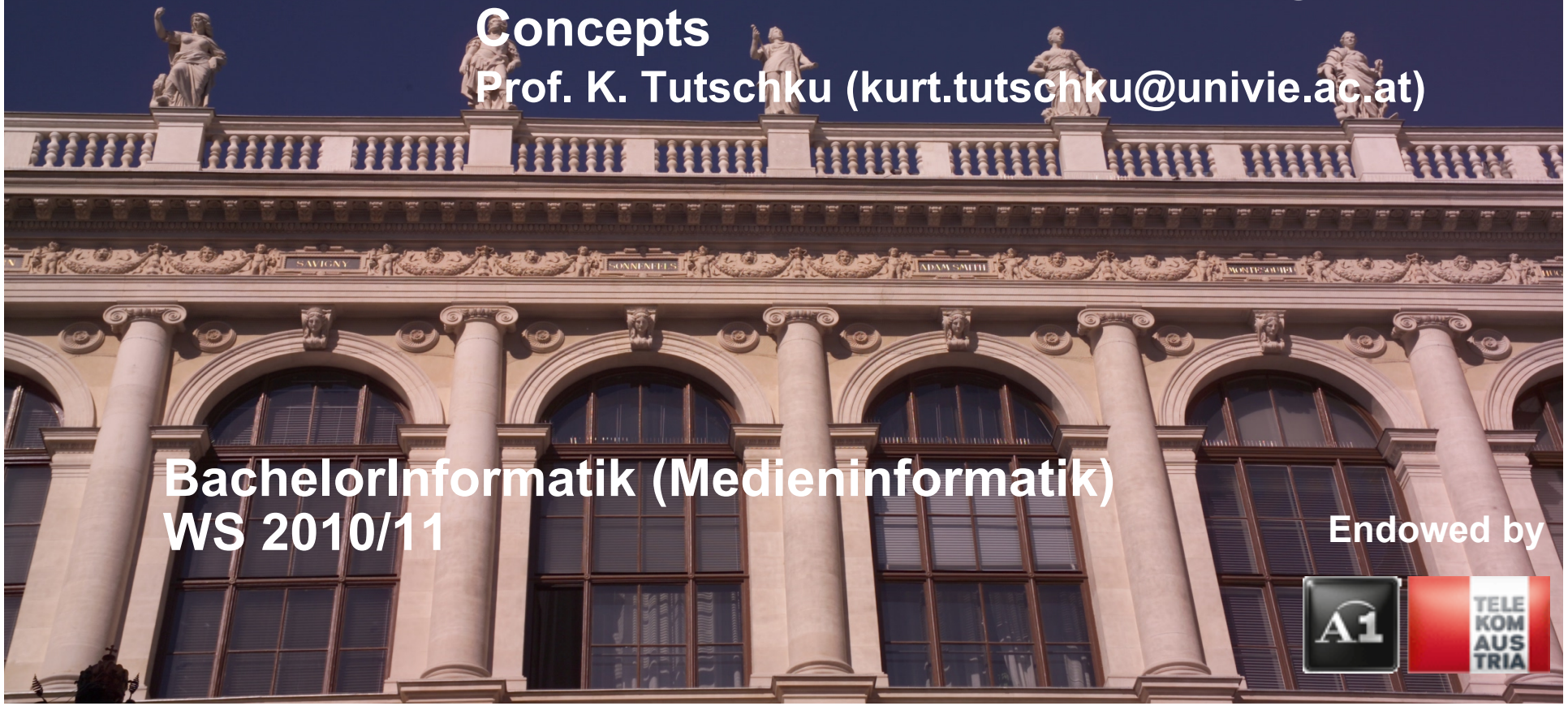
Chair for Future Communication
Prof. Dr. K. Tutschku
Institute for Distributed and Multimedia Systems
Faculty for ComputerScience

050069 VO Netzwerktechnologie für Multimedia Anwendungen

Lecture 4: Revision of Networking Concepts

Prof. K. Tutschku (kurt.tutschku@univie.ac.at)



BachelorInformatik (Medieninformatik)
WS 2010/11

Endowed by

universität wien

Chapter 2: Revision of Networking Concepts

Chair of **Future Communication**

endowed by

A1 TELEKOM AUSTRIA

- **Overview:**
- **Protocol layering and Internet protocol stack**
- **Circuit switching vs. packet switching**
- **Connectionless vs. connection-oriented networks, routing, forwarding, and switching**
- **Transport layer protocols**
- **Application layer**
  - Sockets
  - Client-server and peer-to-peer communication
- **Web services**

Sockets 1/2

- **Process**
  - Program running within a host
  - Processes within same host communicate using inter-process communication (defined by OS).
  - Processes in different hosts communicate by exchanging messages
  - Application: process in a hosts
  - Client process: process that initiates communication
  - Server process: process that waits to be contacted

- **Several processes running on the same host**
  - Identification through 16 bit port numbers
  - Example port numbers:
    - HTTP server: 80
    - Mail server: 25

