INF5071 – Performance in Distributed Systems

Introduction & Motivation

29/8 - 2008

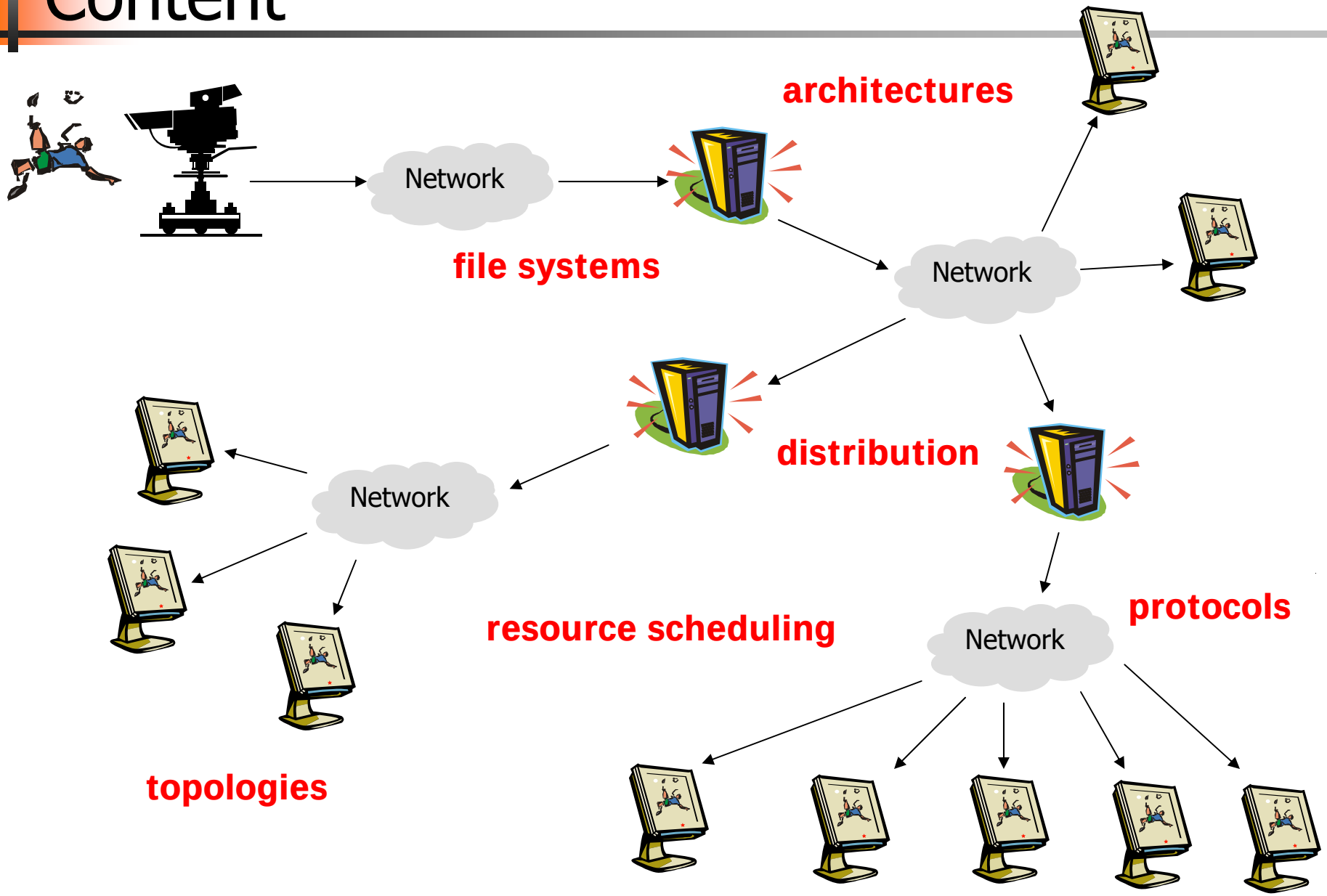
Overview

- About the course
- Application and data evolution
- Architectures
- Machine Internals
- Network approaches
- Case studies

Lecturers and Teaching Assistant

- Paul B. Beskow
  - email: paulbb @ ifi
  - office: Simula 111
- Carsten Griwodz
  - email: griff @ ifi
  - office: Simula 154
- Pål Halvorsen
  - email: paalh @ ifi
  - office: Simula 153

Content


architectures
Network
file systems
Network
distribution
Network
resource scheduling
protocols
Network
topologies

Content

- Applications and characteristics
  (components, requirements, …)
- Server examples and resource management
  (CPU and memory management)
- Storage systems
  (management of files, retrieval, …)

Content

- Protocols with and without Quality of Service (QoS)
  (specific and generic QoS approaches)
- Distribution
  (use of caches and proxy servers)
- Peer-to-Peer
  (various clients, different amount of resources)
- Guest lectures?:
  (architecture, resource utilization and performance, storage and distribution of data, parallelism, etc.)
  - The fast A Microsoft Subsidiary searching system
  - Schibsted media house

Content - student assignment

- Mandatory student assignment
  (will be presented more in-depth later):
  - write a **project plan** describing your assignment
  - write a **report** describing the results and give a **presentation** (probably November 14th)
  - for example (examples from earlier):
    - Transport protocols for various scenarios
    - Network emulators
    - Comparison of Linux schedulers (cpu, network, disk)
    - File system benchmarking (different OSes and file systems)
    - Comparison of methods for network performance monitoring (packet train, packet pair, ping, tcpdump library/pcap, ...)
    - Compare media players (VLC, mplayer, xine, ...)
    - Virtualization
    - ...

  ↳ it has to be **something in the context of performance**!!!

Goals

- Distribution system mechanisms enhancing performance
  - architectures
  - system support
  - protocols
  - distribution mechanisms
  - …
- Be able to evaluate any combination of these mechanisms

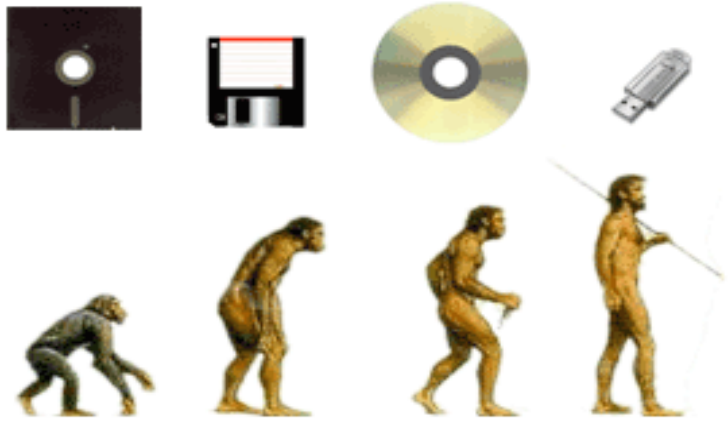
Exam

- Prerequisite:
  approved presentation of student assignment

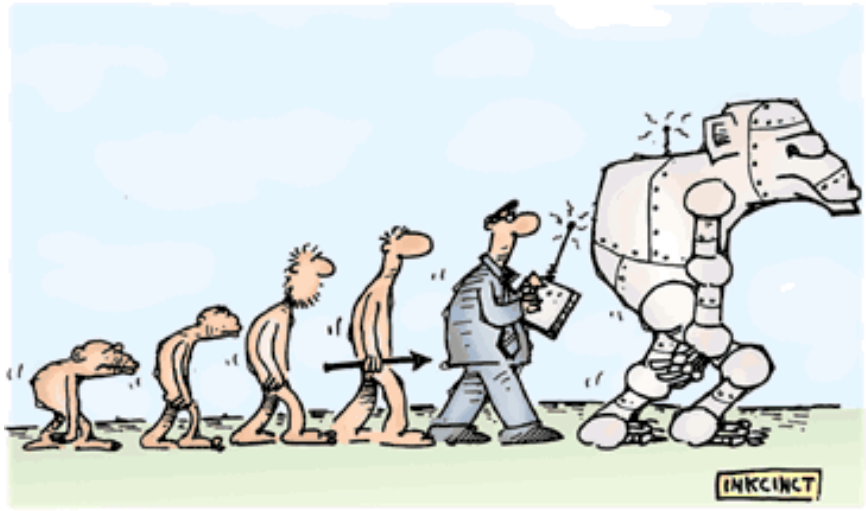
- Oral exam (**early December**):
  - all *transparencies* from lectures

    Note: we do NOT have a book, and you probably do not want to read all the articles the slides are made from!
    ↳ come to the lectures…

  - content of your *own student assignment*



Evolution

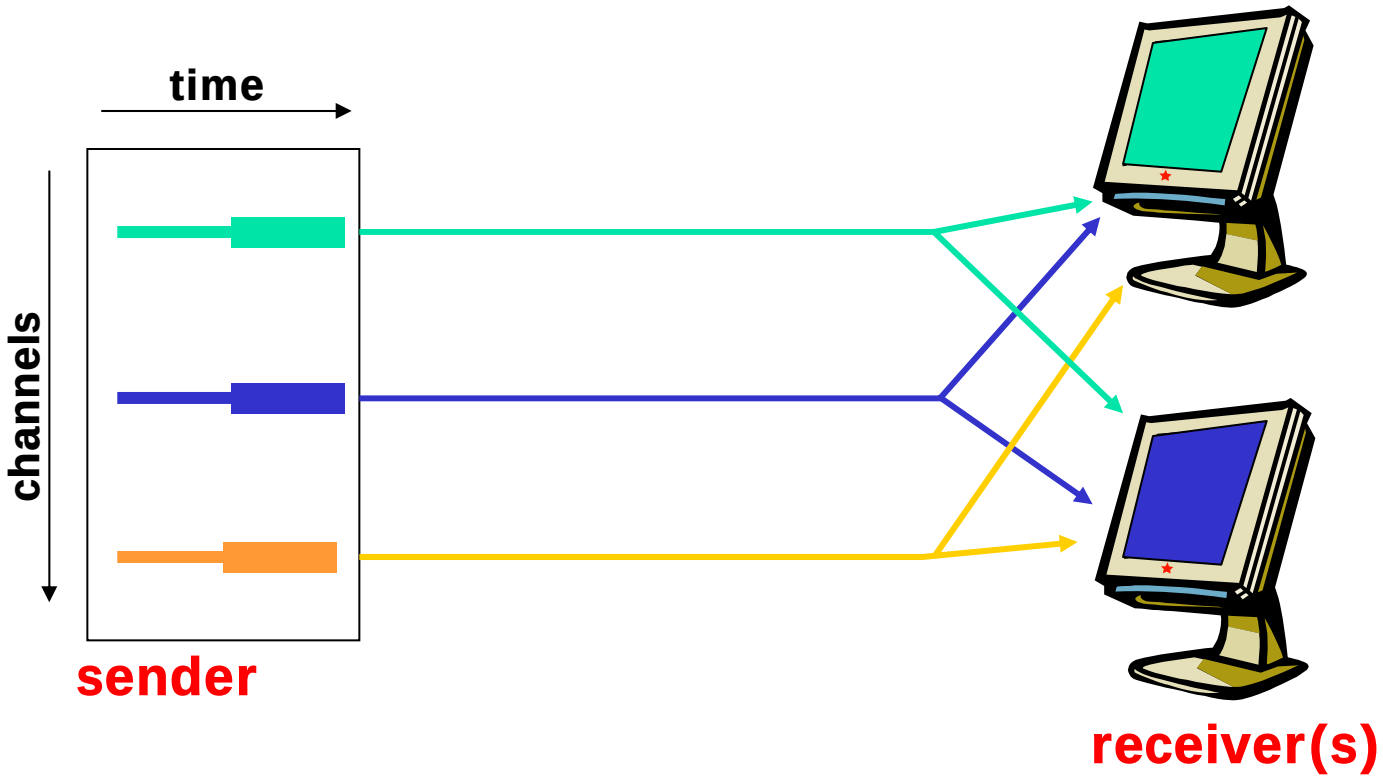


Discrete Data to Continuous Media Data

3D streaming is coming …

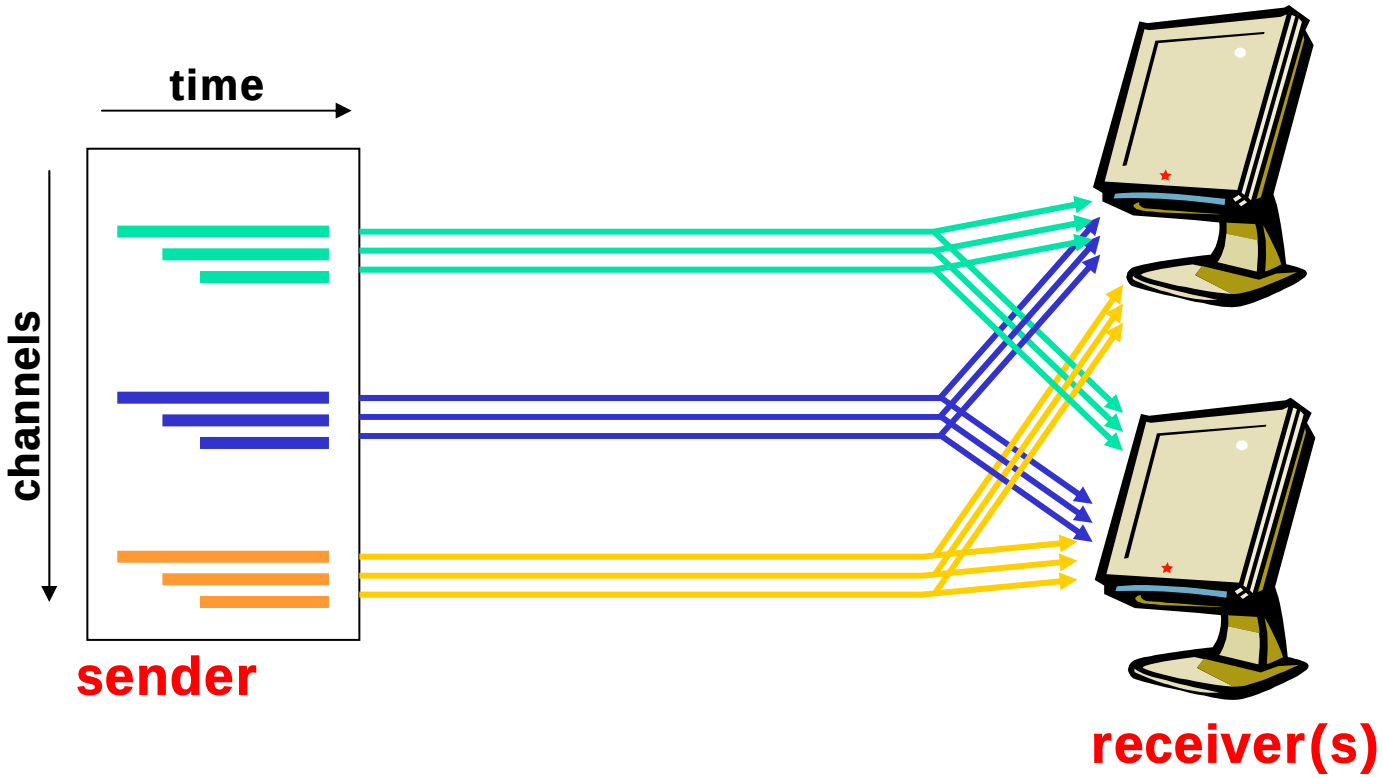
Evolution of (continuous) media streams:

