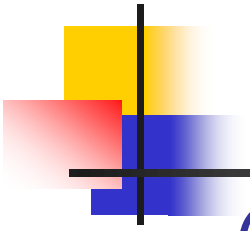
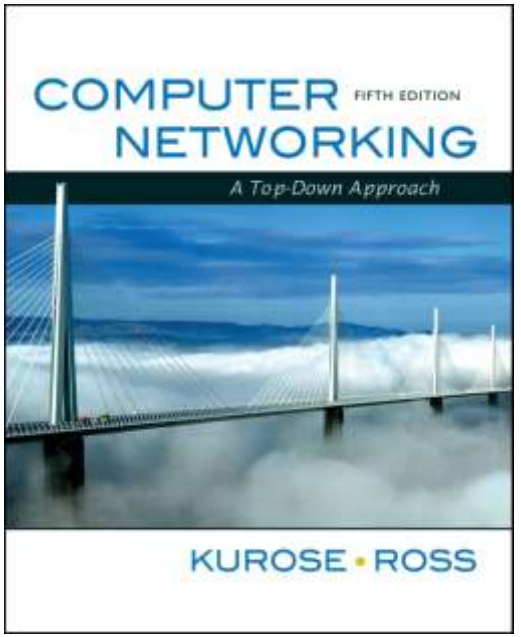


CS234 – Multimedia Networking

Tuesdays, Thursdays 3:30-4:50p.m.
ICS 243

Prof. Nalini Venkatasubramanian
nalini@ics.uci.edu

Chapter 7
Multimedia Networking



Slides adapted from :

Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

Multimedia Systems

- Combination of media
  - continuous and discrete.
  - Levels of media-independence
    - some media types (audio/video) tightly coupled, others not.
  - Computer supported integration
    - timing, spatial and semantic synchronization
- Distributed multimedia communication systems
  - data of discrete and continuous media are broken into individual units (packets) and transmitted.
- Data Stream
  - sequence of individual packets that are transmitted in a time-dependant fashion.
  - Transmission of information carrying different media leads to data streams with varying features
    - Asynchronous
    - Synchronous
    - Isochronous

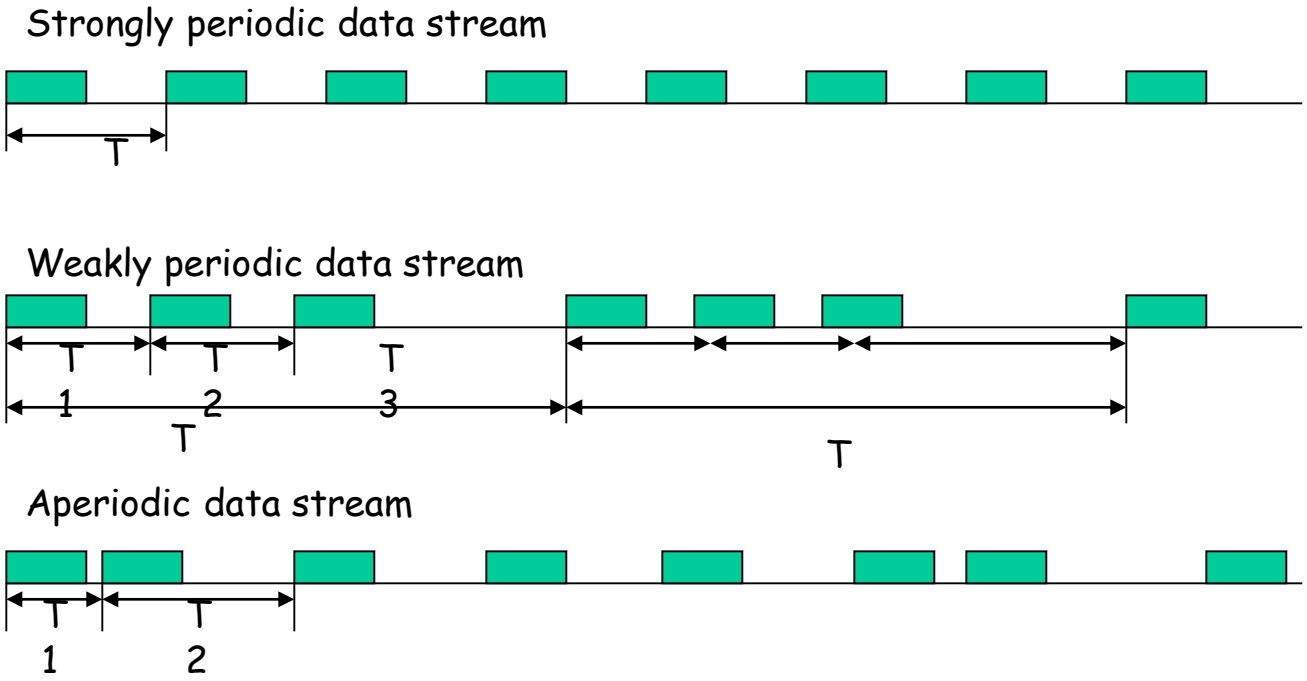
Data Stream Characteristics

- Asynchronous transmission mode
  - provides for communication with no time restriction
  - Packets reach receiver as quickly as possible, e.g. protocols for email transmission
- Synchronous transmission mode
  - defines a maximum end-to-end delay for each packet of a data stream.
  - May require intermediate storage
  - E.g. audio connection established over a network.
- Isochronous transmission mode
  - defines a maximum and a minimum end-to-end delay for each packet of a data stream. Delay jitter of individual packets is bounded.
  - E.g. transmission of video over a network.
  - Intermediate storage requirements reduced.

Data Stream Characteristics

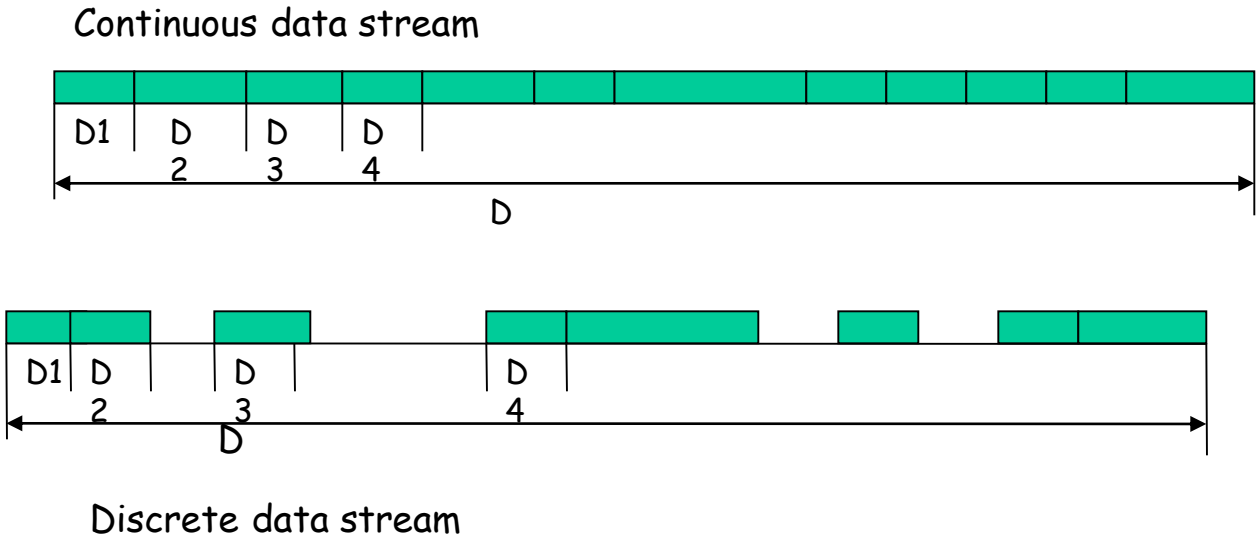
- Data Stream characteristics for continuous media can be based on
  - Time intervals between complete transmission of consecutive packets
    - Strongly periodic data streams - constant time interval
    - Weakly periodic data streams - periodic function with finite period.
    - Aperiodic data streams
  - Data size - amount of consecutive packets
    - Strongly regular data streams - constant amount of data
    - Weakly regular data streams - varies periodically with time
    - Irregular data streams
  - Continuity
    - Continuous data streams
    - Discrete data streams

Classification based on time intervals



Classification based on packet size

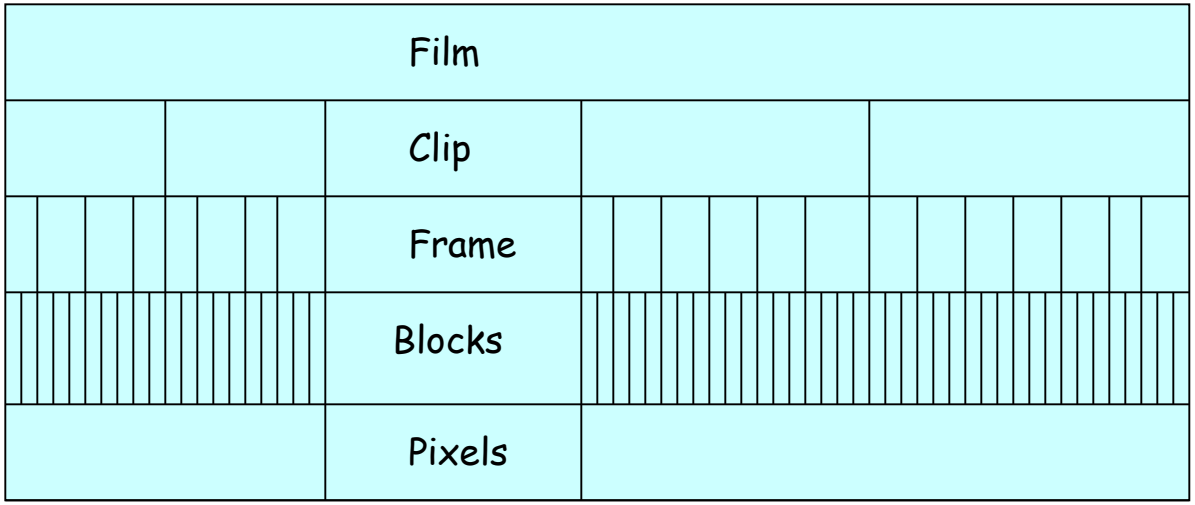
Classification based on continuity



Logical Data Units

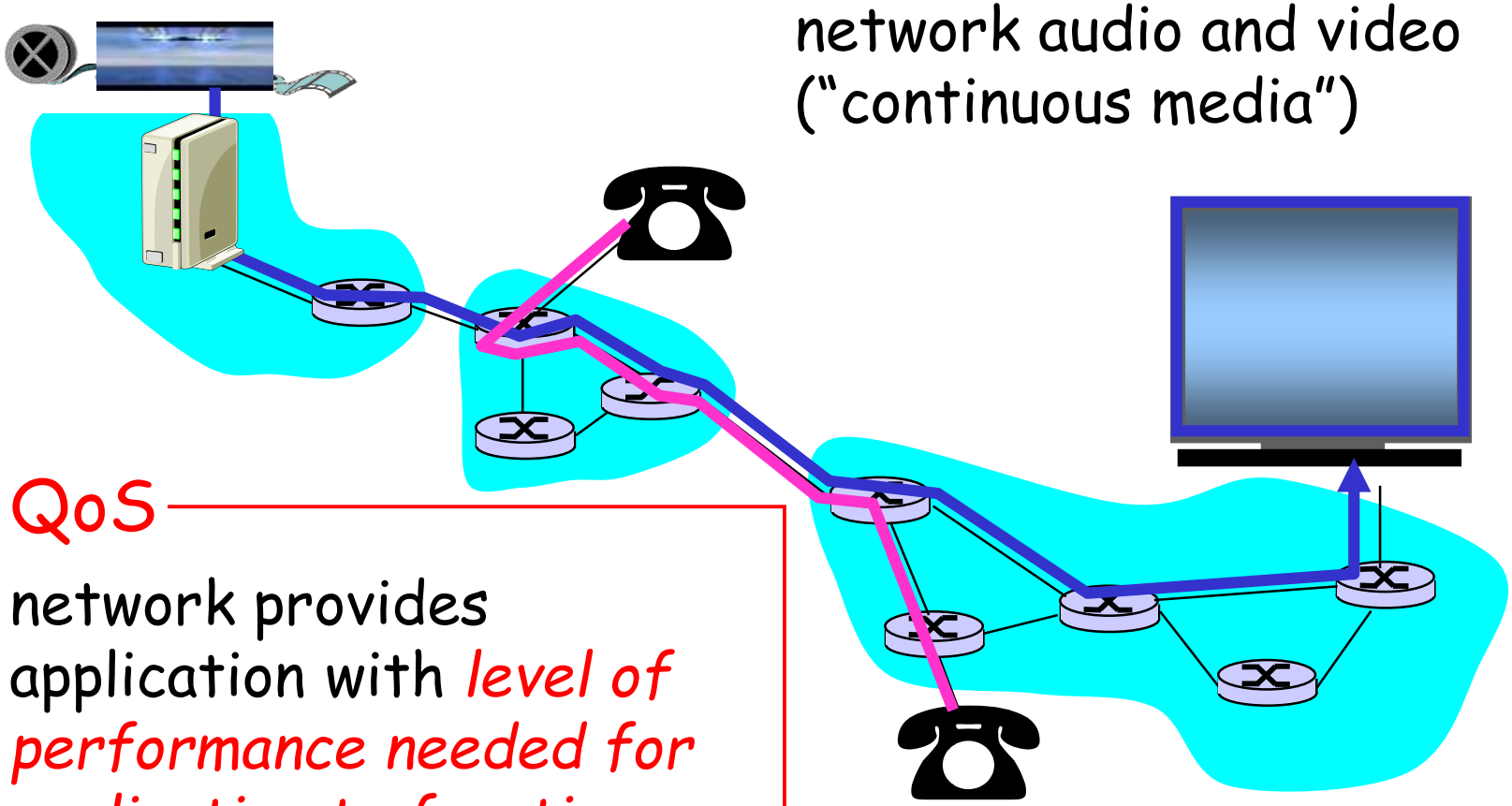
- Continuous media consist of a time-dependent sequence of individual information units called Logical Data Units (LDU).
  - a symphony consists of independent sentences
  - a sentence consists of notes
  - notes are sequences of samples
- Granularity of LDUs
  - symphony, sentence, individual notes, grouped samples
  - film, clip, frame, raster, pixel
- Duration of LDU:
  - open LDU - duration not known in advance
  - closed LDU - predefined duration

Granularity of Logical Data Units



Multimedia and 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.*

Goals

Principles

- classify multimedia applications
- identify network services applications need
- making the best of best effort service

Protocols and Architectures

- specific protocols for best-effort
- mechanisms for providing QoS
- architectures for QoS

Outline

- Multimedia networking applications
- Requirements for Multimedia Communication
  - User and application requirements
  - Processing and protocol constraints
  - Mapping to OSI layers
- Network QoS and Resource Management
  - Providing multiple classes of service
  - Negotiation, Translation, Admission
  - Traffic Shaping, Rate Control, Error Control
  - Monitoring, Adaptation
  - Protocols for real-time interactive applications (RTP,RTCP,SIP)
- Other Case Studies
  - Fast Ethernet, FDDI, DQDB, ATM

MM Networking Applications

Classes of MM applications:

1) stored streaming
2) live streaming
3) interactive, real-time

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

Fundamental characteristics:

- typically **delay sensitive**
  - end-to-end delay
  - delay jitter
- **loss tolerant**: infrequent losses cause minor glitches
- antithesis of data, which are loss *intolerant* but delay *tolerant*.