Sockets 2/2

- **Host**
  - Identification through unique 32 bit IP address
- **Socket**
  - End-point of an Internet Protocol-based communication
  - Components
    - Protocol (TCP, UDP, raw IP)
    - Local IP address
    - Local port
    - Remote IP address
    - Remote port
  - The remote address can be either
    - any valid IP address, or
    - 0.0.0.0 for listening sockets, or
    - 255.255.255.255 for broadcasting sockets
  - Process sends/receives messages to/from its socket

Application Layer Protocols 1/2

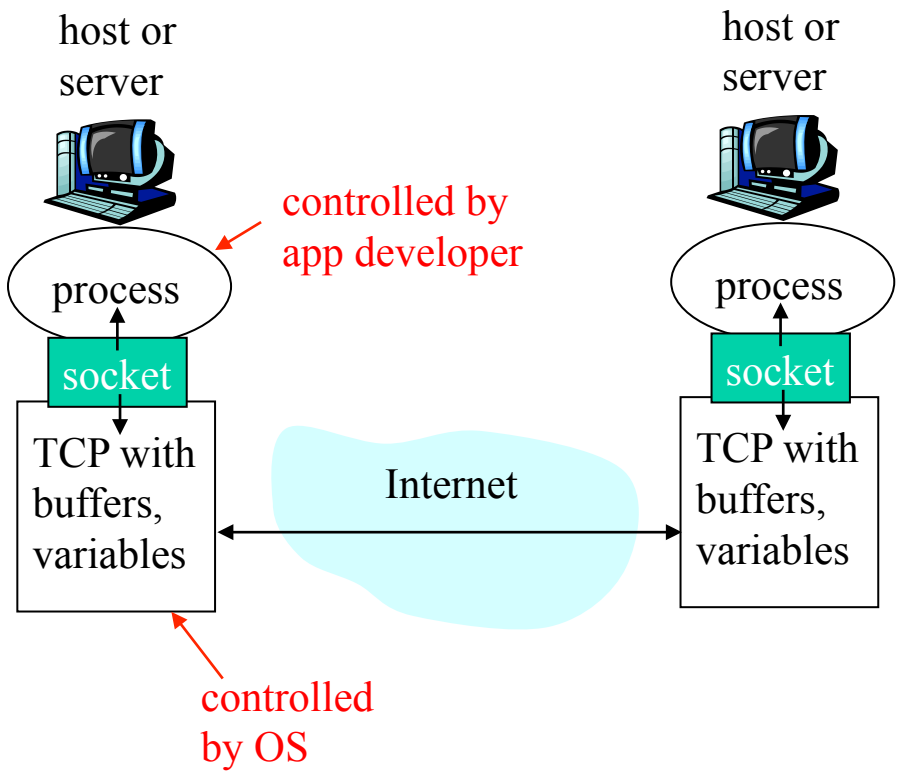
- **Application layer protocols**
  - Run on different end systems
  - Communicate over a network
  - e.g., Web: Web server software communicates with browser software
  - No interaction from core network devices required
  - New software only in end-points needed: rapid deployment possible
- **Public-domain protocols**
  - defined in RFCs
  - allows for interoperability
  - E.g., HTTP, SMTP
- **Proprietary protocols**
  - E.g., KaZaA

Application Layer Protocols 2/2

- **Application layer protocols define**
  - Types of messages exchanged
    - E.g., request & response messages
  - Syntax of message types
    - What fields in messages & how fields are delineated
  - Semantics of the fields
  - State machines
    - Rules for when and how processes send & respond to messages
  - Preferred transport protocol for msgs
  - Preferred port numbers to be contacted
    - E.g., http on port 80

Network Applictions

- **Create a new network application**
  - Define new protocol
  - Deploy software and start it
  - Start communication by contacting those hosts on the right port
- **Some network applications**
  - E-mail
  - Web
  - Instant messaging
  - Remote login
  - P2P file sharing
  - Multi-user network games
  - Streaming stored video clips
  - Internet telephone
  - Real-time video conference
  - Massive parallel computing