Television (Broadcast)



- **analog or digital**
- **traditionally, one program per channel**
  - ❑ analog use frequency division multiplexing only
  - ❑ digital may additionally use time division multiplexing inside one frequency (several programs per channel)

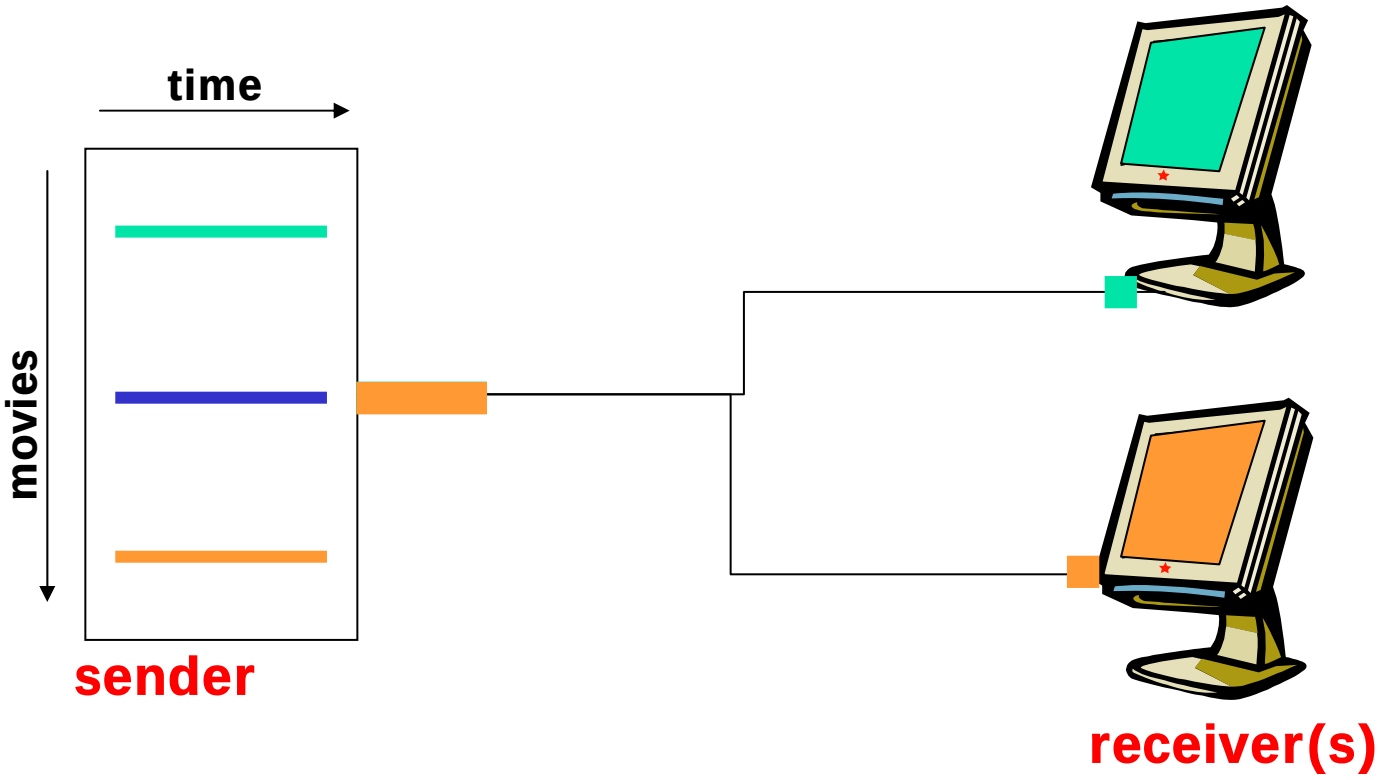
Near Video-on-Demand (NVoD)



- **analog or digital broadcasting**
- **one program over multiple channels**
- **time-slotted emission of the program**

Evolution of (continuous) media streams:

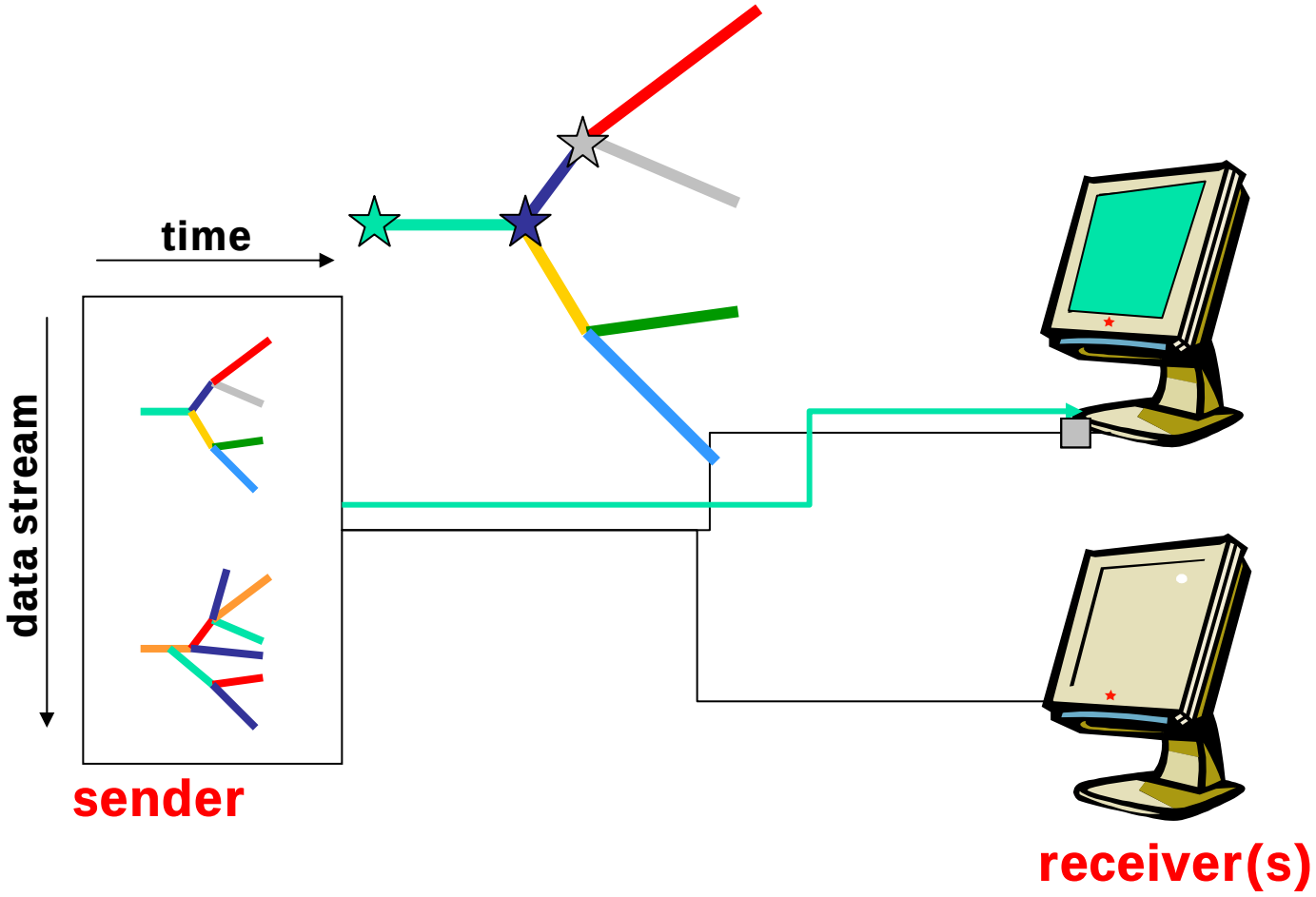
(True) Video-on-Demand (VoD)



- **digital uni- or multicasting**
- **control channels**

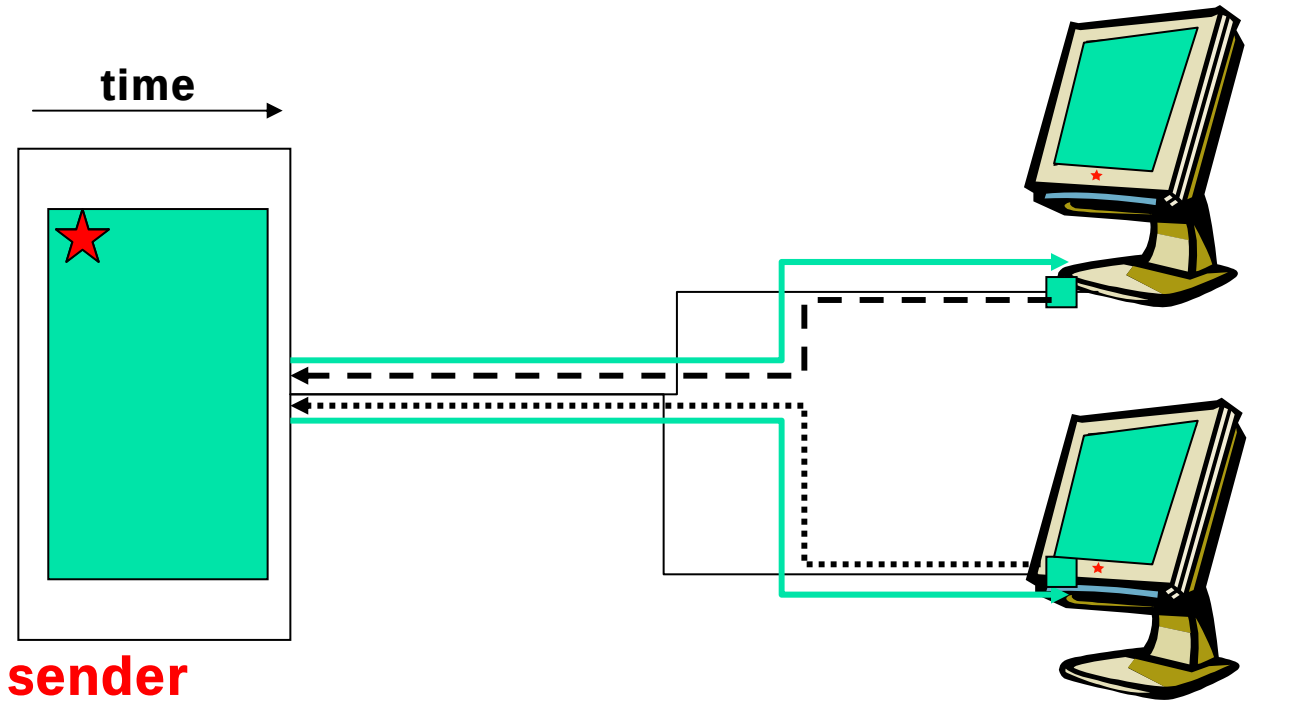
Evolution of (continuous) media streams:

"Interactive Vision"