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 bits back to analog signal:
  - some quality reduction

Example rates

- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up

A few words about video compression

- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- redundancy
  - spatial (within image)
  - temporal (from one image to next)

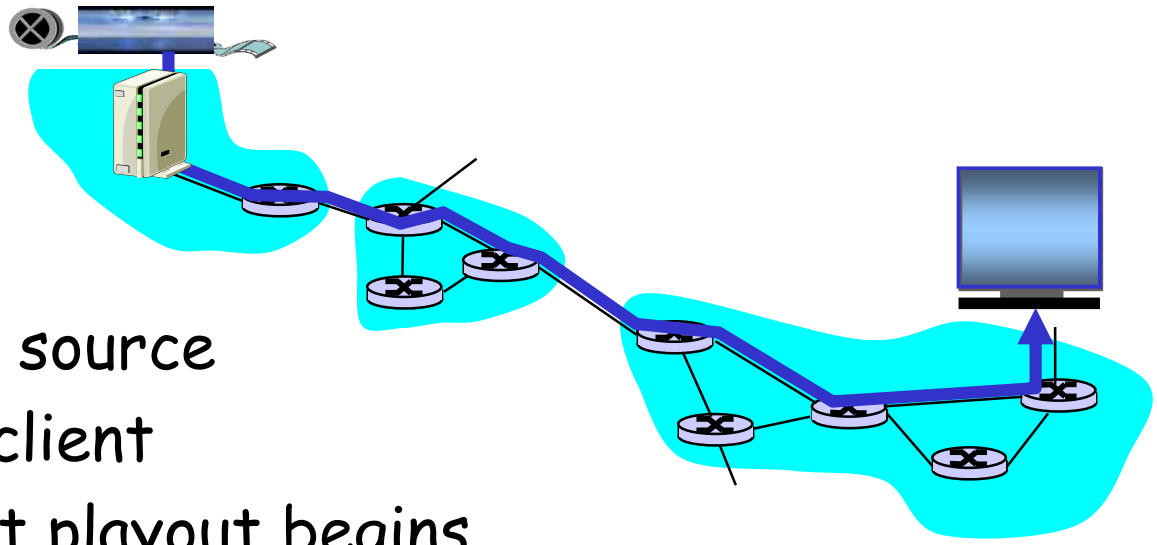
Examples:

- 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

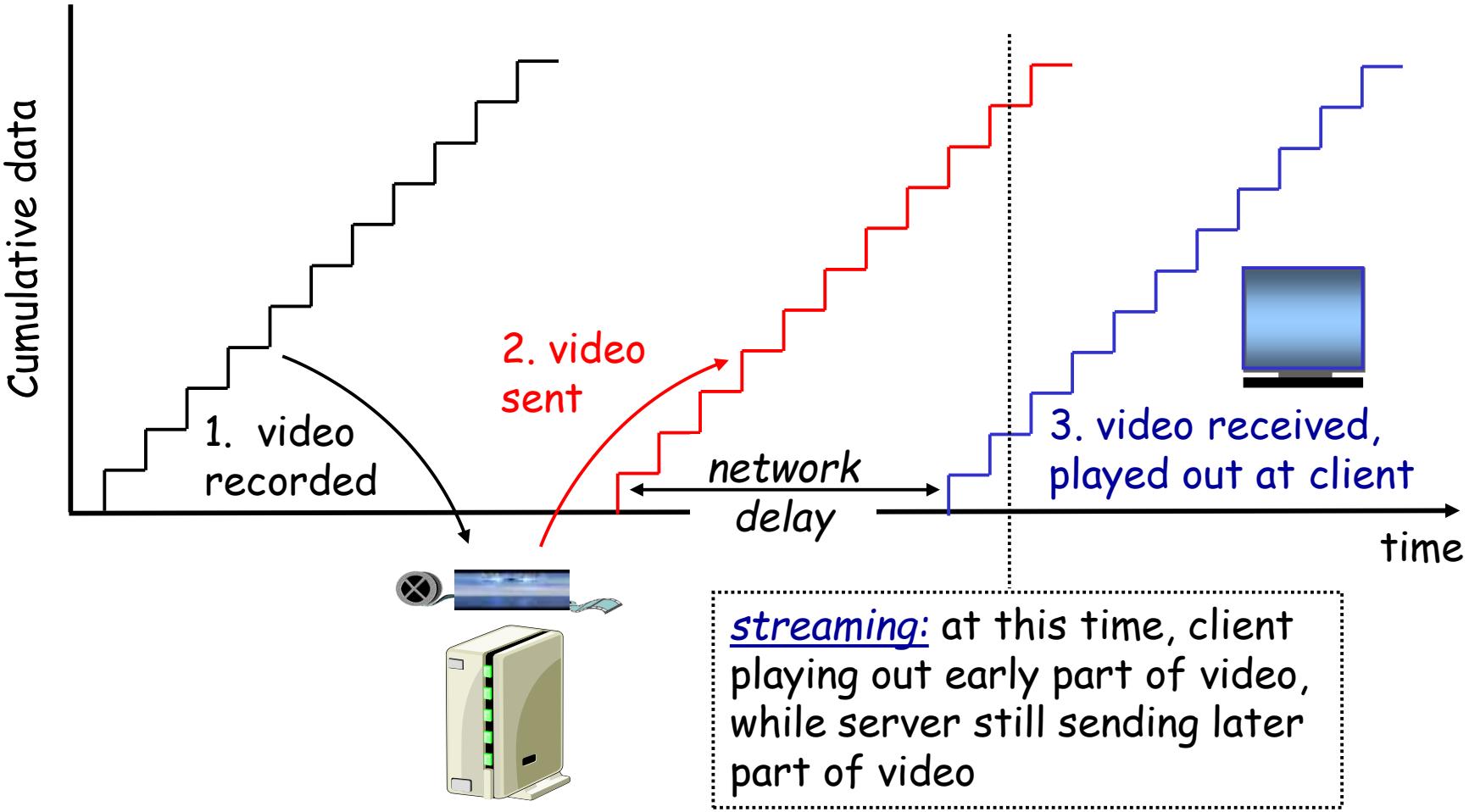
Streaming Stored Multimedia



Stored streaming:

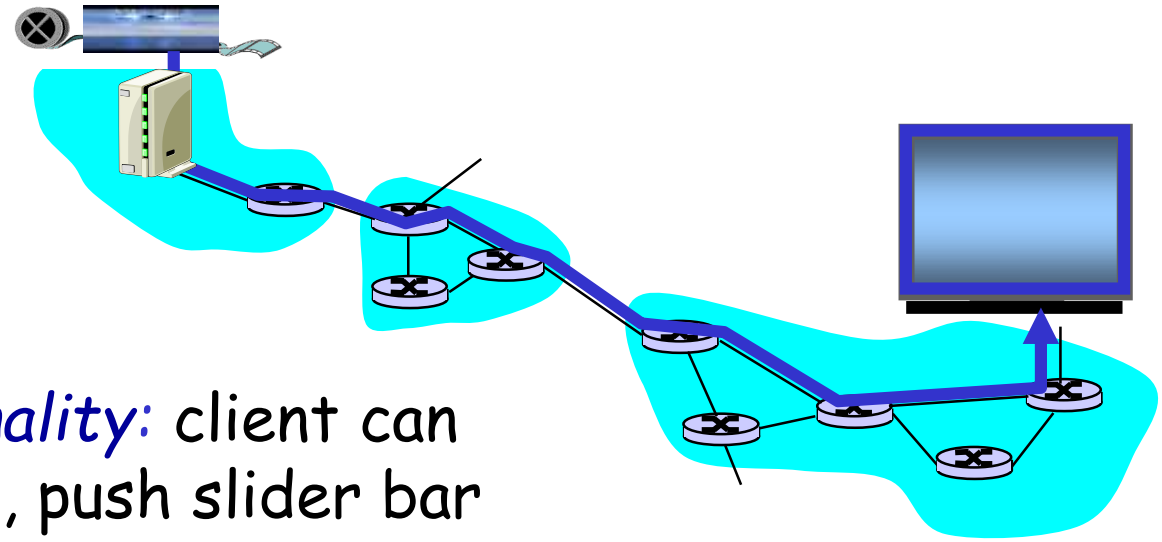
- media stored at source
- transmitted to client
- *streaming:* client playout begins *before* all data has arrived
  - timing constraint for still-to-be transmitted data: in time for playout

Streaming Stored Multimedia: What is it?



streaming: at this time, client playing out early part of video, while server still sending later part of video

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

- timing constraint for still-to-be transmitted data: in time for playout

Streaming *Live* Multimedia

Examples:

- Internet radio talk show
- live sporting event

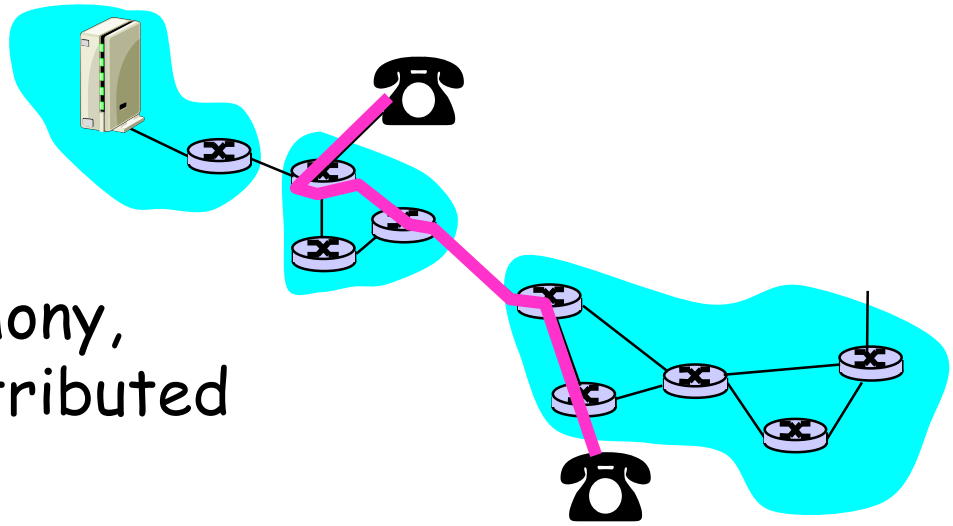
Streaming (as with streaming *stored* multimedia)

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

Interactivity

- fast forward impossible
- rewind, pause possible!

Real-Time Interactive 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?

Requirements on Services and Protocols

- Audio/Video communication needs to be bounded by deadlines or defined by a time interval
  - End-to-end jitter must be bounded
  - End-to-end guarantees are required
- Synchronization mechanisms for different data streams are required
- Communication capability is required
  - Communication of discrete data should not starve
- Fairness principle among applications, users and hosts is required
- Variable bit rate traffic support is required

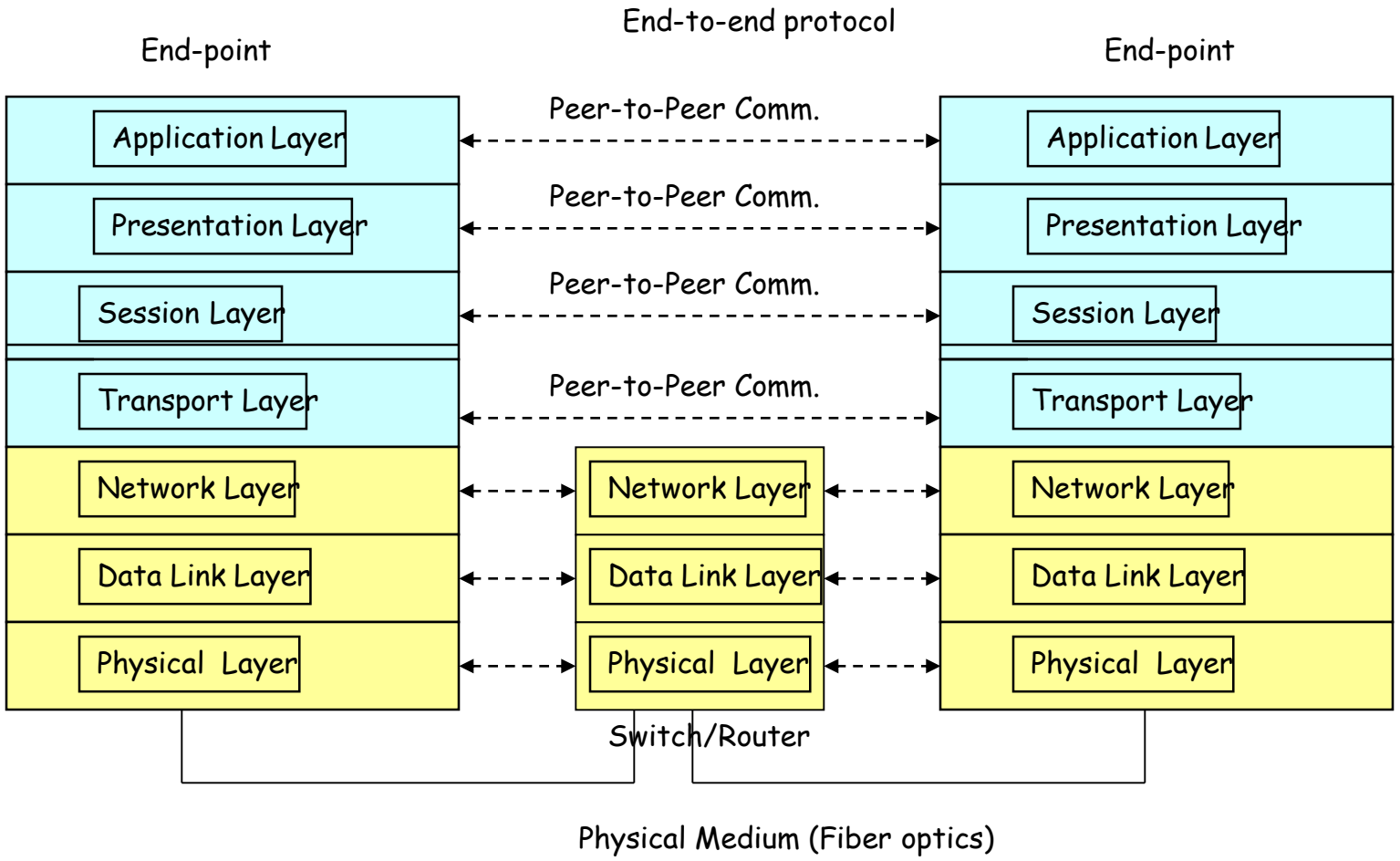
User and Application Requirements

- Data Throughput
  - Application data have stream like behavior with high throughput
  - Need to manipulate large APDU (application protocol data units in real-time)
- Fast Data Forwarding
  - The faster a communication system can transfer a packet, fewer packets have to be buffered
- Service Guarantees - Proper resource management
- Multicasting
  - Efficient sharing of resources
  - Useful for reaching groups of users in applications such as video conferencing.