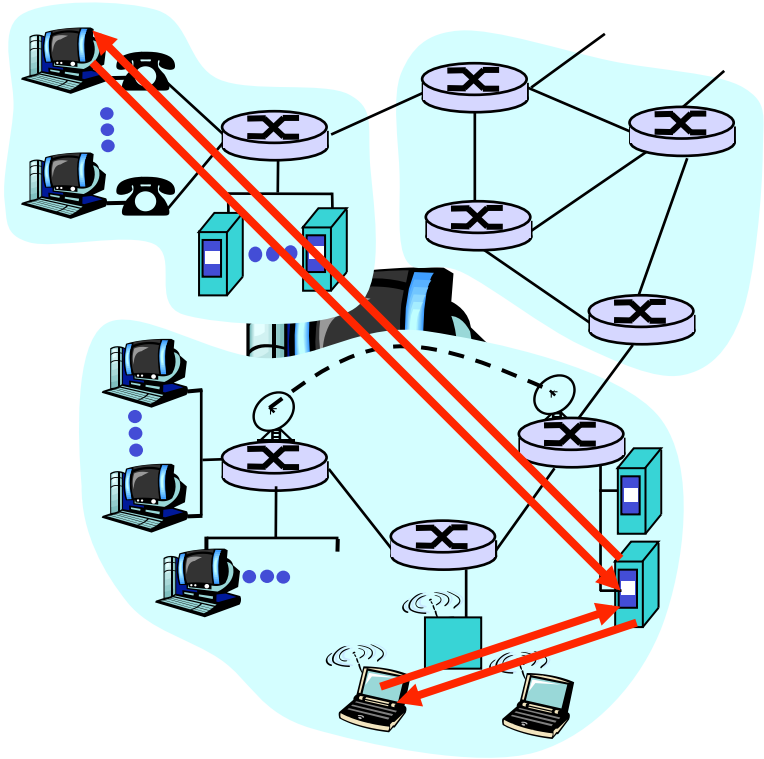
Client-Server Architecture

Server

- Always-on host
- Permanent IP address
- Server farms for scaling

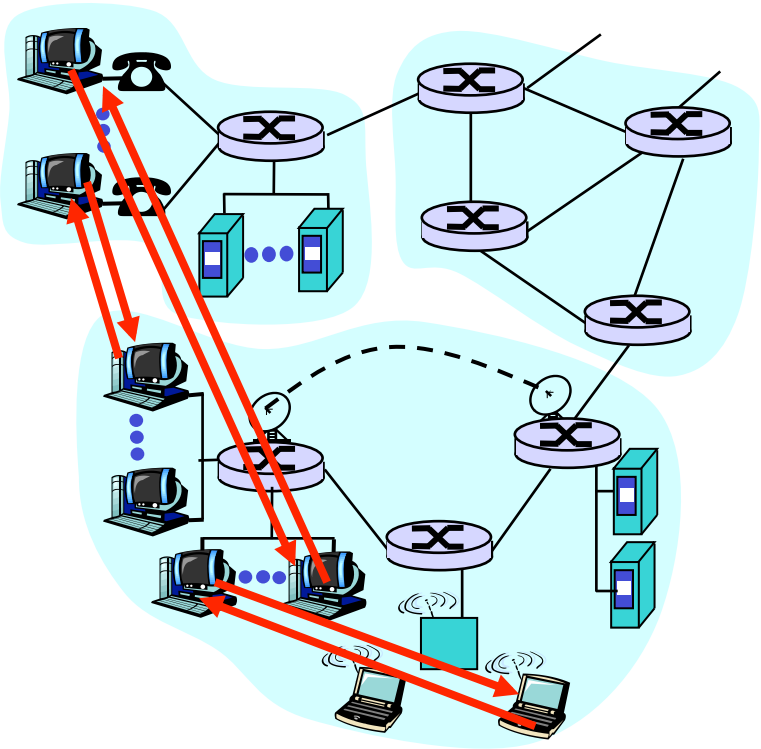
Clients

- Communicate with server
- May be intermittently connected
- May have dynamic IP addresses
- Do not communicate directly with each other



P2P Architecture

- Arbitrary end systems directly communicate
- Peers are intermittently connected and change IP addresses
- Note: applications with P2P architectures have client processes & server processes
- Highly scalable
- But difficult to manage



Was ist P2P?

- **Def.: Ein Peer-to-Peer System ist ein selbstorganisierendes, verteiltes System aus miteinander verbundenen, gleichen und autonomen Knoten, zur gemeinsamen Nutzung von verteilten Ressourcen in einem Netzwerk ohne eine zentrale Instanz.**

- **Peer (engl. Gleichgestellter)**
- **Peer-to-Peer vs. Server-Client Prinzip**
- **Charakteristika von P2P Systemen:**
  - Peers sind sowohl Clients als auch Server
  - Dezentralisierung
  - Nutzung nicht benötigter Ressourcen
  - Transient Connectivity
  - Autonomie der Peers
  - Keine Globale Sicht

Bewertung von P2P

- **Vorteile:**
  - Skalierbarkeit
  - Gemeinsame Nutzung von Ressourcen
  - Robuster gegen den zufälligen Ausfall einzelner Komponenten
- **Probleme:**
  - Peers und Verbindungen sind nicht zuverlässig
  - Hoher Kommunikationsaufwand

Anwendung von P2P

- **Aktuelle Anwendungsbereiche**
  - Filesharing (Bittorrent, KaZaa, Napster, eDonkey)
  - Instant-Messaging (CSpace)
  - Data Storage (PAST, OceanStore, FarSite)
  - Grid/Distributed Computing (Seti@Home, Folding@Home)
  - Collaboration / Groupware (Groove)
  - Kontrolle von Netzen (Tutschku et al, 2003 –)