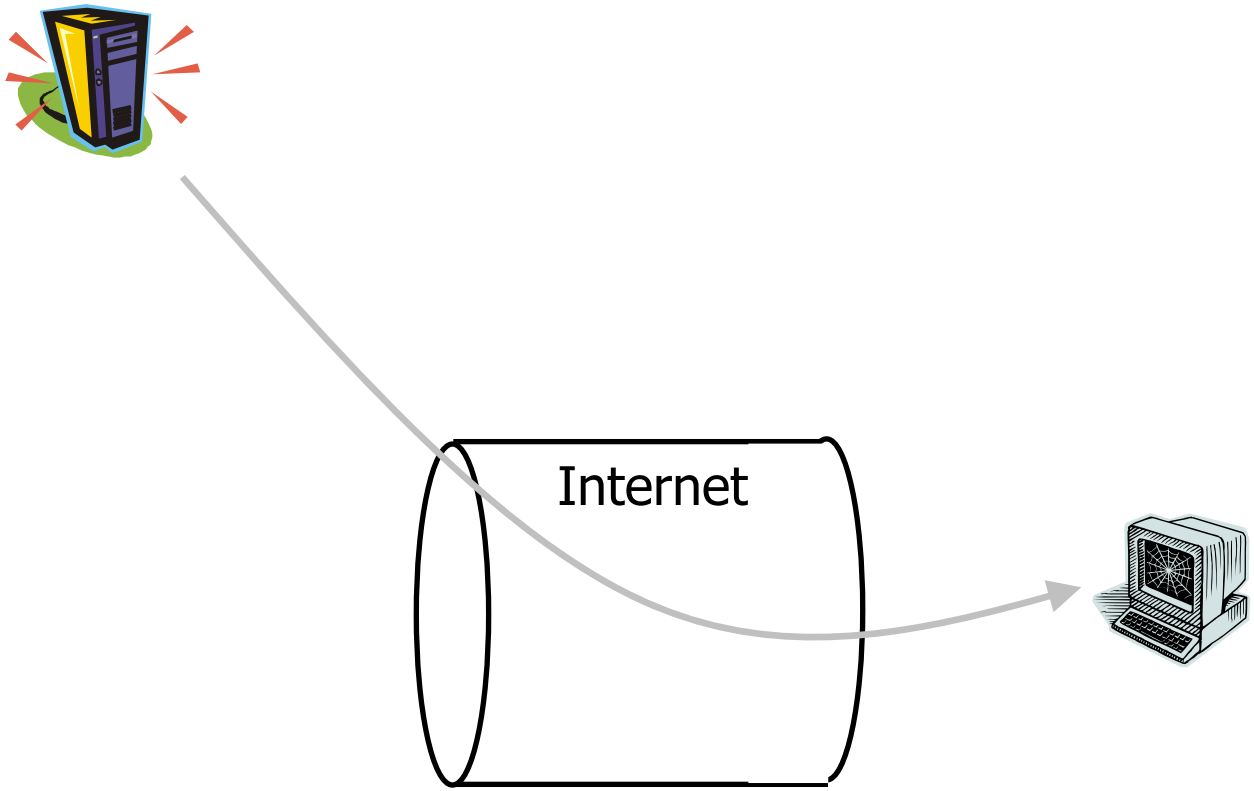
- **digital uni- or multicasting**
- **control channels**
- ***fixed non-linear* data streams**

Evolution of (continuous) media streams: "Cyber Vision"



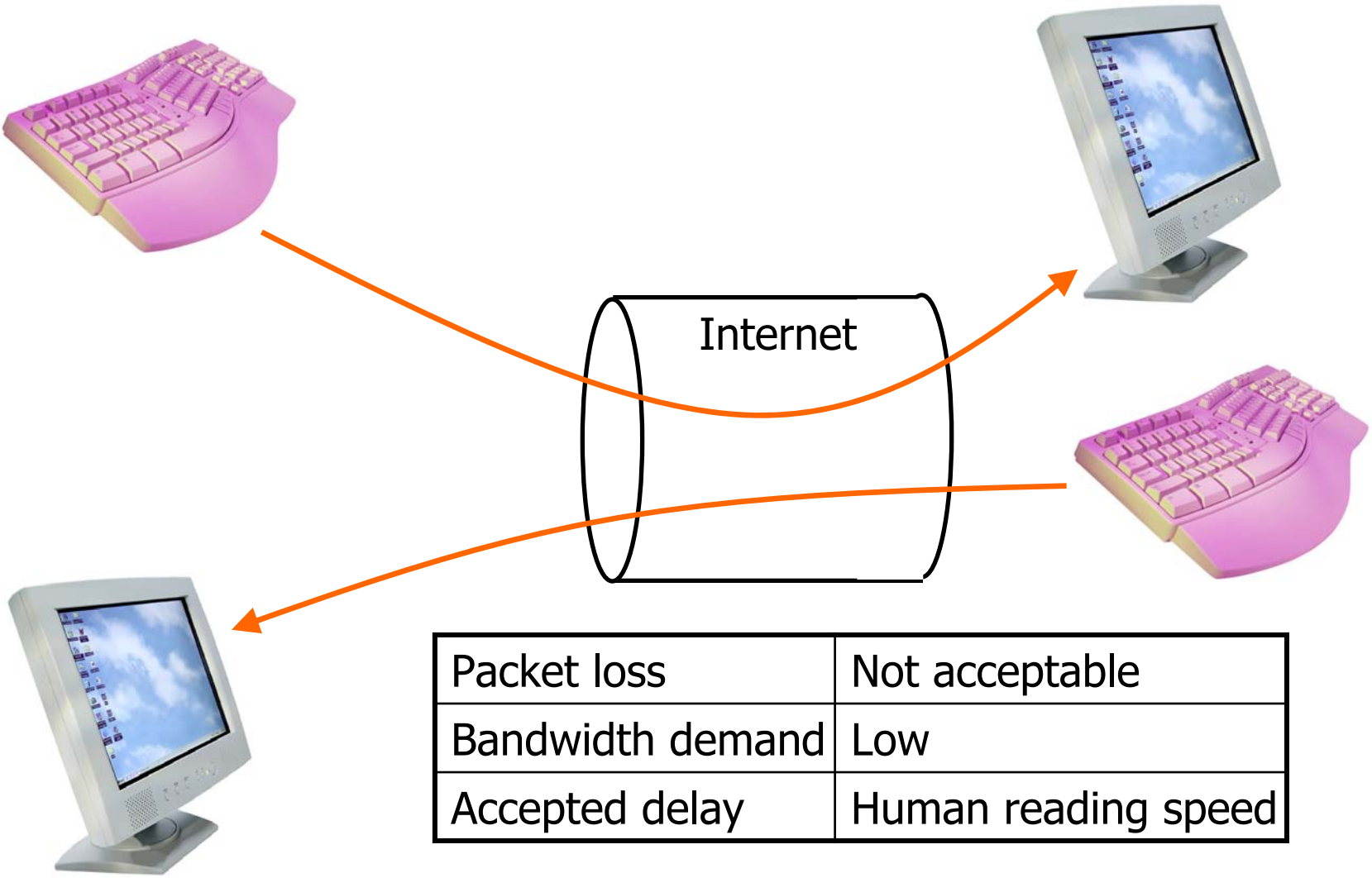
- **digital uni- or multicasting**
- **control channels**
- ***variable non-linear* "media", e.g.,**
  - **games, virtual reality, ...**

File download and Web browsing



Packet loss	Not acceptable
Bandwidth demand	Low (?)
Accepted delay	Medium – High (?)

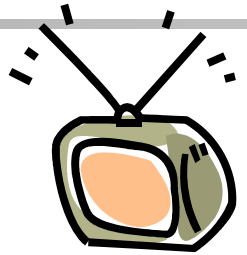
Textual commands and textual chat

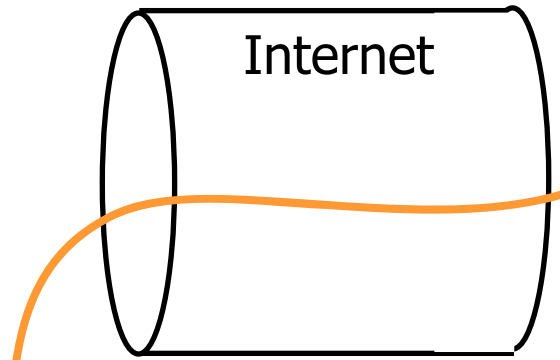


Packet loss	Not acceptable
Bandwidth demand	Low
Accepted delay	Human reading speed

Live and on-Demand Streaming

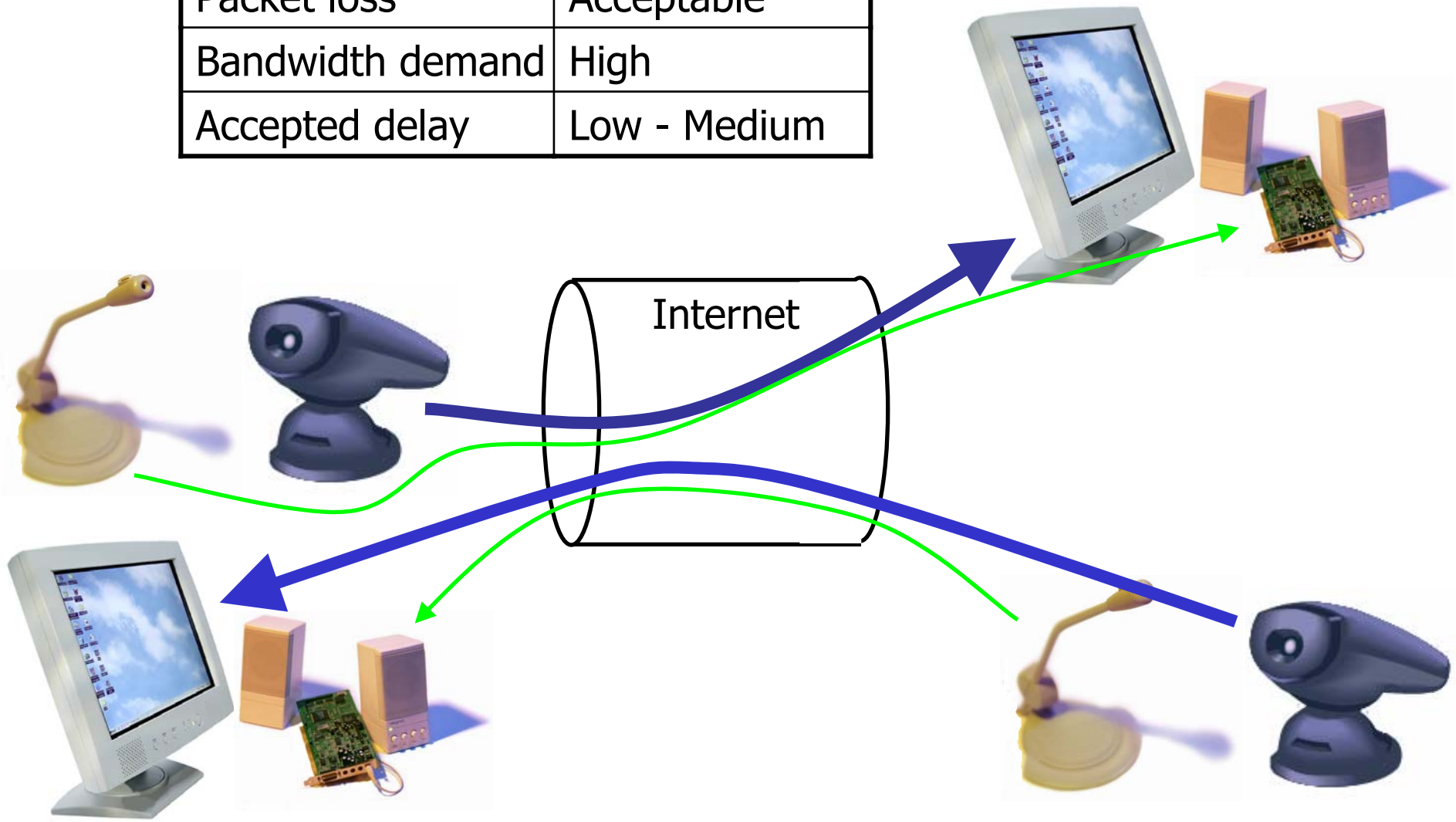
Packet loss	Acceptable
Bandwidth demand	High
Accepted delay	Medium