Processing and protocol constraints

- Adapter-to-adapter transmission
  - achieves fast transmission
  - does not allow control over streams and QoS control
- Data movement in protocol stack
  - requires expensive data copying
  - need to explore other buffer management techniques and strategies.
- Segmentation and reassembly
  - part of the protocol stack - these operations must be done efficiently.
- Retransmission error recovery
- Underlying network
  - may provide many transmission modes

OSI Layering



Mapping of Requirements to OSI

- Physical Layer
  - defines transmission methods of individual bits over a physical medium.
  - For multimedia, need high bandwidth and minimal delay upto gigabit/terabit tranmission rates
    - ATM switched with SONET physical layer deliver upto 2.4 and higher Gbps
- Data Link Layer
  - defines transmission of blocks called data frames
    - defines access protocols to physical medium, flow control and block synchronization
    - E.g MAC (medium-access-control) sublayer defines Timed Token rotation protocol in Token Ring/FDDI and CSMA/CD protocol in Fast Ethernet
    - Audio/video require reservations and throughput guarantees at this layer
    - can also define mechanisms for error correction at this layer

Mapping of requirements to OSI

- Network Layer
  - defines transmission of information blocks called packets
  - Services in this layer include addressing, inter-networking, error-handling, network management, congestion control, sequencing of packets, multi-casting
  - Audio/video require reservation and guarantees at this layer.
    - These requests for guarantees are defined by appropriate network QoS parameters.
  - Audio/video requires connection-oriented behavior where reservations are made during connection setup.
  - Reservation must be done along the path between the communication stations.
  - Network QoS must be negotiated at this layer.

Mapping of MM requirements to OSI

- Transport Layer
  - provides a process to process connection
  - In this layer, the network QoS is enhanced
    - If the network service is poor, the transmission layer bridges the gap between what the transport user wants and what the network provides.
  - Error handling is based on process-to-process communication.
  - Error handling should not include retransmission for audio/video because this mechanism introduces high end-to-end delay.
  - Synchronization and rate control should be supported.

Mapping of MM requirements to OSI

- Session Layer
  - This layer guarantees existence of multimedia connections during a whole multimedia session
    - provides synchronization within a stream and among streams
    - provides support for point-to-point session and multicast sessions.
- Presentation Layer
  - This layer abstracts from different formats
  - Includes services for transformation between application specific formats and the agreed upon transport format.
  - Audio/video conversation is needed because many formats exist.

Mapping of MM requirements to OSI

- Application Layer
  - Audio/video need support for real-time access and transmission
  - Audio/video services supported in this layer include playback, record, fast forward, rewind, pause etc..

Outline

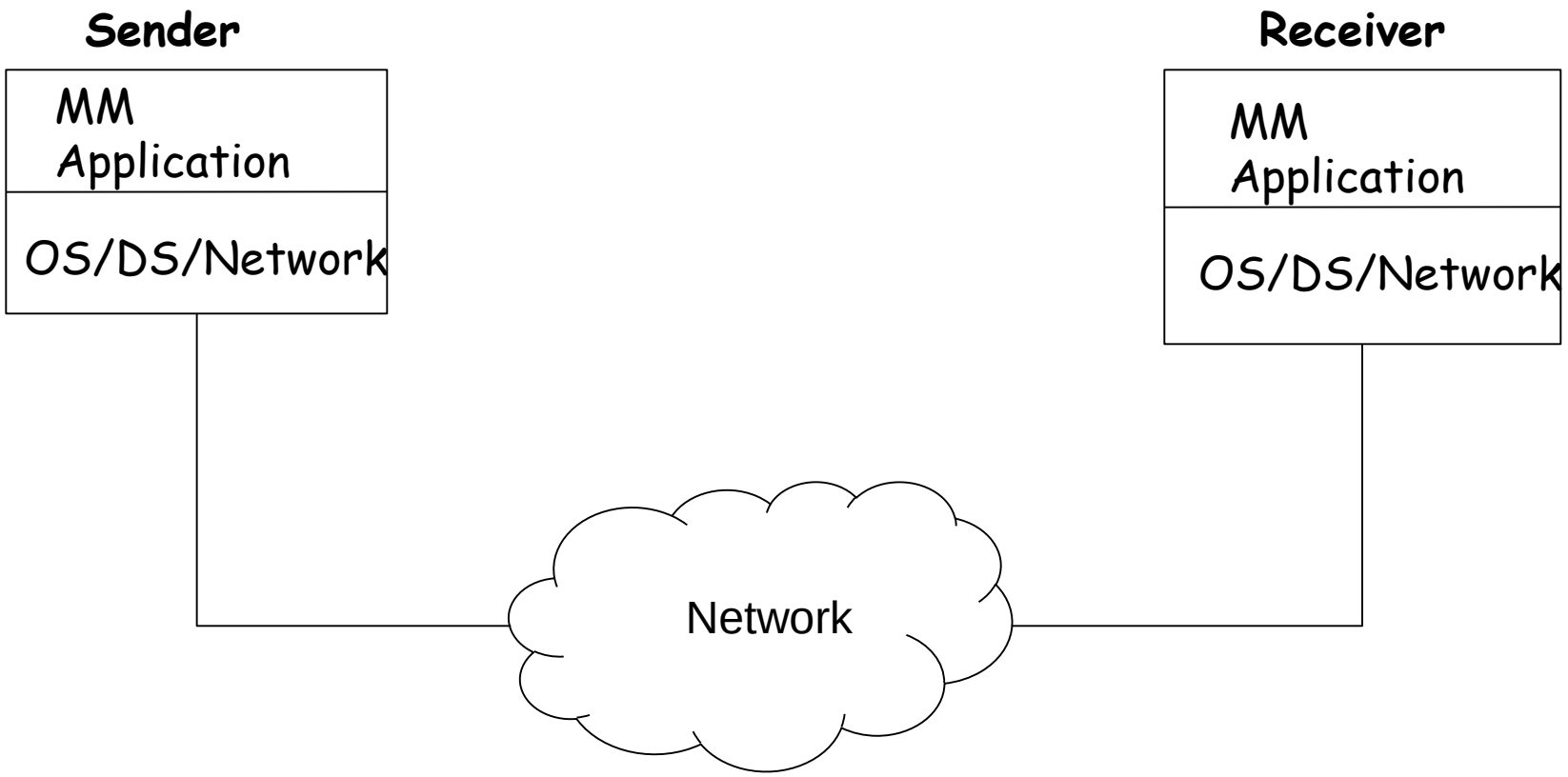
- Multimedia networking applications
- Requirements for Multimedia Communication
  - User and application requirements
  - Processing and protocol constraints
  - Mapping to OSI layers

- Network QoS and Resource Management
  - Providing multiple classes of service
  - Negotiation, Translation, Admission
  - Traffic Shaping, Rate Control, Error Control
  - Monitoring, Adaptation
- MM over Internet
  - Protocols for real-time interactive applications (RTP,RTCP,SIP)
- Other Case Studies
  - Fast Ethernet, FDDI, DQDB, ATM

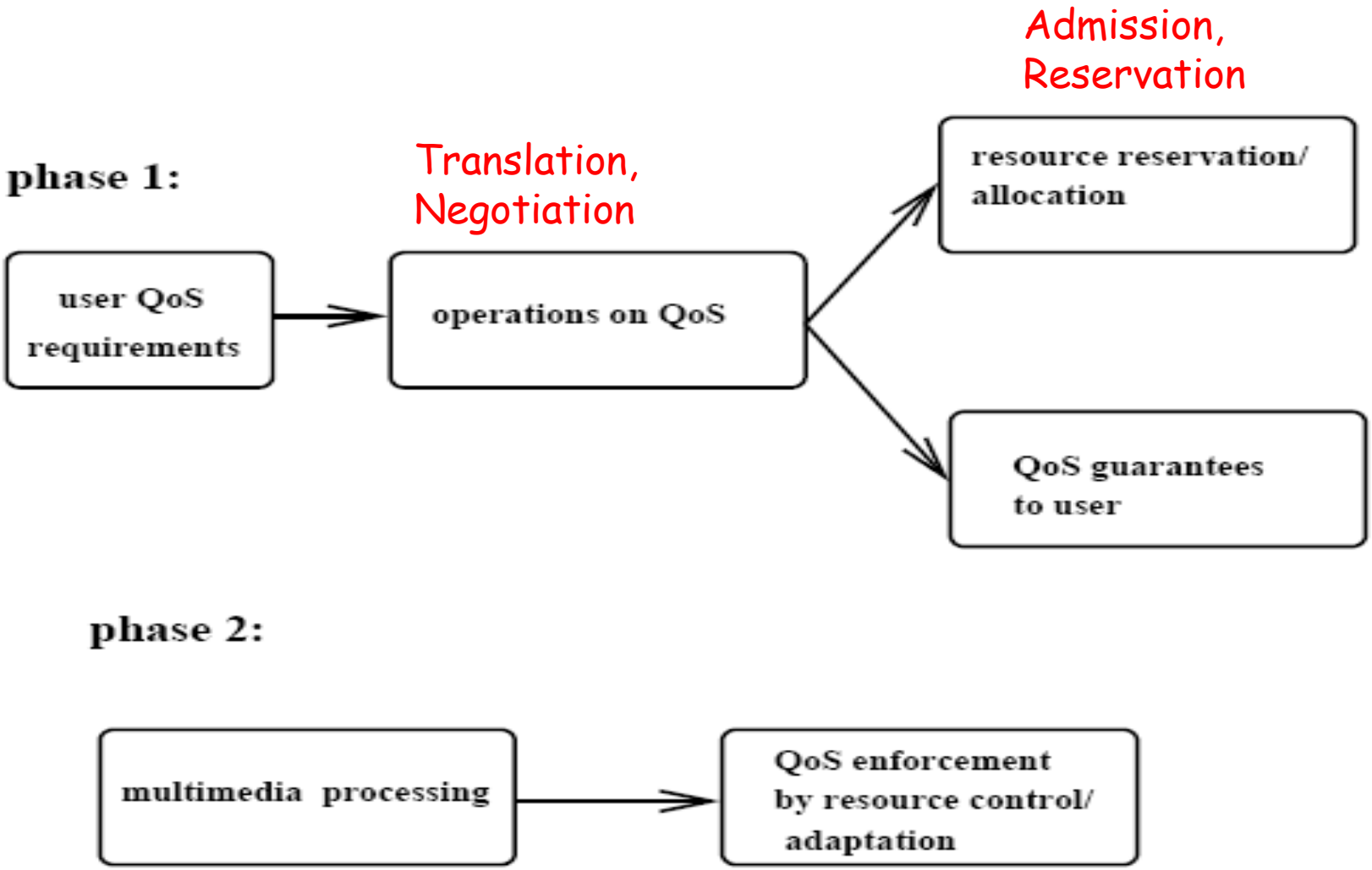
Network QoS and resource management

- Network QoS parameters include:
  - end-to-end delay, jitter, packet rate, burst, throughput, packet loss.
- For establishment of a multimedia call, the following tasks must be performed:
  - Application/user defines QoS parameters
  - QoS parameters must be distributed and negotiated
  - QoS parameters must be translated between the different layers.
  - QoS parameters must be mapped to resource requirements.
  - Required resources must be admitted, reserved and allocated along the path between the sender and receiver(s).

Multimedia System/Network



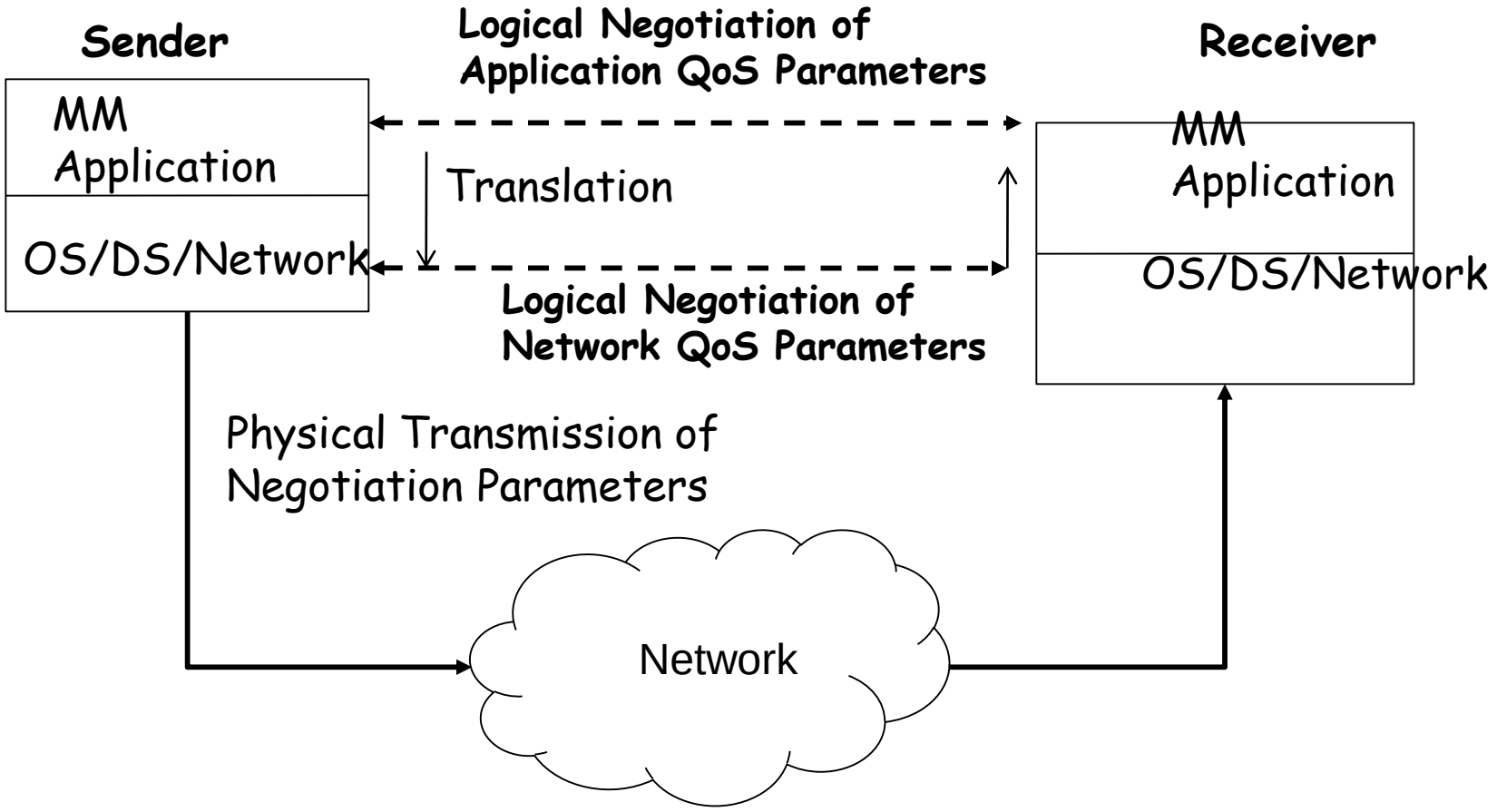
Relation between QoS and Resources (Phase 1)