P2P Overlay Network

- **Overlay Network ist ein logisches Netz**
- **Setzt auf einem physikalische Netz (Internet) auf**
- **Besteht aus allen teilnehmenden Peers (Knoten)**
- **Jeder Knoten speichert eine Liste von Nachbarknoten**
- **Knoten sind durch (virtuelle) Kanten miteinander verbunden**

P2P Klassifikation

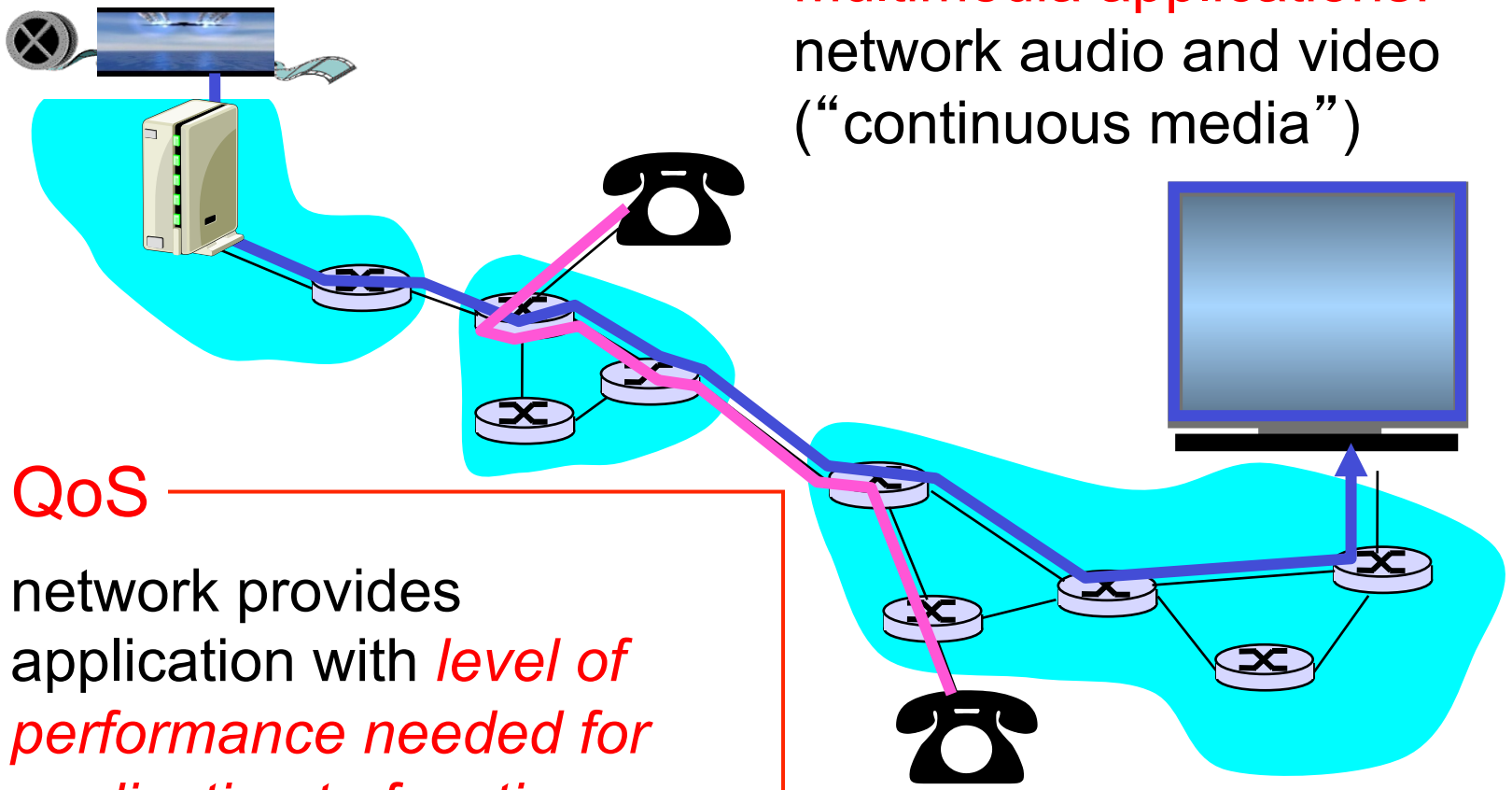
<table>
<tr><td rowspan="3">Client-Server</td><td colspan="4">Peer-to-Peer</td></tr>
<tr><td colspan="3">Unstructured P2P</td><td>Structured P2P</td></tr>
<tr><td colspan="2">1. Generation</td><td>2. Generation</td><td>3. Generation</td></tr>
<tr><td>Centralized P2P</td><td>Pure P2P</td><td>Hybrid P2P</td><td>DHT-based</td></tr>
<tr><td>www</td><td>Napster</td><td>Gnutella 0.4, Freenet</td><td>Gnutella 0.6</td><td>Cord, CAN, Pastry</td></tr>
</table>

Chapter 3: Multimedia Networking

Overview

- 3.1 Multimedia Networking Applications
- 3.2 Streaming stored audio and video
- 3.3 Real-time Multimedia: Internet Phone study
- 3.4 Protocols for Real-Time Interactive Applications
  - RTP,RTCP
- 3.5 IP Telefony, SIP, and H.323
- 3.6 Distributing Multimedia: content distribution networks

Multimedia, Quality of Service: What is it?