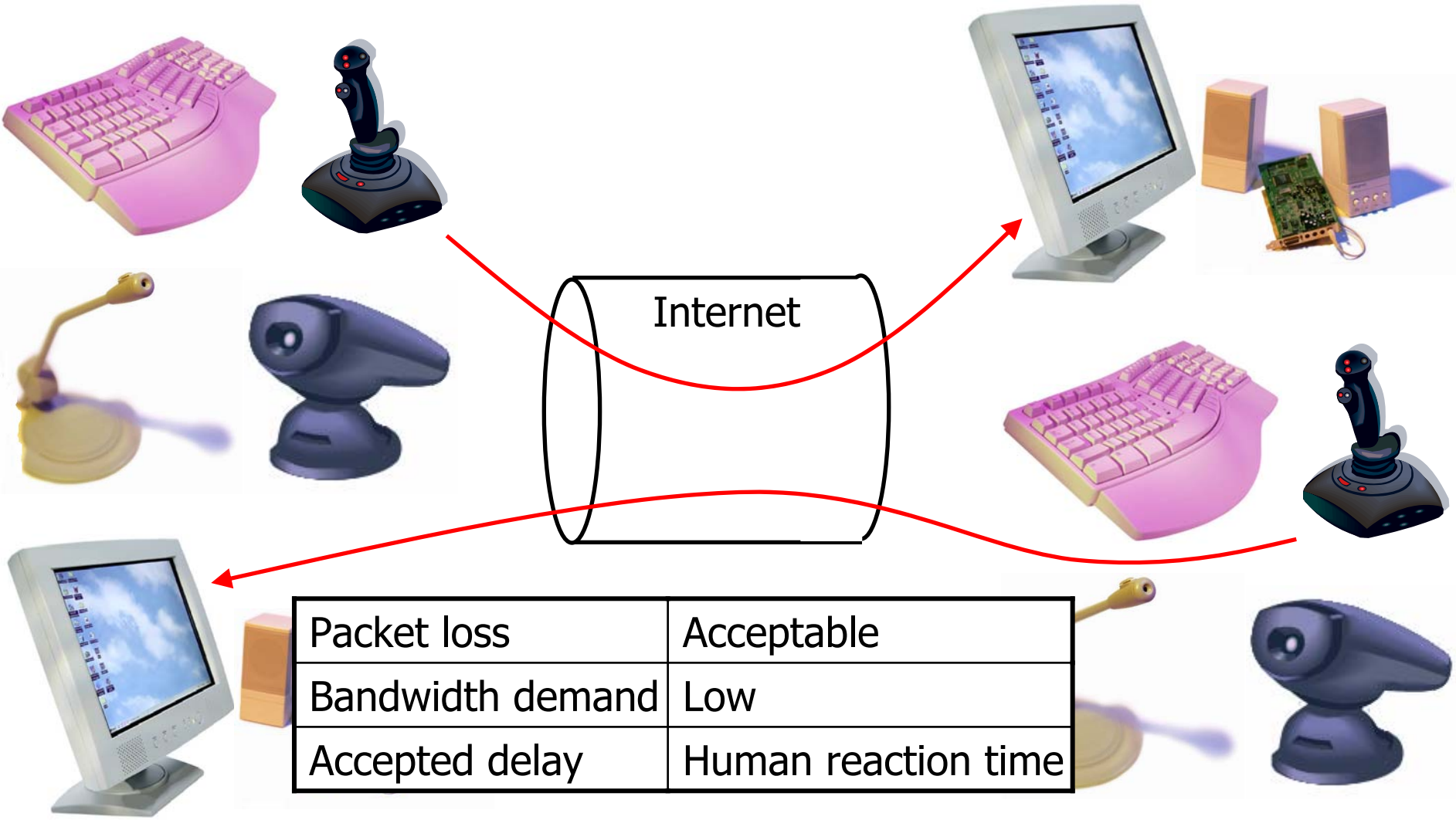


AV chat and AV conferencing

Packet loss	Acceptable
Bandwidth demand	High
Accepted delay	Low - Medium



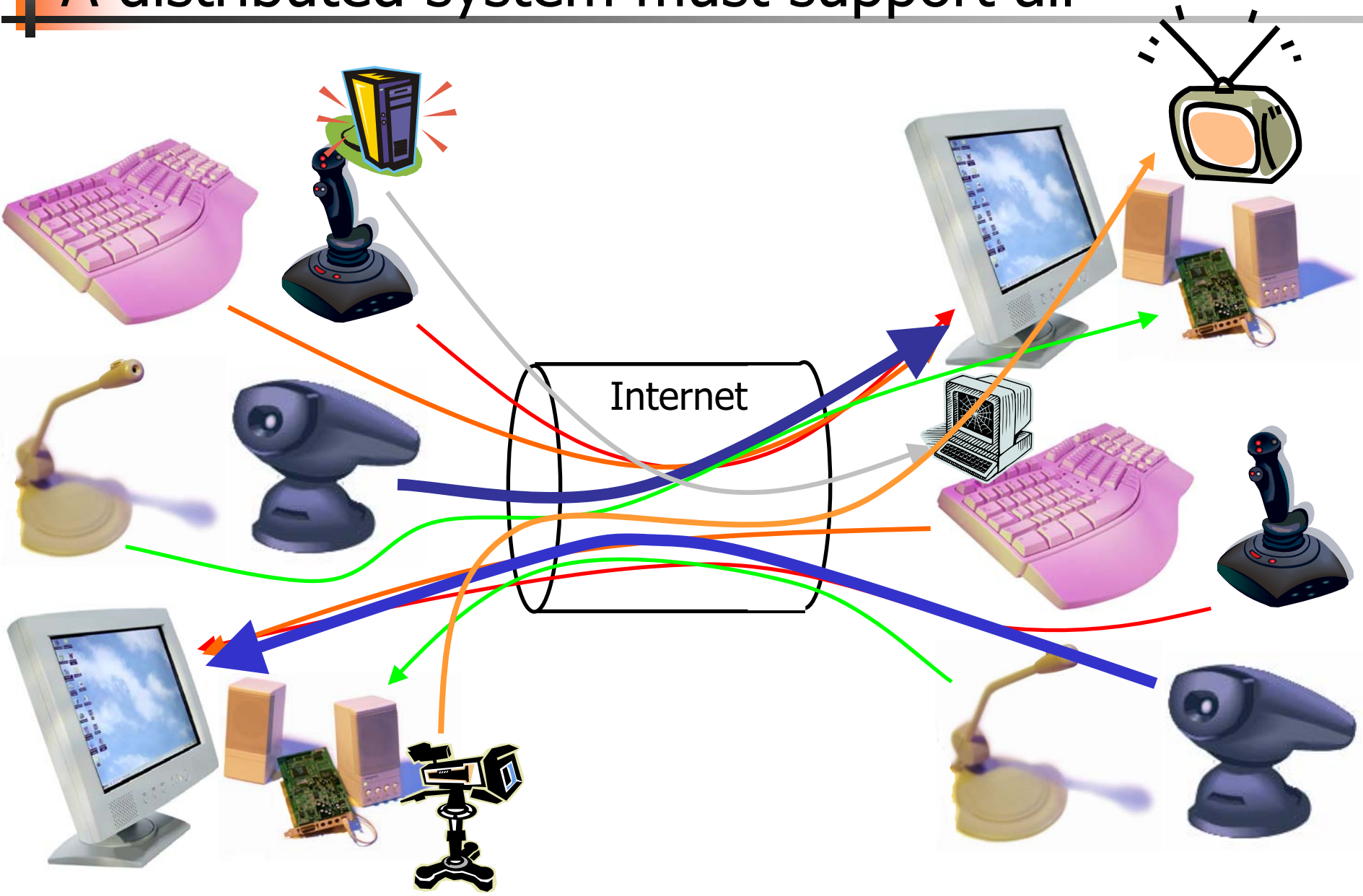
Haptic Interaction



Packet loss	Acceptable
Bandwidth demand	Low
Accepted delay	Human reaction time

Evolution & Requirements:

A distributed system must support all



Different Views on Requirements

- Application / user
  - QoS – time sensitivity?
  - resource capabilities – bandwidth, latency, loss, reliability, ...
  - best possible perception
- Business
  - scalability
  - reliability
- Architectural
  - topology
  - cost vs. performance

Technical Challenges

- Servers (and proxy caches)
  - **storage**
    - continuous media streams, e.g.:
      - 4000 movies * 90 minutes * 15 Mbps (HDTV) = 40.5 TB
      - 2000 CDs * 74 minutes * 1.4 Mbps = 1.4 TB
    - metrological data, physics data, …
    - web data – people put everything out nowadays
  - **I/O**
    - many concurrent clients
    - real-time retrieval
    - continuous playout
      - DVD (~4Mbps)
      - HDTV (~15Mbps)
    - current examples of capabilities
      - disks:
        - mechanical: e.g., Seagate X15 - ~400 Mbps
        - SSD: e.g., MTRON Pro 7000 – ~1.2 Gbps
      - network: Gb Ethernet (1 and 10 Gbps)
      - bus(ses):
        - PCI 64-bit, 133Mhz (8 Gbps)
        - PCI-Express (2 Gbps each direction/lane, 32x = 64 Gbps)
  - **computing in real-time**
    - encryption
    - adaptation
    - transcoding
    - …

Technical Challenges

- User end system
  - real-time processing of data
    (e.g., 1000 MIPS for an MPEG-II decoder)
  - storage of media/web files
  - request/response delay (< 150 ms for videophon
  - high data rates, e.g., MPEG-II DVD qualit
    - max. total video data rate of ~10
    - average transport strea ers, error protection)
    - max. user rate signals)
  - m tes and share its resources with the rest of the

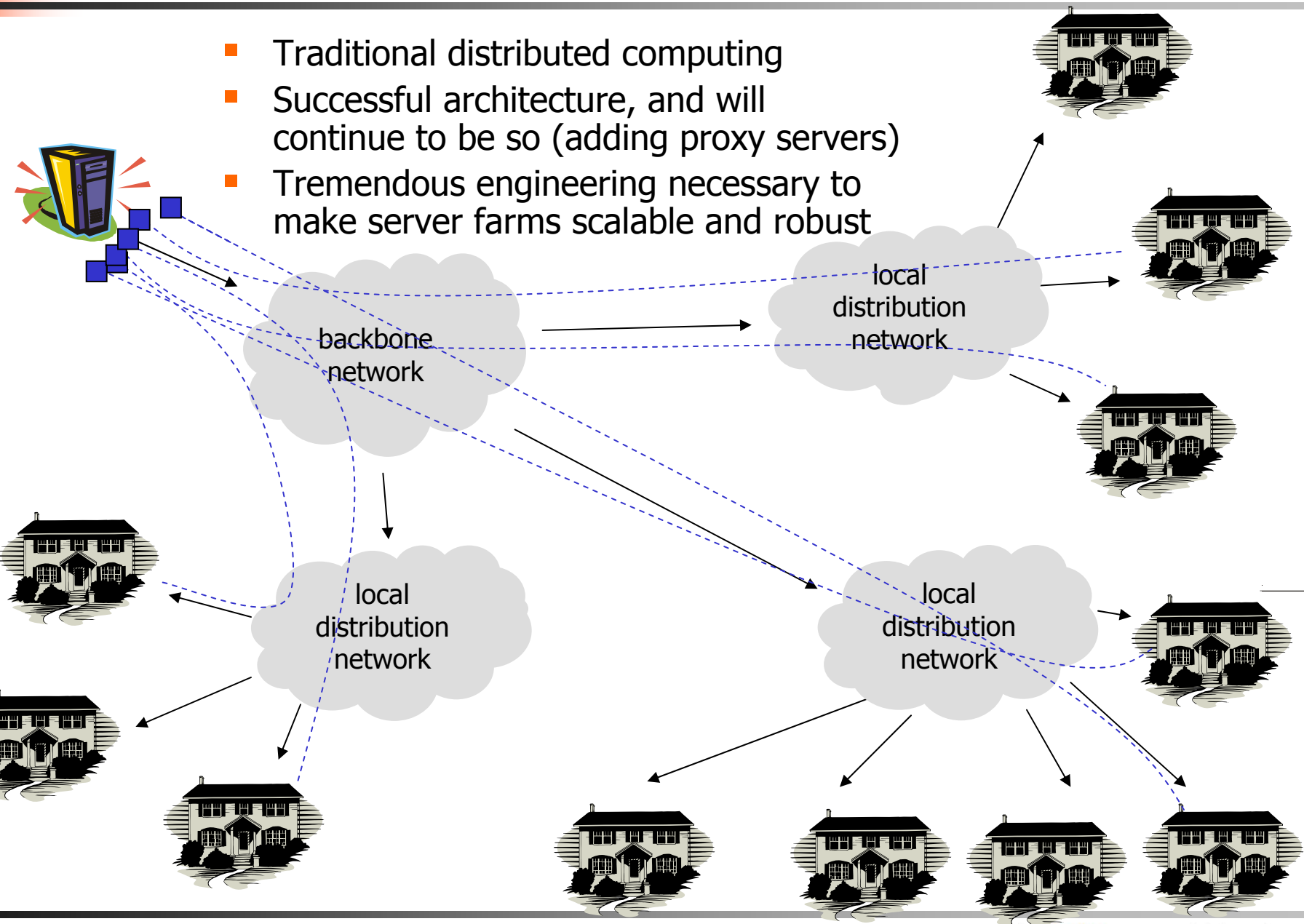
Thus, we will mostly concentrate on **server** – and **network** mechanisms

- Net
  - real-time transport of media data
  - high rate downloads
  - TCP fairness
  - mobility
  - ...

Traditional Distributed Architectures

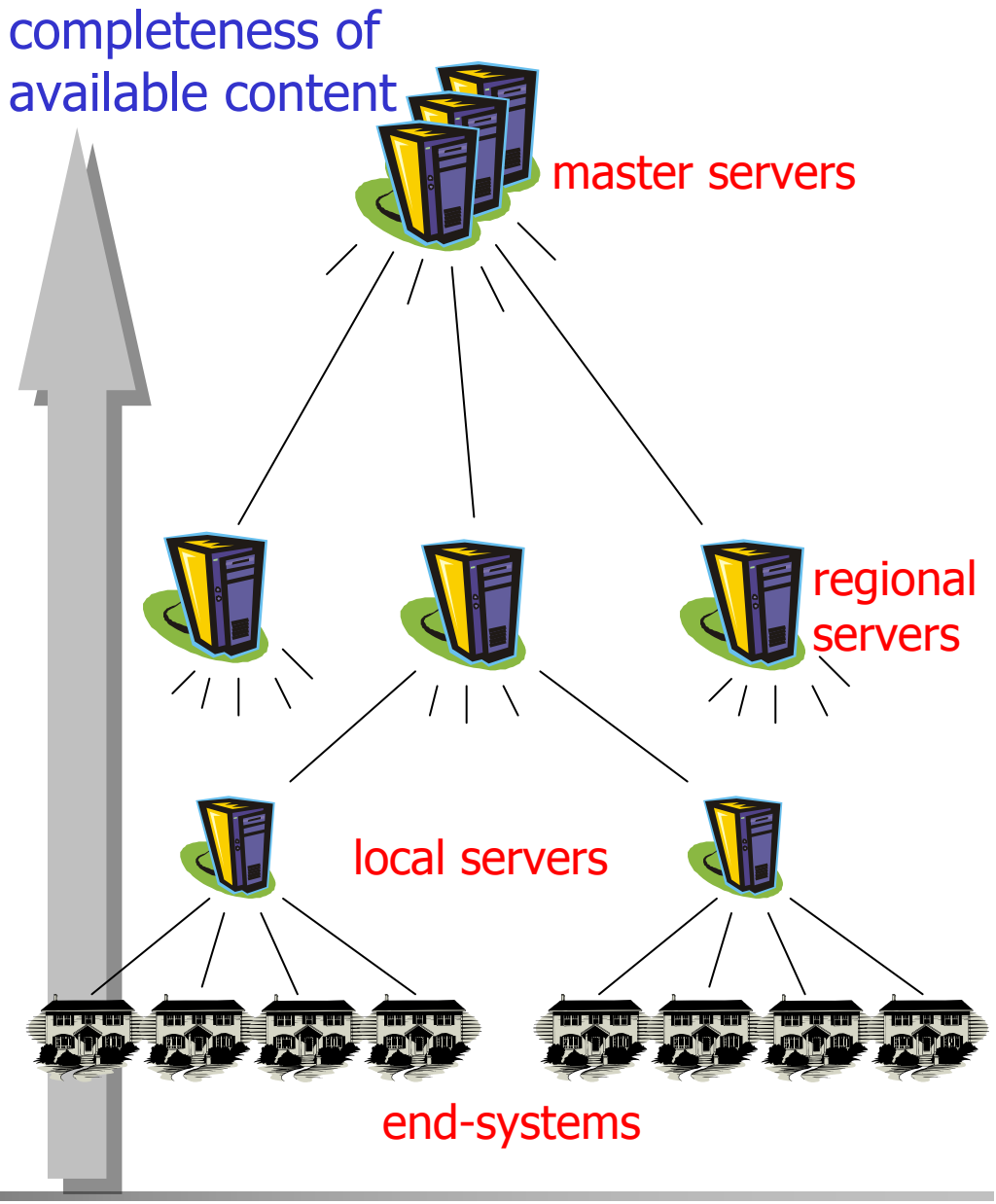
Client-Server

- Traditional distributed computing
- Successful architecture, and will continue to be so (adding proxy servers)
- Tremendous engineering necessary to make server farms scalable and robust



