Advanced Computer Networks

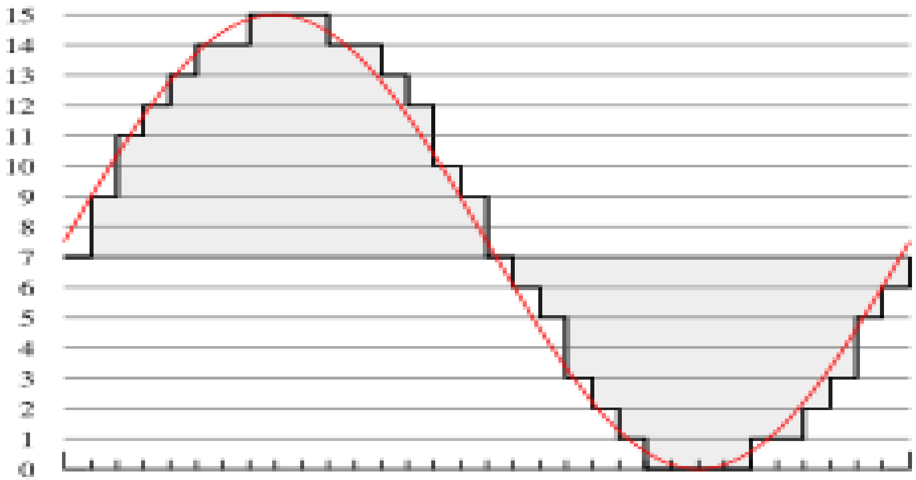
Multimedia Networking

Outline

- Digital audio and video
  - Sampling, quantizing, and compressing
- Multimedia applications
  - Streaming audio and video for playback
  - Live, interactive audio and video
- Multimedia transfers over a best-effort network
  - Tolerating packet loss, delay, and jitter
  - Forward error correction and playout buffers
- Improving the service the network offers
  - Marking, policing, scheduling, and admission control

Digital Audio

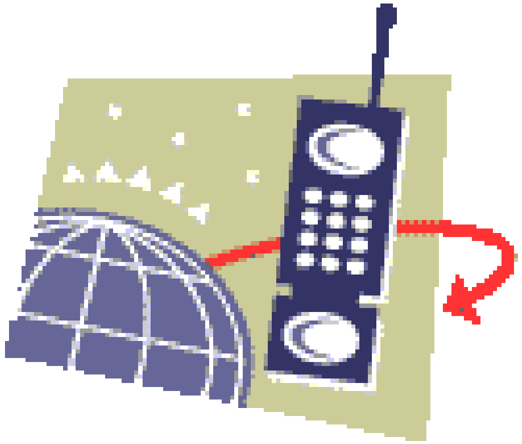
- Sampling the analog signal
  - Sample at some fixed rate
  - Each sample is an arbitrary real number
- Quantizing each sample
  - Round each sample to one of a finite number of values
  - Represent each sample in a fixed number of bits



4 bit representation (values 0-15)

Audio Examples

- Speech
  - Sampling rate: 8000 samples/second
  - Sample size: 8 bits per sample
  - Rate: 64 kbps



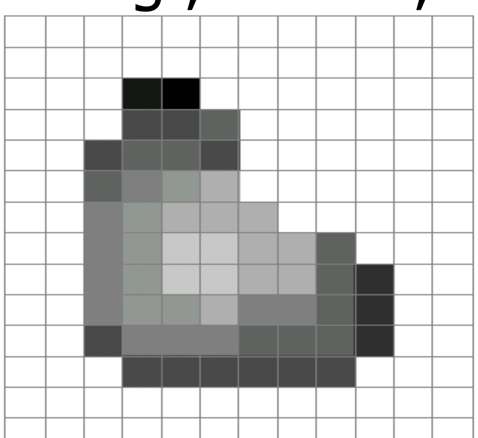
- Compact Disc (CD)
  - Sampling rate: 44,100 samples/second
  - Sample size: 16 bits per sample
  - Rate: 705.6 kbps for mono,
    1.411 Mbps for stereo

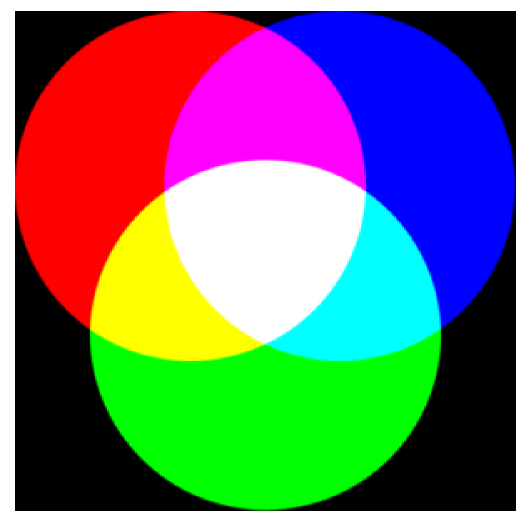
Audio Compression

- Audio data requires too much bandwidth
  - Speech: 64 kbps is too high for a dial-up modem user
  - Stereo music: 1.411 Mbps requires too much bandwidth
- Compression to reduce the size
  - Remove redundancy
  - Remove details that human tend not to perceive
- Example audio formats
  - Speech: GSM (13 kbps), G.729 (8 kbps), and G.723.3 (6.4 and 5.3 kbps)
  - Stereo music: MPEG 1 layer 3 (MP3) at 96 kbps, 128 kbps, and 160 kbps

Digital Video

- Sampling the analog signal
  - Sample at some fixed rate (e.g., 24 or 30 times per sec)
  - Each sample is an image
- Quantizing each sample
  - Representing an image as an array of picture elements
  - Each pixel is a mixture of colors (red, green, and blue)
  - E.g., 24 bits, with 8 bits per color

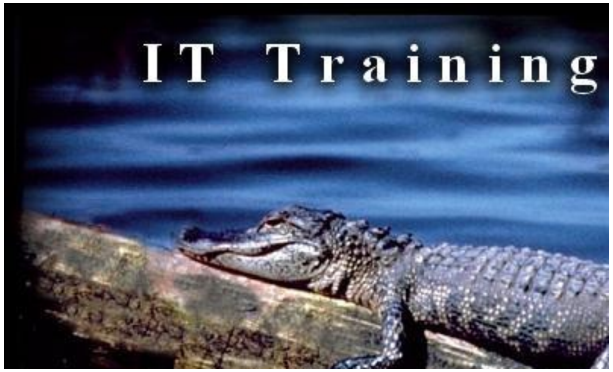





The
2272 x 1704
hand
The
320 x 240
hand

Video Compression: Within an Image

- Image compression
  - Exploit spatial redundancy (e.g., regions of same color)
  - Exploit aspects humans tend not to notice
- Common image compression formats
  - Joint Pictures Expert Group (JPEG)
  - Graphical Interchange Format (GIF)



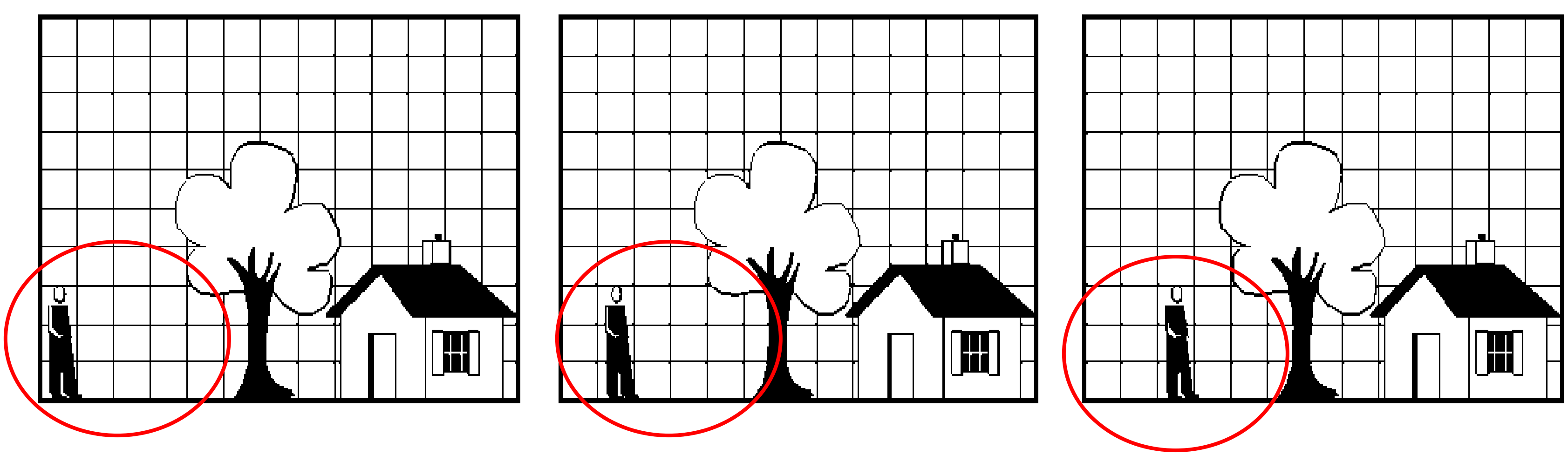
Uncompressed: 167 KB

Good quality: 46 KB

Poor quality: 9 KB

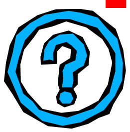
Video Compression: Across Images

- Compression across images
  - Exploit temporal redundancy across images
- Common video compression formats
  - MPEG 1: CD-ROM quality video (1.5 Mbps)
  - MPEG 2: high-quality DVD video (3-6 Mbps)
  - Proprietary protocols like QuickTime and RealNetworks



Consideration of Networked Multimedia

- Characteristics of multimedia information
  - Large data volume



Exercise: What is the size of a video clip of 60 minutes if the frame size is 640*480, the pixel depth is 24, and the frame rate is 24 fps?

  - Real-time property
    - Continuous display
    - Delay requirement of multimedia applications
- Properties of current Internet
  - Limitation of bandwidth
  - Best effort network, cannot guarantee quality of multimedia applications
  - Heterogeneity
    - Different user requirements
    - Different user network conditions

Consideration of Networked Multimedia

- Requirements of multimedia applications on the network
  - Delay requirement
  - Quality requirement
    - Satisfactory quality of media presentation
    - Synchronization requirement
    - Continuous requirement (no jerky video/audio)
    - Can tolerant some degree of information loss
- Challenges of multimedia networking
  - Media size vs. bandwidth limit of the network
  - User requirement of multimedia application vs. best-effort network
  - How to meet different requirements of different users?