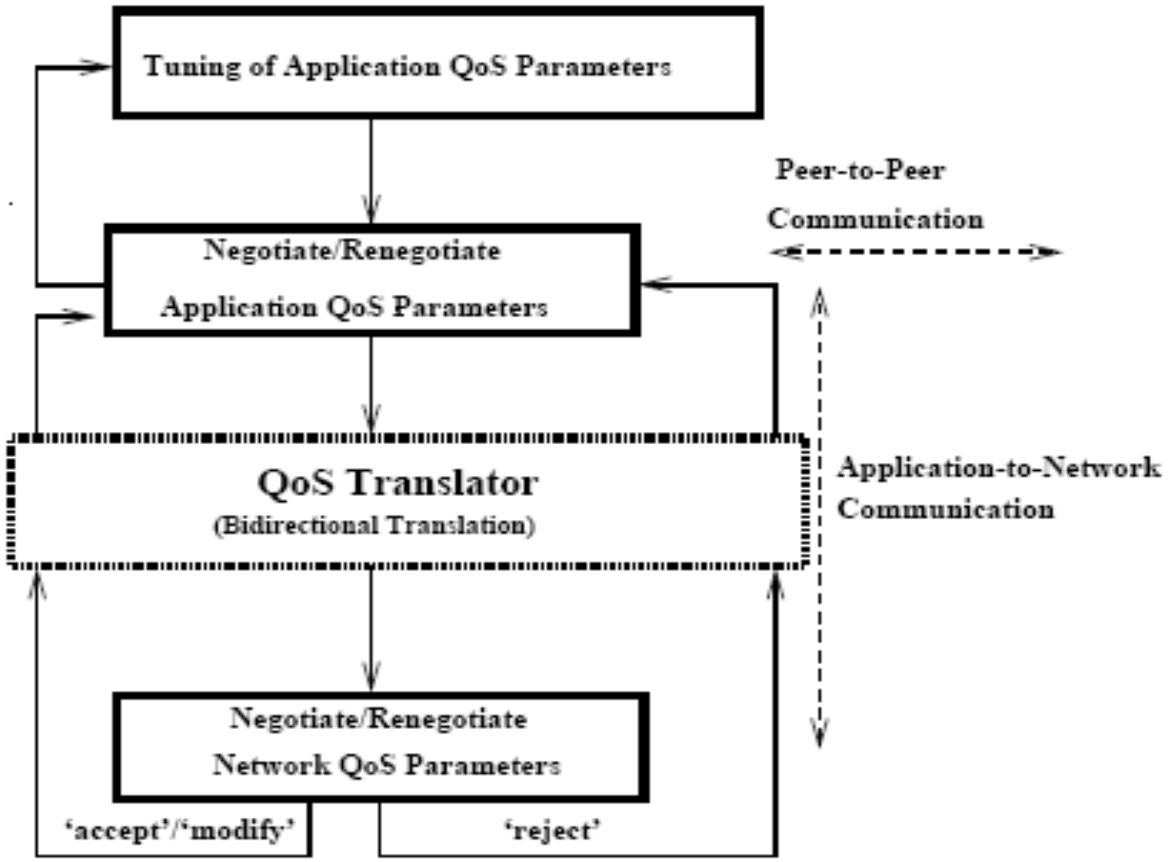
Phase 1: Establishment Phase (QoS Operations)

- **QoS Translation** at different Layers
  - User-Application
  - Application-OS/Transport Subsystem
- **QoS Negotiation**
  - Negotiation of QoS parameters among two peers/components

Phase 1: Connection Establishment



QoS Operations within Establishment Phase



User/Application
QoS Translation

Overlay P2P
QoS Negotiation

Application/Transpor
QoS Translation

QoS Negotiation in
Transport Subsystem

Example

- Video Stream Quality:
  - Frame size: 320x240 pixels, 24 bits (3 Bytes per pixel)
  - Application frame rate RA: 20 fps
- Translate to Network QoS if
  - Assume network packet size is 4KBytes
  - Network packet rate (RN):= ⌈320x240x3⌉ bytes / 4096 bytes

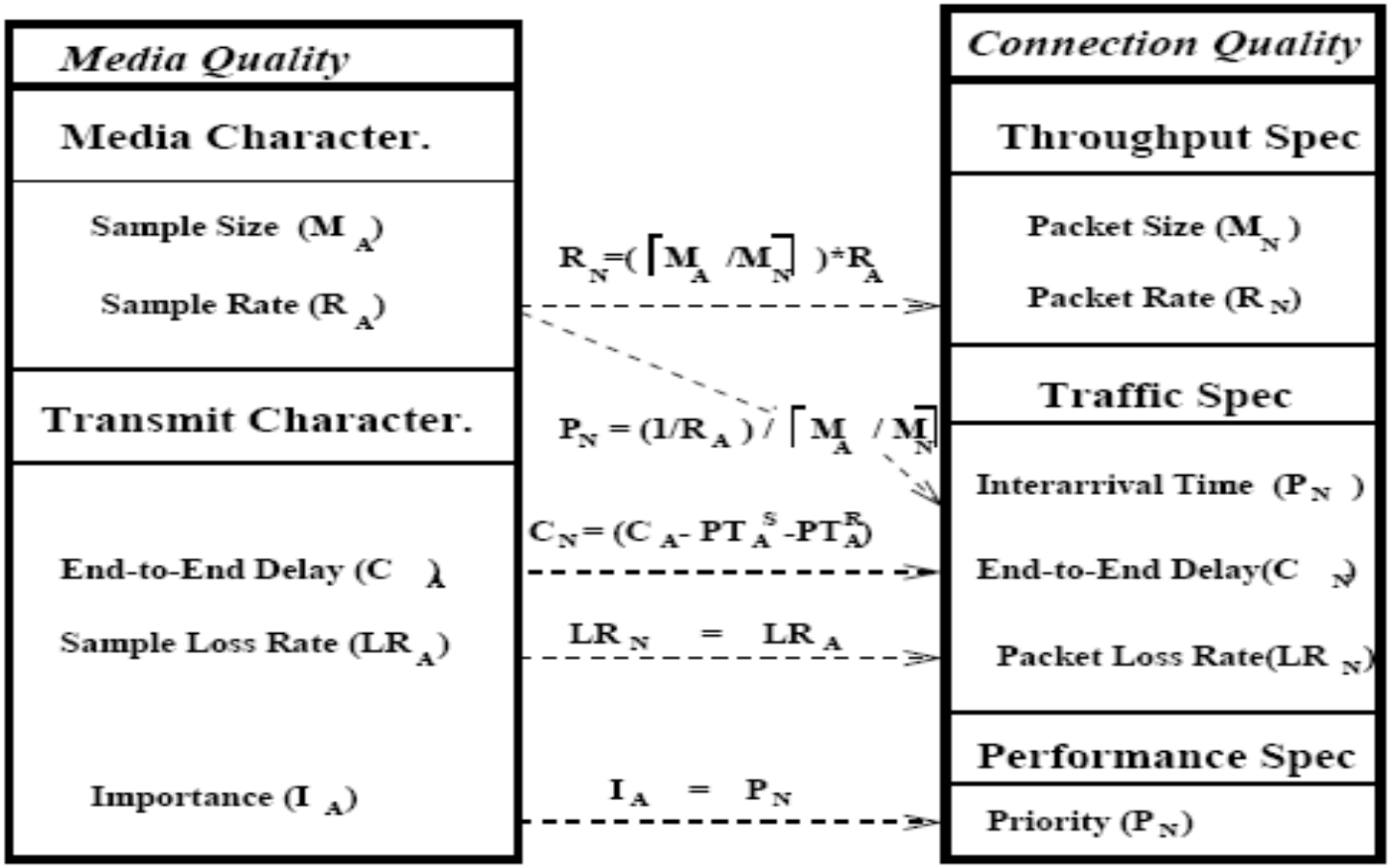
Negotiation and translation

- For negotiation of network QoS, use peer-to-peer negotiation and triangular negotiation
- QoS Translation
  - happens between QoS parameters specified in the application layer and required in the transport/network layer.
    - (frame size M_a, frame rate R_a) to (throughput B_n, packet rate R_n)
    - Assume frame size of 320x240 pixels, 8bits/pixel, frame rate 10fps. Assume packet size (M_n) is 4Kbytes.
    - Throughput of the application is B_a = M_a * R_a = 6,144,000 bits/sec
    - Packet rate R_n = ($\lceil M\_a/M\_n \rceil$) * R_a = 190 packets/sec
    - Network bandwidth B_n = M_n * R_n

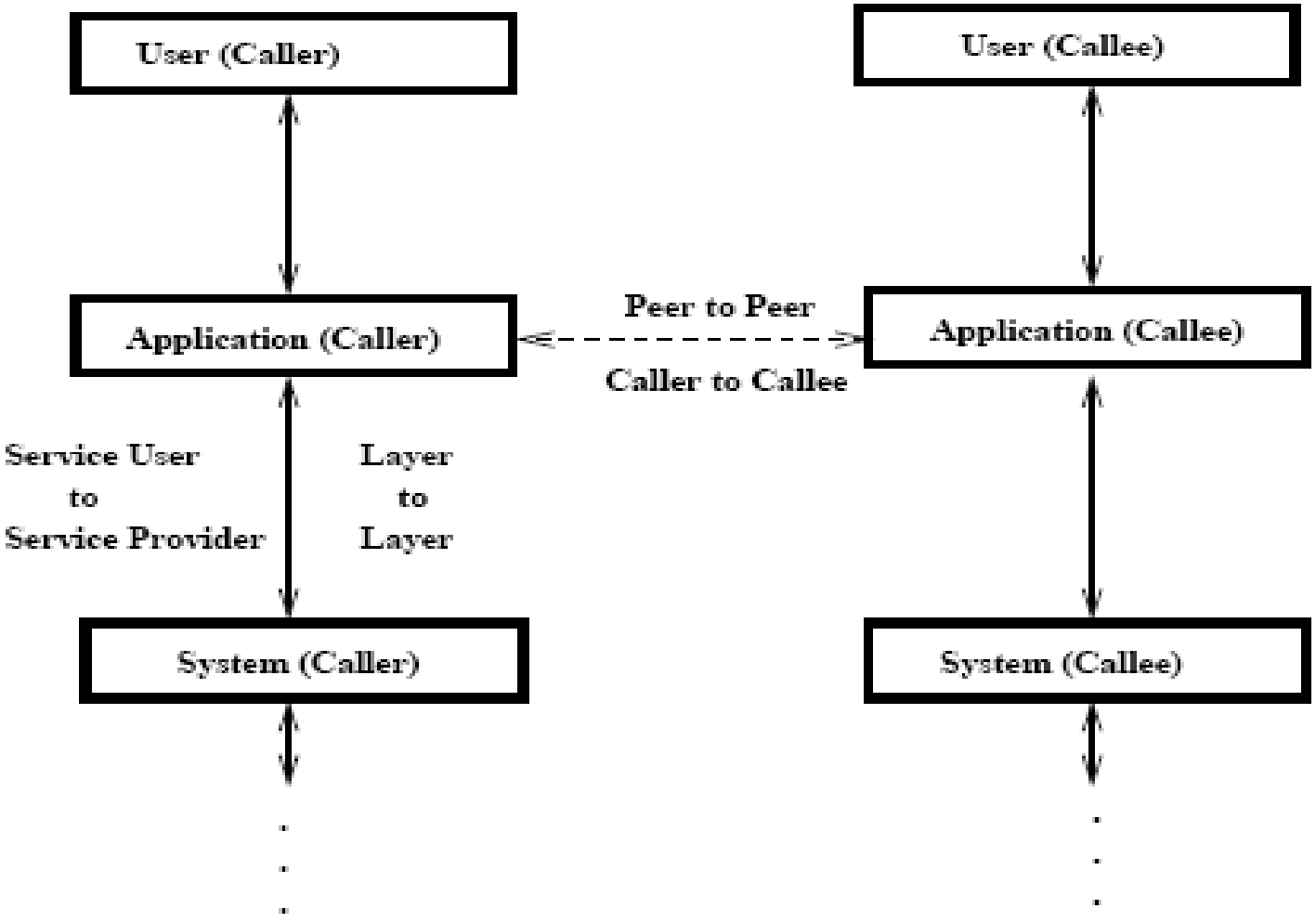
Reverse translation

- Reverse translation is useful for adaptation and media scaling
  - computes from (throughput, packet rate) the (frame-size, frame-rate)
  - reverse translation is not unambiguous
  - One can scale down either the frame size or the frame rate.

Layered Translation (Example)