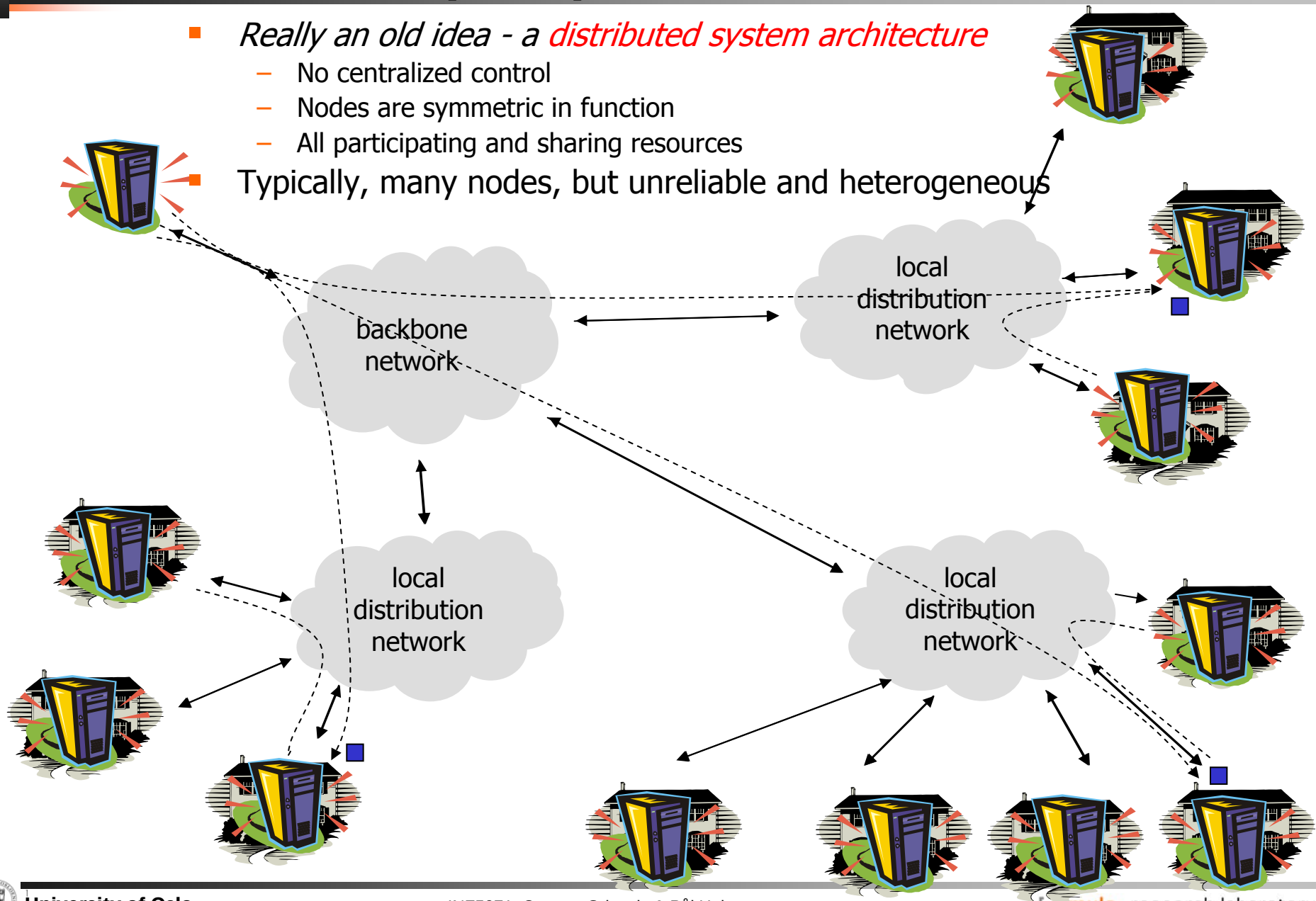
Server Hierarchy

- Intermediate nodes or proxy servers may offload the main master server
- Popularity of data:
  not all are equally popular – most request directed to only a few (Zipf distribution)
- Straight forward hierarchy:
  - popular data replicated and kept close to clients
  - locality vs. communication vs. node costs



Peer-to-Peer (P2P)

- *Really an old idea - a distributed system architecture*
  - No centralized control
  - Nodes are symmetric in function
  - All participating and sharing resources
- Typically, many nodes, but unreliable and heterogeneous

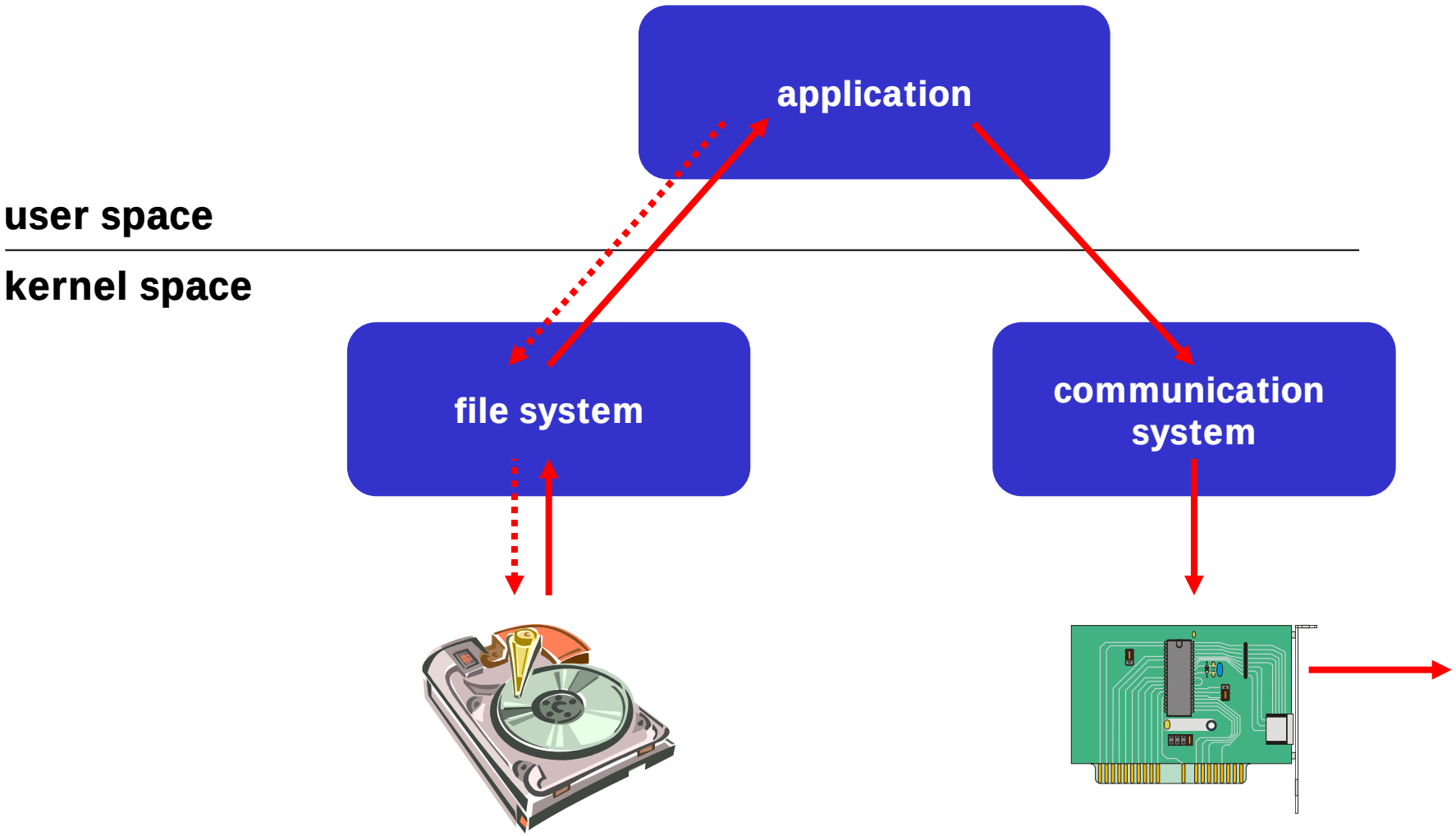


Topologies

- Client / server
  - easy to build and maintain
  - severe scalability problems
- Hierarchical
  - complex
  - potential good performance and scalability
  - consistency challenge
  - cost vs. performance tradeoff
- P2P
  - complex
  - low-cost (for content provider!!)
  - heterogeneous and unreliable nodes
- We will in later lectures look at different issues for all these