11

Multimedia Networking Systems

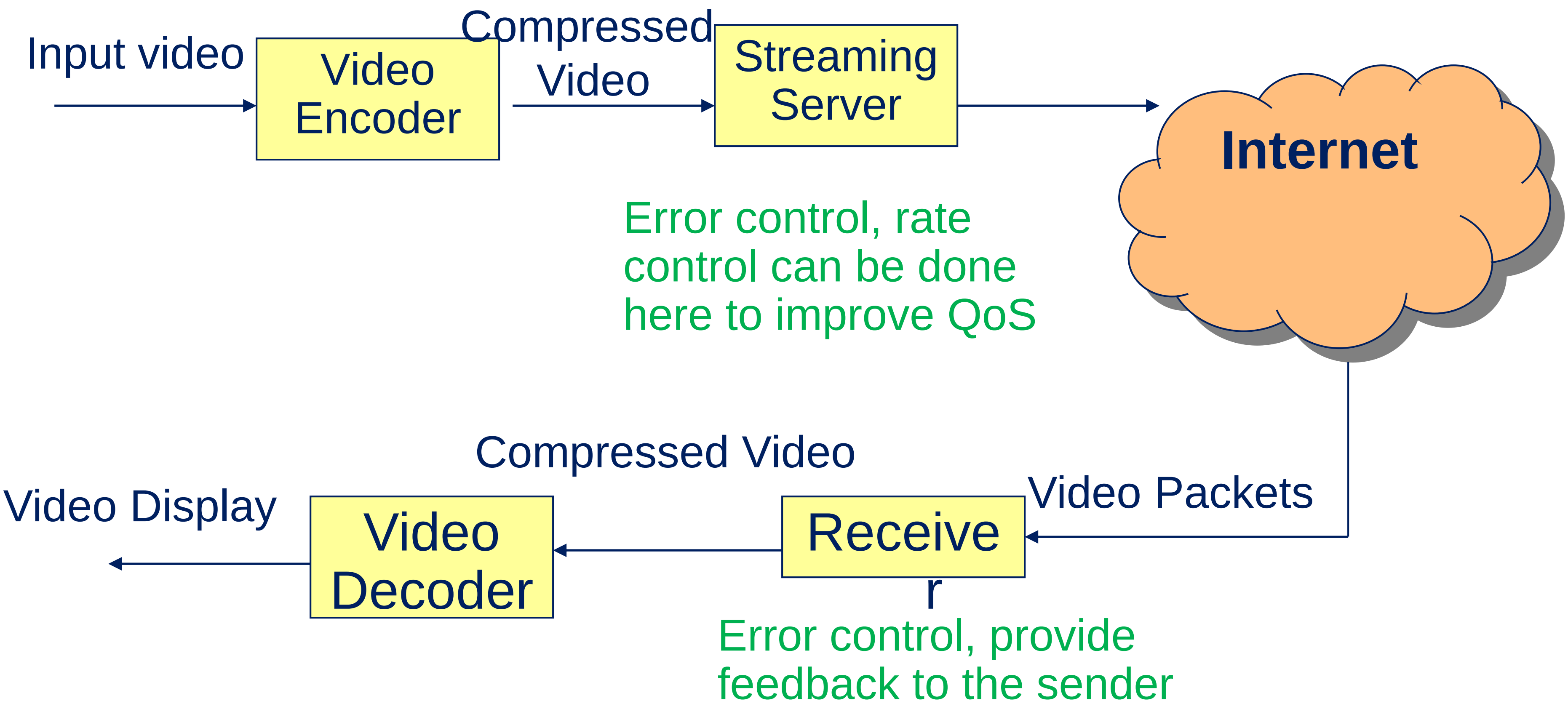
- Live media transmission system
  - Capture, compress, and transmit the media on the fly
- Send stored media across the network
  - Media is pre-compressed and stored at the server. This system delivers the stored media to one or multiple receivers.
- Differences between the two systems
  - For live media delivery:
    - Real-time media capture, need hardware support
    - Real-time compression– speed is important
    - Compression procedure can be adjusted based on network conditions
  - For stored media delivery
    - Offline compression – better compression result is important
    - Compression can not be adjusted during transmission

12

Transferring Audio and Video Data

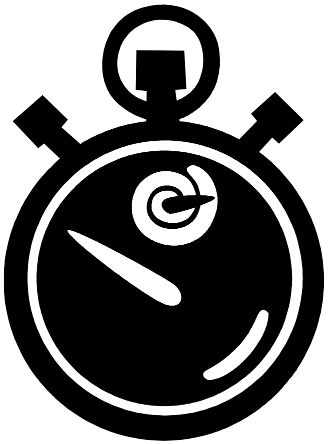
- Simplest case: just like any other file
  - Audio and video data stored in a file
  - File downloaded using conventional protocol
  - Playback does not overlap with data transfer
- A variety of more interesting scenarios
  - Live vs. pre-recorded content
  - Interactive vs. non-interactive
  - Single receiver vs. multiple receivers
- Examples
  - Streaming audio and video data from a server
  - Interactive audio in a phone call

Generic Media Streaming System



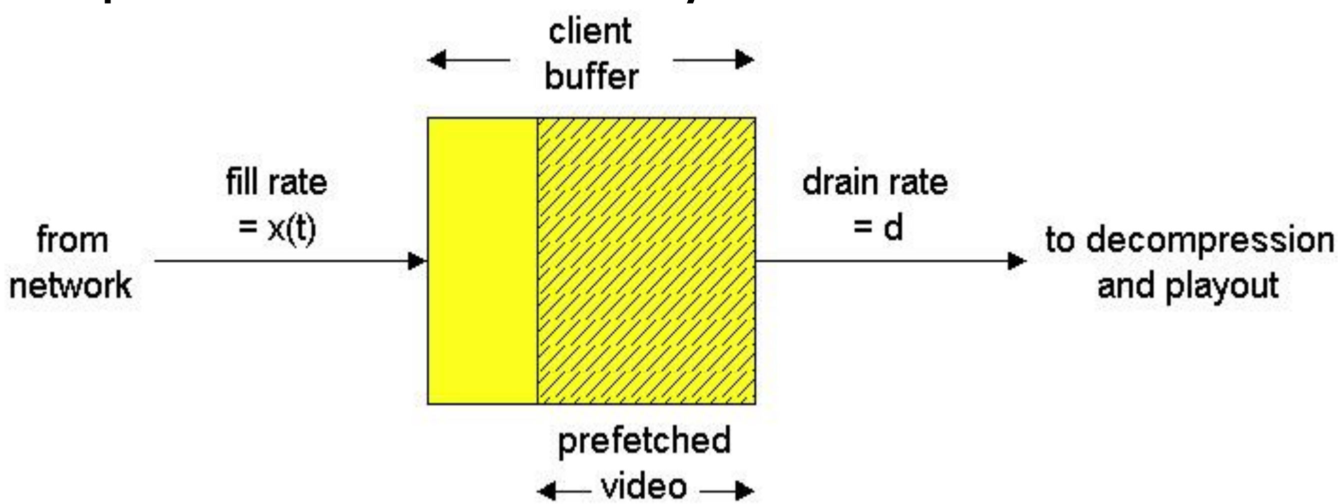
Streaming Stored Audio and Video

- Client-server system
  - Server stores the audio and video files
  - Clients request files, play them as they download, and perform VCR-like functions (e.g., rewind and pause)
- Playing data at the right time
  - Server divides the data into segments
  - … and labels each segment with timestamp or frame id
  - … so the client knows when to play the data
- Avoiding starvation at the client
  - The data must arrive quickly enough
  - … otherwise the client cannot keep playing

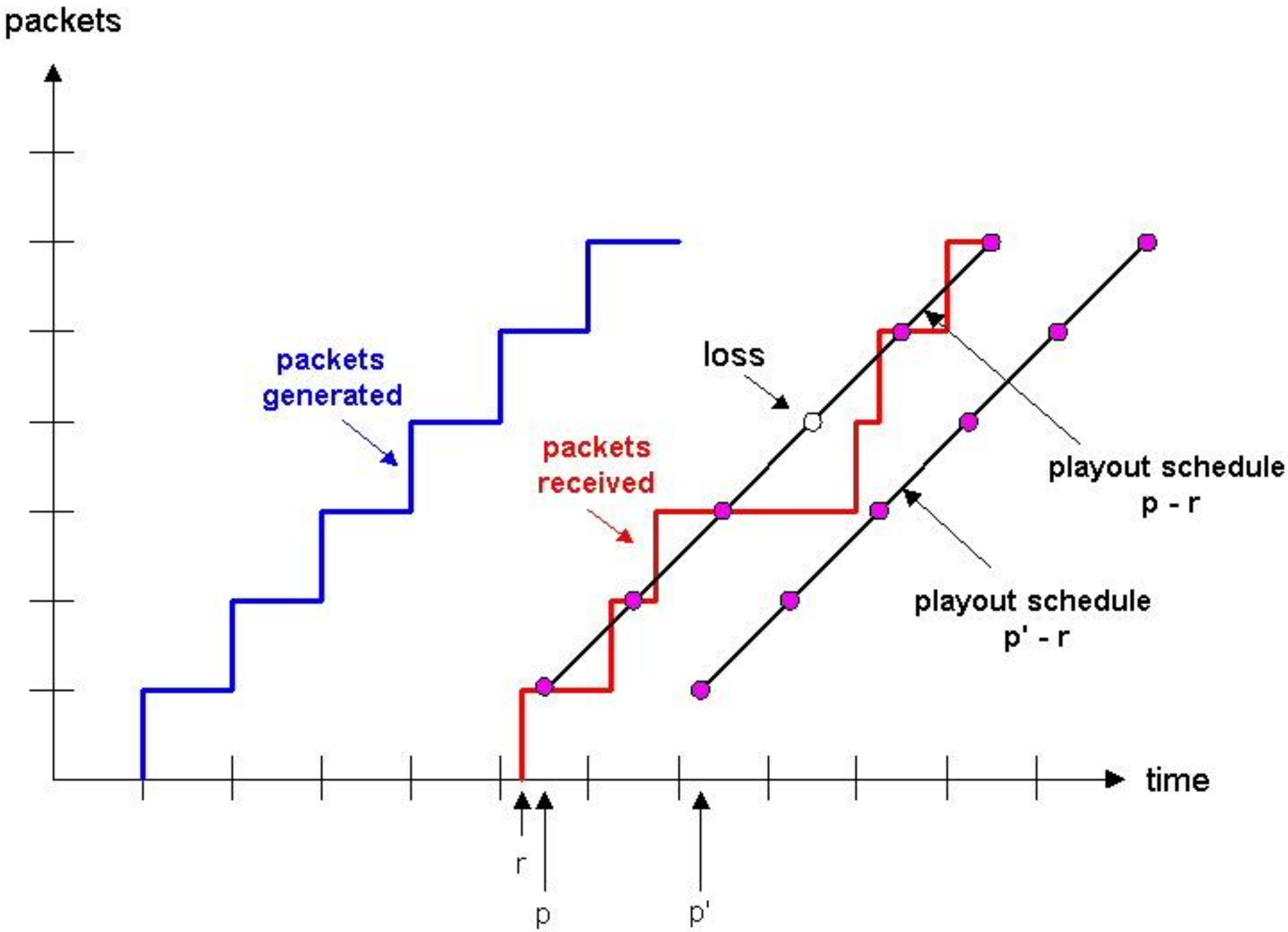


Playout Buffer

- Client buffer
  - Store the data as it arrives from the server
  - Play data for the user in a continuous fashion
- Playout delay
  - Client typically waits a few seconds to start playing
  - … to allow some data to build up in the buffer
  - … to help tolerate some delays down the road