Multimedia applications:
network audio and video
("continuous media")

QoS

network provides application with *level of performance needed for application to function.*

MM Networking Applications

Classes of MM applications:

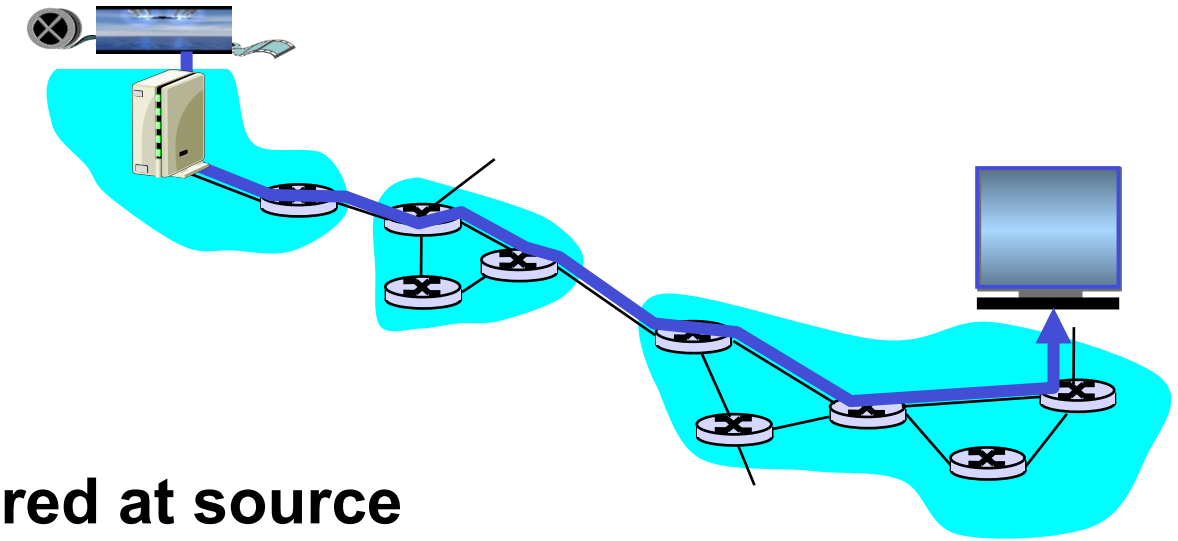
- **Streaming stored audio and video**
- **Streaming live audio and video**
- **Real-time interactive audio and video**

Fundamental characteristics:

- **Typically delay sensitive**
  - end-to-end delay
  - delay jitter
- **But loss tolerant: infrequent losses cause minor glitches**
- **Antithesis of data, which are loss intolerant but delay tolerant**