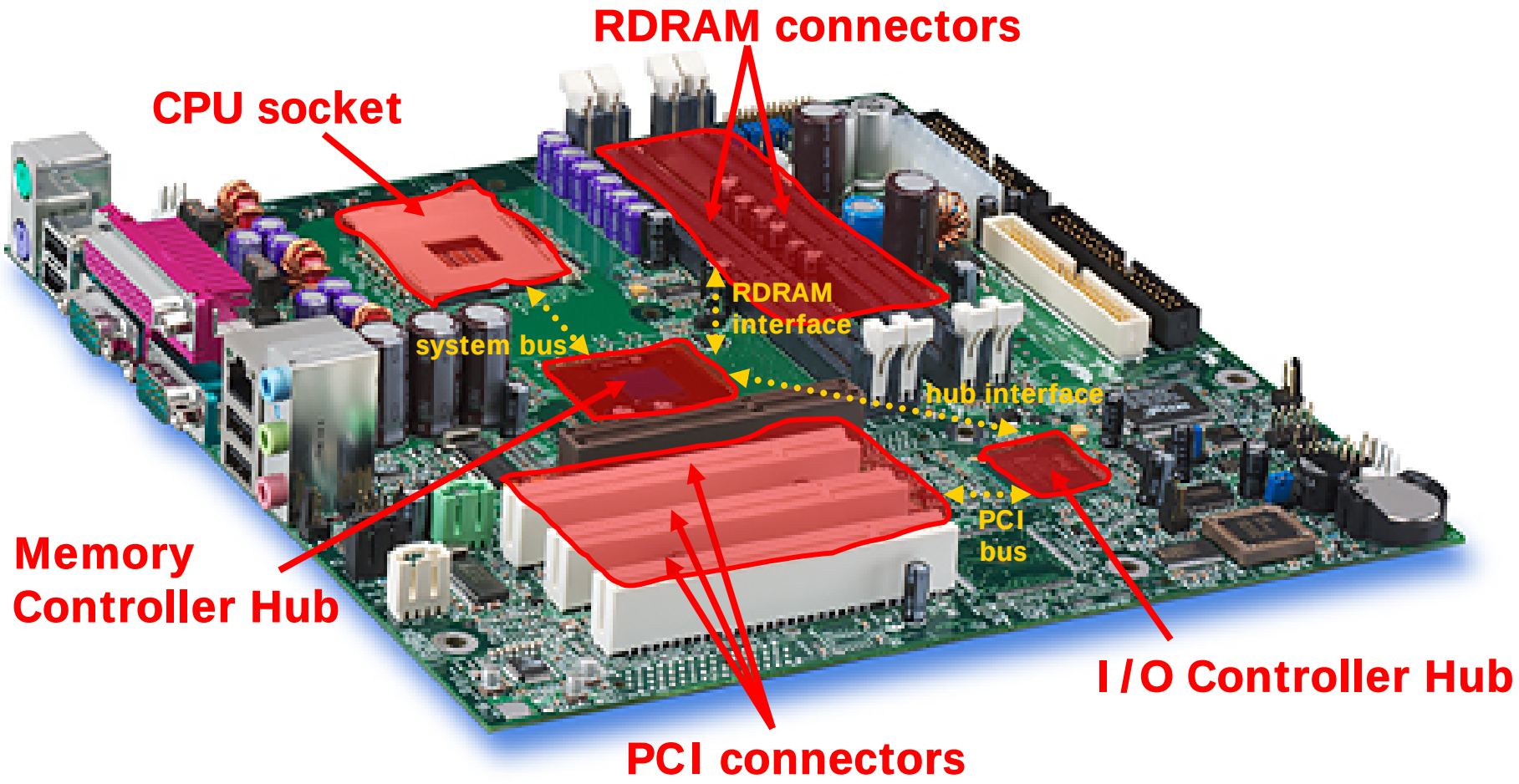
Traditional Server Machine Internals

General OS Structure and Retrieval Data Path



Example:
Intel Hub Architecture (850 Chipset) – I

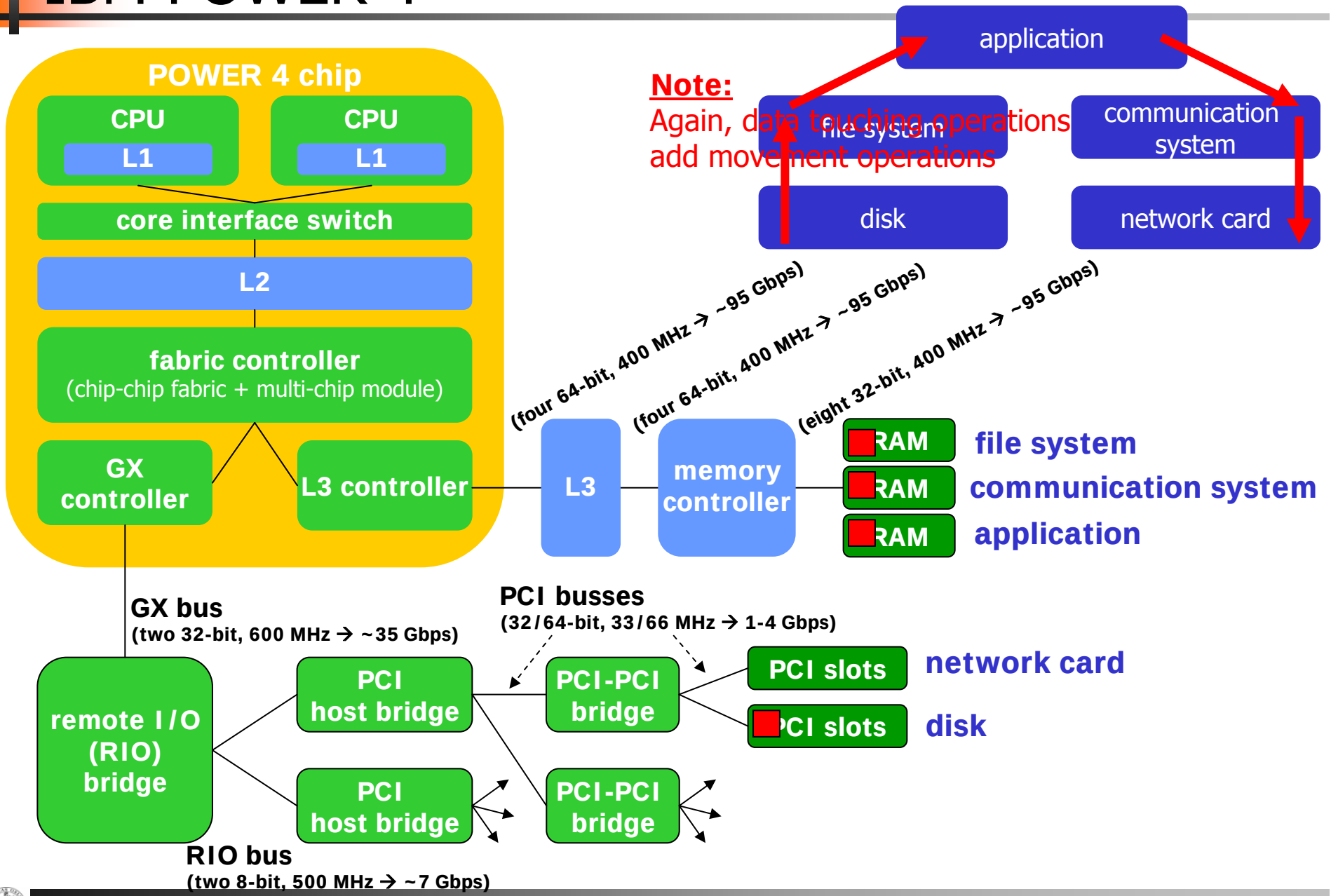
Intel D850MD Motherboard:



Example:
Intel Hub Architecture (850 Chipset) – II

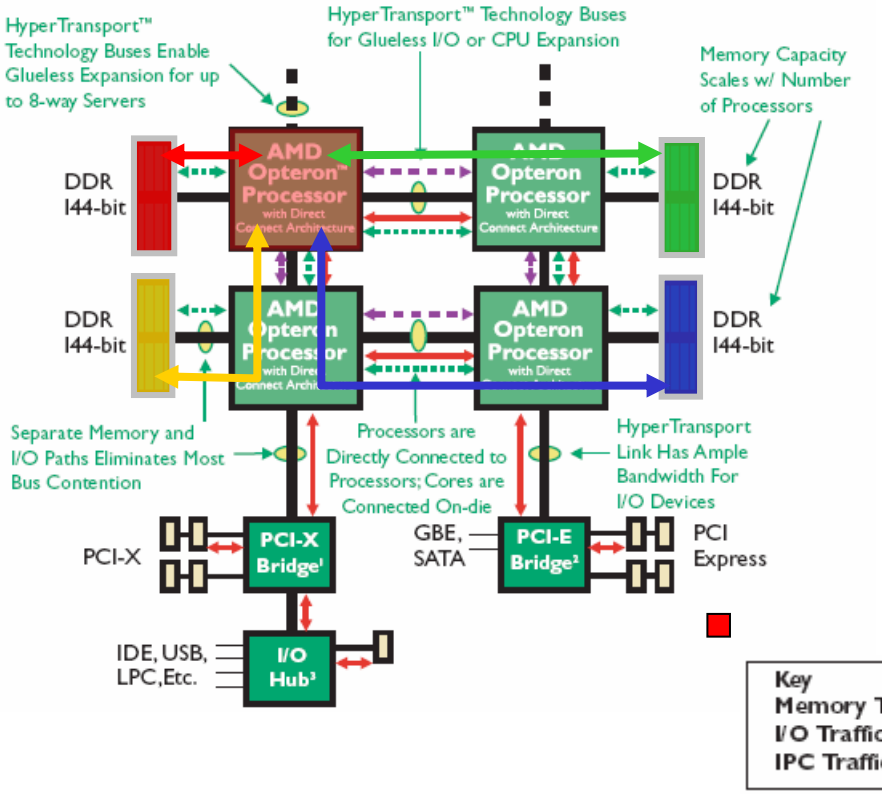
Note:
these transfers only show data movement between sub-systems and not the commands themselves. Additionally, data touching operations within a sub-system will require that data is moved from memory and to the CPU, e.g.:
- checksum calculation
- encryption
- data encoding
- forward error correction

Example: IBM POWER 4

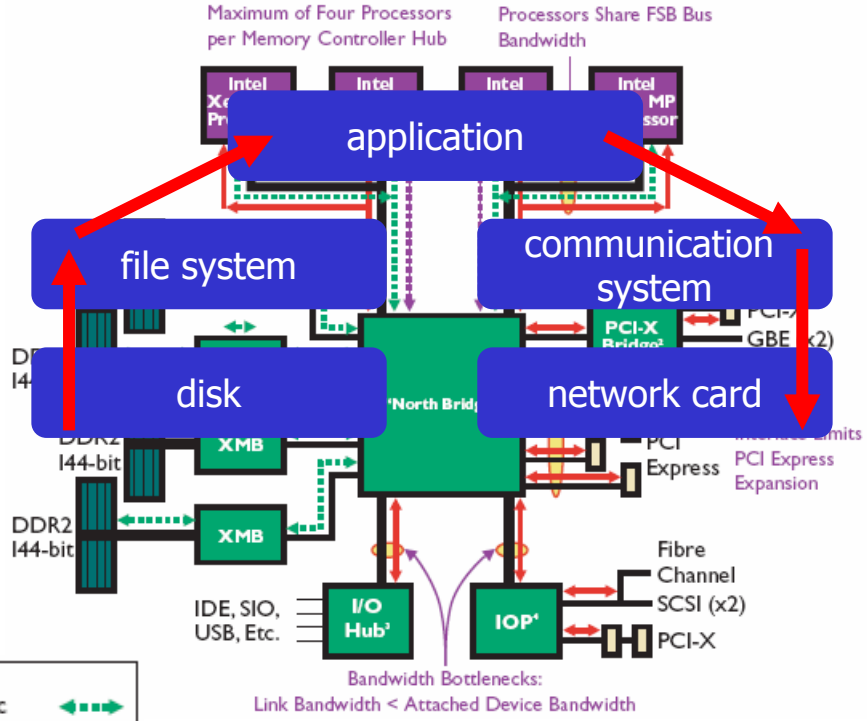


Example:
AMD Opteron & Intel Xeon MP 4P servers

AMD Opteron™ Processor-based 4P Server



Intel Xeon MP Processor-based 4P Server



☞ Know your hardware –
different configuration may have different bottlenecks

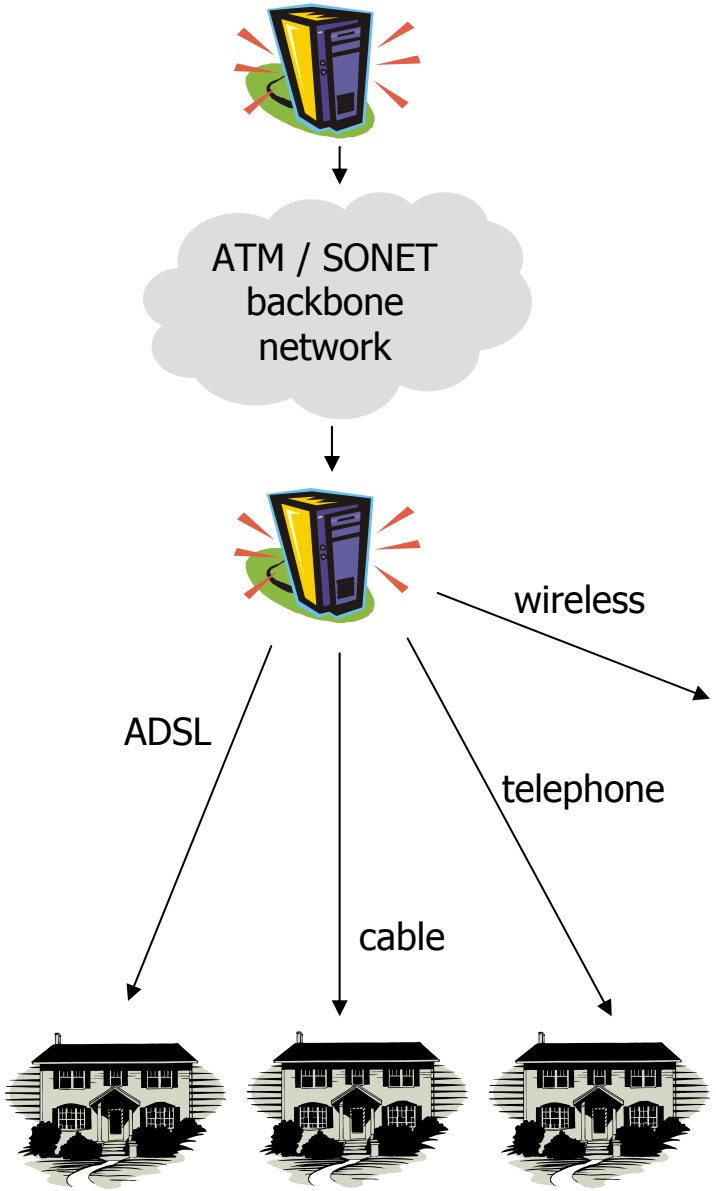
Server Internals Challenges

- *Data retrieval from disk and push to network for many users*
- Important resources:
  - memory
  - busses
  - CPU
  - storage (disk) system:
  - communication system
  - …
- Stable operations:
  - redundant HW
  - multiple nodes
- Much can be done to optimize resource utilization, e.g., scheduling, placement, caching/prefetching, admission control, merging concurrent users, …
- We will in later lectures look at several of these


Network Approaches

Network Architecture Approaches

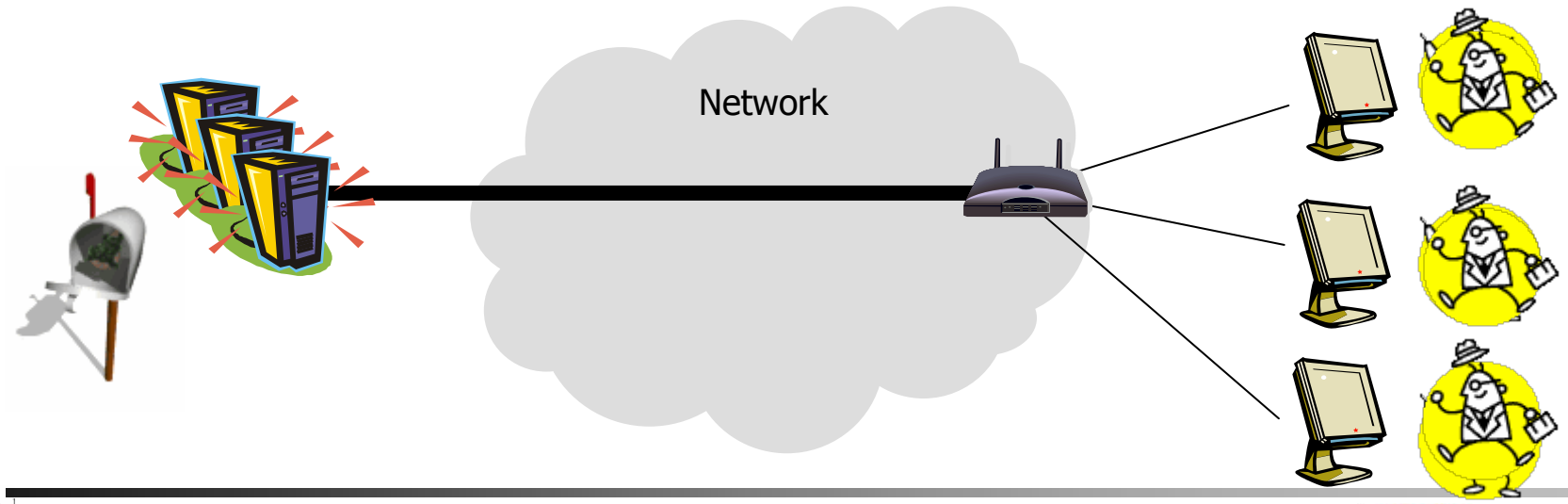
- WAN backbones
  - SONET
  - ATM
- Local distribution network
  - ADSL (asymmetric digital subscriber line)
  - FTTC (fiber to the curb)
  - FTTH (fiber to the home)
  - HFC (hybrid fiber coax) (=cable modem)
  - E-PON (Ethernet passive optical network)
  - …
- Has to be aware of different capabilities
  - loss rate
  - bandwidth
  - possible asymmetric links
  - distance
  - load
  - ….