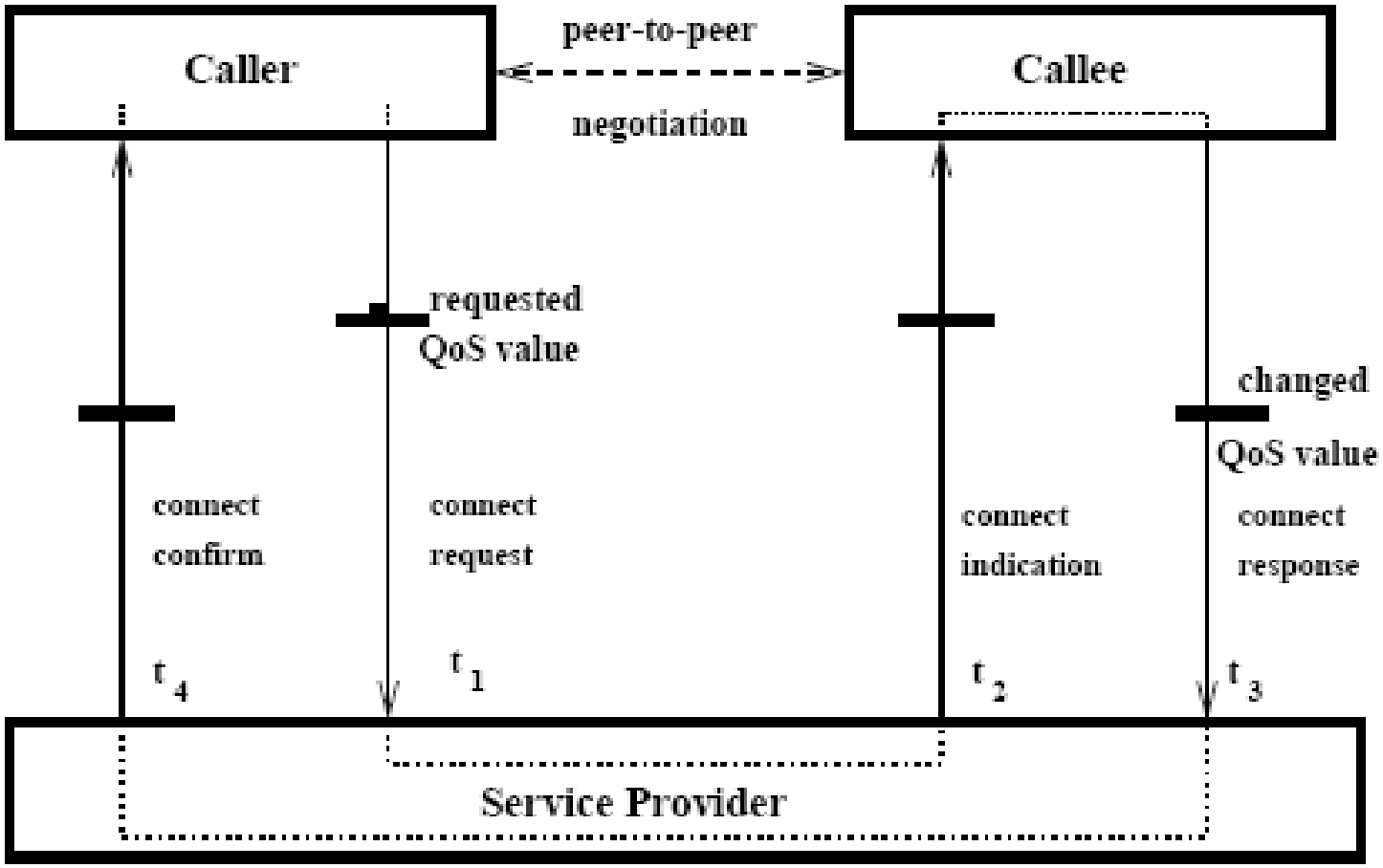
QoS Negotiation



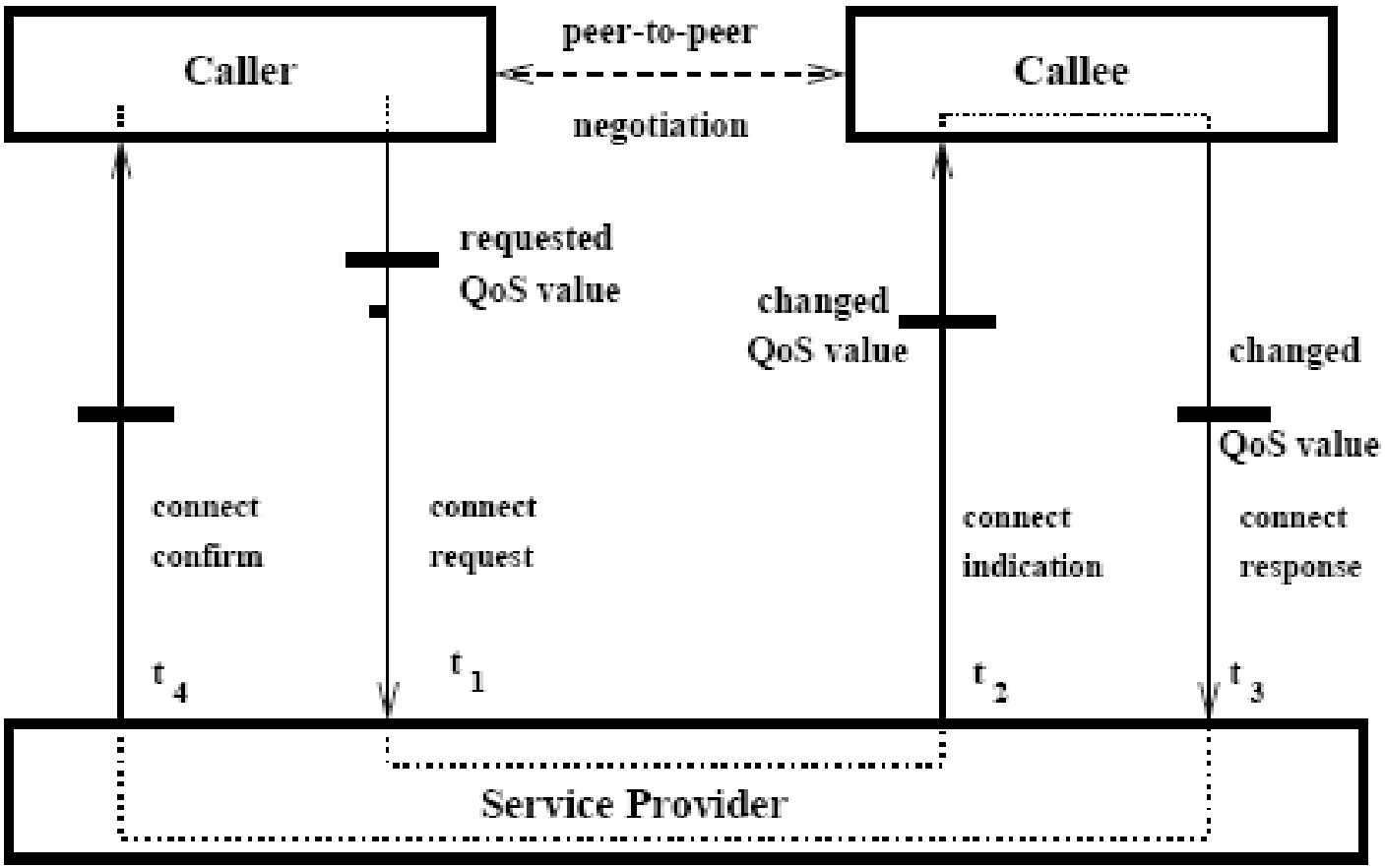
Different Types of Negotiation Protocols

- Bilateral Peer-to-Peer Negotiation
  - Negotiation of QoS parameters between equal peers in the same layer
- Triangular Negotiation
  - Negotiation of QoS parameters between layers
- Triangular Negotiation with Bounded Value

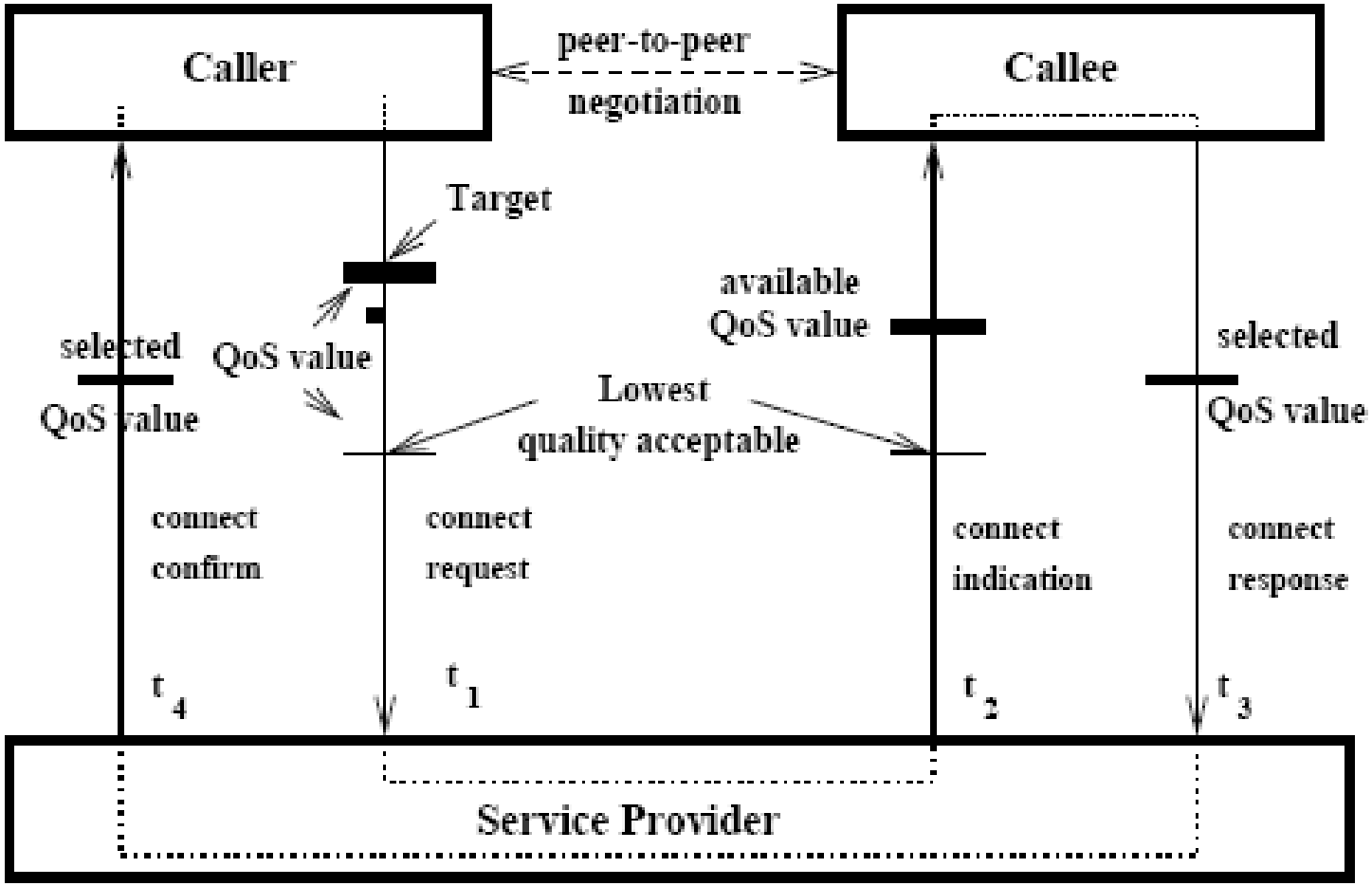
Bilateral QoS Negotiation



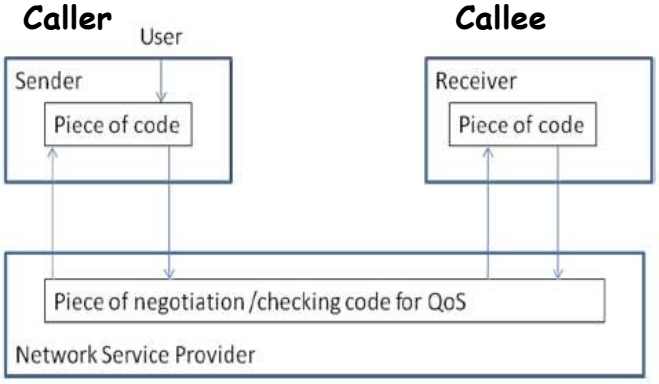
Triangular QoS Negotiation



Triangular Negotiation with Bounded Value



Triangular Negotiation Protocol (Pseudo-Code Example)



Caller Pseudo-Code

```
get-from-user(minQoS, desiredQoS);
send-request-to-service-provider(minQoS, desiredQoS,peer-destination);
wait-for-response-from-service-provider(answer,targetQoS);
if (answer==reject) then let user know about the rejection;
if (answer==accept) && (targetQoS >=minQoS), then negotiateQoS:= targetQoS;
```

Network-Service Provider Pseudo-Code

```
get-from-caller(minQoS,desiredQoS);
//check minQoS and desiredQoS against its own capability (availableQoS),
what QoS the network provider can provide;
if availableQoS < minQoS then
   { send-response-to-caller(reject,0);
     exit();
   }
if availableQoS >=minQoS then
   { desiredQoS:= availableQoS;
     //reserve the resources for available QoS;
     send-request-to-callee(minQoS,desiredQoS,peer-destination);
   }
wait-for-response-from-callee(answer,targetQoS);
if (answer == reject) then
   { //release reserved resources for available QoS;
     send-response-to-caller(reject,0);
   }
if (answer == accept) then
   { //allocate the resources for target QoS;
     send-response-to-caller(accept,targetQoS);
   }
```

Callee Pseudo-Code

```
get-from-network-provider(minQoS,desiredQoS);
// check(minQoS,desiredQoS) against its own resources;
if availableQoS < minQoS then
   { send-response-to-network-provider(reject,0);
     exit();
   }
if availableQoS >=minQoS then
   { targetQoS:=availableQoS;
     send-response-to-network-provider(accept,targetQoS);
   }
```

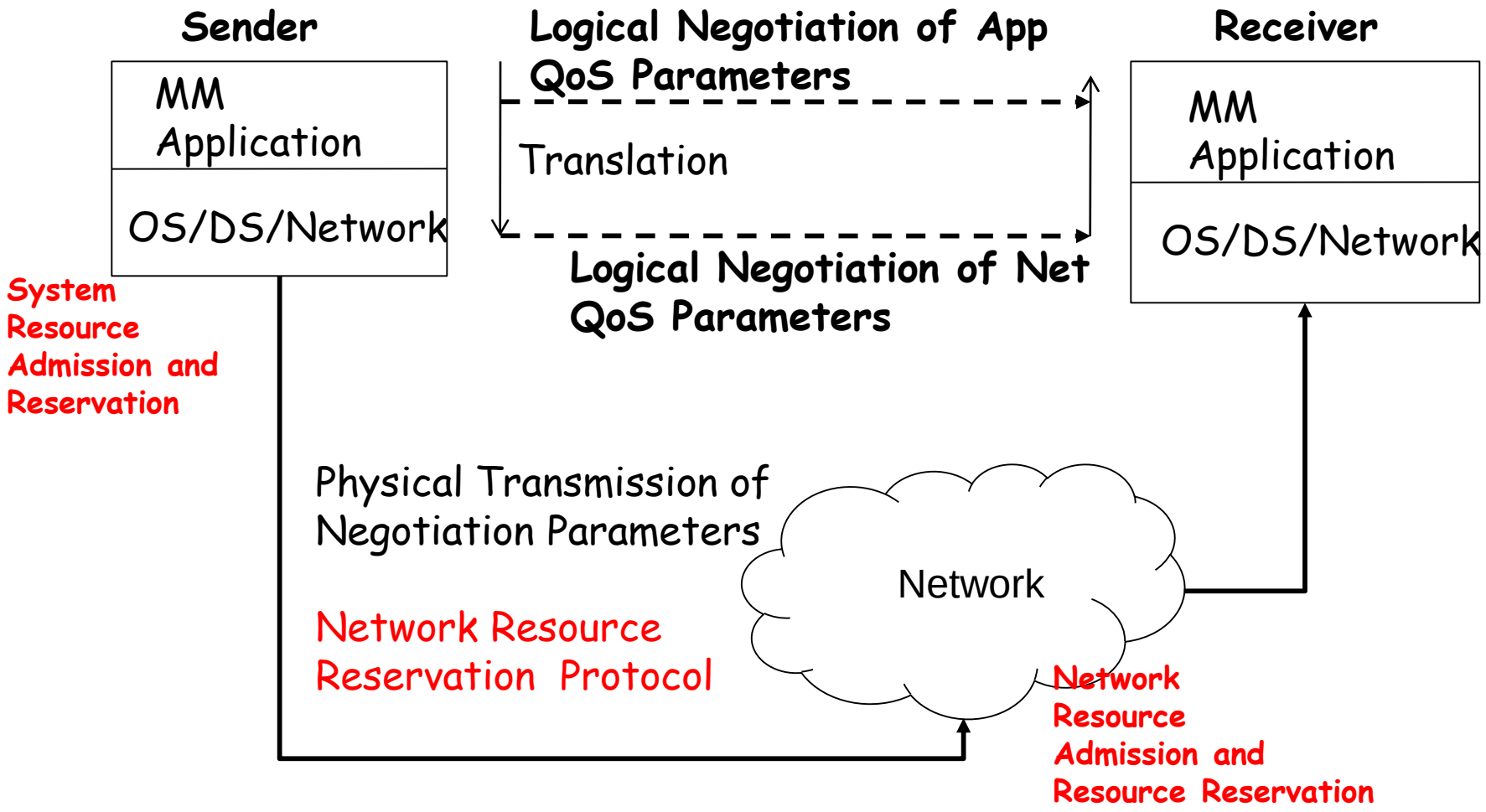
Multimedia Resource Management

- Resource managers with operations and resource management protocols
  - Various operations must be performed by resource managers in order to provide QoS
- Phase 1: Establishment Phase (resource operations)
  - Operations are executed where schedulable units utilizing shared resources must be admitted, reserved and allocated according to QoS requirements
- Phase 2: Enforcement Phase
  - Operations are executed where reservations and allocations must be enforced, and adapted if needed