Influence of Playout Delay

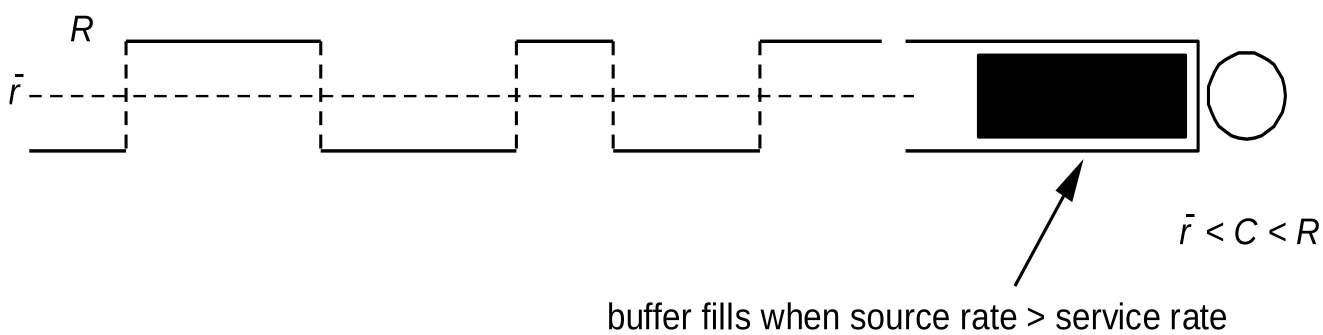


Requirements for Data Transport

- Delay
  - Some small delay at the beginning is acceptable
  - E.g., start-up delays of a few seconds are okay
- Jitter
  - Variability of packet delay within the same packet stream
  - Client cannot tolerate high variation if the buffer starves
- Loss
  - Small amount of missing data does not disrupt playback
  - Retransmitting a lost packet might take too long anyway

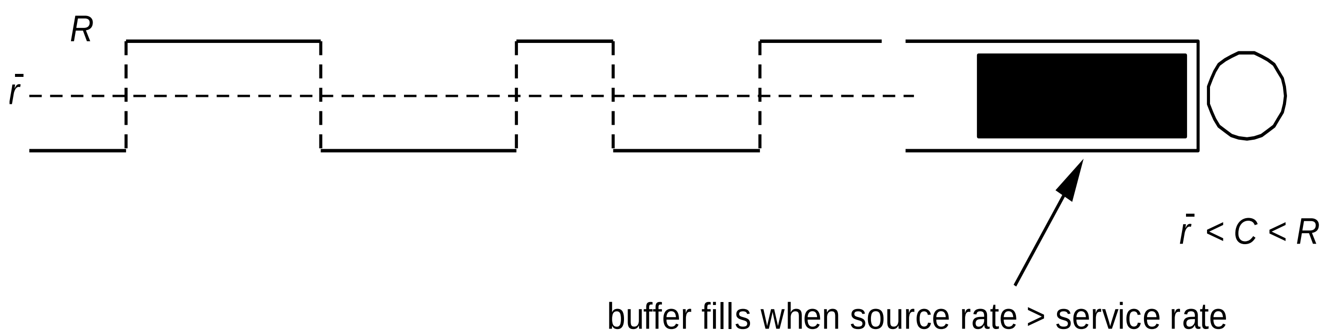
Voice Call Processing & Delays

- Sound sampled and encoded
  - Constant Bit Rate (CBR), e.g. 64Kbps (8000*8b)
- Voice activity detection (and compression?)
  - Discard (long, e.g. >0.2s) inactive periods
  - On-off Variable Bit Rate (VBR)
- Question: what link rate *C* is needed?



Voice Call Processing & Delays (cont')

- Sound sampled and CBR-encoded
- Voice activity detection, compression → VBR
- Shaping
  - Specifications of sent traffic
  - Allows efficient rates, buffers
- Packetization (and padding if necessary)

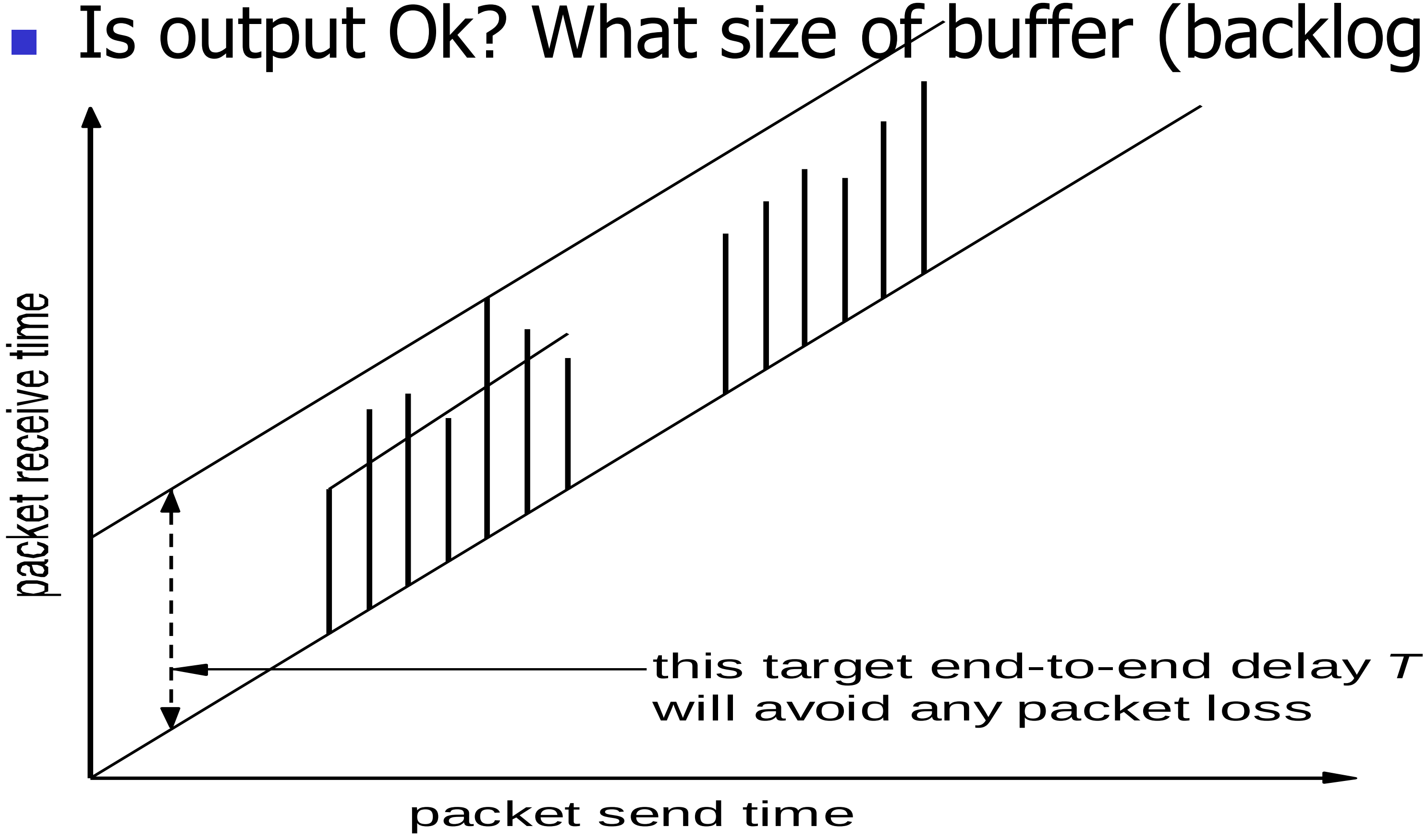


Voice Call Processing & Delays (cont')

- Sender:
  - Sampling and CBR-encoding
  - Voice Activity Detection, compression → VBR
  - Shaping
  - Packetization (and padding if necessary)
- Network (queues, transmission, propagation)
- Receiver
  - Immediate or delayed playout?

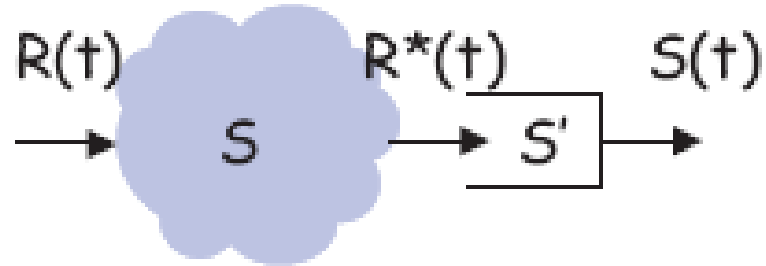
Playout Delay to `Hide` Jitter

- Lines: actual packet delays
- Is output Ok? What size of buffer (backlog)?