Network Challenges

- Goals:
  - network-based distribution of content to consumers
  - bring control to users

- Distribution in LANs is more or less solved:
  OVERPROVISIONING works
  - established in studio business
  - established in small area (hotel/hospital/plane/…) businesses



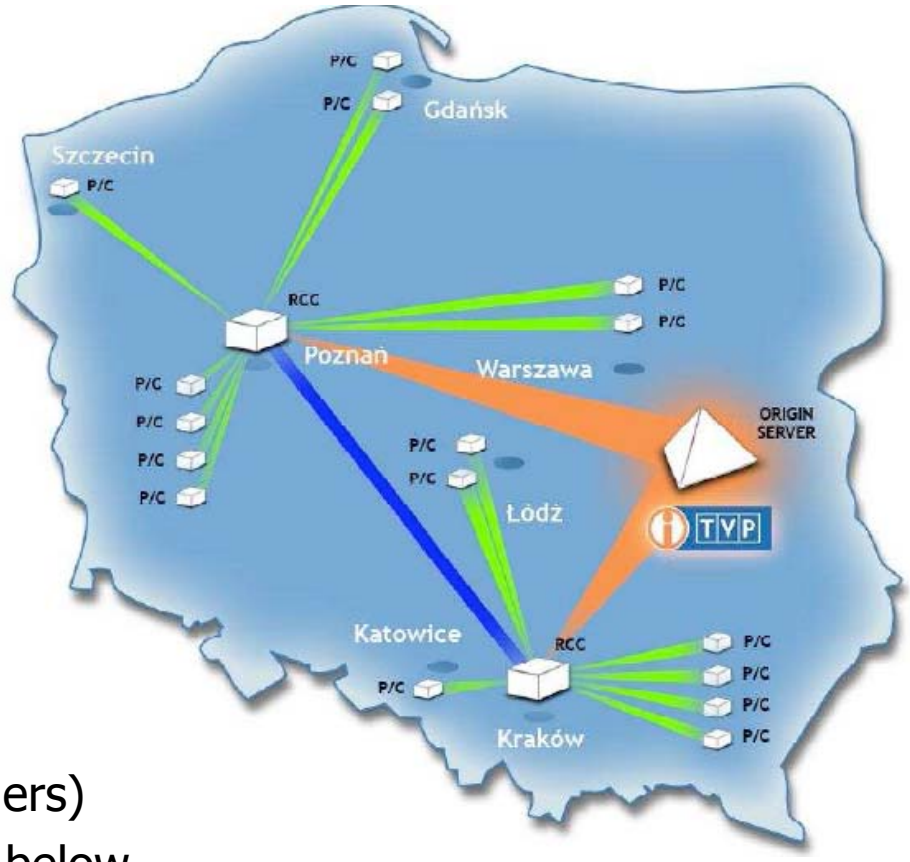
Network Challenges

- WANs are not so easy
  - overprovisioning of resources will NOT work
  - no central control of delivery system
  - too much data
  - too many users
  - too many different systems
- Different applications and data types have different requirements and behavior
- What kind of services offered is somewhat dependent on the used protocols
- We will in later lectures look at different protocols and mechanisms

Case Studies: Application Characteristics

iTVP

- Country-wide **IP TV and VoD** in Poland

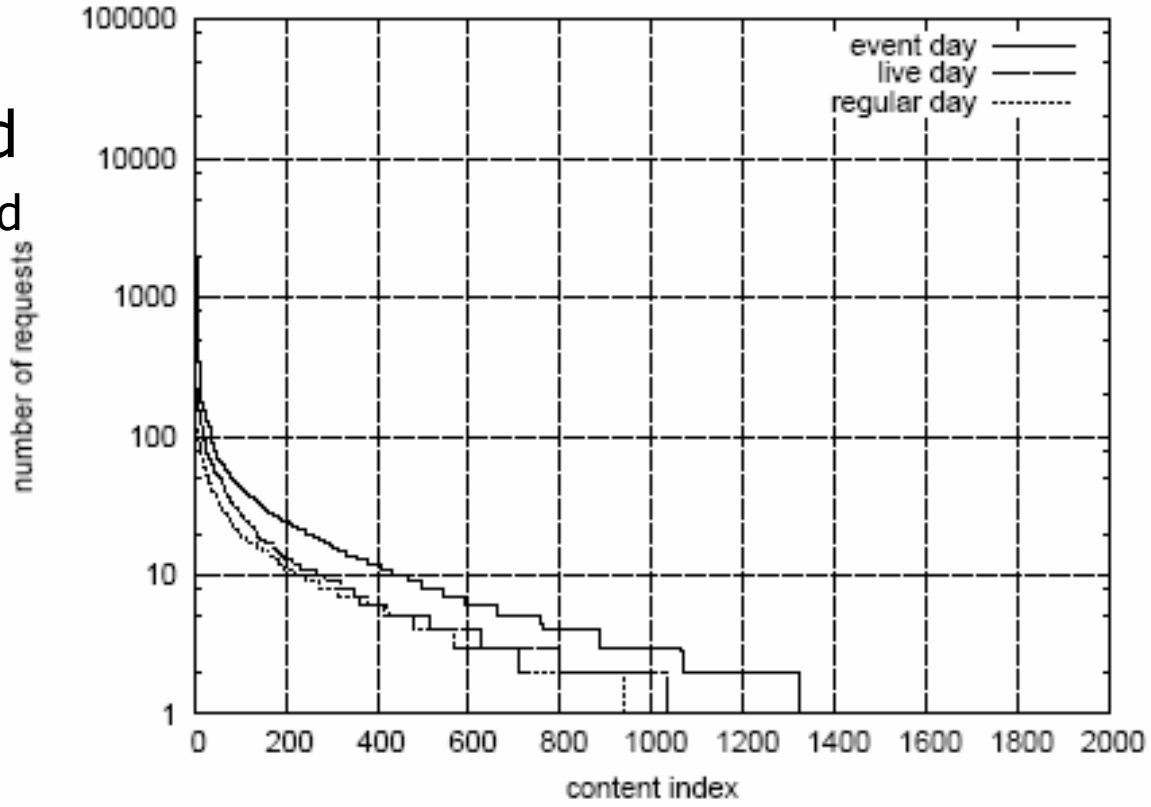


  - live & VoD
  - hierarchical structure with caching
    - **origin server**
    - **regional content centers (RCC)** (receiving data from content providers)
    - a number of **proxy caches (P/C)** below (handling requests from users)
  - different quality levels of the video – up to 700 Kbps
  - observations over several months

iTVP: Popularity Distribution

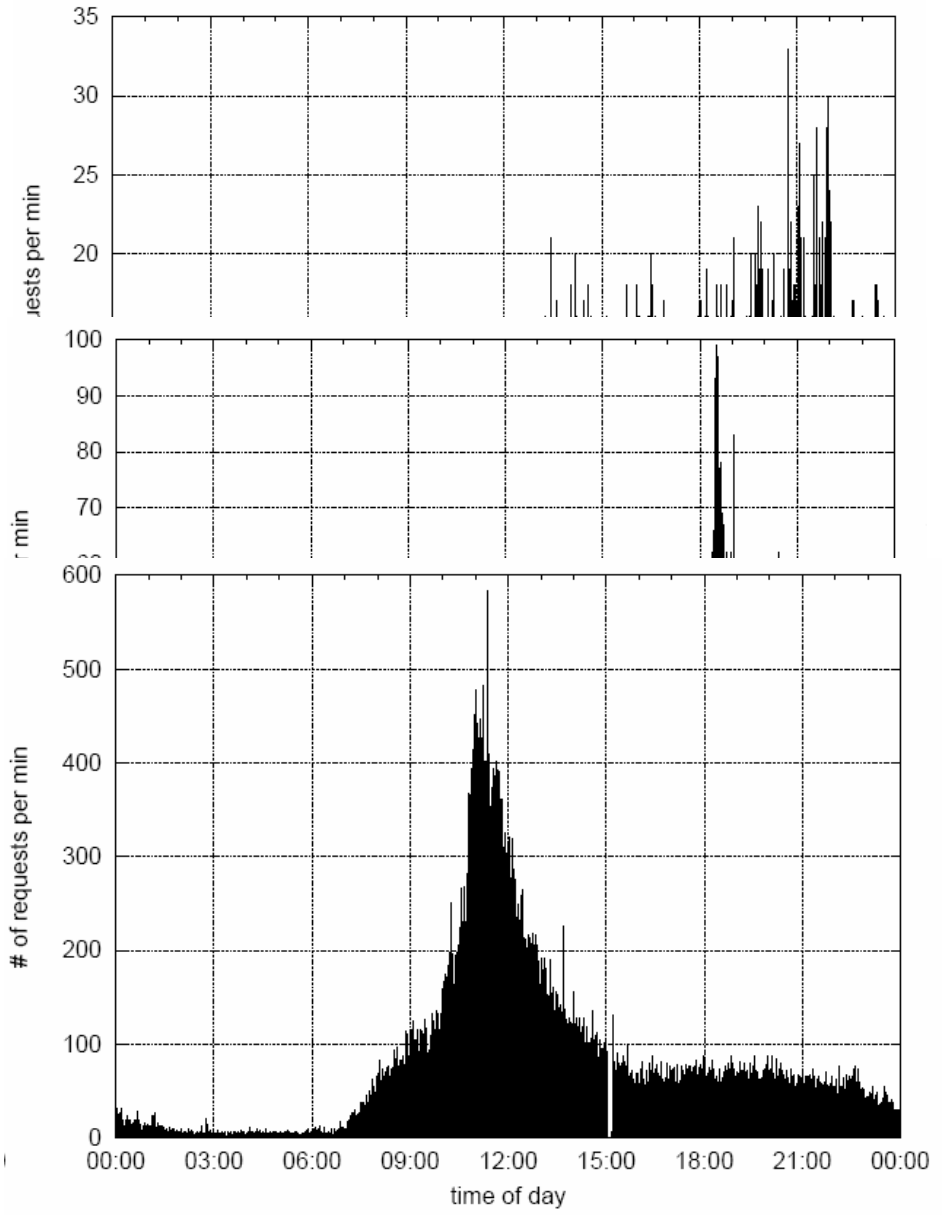
- Popularity of media objects according to Zipf, i.e., most accesses are for a few number of objects
- The object popularity decreases as time goes

- During a 24-hour period
  - up to 1500 objects accessed
  - ~1200 accesses for the most popular



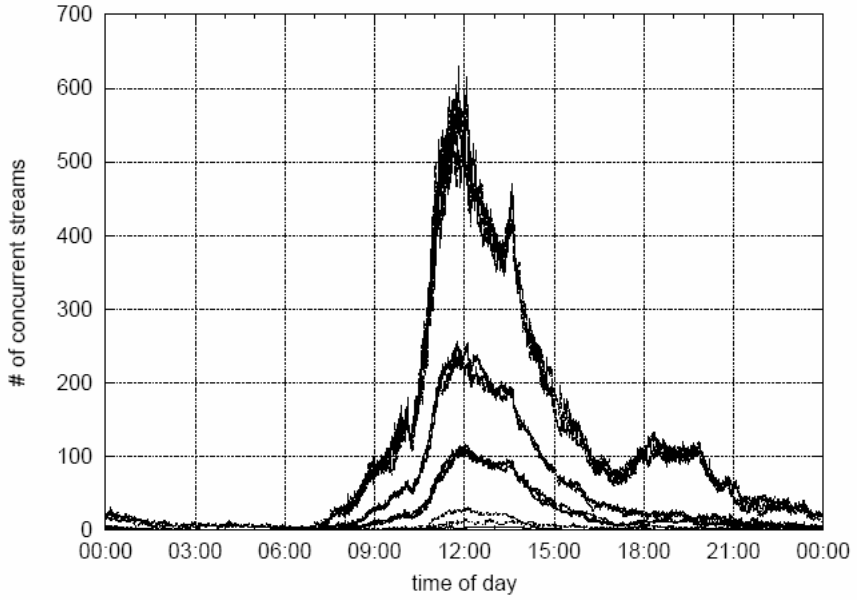
iTVP: Access Patterns

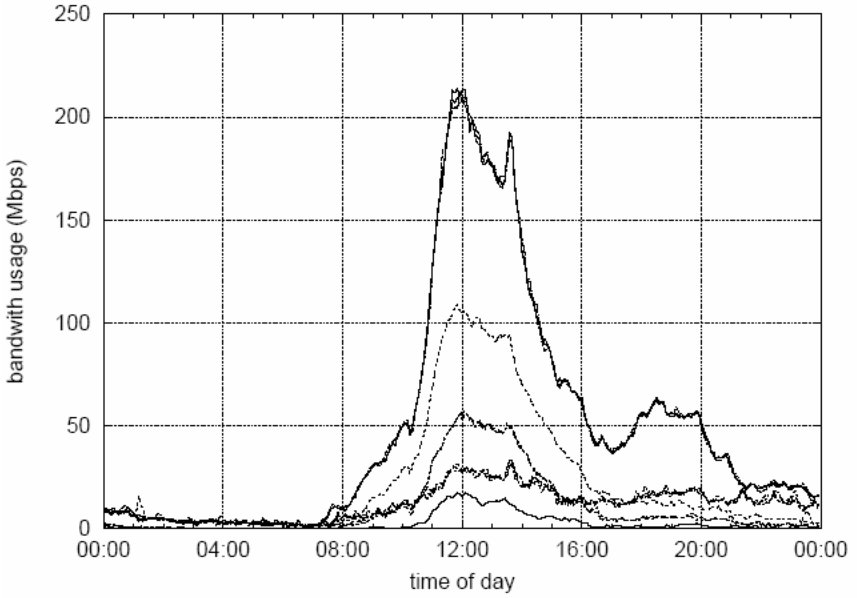
- Regular days
  - low in the morning, high in the evening
  - typical 30 requests per minute
  - the most popular items had an **average** of 300 accesses per day
  - an average total of 11.500 accesses per day
- Live transmissions
  - higher request rate
  - an average total of 18.500 accesses per day
  - 20% accesses to the most popular content
- Event transmissions
  - several hundreds accesses per minute during event transmission
  - an average total of 100.000+ accesses per day
  - 50% accesses to the most popular content



iTVP: Concurrency and Bandwidth

- The number of concurrent users vary, e.g., for a single proxy cache
  - event: up to 600
  - regular: usually less than 20
- Transfers between nodes are on the order of several Mbps, e.g.,
  - event:
    - single proxy: up to 200 Mbps
    - whole system: up to 1.8 Gbps
  - regular:
    - single proxy: around 60 Mbps
    - whole system: up to 400 Mbps