Phase 1: Resource Preparation Operations

- QoS to Resource Mapping
  - Need translation or profiling (e.g., how much processing CPU cycles, i.e., processing time, it takes to process 320x240 pixel video frame)
- Resource Admission
  - Need admission tests to check availability of shared resources
- Resource Reservation
  - Need reservation mechanisms along the end-to-end path to keep information about reservations
- Resource Allocation

Phase 1: Connection Establishment



Admission Tests

- Task (System) schedulability tests for CPU resources
  - This is done for delay guarantees
- Network Packet schedulability tests for sharing host network interfaces, network switches
  - This is done for network delay and jitter guarantees
- Spatial tests for memory/buffer allocation
  - This is done for delay and reliability guarantees
- Network Link bandwidth tests
  - This is done for network throughput guarantees

Admission Control

- Throughput QoS maps to bandwidth resource.
- Packet-rate and error-rate map to scheduling and buffer resources
- Bandwidth allocation
  - Let b_i be the reserved bandwidth for the ith connection and B_max the maximal bandwidth at the network interface.
  - The admission test is

$$\sum_{i=1,n} b\_i \leq B\_max$$

Bandwidth Allocation

- In an iterative fashion, we consider
  - AllocatedBW_i be the bandwidth already allocated to the ith connection
  - RequestedBW_j be the bandwidth requested by the jth connection
  - Let
    - AvailableBW = B_max - $\sum_{i=1,n}$ AllocatedBW_i where i is not equal to j.
  - The admission control test is
    - RequestedBW_j <= AvailableBW

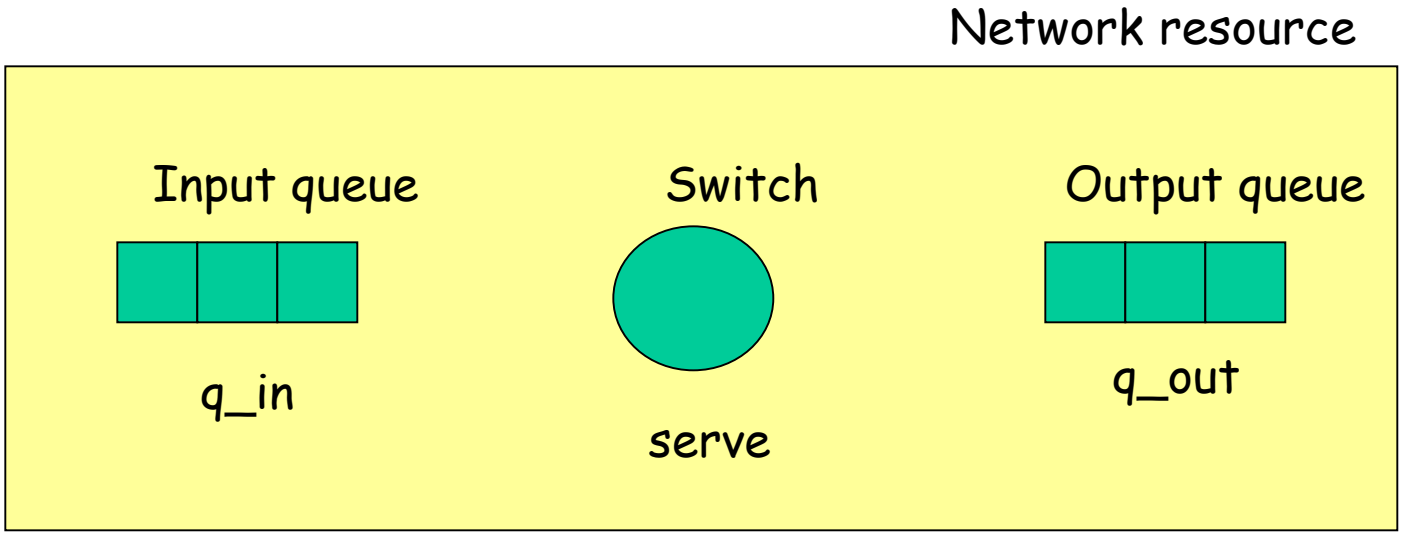
Example - Admission Test

- Consider an ATM host interface
  - B_max = 130Mbps (actual physical bandwidth of OC-3 host interface is 155Mbps, but at the network layer one gets approximately 130Mbps).
  - Let b_1 of virtual circuit (vci1) = 1Mbps, b_2 of virtual circuit 2 (vci2) = 64kbps, b_3 of vci3 = 10Mbps
  - the admission test is 11.064Mbps < 130Mbps and all three connections are admitted.

Admission Control

- At network nodes (e.g. switches) we need to make scheduling decisions when admitting new streams
  - need to make schedulability tests available
  - Note that scheduling algorithms running on intermediate network noded are always non-preemptive.
- To schedule a packet through a network node on time, consider
  - e_i is the processing time of packet i in microseconds
  - Then the scheduling admission test is
    - $\sum_{i=1,n} e\_i \leq 1$, where I are all packets needed to be scheduled within the considered second.

Scheduling Scenario at Network Node



e_i = q_in + serve + q_out

Network Admission Control

- The processing time
  - e_i = q_in + serve + q_out at a network node consists of
    - q_in - the queueing delay of a connection packet in the input queue
    - q_out - the queueing delay of a connection packet in the output queue
    - serve - service time (equivalent to the switching time in a switch resource) of packet i.
  - The serve time at a node is often constant due to hardware implementation. q_in, q_out times are variable times and depend on queue occupancy
    - $q = N/\lambda$ - Little's theorem
    - N is the occupancy of the queue in number of messages and λ is the arrival rate in messages per second to the queue.

Resource Reservation and Allocation

- Types of reservations
  - Pessimistic approach – Worst case reservation of resources
  - Optimistic approach – Average case reservation of resources
  - Also sender vs. receiver oriented reservation protocol
- To implement resource reservation we need:
  - Resource table
    - to capture information about managed table (e.g., process management PID table)
  - Reservation table
    - to capture reservation information
  - Reservation function
    - to map QoS to resources and operate over reservation table

Network Resource Reservation

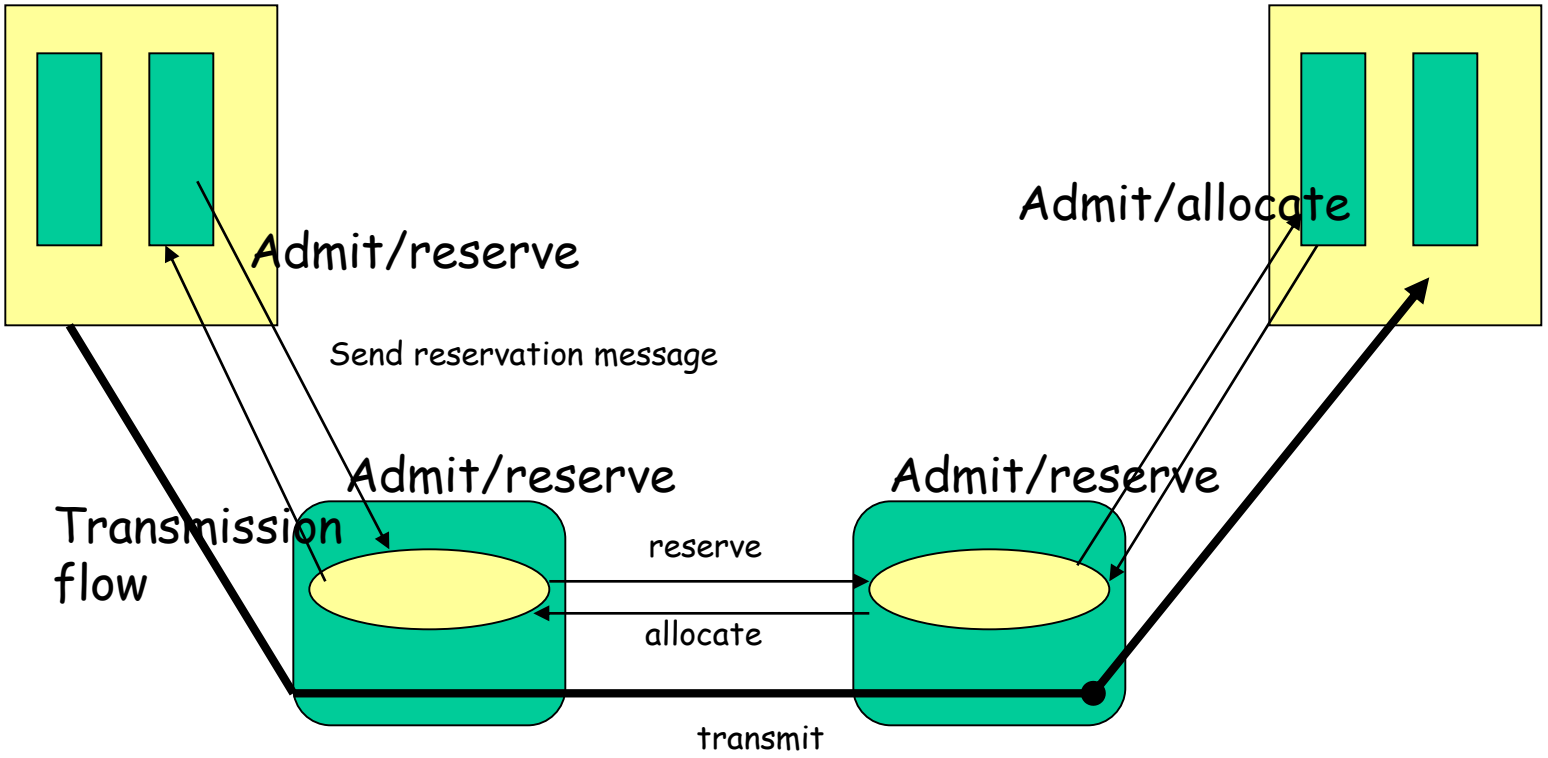
- Bandwidth reservation
  - Pessimistic
    - maximal bandwidth allocation
      - $M\_a = \max\_i(M\_ai)$
      - $B\_n = M\_n * (\lceil M\_a/M\_n \rceil) * R\_a$
  - Optimistic
    - average bandwidth allocation
      - $M\_a = 1/n \sum M\_ai$
      - $B\_n = M\_n * (\lceil M\_a/M\_n \rceil) * R\_a$

$i=1,n$

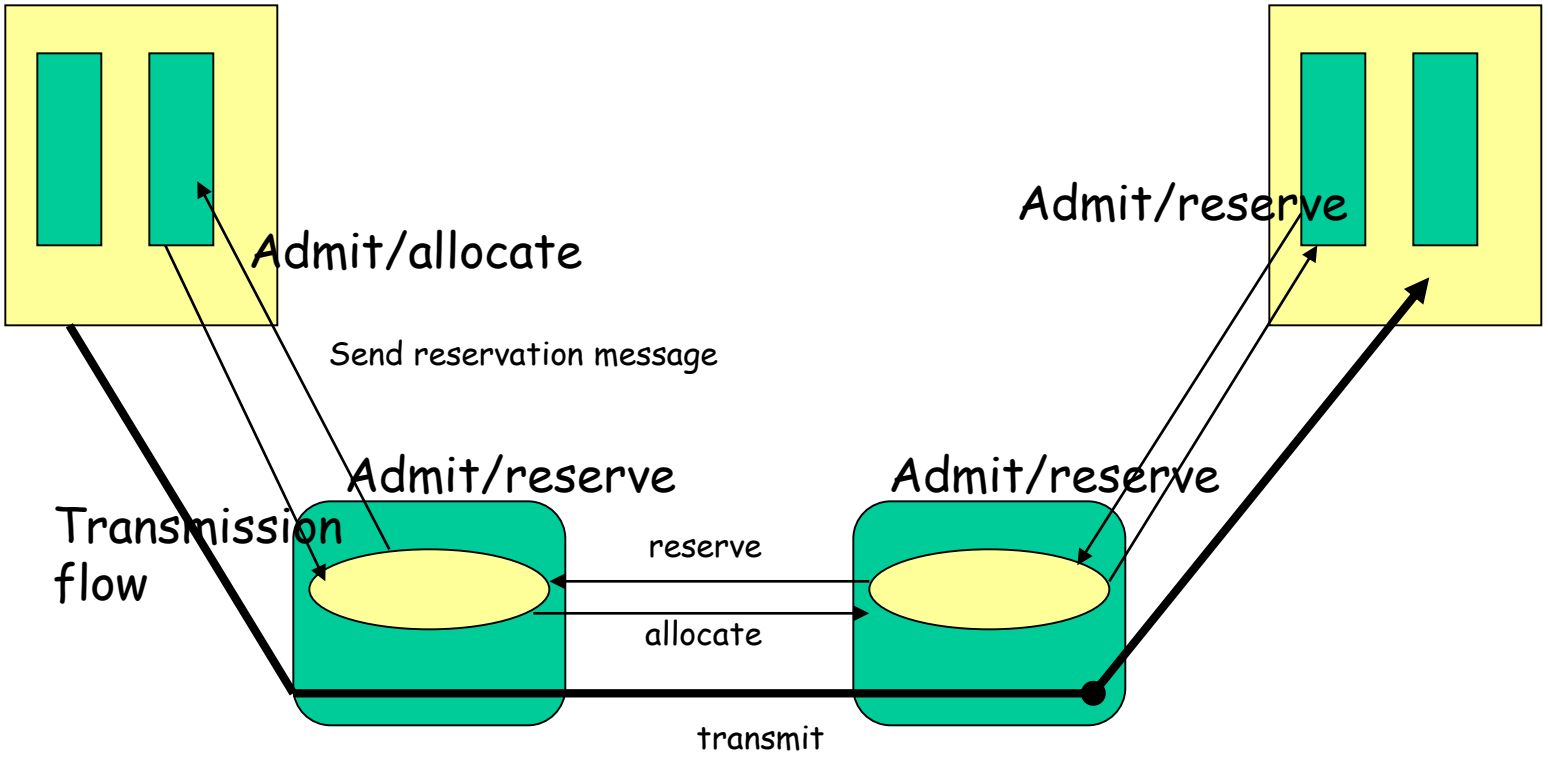
Reservation/Allocation protocols

- Sender-oriented vs. Receiver oriented protocol
  - Sender oriented reservation
    - sender transmits a QoS specification to the targets
    - intermediate routers and targets may adjust the QoS spec wrt available resources before the QoS specification is transmitted back to the sender.
  - Receiver oriented reservation
    - receiver describes resource requirements in a QoS specification and sends it to the sender in a "reservation" message.
    - Assumes that sender has sent a path message before, providing information about outgoing data.