Playout Buffer Model

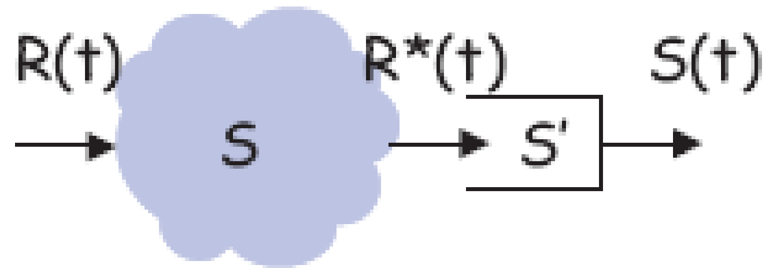
- Network system S:
  - CBR input: $R(t)=rt$
  - Delay variation (jitter) bounded by Δ
  - Namely for some $D>0$ holds: $R(t) \leqq R^*(t+D+\Delta)$ and $R^*(t+D-\Delta) \leqq R(t)$

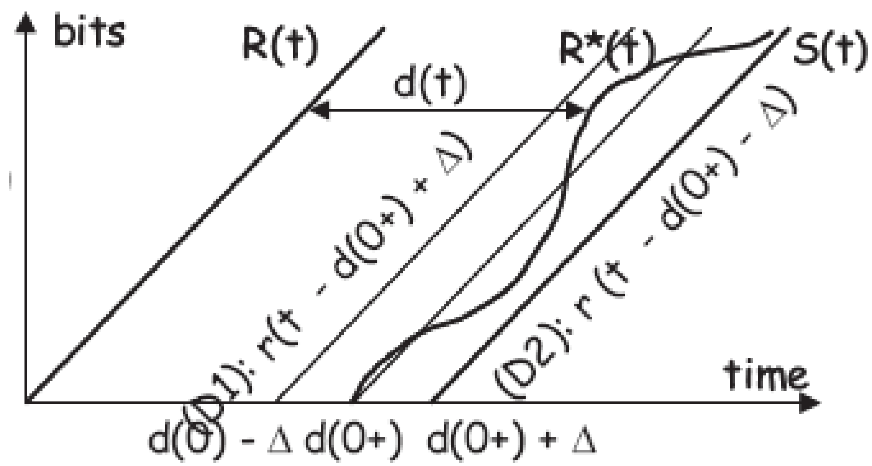


- Playout buffer S':
  - Receive first bit at $d(0)$, delay it by Δ
  - Then, serve at constant rate r, when not empty
  - ➔ if never empty after $d(0)$, then $S(t)=r(t-\Delta-d(0))^+$

Playout Buffer Analysis

- Claim: for every $t>d(0)$ holds $0<R^*(t)-S(t)<2\Delta r$
  - Playout buffer always non-empty, produces at rate r
  - Buffer size of $2\Delta r$ is sufficient (to avoid overflow)

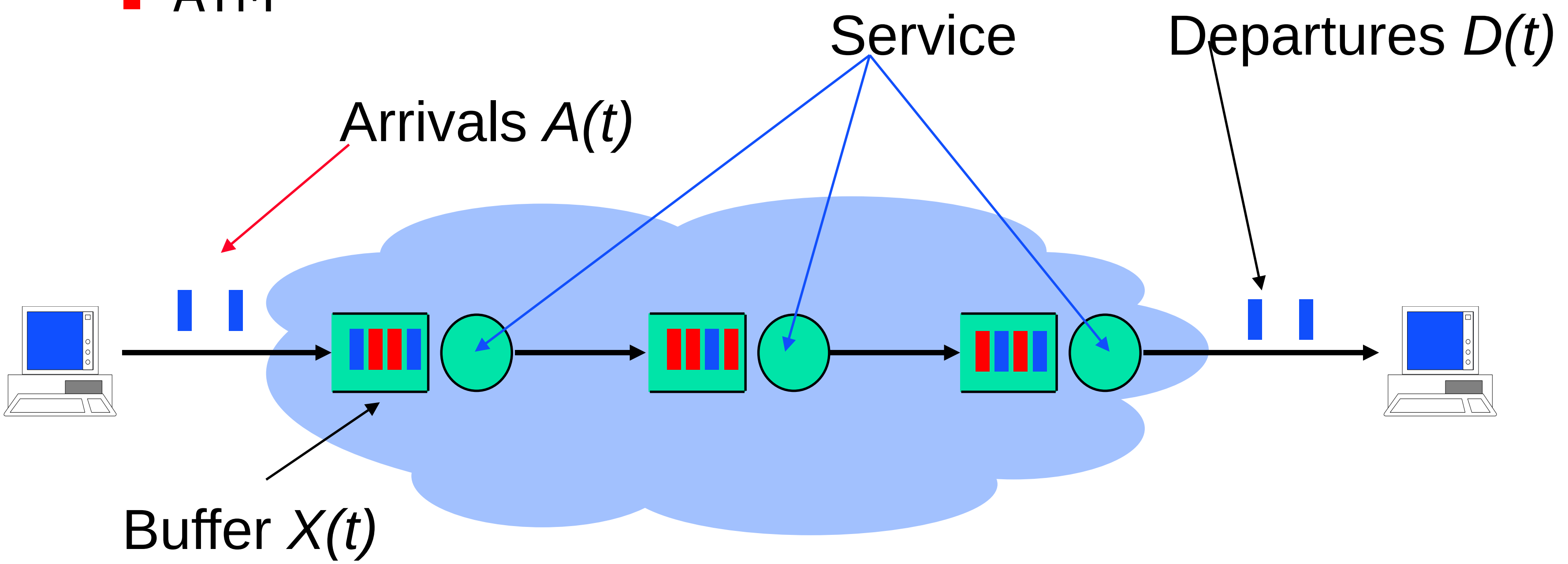




http://ica1www.epfl.ch/PS_files/netCalBookv4.pdf

Network Calculus

- Network Calculus: Worst-case analysis of Arrivals, Service, Buffering and Departures
- Used in design of QoS protocols
  - Internet: mainly IntServ, DiffServ
  - ATM



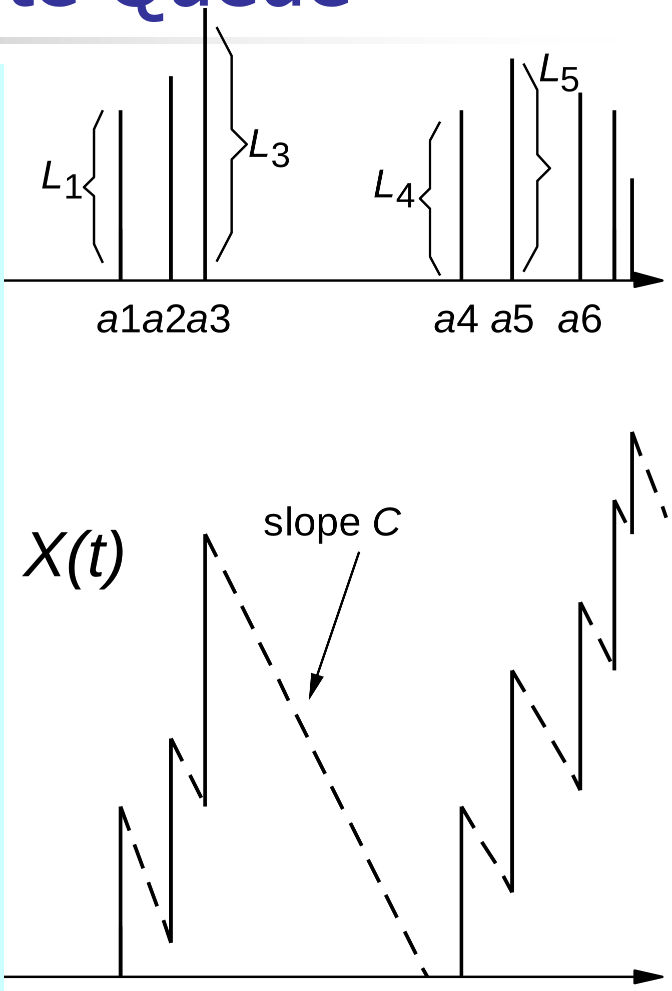
Model and notation (contd.)

- Assume output queueing at the switch or router
- Link rate C bits per second, packet length L bits per packet, packet transmission time L/C
- Assume store and forward switch and packet has arrived only after it is completely recieved.
- A link is multiplexed, shared by many other streams, multiple streams could 'arrive' simultaneously and queueing delay will occur.
- *Work conserving*, scheduler – if there are bits in the buffer then there is transmission.
- Infinite buffers – there is always space for an arriving packet

Also called non-idling scheduler

Buffer Analysis: Fixed Rate Queue

- Bits sent (removed) at C bits/second (unless empty)
- a_k: arrival of (complete) packet k
- $L_k \in [0,\infty)$: bit-length of packet k
- $A(t)$, $D(t)$: total arrivals (deliveries) till time t; $A(0_-)=0$
  - t_- : just before time t
- $X(t)$: queued bits at time t



Some simple analysis

For work-conserving/non-idling schedulers

- $X(t)$ depends on a_k and L_k and not on order of service
- Busy and Idle period durations are invariant with the scheduling discipline.
- The same packets depart during a busy period irrespective of the scheduling discipline – different scheduling disciplines order the departures differently.
- Time average of $X(t)$ is denoted by $\bar{X}(t_1, t_2)$ and defined as

$$\bar{X}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} X(u)du$$

is also invariant to scheduling discipline

Some simple analysis (contd.)

- Packets (or bits) in the buffer can belong to different sessions (streams). Let $X_s(t)$ denote the number of bits from stream s in the buffer.

$$X(t) = \sum_{\text{sessions } s} X_s(t)$$

$$\bar{X}(t_1, t_2) = \sum_{\text{sessions } s} \bar{X}_s(t_1, t_2)$$

- This is the *Conservation Law* $\Rightarrow$ Average number of bits in the buffer of a work conserving scheduler is invariant to the scheduling discipline.
- Buffer occupancy represents delay. Thus average delay (of what? bits/pkts) is invariant to scheduling discipline.