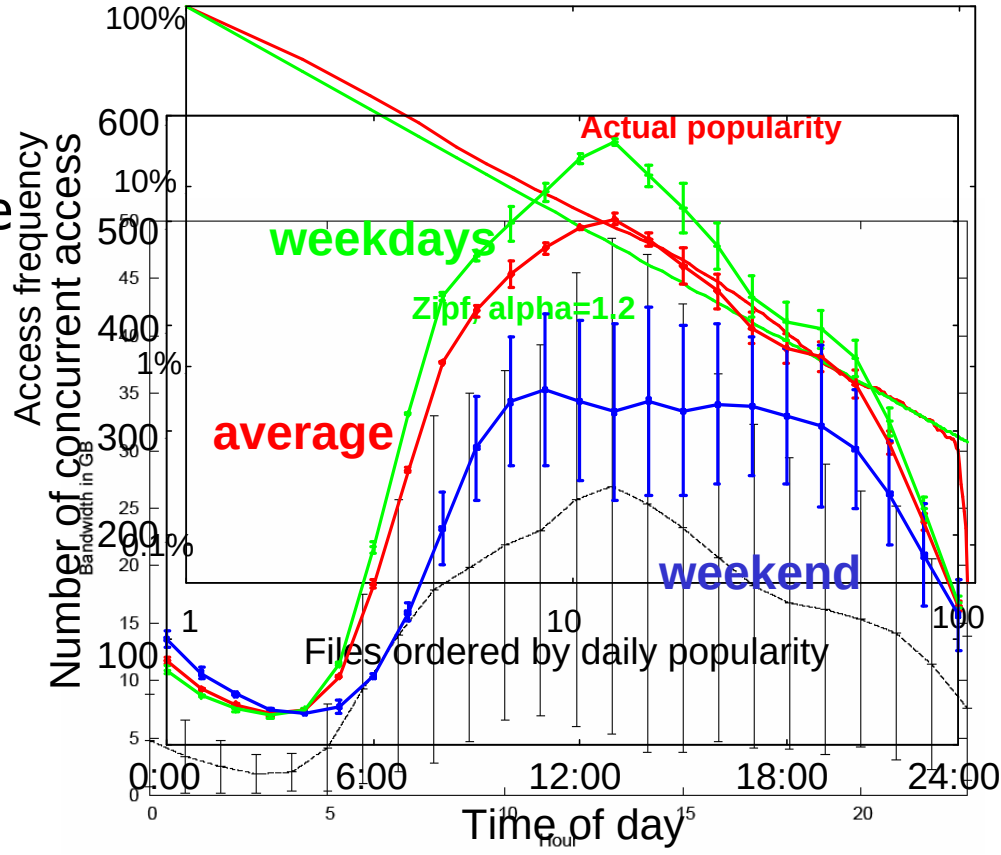




Verdens Gang (VG) TV: News-on-Demand

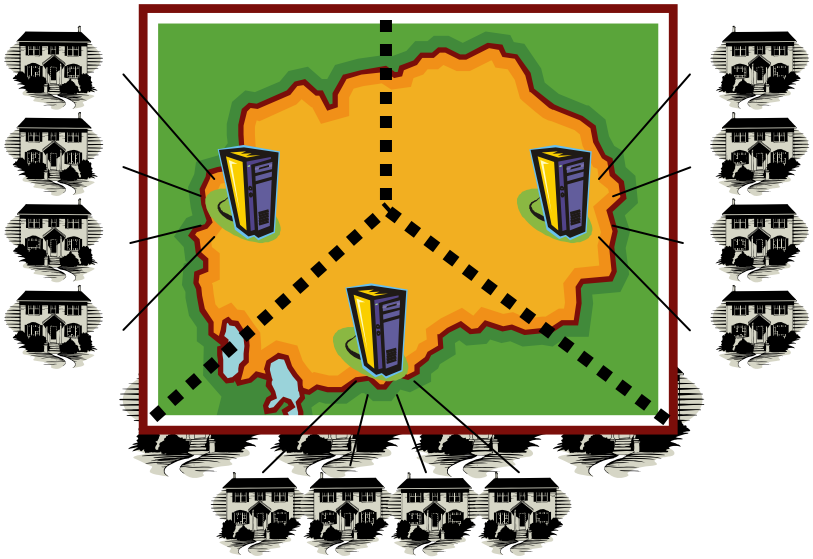
- Client-server
- Microsoft Media Server protocol (over UDP, TCP or HTTP)

- From a 2-year log of client accesses for news videos Johnsen et. al. found
  - Approximated Zipf distributed popularity, but more articles are popular
  - Access pattern dependent on time of day and day of week
  - Large bandwidth requirements, i.e., several GBs per hour



Funcom's Anarchy Online

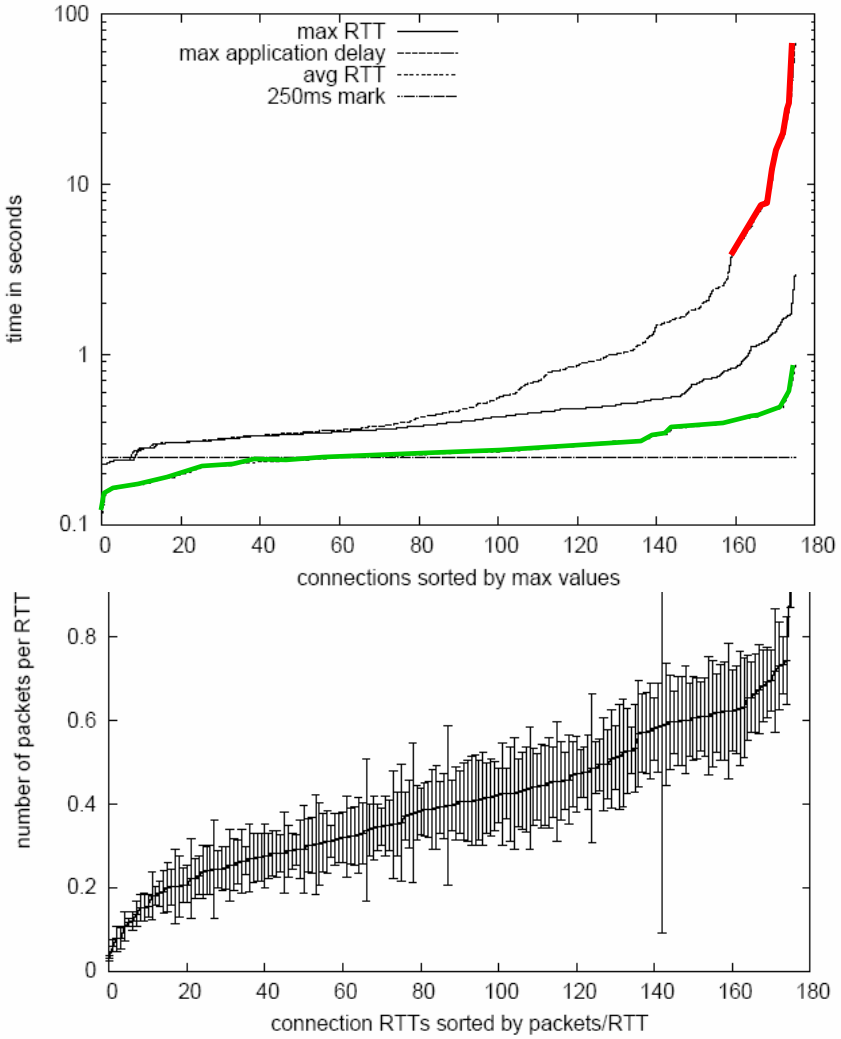
- World-wide **massive multiplayer online roleplaying game**
  - client-server
    - point-to-point TCP connections



    - virtual world divided into many regions
    - one or more regions are managed by one machine

Funcom's Anarchy Online

- For a given region in a one hour trace we found
  - ~175 players (from three continents??)
  - average **layer 3** RTT somewhat above 250 ms
    ↳ OK
  - a worst-case **application** delay of 67 s (!)
    ↳ loss results in a players nightmare
  - less than 4 packets per second
  - small packets: ~120 B
    ↳ thins streams

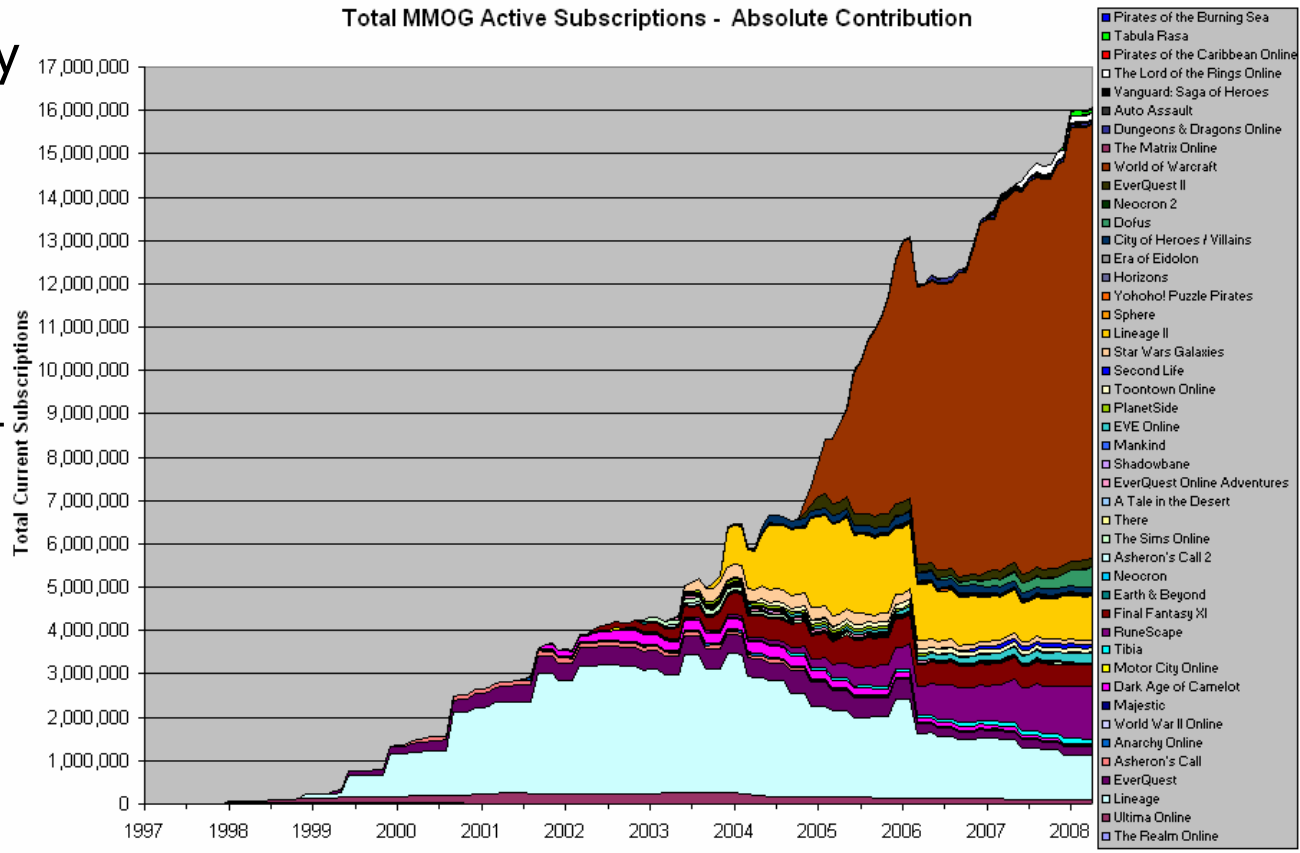


Application Characteristics

- Movie-on-Demand and live video streaming
  - Access pattern according to Zipf
  - high rates, many and large packets
  - many concurrent users (Blockbuster online – 2.2 million users)
  - extreme peeks
  - timely, continuous delivery
- News-on-Demand streaming
  - daily periodic access pattern – close to Zipf
  - similar to other video streaming
- ...

Application Characteristics

- Games
  - low rates, few and small packets, especially MMOGs:
    - < 10 packets per second
    - ~100 bytes payload per packet
  - interactive
  - low latency delivery (100 – 1000 ms)
  - many concurrent users
    - MMOGs in total – > 16 million
    - WoW – > 9 million



Picture Today!

Summary

- Assumptions:
  - overprovisioning of resources will NOT work
- Systems:
  - need for interoperability – not from a single source
  - need for co-operative distribution systems
- Huge amounts of data:
  - billions of web-pages (11.5 billion indexable web pages January 2005)
  - billions of downloadable articles
  - thousands of movies (estimated 65000 in 1995!! H/Bollywood = 500/1000 per year)
  - data from TV-series, sport clips, news, live events, …
  - games and virtual worlds
  - music
  - home made media data shared on the Internet
  - …

Summary

- Applications and challenges in a distributed system
  - different requirements
  - different architectures
  - different devices
  - different capabilities
  - …
  - and it keeps growing!!!!

- Performance issues are important…!!!!