Sender oriented reservation protocol



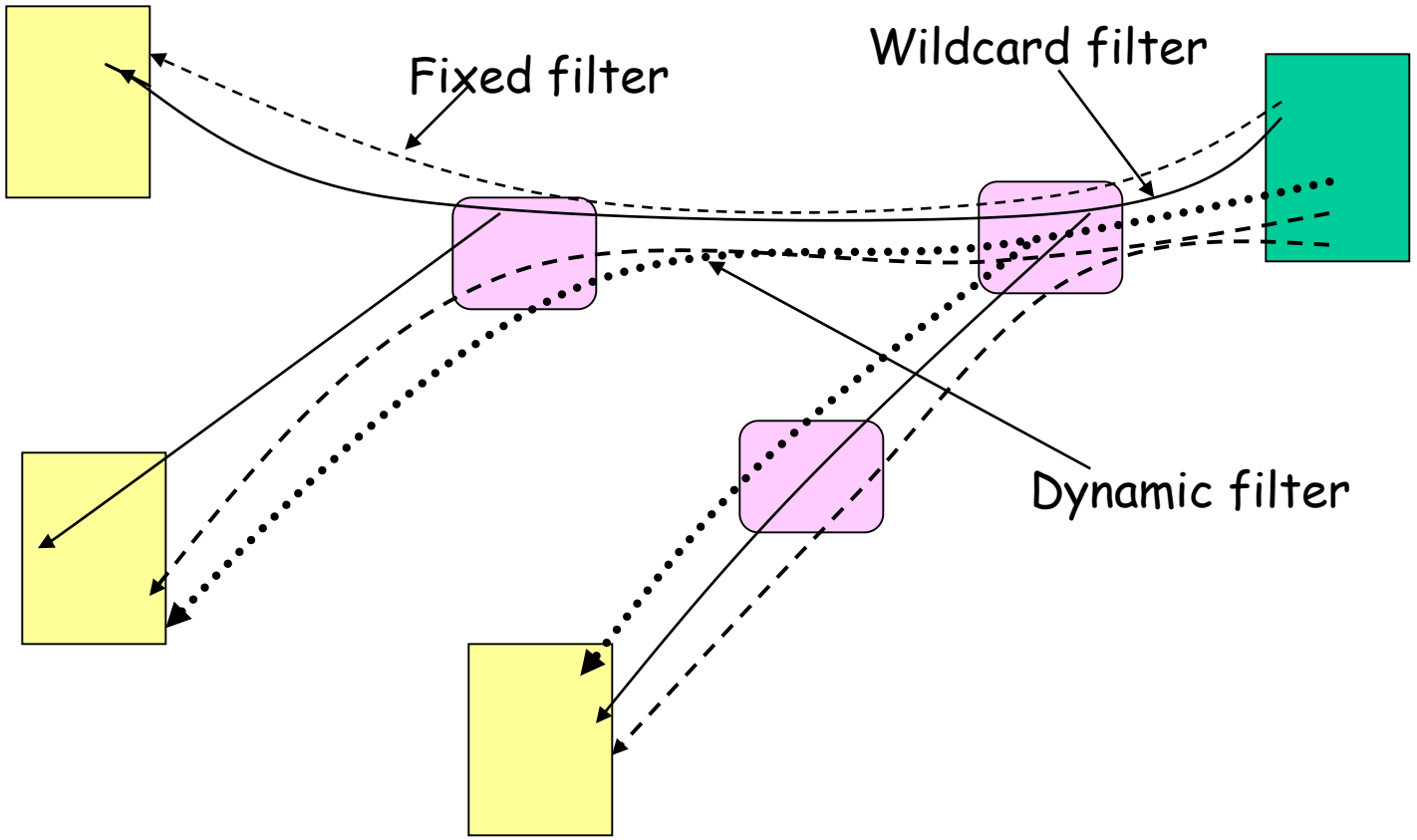
Receiver oriented reservation protocol



Reservation Styles

- Represents the creation of a path reservation and time when senders and receivers perform QoS negotiation and resource reservation
- Sender based reservation
  - single reservation or multicast reservation
- The IETF standard defined three types of reservation styles (RSVP) for receiver oriented reservation
  - Wildcard Style
    - allows receiver to create a single reservation along each link shared among all senders for the given session
  - Fixed Filter Style
    - allows each receiver to create a single reservation from a particular sender whose data packets it wants to receive
  - Dynamic Filter Style
    - allows each receiver to create N reservations to carry flows from up to N different senders. This style allows the receiver to do channel switching

Reservation Styles



End-to-end Error Control

- Many MM communication systems offer unreliable transport
  - UDP/IP protocol was used for transmitting digital audio over the Internet
  - Tenet protocol suite's transport protocols provide unreliable but timely delivery for MM communication
- Reliability needed in multimedia communication
  - Decompression
    - many compression schemes cannot tolerate loss
  - Human perception
    - loss of audio detected very quickly
  - Data Integrity
    - recording application - cannot recover from error in first recording

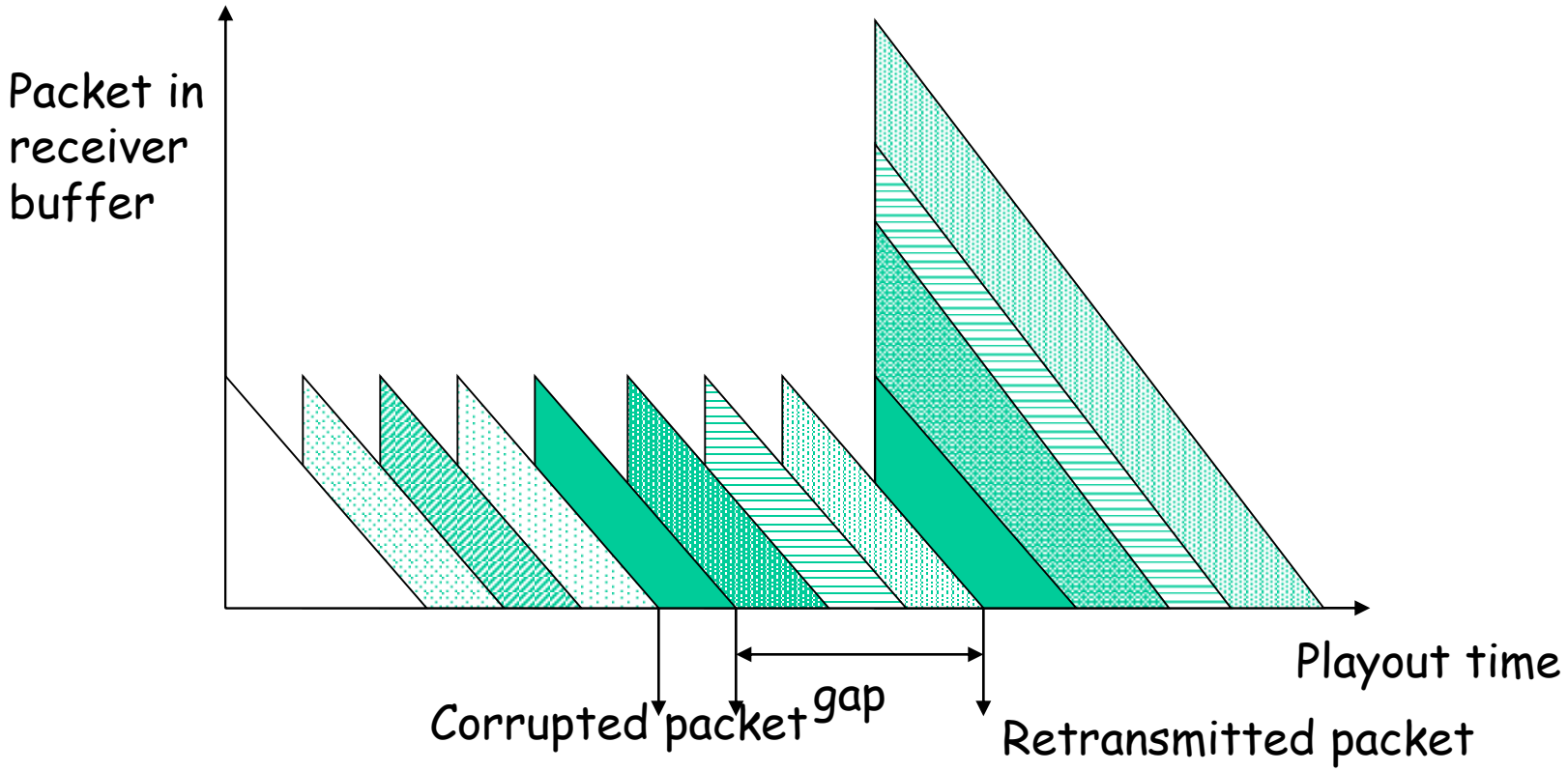
End-to-end Error Control

- Error Detection
  - Traditional mechanisms
    - checksumming, PDU sequencing.
    - Allow detection of data corruption, loss etc. at lower level
  - MM needs
    - byte error detection at the application PDU level
    - time detection - late PDU is useless
- Error Correction
  - Traditional mechanisms
    - retransmission using acknowledgement schemes and/or window based flow control
    - amount of data to be stored at sender too large
    - sender may be forced to suspend transmission (window based)
    - retransmitted data may be too late
    - not designed for multicasting

MM Error Correction Algorithms

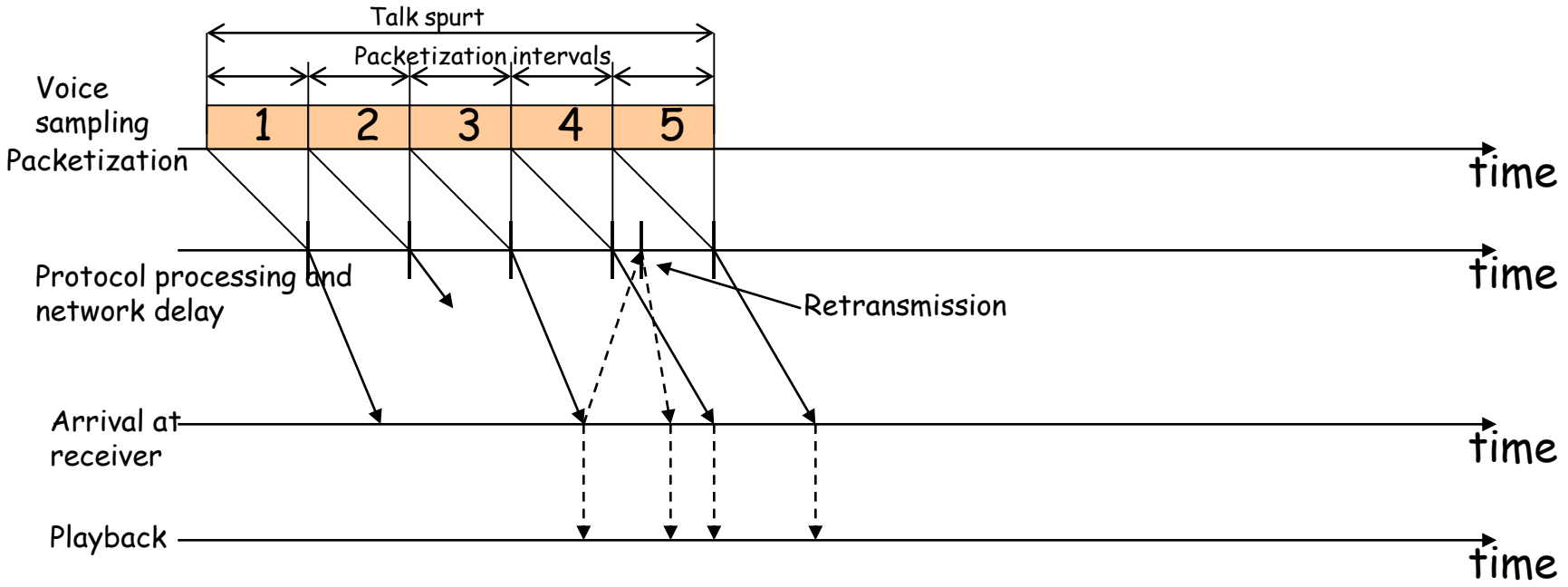
- Go-back-N retransmission
  - If PDU j is lost, sender will go back to j and restart transmission from j (if j<=n).
  - Problems - gap introduction, violation of throughput guarantees
- Selective retransmission
  - receiver sends negative ack if PDU j<=n is lost. Sender retransmits only those PDUs reported missing
  - receiver has to store successfully delivered PDU until all previous Pdus have been delivered successfully.
- Partially reliable streams
  - limits number of packets that will be retransmitted in a time interval.
- Forward error correction
  - sender sends additional information that the receiver can locate and correct bits or bit sequences. Requires H/w support
- Priority channel coding
  - separate medium into multiple data streams with diff priorities
- Slack automatic repeat request.
  - Retransmission of lost voice packets in high speed LANs.

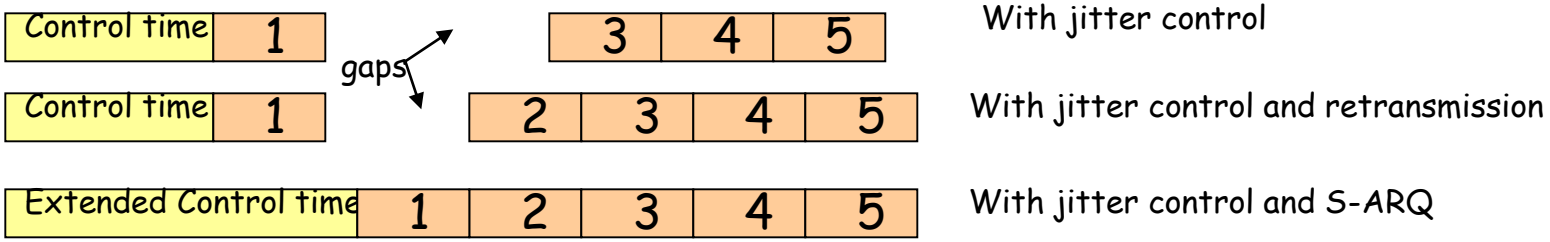
Go-back-N retransmission scheme



Gap problem in Go-back-N transmission scheme

Slack Automatic Repeat Request





Monitoring

- Network Management
  - consists of monitoring agents at every intermediate node that:
    - gather information and store it in MIB (management information base)
    - exchange information among each other
    - convey information to other resource managers
  - Standard network management (administration) protocols
    - CMIS/CMIP (common management information services and protocols) for wide-area networks
    - SNMP (Simple network management protocol) - IP based
  - Monitoring for MM transmission - possible QoS violations
    - monitoring variables should be optional
    - must be able to turn monitoring on and off.

Adaptation Schemes

- Network Adaptation
  - For network to adapt, we need efficient routing and resource allocation.
  - Load balancing scheme needs services such as
    - routing, performance monitoring (detecting load changes), dynamic re-routing (changing the route), load balancing control (making a decision to re-route)
- Source Adaptation
  - Feedback from the network to the source needed or feedback from other peer
  - adaptive rate control
  - traffic shaping
  - hierarchical coding

Adaptive Applications

- Essential Idea
  - Instead of requiring the network to make strict performance guarantees, the application asks for loose performance guarantees and the application changes its behavior to accommodate to how the network is currently delivering data.
- Example application - VAT (voice conferencing system)
  - experimental use over Internet
  - Challenge of supporting a phone conversation - maintaining correct spacing between samples
  - To avoid garbled output due to variation in transit times through network, adaptive applications buffer the voice samples at the receiver. The inter-sample timing is recreated by the receiver before the samples are played.
  - Vat recreates timing by having sender time-stamp each sample. Receiver used time-stamps to restore the inter-sample timing.

Adaptive Applications (cont.)

- Make receiving buffers large enough
  - samples delayed in the network will arrive in time to be played.
    - If the network delay for a sample varies between 50-100ms, receiver buffer must store up to 50ms worth of data.
    - Voice samples that arrive in less than 100ms are buffered until 100ms have elapsed since they were sent and then played.
  - Choosing a Playback point - time at which voice samples are played back is hard
    - vat changes the playback point during conversation in response to the network delays it observes.
    - If all samples are arriving late, increase playback point
    - If all samples are arriving well before they are played, move the playback point back.