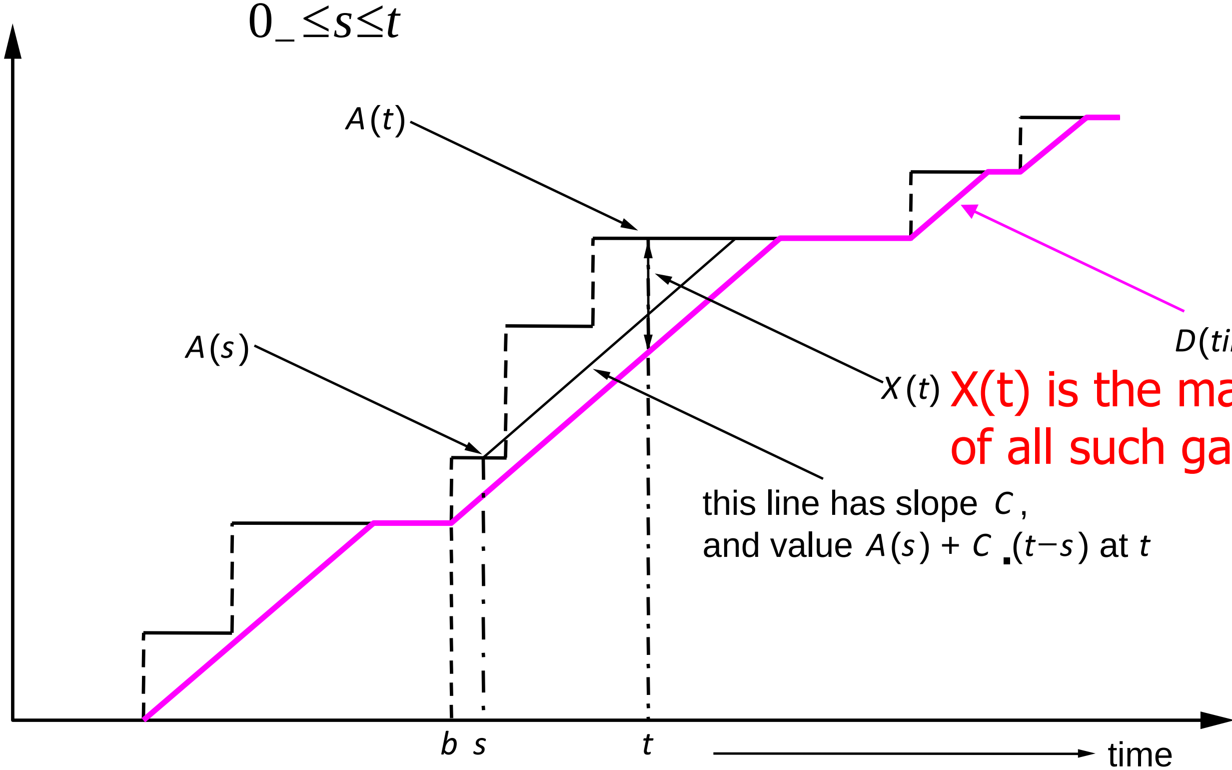
Finite buffer: a first look

- If an arriving packet can cause the buffer capacity to be exceeded then
  - The arriving packet may be dropped
  - A packet already in the buffer may be pushed out
- Lost packets cause distortion in stream traffic and delay because of the need to recover the loss, in elastic traffic
- Performance measure of interest: Loss rate
- Large buffers are not always good – in stream applications, a packet with large delay is useless at the destination
- With finite buffers, say B, $X(t)$ will never exceed B and packets that will cause it to exceed B are dropped.

Reich's Equation: Illustration

Assume that just before time 0 (i.e., at 0-), the buffer is empty: X(0-) = 0. For all t ≥ 0,

$$X(t) = \sup_{0_- \le s \le t} (A(t) - A(s) - C \cdot (t - s))$$

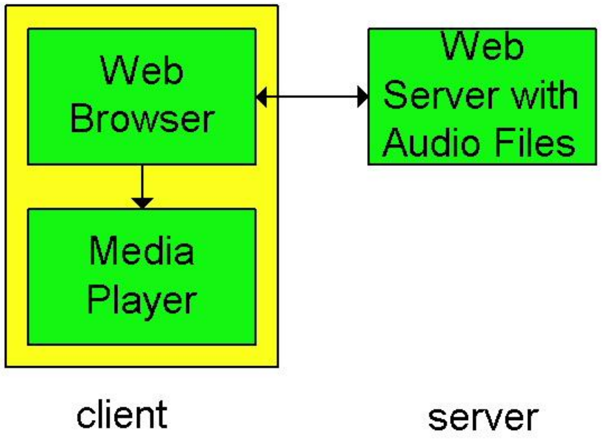


Let S be a subset of real numbers R. Then ∃u ∈ R, such that (1) for all s ∈ S, s ≤ u, and (2) for every b such that b upper bounds S, u ≤ b (u is the least upper bound of S). u is called a supremum of the set S.

Communication Networking – an analytical approach, Kumar, Manjunath and Kuri, Elsevier, 2004 (mainly ch. 4)

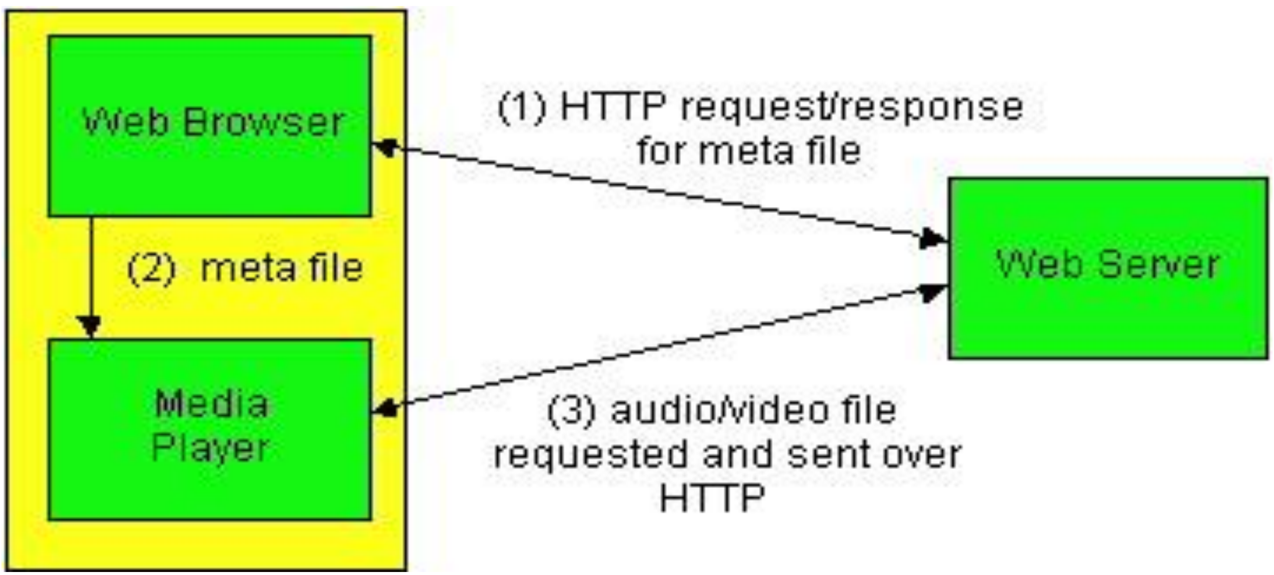
Streaming From Web Servers

- Data stored in a file
  - Audio: an audio file
  - Video: interleaving of audio and images in a single file
- HTTP request-response
  - TCP connection between client and server
  - Client HTTP request and server HTTP response
- Client invokes the media player
  - Content-type indicates the encoding
  - Browser launches the media player
  - Media player then renders the file



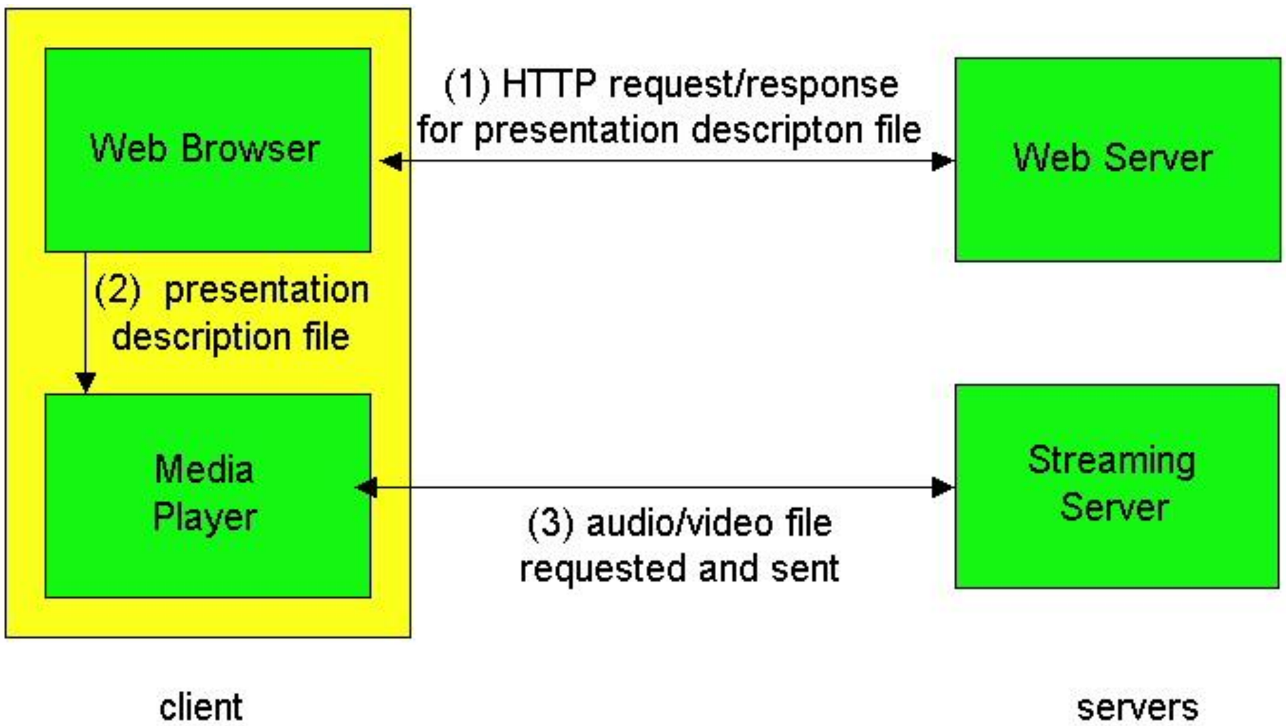
Initiating Streams from Web Servers

- Avoid passing all data through the Web browser
  - Web server returns a meta file describing the object
  - Browser launches media player and passes the meta file
  - The player sets up its own connection to the Web server



Using a Streaming Server

- Avoiding the use of HTTP (and perhaps TCP, too)
  - Web server returns a meta file describing the object
  - Player requests the data using a different protocol



TCP is Not a Good Fit

- Reliable delivery
  - Retransmission of lost packets
  - … even though it may not be useful
- Adapting the sending rate
  - Slowing down after a packet loss
  - … even though it may cause starvation at the client
- Protocol overhead
  - TCP header of 20 bytes in every packet
  - … which is large for sending audio samples
  - Sending ACKs for every other packet
  - … which may be more feedback than needed

Better Ways of Transporting Data

- User Datagram Protocol (UDP)
  - No automatic retransmission of lost packets
  - No automatic adaptation of sending rate
  - Smaller packet header

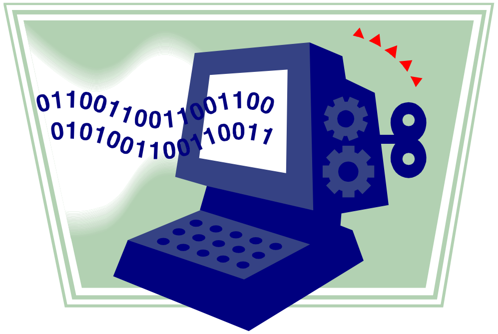
- UDP leaves many things up to the application
  - When to transmit the data
  - How to encapsulate the data
  - Whether to retransmit lost data
  - Whether to adapt the sending rate
  - … or adapt the quality of the audio/video encoding