Jitter is the variability of packet delays within the same packet stream

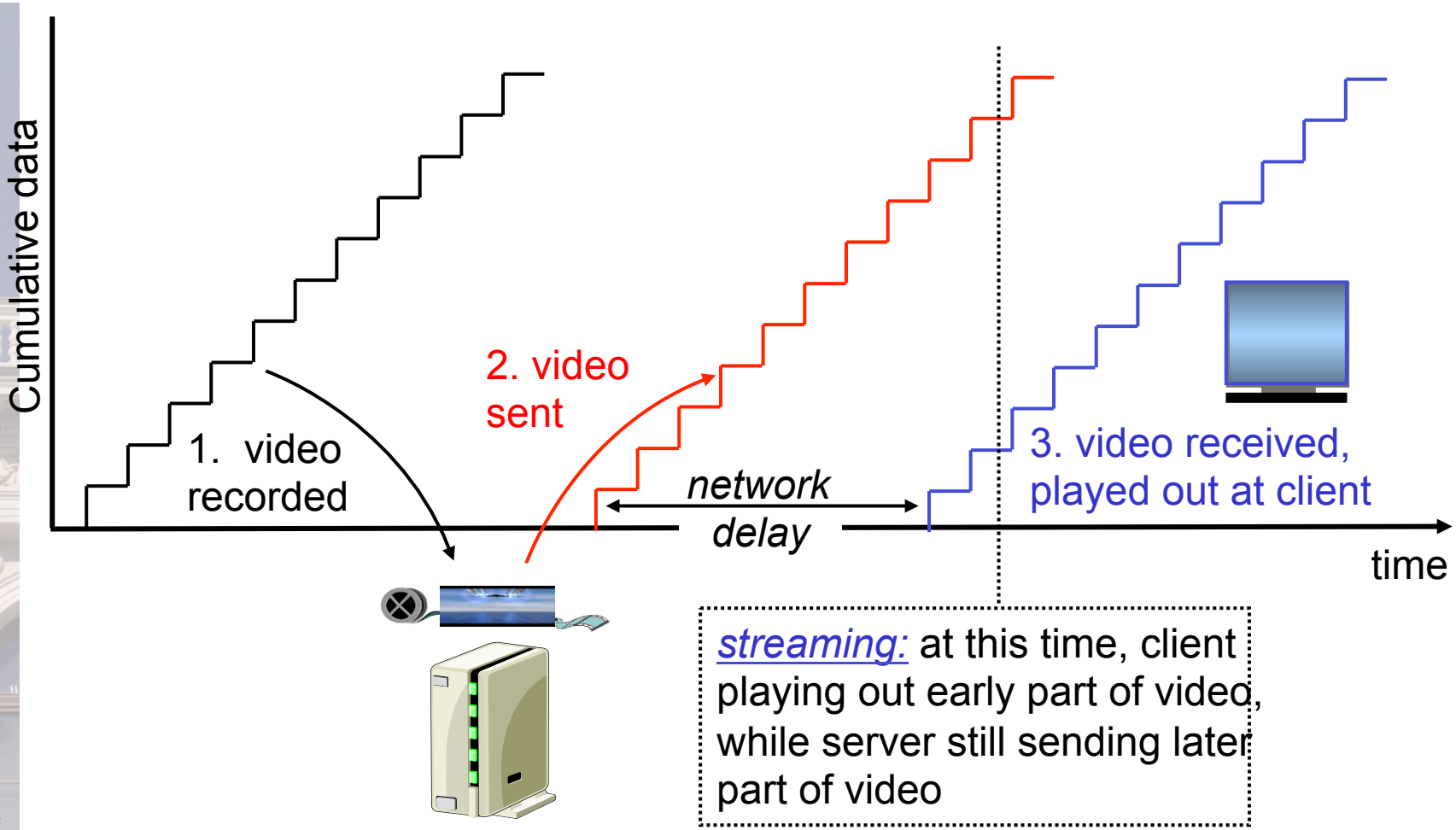
Streaming Stored Multimedia



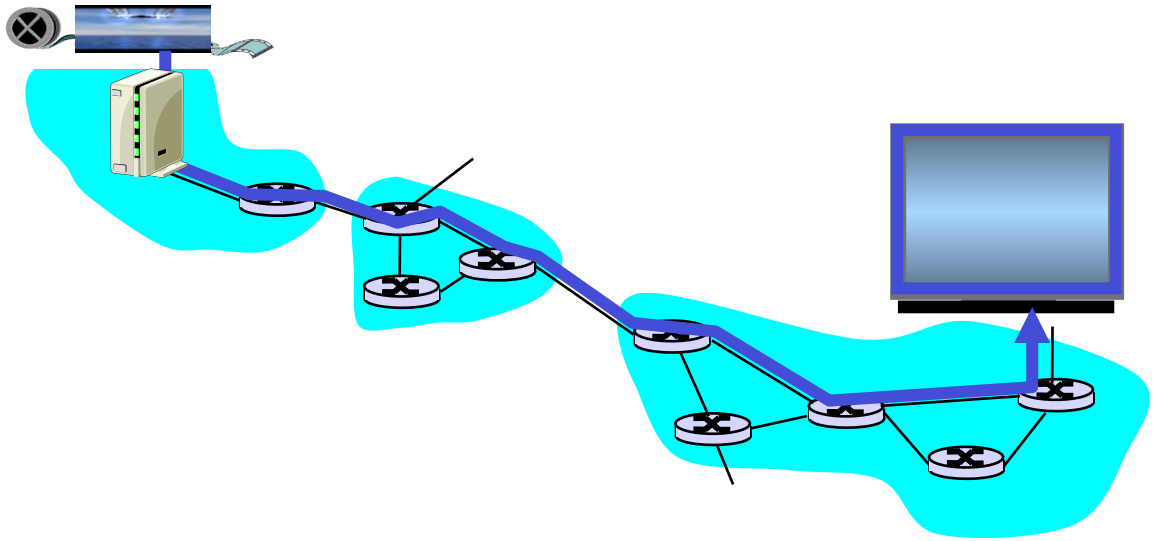
Streaming:

- **media stored at source**
- **transmitted to client**
- **streaming: client playout begins *before* all data has arrived**
- **timing constraint for still-to-be transmitted data: in time for playout**

Streaming Stored Multimedia: What is it?