Recovering From Packet Loss

- Loss is defined in a broader sense
  - Does a packet arrive in time for playback?
  - A packet that arrives late is as good as lost
  - Retransmission is not useful if the deadline has passed
- Selective retransmission
  - Sometimes retransmission is acceptable
  - E.g., if the client has not already started playing the data
  - Data can be retransmitted within the time constraint

Forward Error Correction (FEC)

- Forward error correction
  - Add redundant information to the packet stream
  - So the client can reconstruct data even after a loss
- Send redundant chunk after every n chunks
  - E.g., extra chunk is an XOR of the other n chunks
  - Receiver can recover from losing a single chunk
- Send low-quality version along with high quality
  - E.g., 13 kbps audio along with 64 kbps version
  - Receiver can play low quality version if the high-quality version is lost

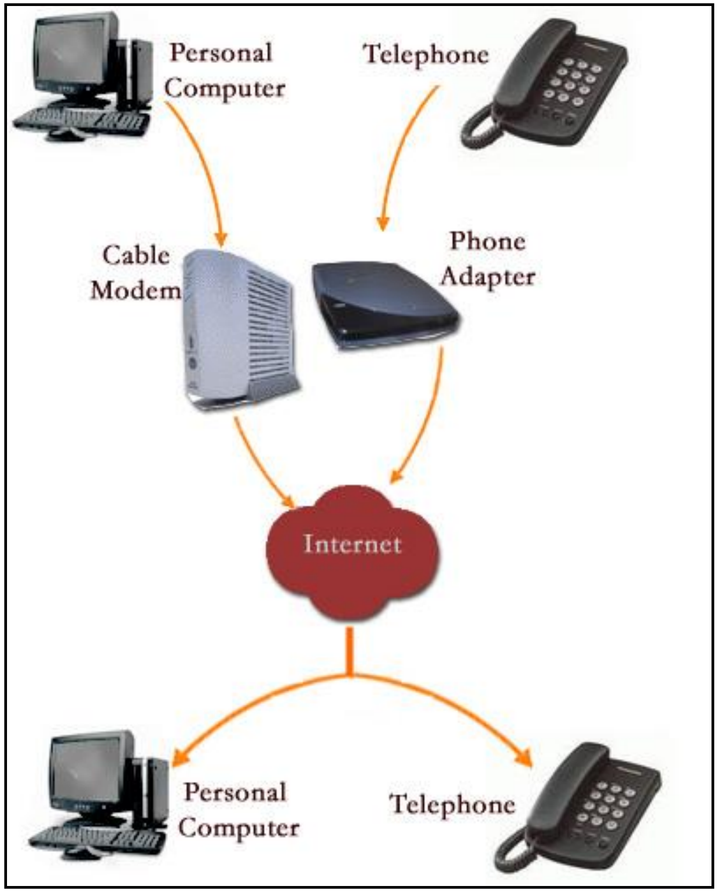


Interactive Audio and Video

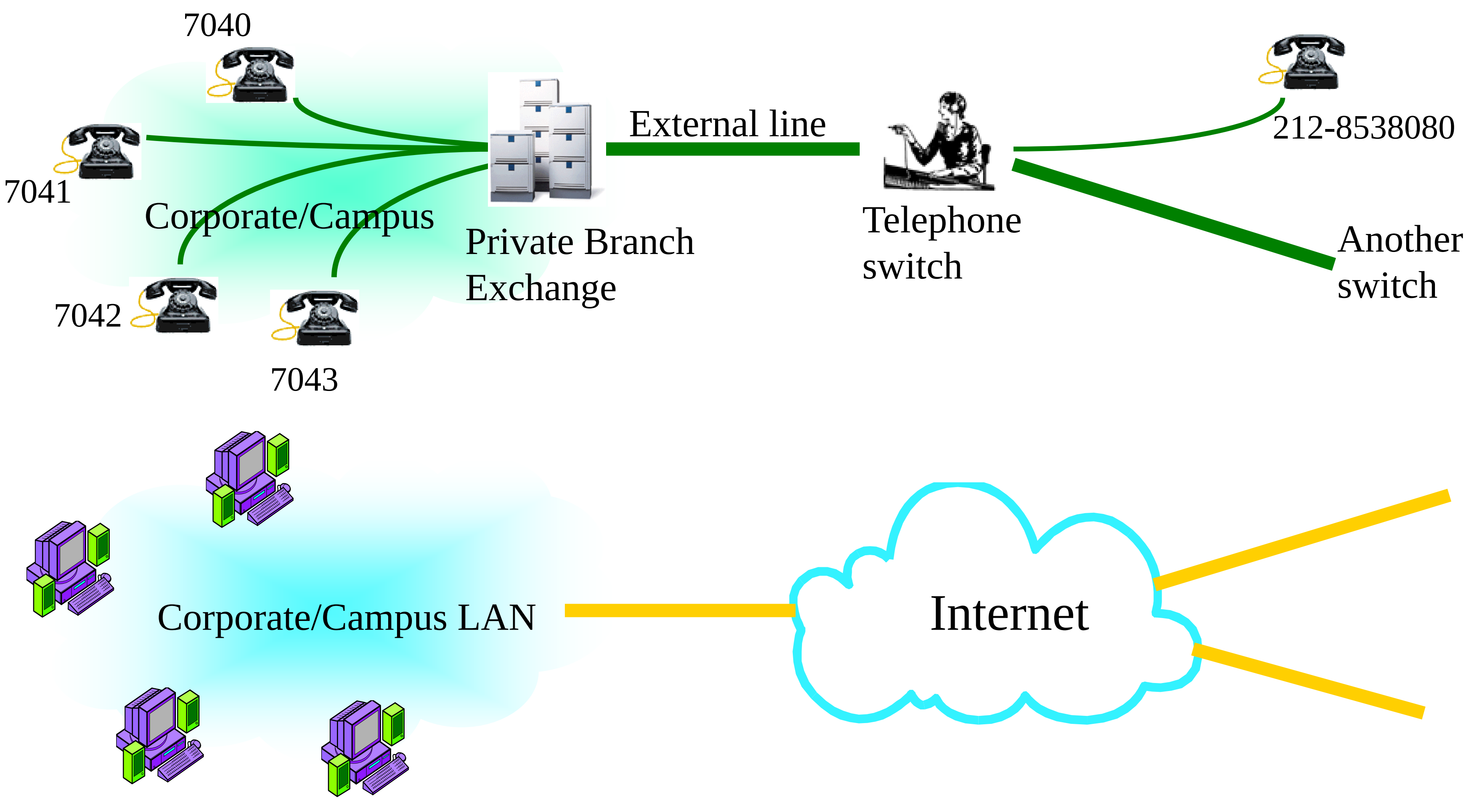
- Two or more users interacting
  - Telephone call
  - Video conference
  - Video game
- Strict delay constraints
  - Delays over 150-200 msec are very noticeable
  - … and delays over 400 msec are a disaster for voice
- Much harder than streaming applications
  - Receiver cannot introduce much playout delay
  - Difficult if the network does not guarantee performance

Voice Over IP (VoIP)

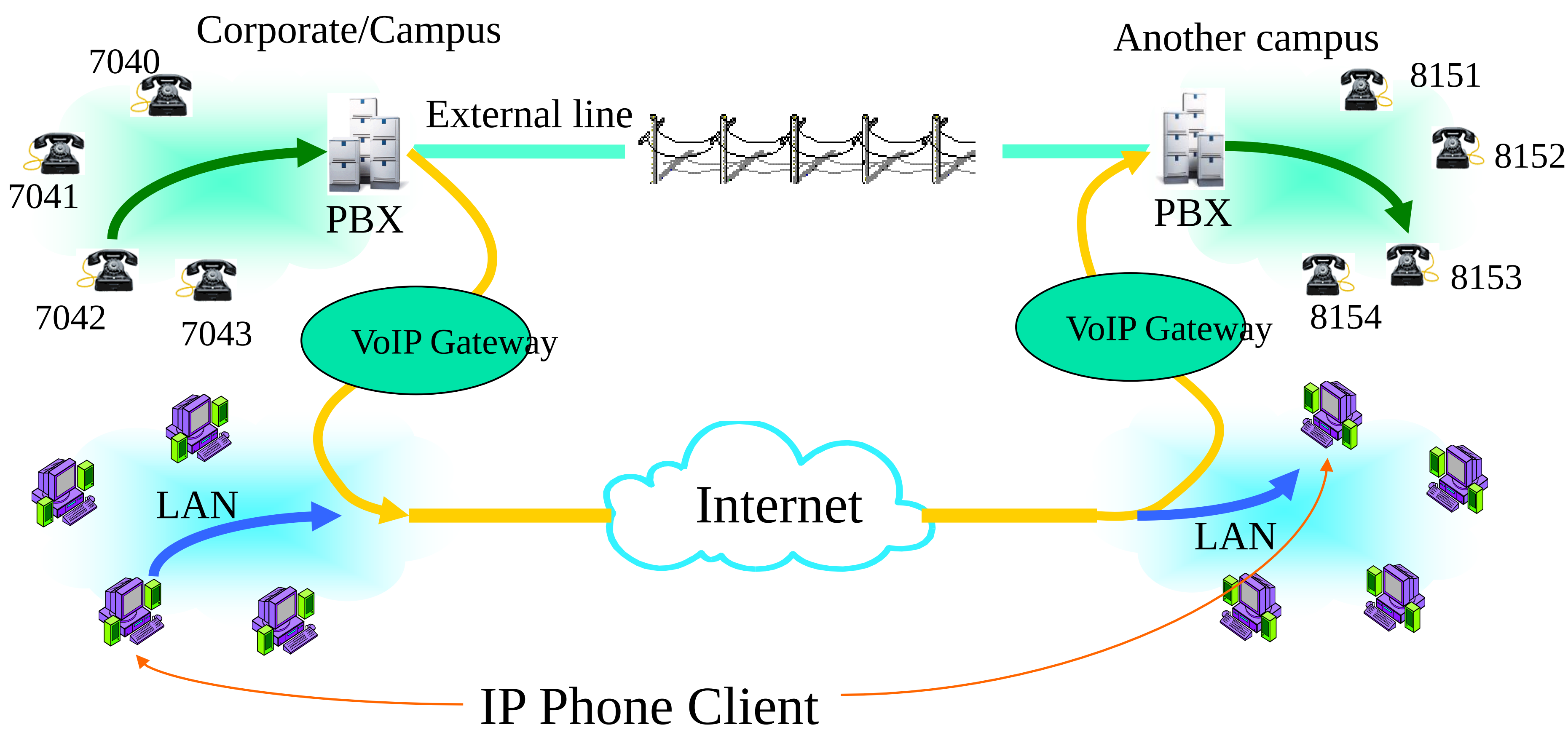
- Delivering phone calls over IP
  - Computer to computer
  - Analog phone to/from computer
  - Analog phone to analog phone
- Motivations for VoIP
  - Cost reduction
  - Simplicity
  - Advanced applications
    - Web-enabled call centers
    - Collaborative white boarding
    - Do Not Disturb, Locate Me, etc.
    - Voicemail sent as e-mail



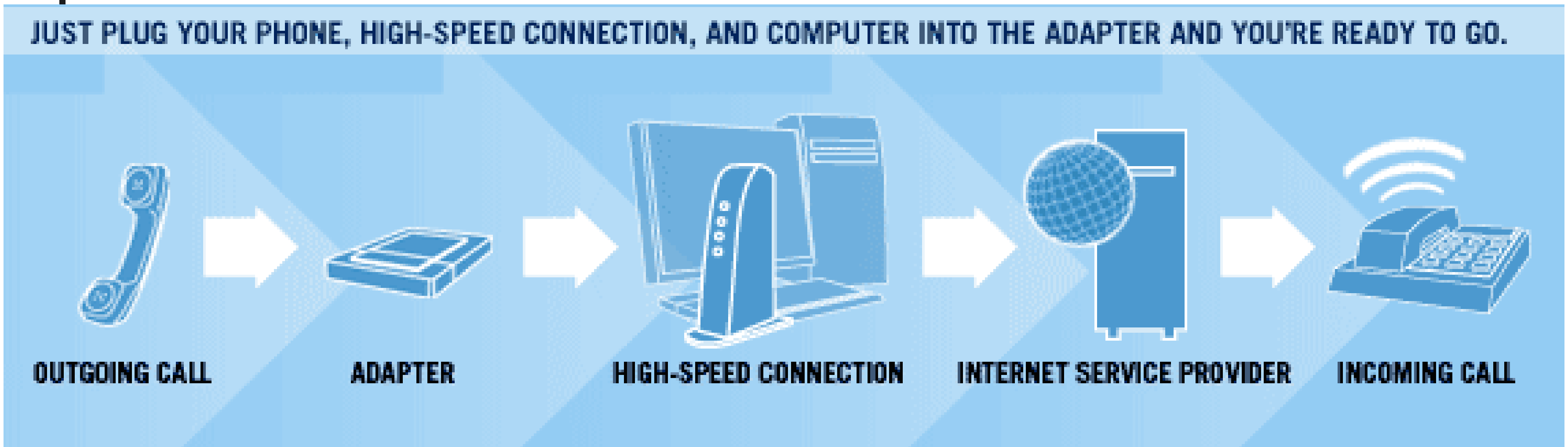
Traditional Telecom Infrastructure



VoIP Gateways



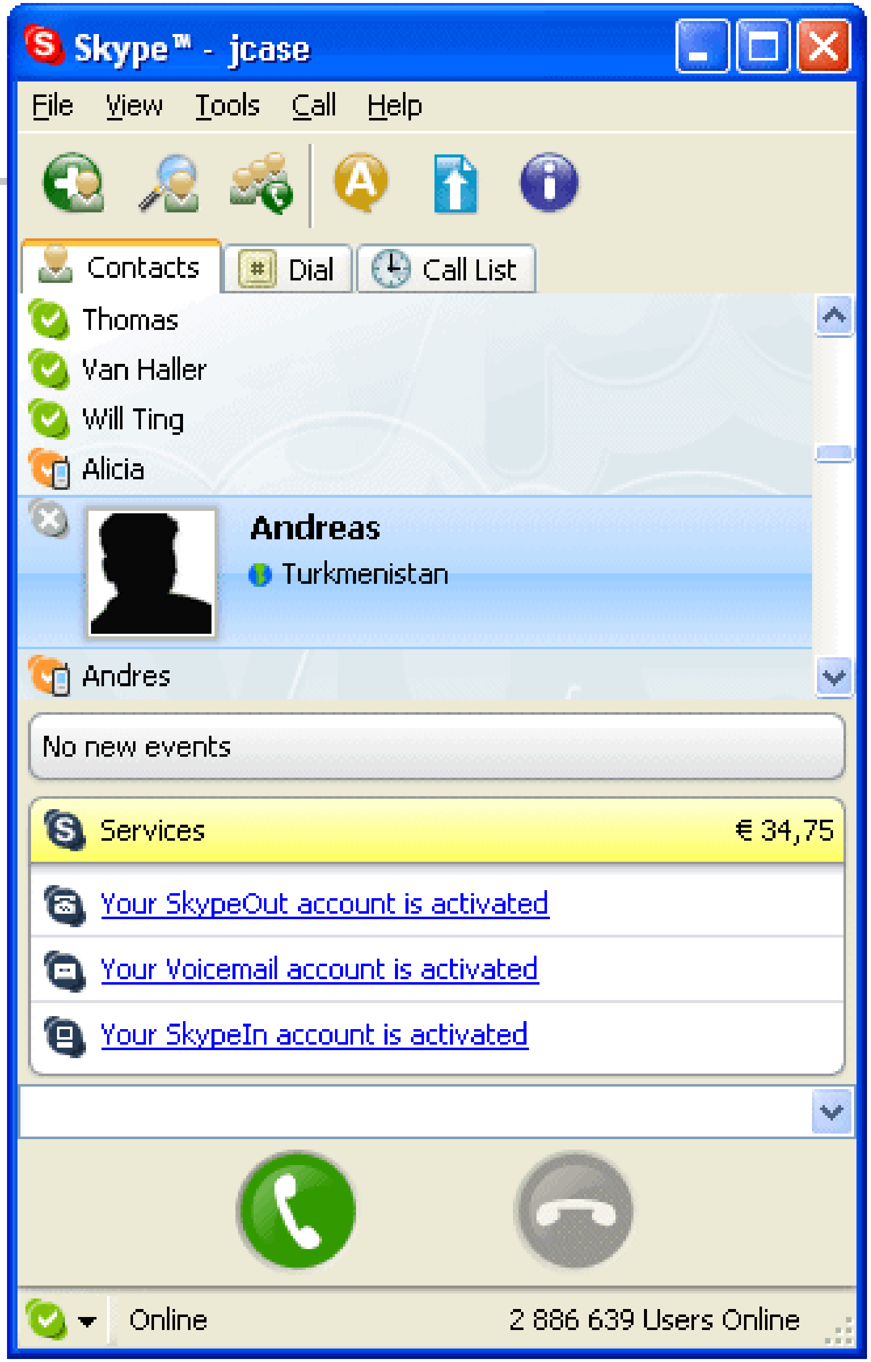
VoIP With an Analog Phone



- Adapter
  - Converts between analog and digital
  - Sends and receives data packets
  - Communicates with the phone in standard way

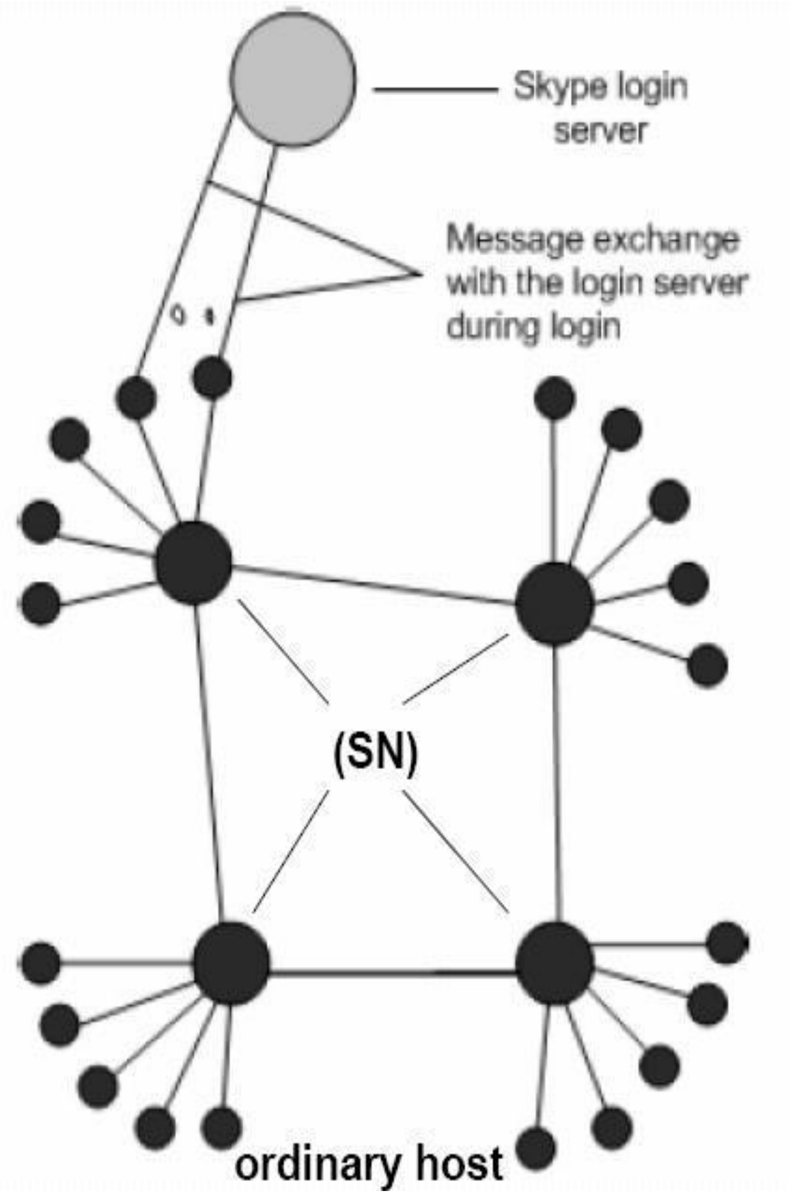
Skype

- Niklas Zennström and Janus Friis in 2003
- Developed by KaZaA
- Instant Messenger (IM) with voice support
- Based on peer-to-peer (P2P) networking technology



Skype Network Architecture

- Login server is the only central server (consisting of multiple machines)
- Both ordinary host and super nodes are Skype clients
- Any node with a public IP address and having sufficient resources can become a super node
- Skype maintains their own super nodes



Challenges of Firewalls and NATs

- Firewalls
  - Often block UDP traffic
  - Usually allow hosts to initiate connections on port 80 (HTTP) and 443 (HTTPS)
- NAT
  - Cannot easily initiate traffic to a host behind a NAT
  - … since there is no unique address for the host
- Skype must deal with these problems
  - Discovery: client exchanges messages with super node
  - Traversal: sending data through an intermediate peer