Streaming Stored Multimedia: Interactivity



- ***VCR-like functionality:*** **client can pause, rewind, FF, push slider bar**
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK
  - RTSP often used (more later)
- **timing constraint for still-to-be transmitted data: in time for playout**

Streaming Live Multimedia

Examples:

- Internet radio talk show
- Live sporting event

Streaming

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity

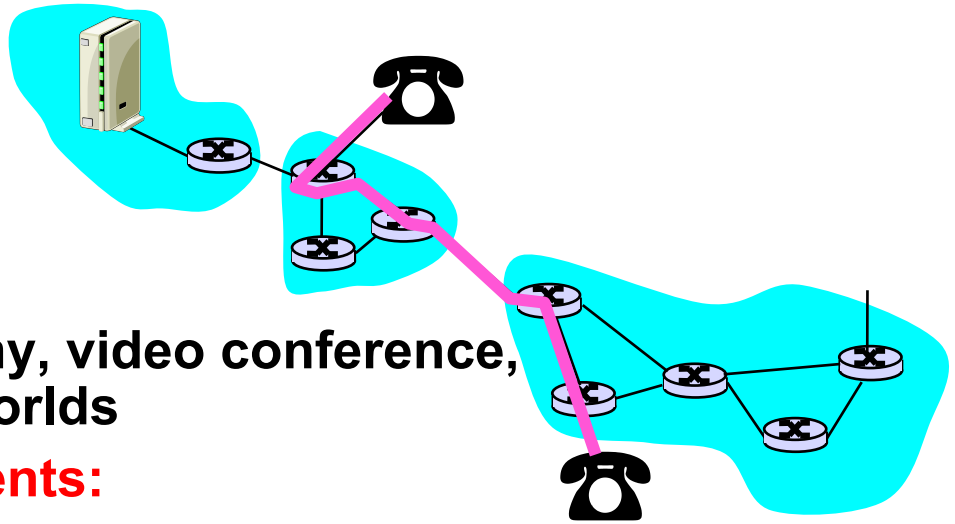
- fast forward impossible
- rewind, pause possible!

universität wien

Chair of Future Communication

endowed by A1 TELEKOM AUSTRIA

Interactive, Real-Time Multimedia



- **applications: IP telephony, video conference, distributed interactive worlds**
- **end-end delay requirements:**
  - audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays
    - higher delays noticeable, impair interactivity
- **session initialization**
  - how does callee advertise its IP address, port number, encoding algorithms?

Multimedia Over Today's Internet

TCP/UDP/IP: "best-effort service"

- ***no*** **guarantees on delay, loss**

But you said multimedia apps requires QoS and level of performance to be effective!

Today's Internet multimedia applications use application-level techniques to mitigate (as best possible) effects of delay, loss

How should the Internet evolve to better support multimedia?

Integrated services philosophy:

- Fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- Requires new, complex software in hosts & routers

Laissez-faire

- no major changes
- more bandwidth when needed
- content distribution, application-layer multicast
  - application layer

Differentiated services philosophy:

- Fewer changes to Internet infrastructure, yet provide 1st and 2nd class service.

What's your opinion?

A few words about audio compression

- **Analog signal sampled at constant rate**
  - telephone: 8,000 samples/ sec
  - CD music: 44,100 samples/sec
- **Each sample quantized, i.e., rounded**
  - e.g., 2^8=256 possible quantized values
- **Each quantized value represented by bits**
  - 8 bits for 256 values
- **Example: 8,000 samples/ sec, 256 quantized values --> 64,000 bps**
- **Receiver converts it back to analog signal:**
  - some quality reduction

Example rates

- **CD: 1.411 Mbps**
- **MP3: 96, 128, 160 kbps**
- **Internet telephony: 5.3 - 13 kbps**

A few words about video compression

- **Video is sequence of images displayed at constant rate**
  - e.g. 24 images/sec
- **Digital image is array of pixels**
- **Each pixel represented by bits**
- **Redundancy**
  - spatial
  - temporal

Examples:

- **MPEG 1 (CD-ROM) 1.5 Mbps**
- **MPEG2 (DVD) 3-6 Mbps**
- **MPEG4 (often used in Internet, < 1 Mbps)**

Research:

- **Layered (scalable) video**
  - adapt layers to available bandwidth

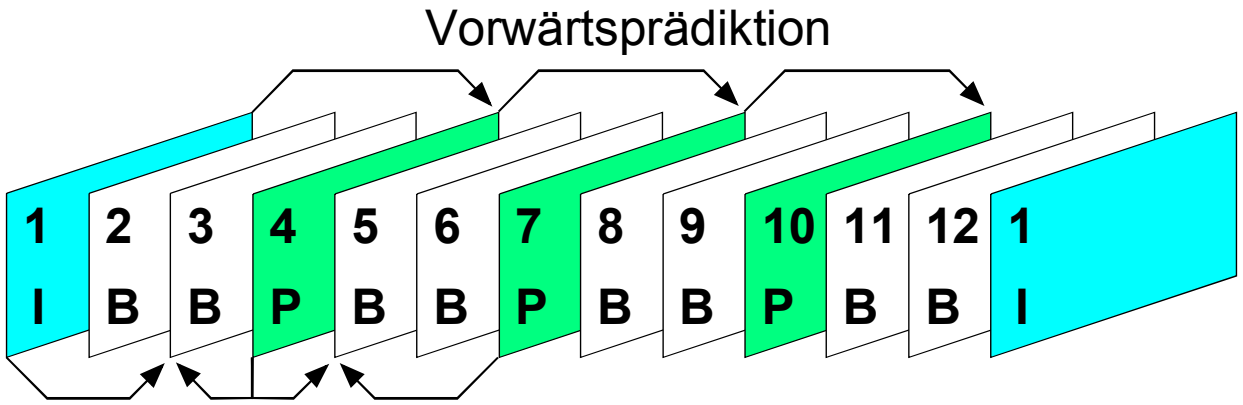
Generation of MPEG Video Sequences

- **Given: sequence of digital images**
- **MPEG compression is combination of**
  - Intra-frame compression (spatial redundancy reduction)
    - Discrete-Consine Transformation (DCT):
      8x8 pixel blocks ⇒ DCT ⇒ 8x8 DC coefficients
    - Quantization
    - Zig-zag entropy encoding
  - Inter-frame compression (temporal redundancy reduction)
    - Block-based motion compensation

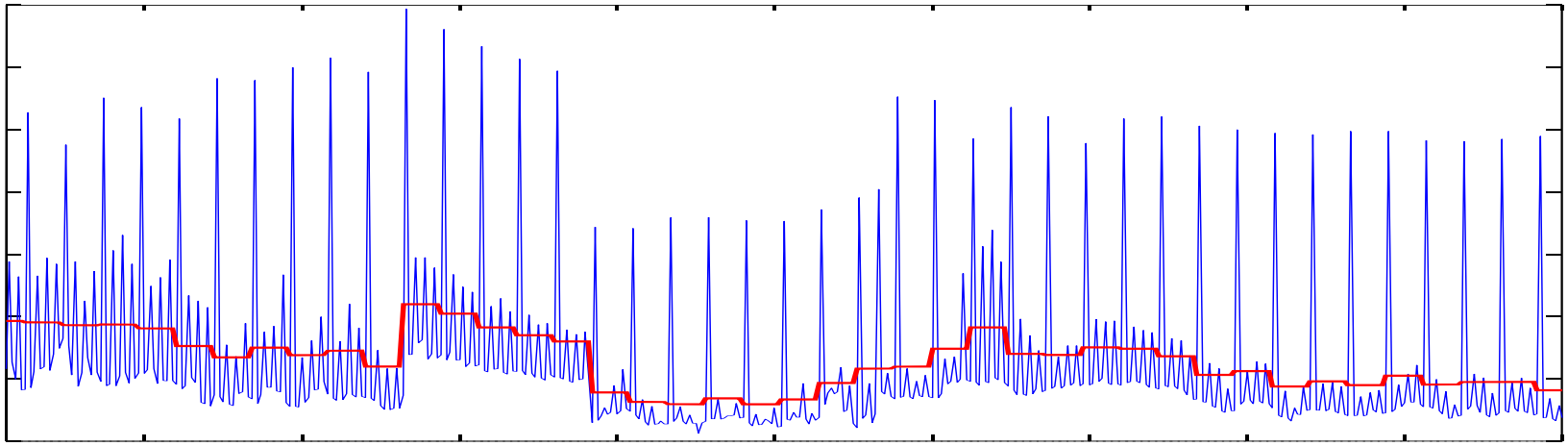
Inter-Frame Dependencies

- **Frame types**
  - Intracoded frames (I-frame)
    - Does not depend on any other frames
    - Most important information, largest frame size
  - Predicted frames (P-frame)
    - Depends on preceding I- or P-frame
    - Medium frame size
  - Bidirectional frames (B-frame)
    - Depends on preceding and succeeding I- or P-frame
    - Small frame size, no other frame depends on it
- **Group of Pictures (GoP)**
  - All frames following and depending on a specific I-frame

Measured Time Series of MPEG Frame Sizes

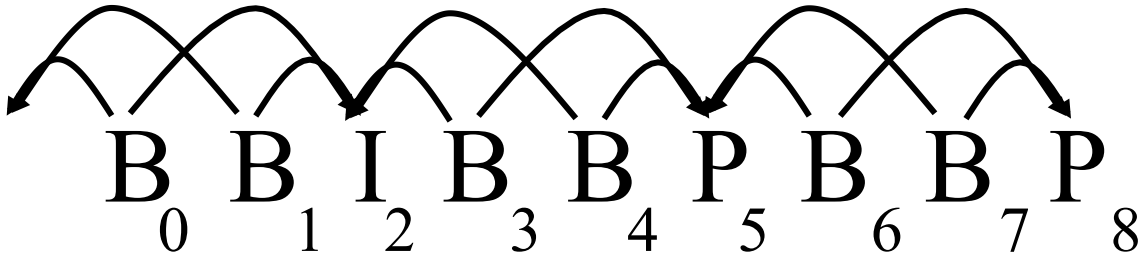


Bidirektionale Prädiktion



Display and Bitstream Order

- **Display order**



- **Bitstream order**