Data Transfer

- UDP directly between the two hosts
  - Both hosts have public IP address
  - Neither host's network blocks UDP traffic
  - Easy: the hosts can exchange UDP packets directly
- UDP between an intermediate peer
  - One or both hosts with a NAT
  - Neither host's network blocks UDP traffic
  - Solution: direct UDP packets through another node
- TCP between an intermediate peer
  - Hosts behind NAT and UDP-restricted firewall
  - Solution: direct TCP connections through another node

Silence Suppression

- What to transfer during quiet periods?
  - Could save bandwidth by reducing transmissions
  - E.g., send nothing during silence periods
- Skype does not appear to do silence suppression
  - Maintain the UDP bindings in the NAT boxes
  - Provide background noise to play at the receiver
  - Avoid drop in the TCP window size
- Skype sends data when call is “on hold”
  - Send periodic messages as a sort of heartbeat
  - Maintain the UDP bindings in the NAT boxes
  - Detect connectivity problems on the network path

Skype Data Transfer

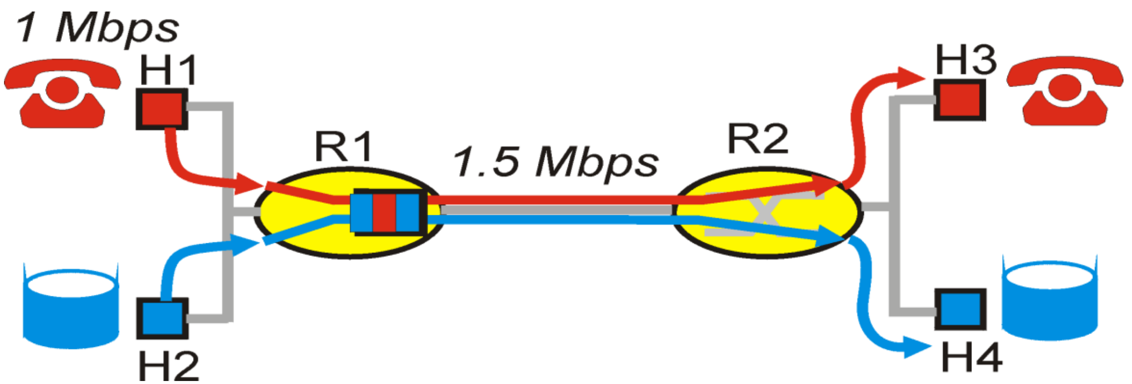
- Audio compression
  - Voice packets around 67 bytes
  - Up to 140 packets per second
  - Around 5 KB/sec (40 kbps) in each direction
- Encryption
  - Data packets are encrypted in both directions
  - To prevent snooping on the phone call
  - … by someone snooping on the network
  - … or by the intermediate peers forwarding data

VoIP Quality

- The application can help
  - Good audio compression algorithms
  - Avoiding hops through far-away hosts
  - Forward error correction
  - Adaptation to the available bandwidth
- But, ultimately the network is a major factor
  - Long propagation delay?
  - High congestion?
  - Disruptions during routing changes?
- Leads to an interest in Quality of Service

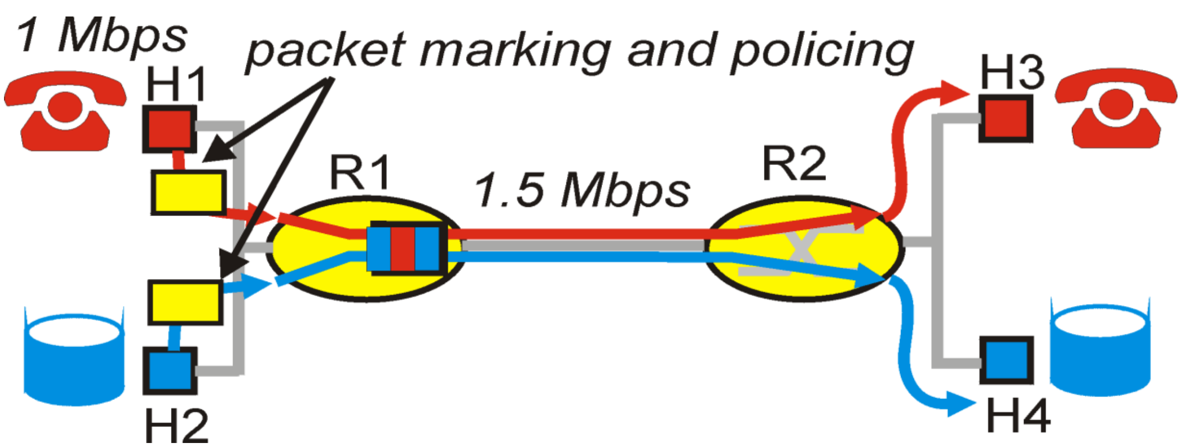
Principles for QoS Guarantees

- Applications compete for bandwidth
  - Consider a 1 Mbps VoIP application and an FTP transfer sharing a single 1.5 Mbps link
  - Bursts of FTP traffic can cause congestion and losses
  - We want to give priority to the audio packets over FTP
- Principle 1: Packet marking
  - Marking of packets is needed for the router
  - To distinguish between different classes



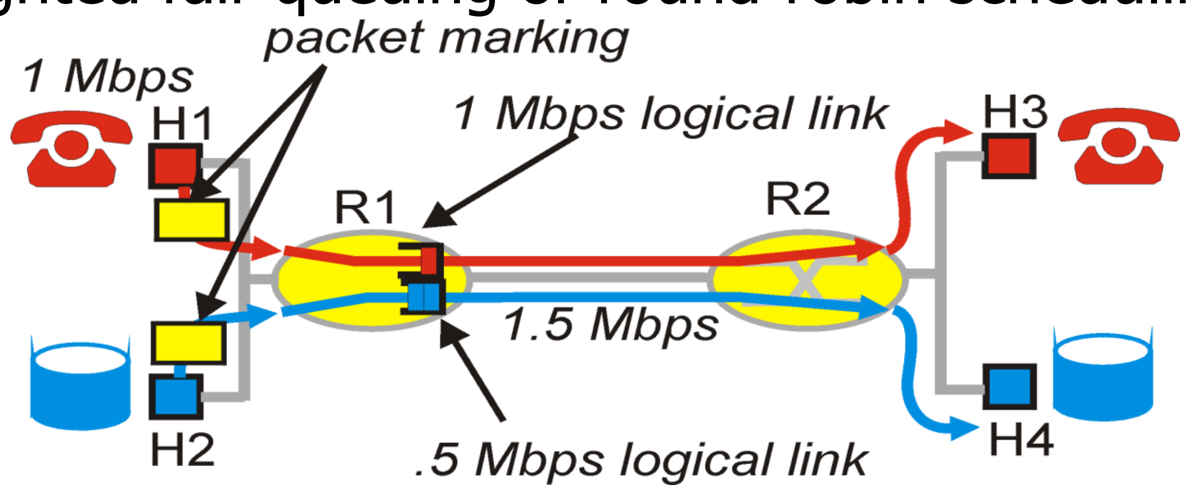
Principles for QoS Guarantees

- Applications misbehave
  - Audio sends packets at a rate higher than 1 Mbps
- Principle 2: Policing
  - Provide protection for one class from other classes
  - Ensure sources adhere to bandwidth restrictions
  - Marking and policing need to be done at the edge



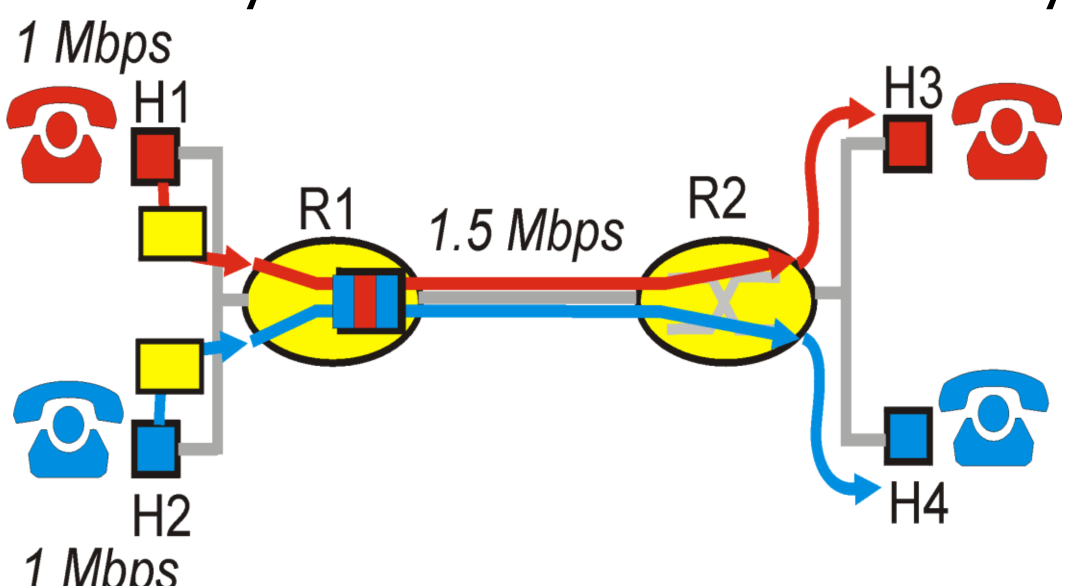
Principles for QoS Guarantees

- Alternative to marking and policing
  - Allocate fixed bandwidth to each application flow
  - But, this can lead to inefficient use of bandwidth
  - … if one of the flows does not use its allocation
- Principle 3: Link scheduling
  - While providing isolation, it is desirable to use resources as efficiently as possible
  - E.g., weighted fair queuing or round-robin scheduling



Principles for QoS Guarantees

- Cannot support traffic beyond link capacity
  - If total traffic exceeds capacity, you are out of luck
  - Degrade the service for all, or deny someone access
- Principle 4: Admission control
  - Application flow declares its needs in advance
  - The network may block call if it cannot satisfy the needs



Quality of Service

- Significant change to Internet architecture
  - Guaranteed service rather than best effort
  - Routers keeping state about the traffic
- A variety of new protocols and mechanisms
  - Reserving resources along a path
  - Identifying paths with sufficient resources
  - Link scheduling and buffer management
  - Packet marking with the Type-of-Service bits
  - Packet classifiers to map packets to ToS classes
- Seeing some deployment within individual ASes
  - E.g., corporate/campus networks, and within an ISP

Summary

- Digital audio and video
  - Increasingly popular media on the Internet
  - Video on demand, VoIP, online gaming, IPTV, …
- Interaction with the network
  - Adapt to delivering the data over a best-effort network
  - Adapt the network to offer better quality-of-service