MM Communication Protocols -Heidelberg Protocol Stack

Heidelberg Continuous Media Realm

Heidelberg Resource Administration Technique

Stream Protocol (ST-II)

Connection-oriented, guaranteed service

ST Control Message Protocol, ST

Resource reservation, flow specification with QoS

MM Communication Protocols -Tenet Protocol Stack

Real-time Message Transport Protocol (RMTP)

connection-oriented, performance guaranteed
unreliable message delivery
Flow control: rate control

Continuous Media Transport Protocol (CMTP)

transport of periodic network traffic with performance guarantees

Real-time Channel Administration Protocol (RCAP)

resource reservation, admission, QoS handling

Real-time Channel Internet Protocol (RTIP)

connection oriented, performance guaranteed
unreliable delivery of packets

MM Communication Protocols - XTP Protocol Stack

Services

connection, transaction, unacknowledged datagram, acknowledged datagram, isochronous stream, bulk data

Users (contexts)

create an association

Flow control

sliding window or rate based flow control
window-based flow control uses a combined mechanism between cumulative acknowledgement and selective acknowledgement

Error control

Mechanisms and policies - can be customized

Connection-oriented transmission

Good for ATM - fast connection establishment
Problems: large headers, software implementation slow

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?

Streaming Stored Multimedia

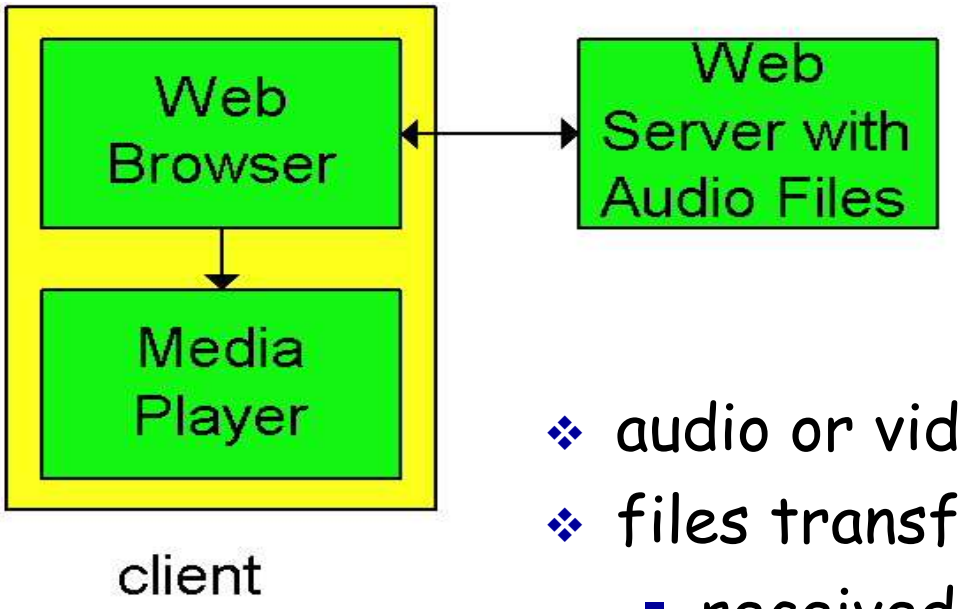
application-level streaming techniques for making the best out of best effort service:

- client-side buffering
- use of UDP versus TCP
- multiple encodings of multimedia

Media Player

- jitter removal
- decompression
- error concealment
- graphical user interface w/ controls for interactivity

Internet multimedia: simplest approach

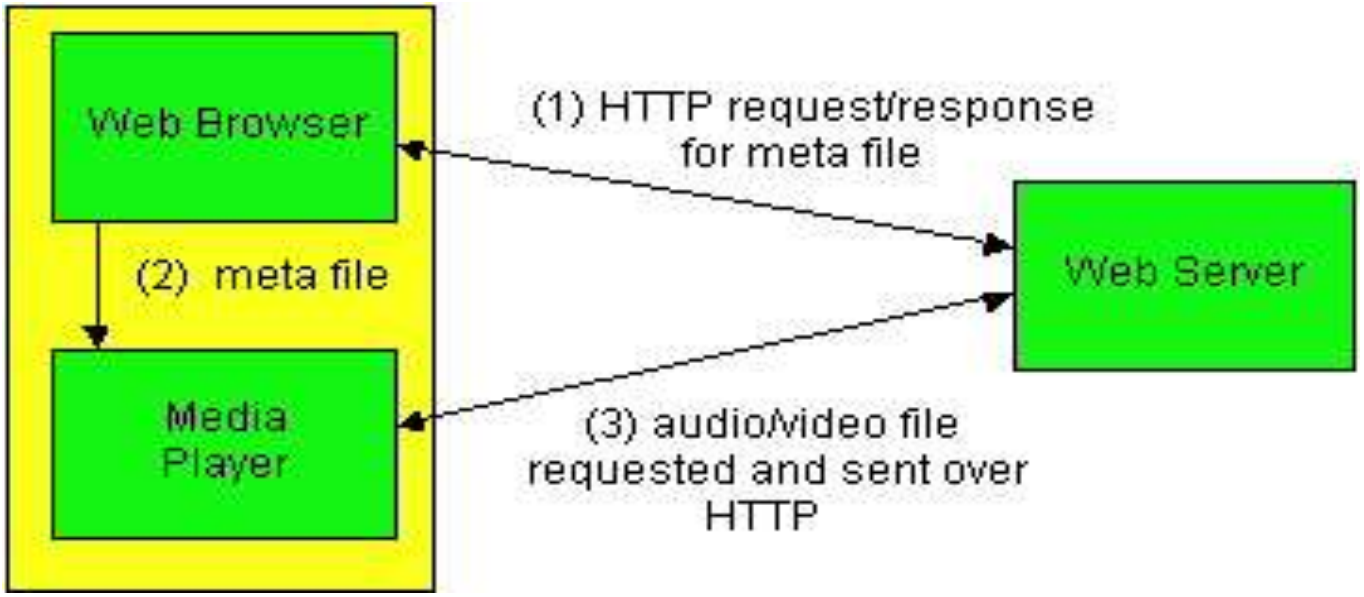


- audio or video stored in file
- files transferred as HTTP object
  - received in entirety at client
  - then passed to player

audio, video not streamed:

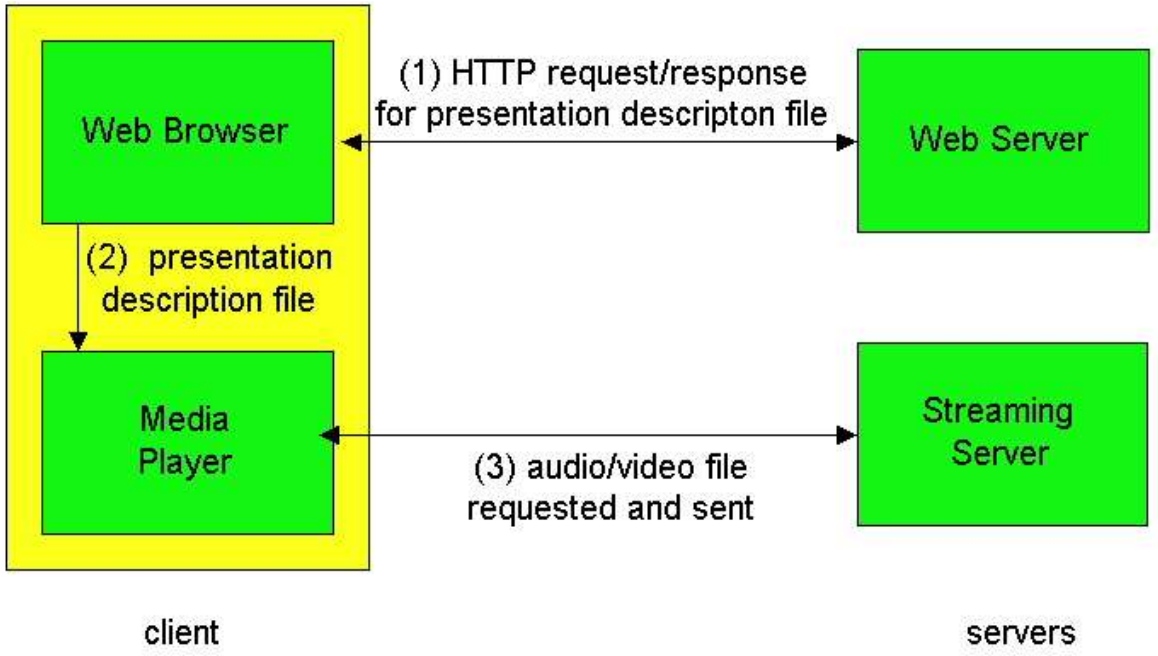
- no, "pipelining," long delays until playout!

Internet multimedia: streaming approach



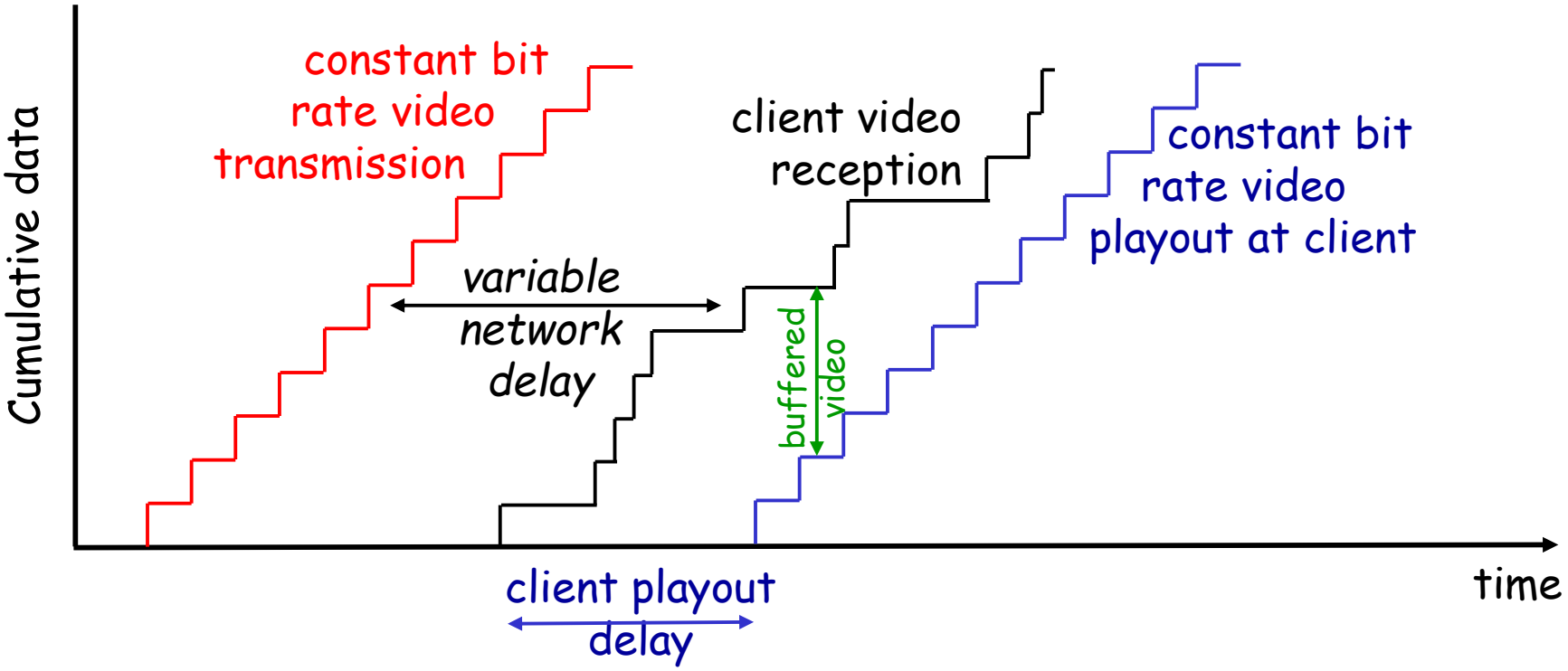
- browser GETs **metafile**
- browser launches player, passing metafile
- player contacts server
- server streams audio/video to player

Streaming from a streaming server



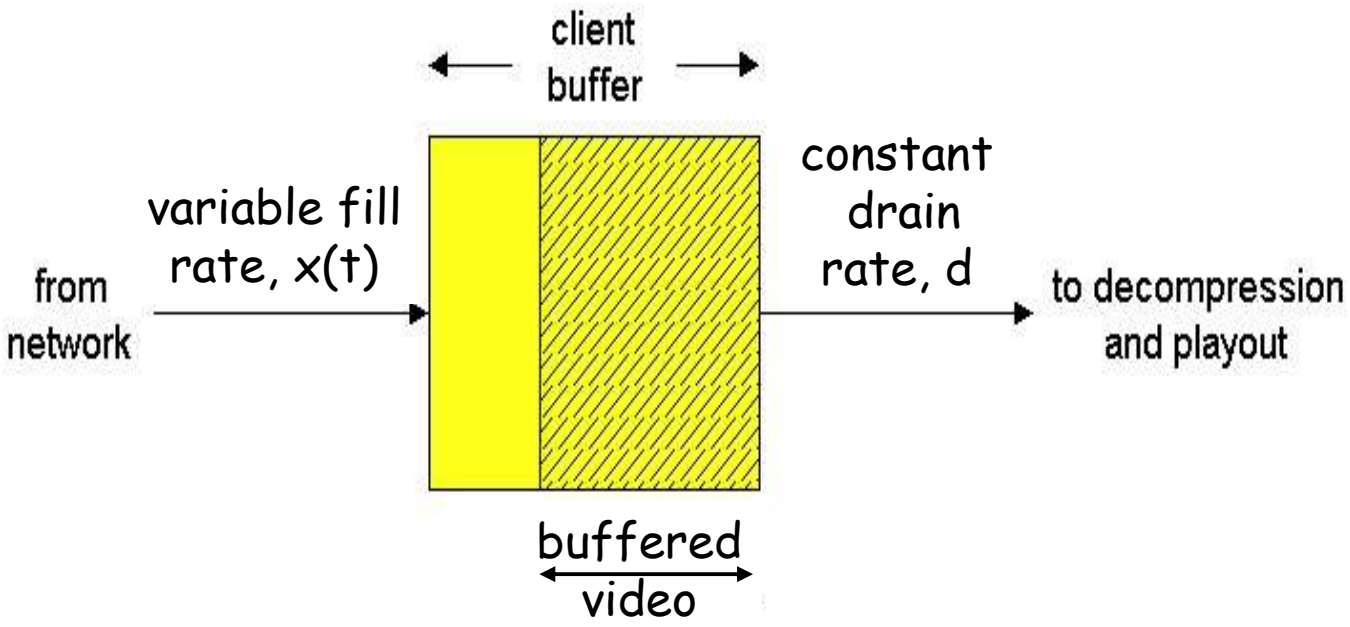
- allows for non-HTTP protocol between server, media player
- UDP or TCP for step (3), more shortly

Streaming Multimedia: Client Buffering



- client-side buffering, playout delay compensate for network-added delay, delay jitter

Streaming Multimedia: Client Buffering



- client-side buffering, playout delay compensate for network-added delay, delay jitter

Streaming Multimedia: UDP or TCP?

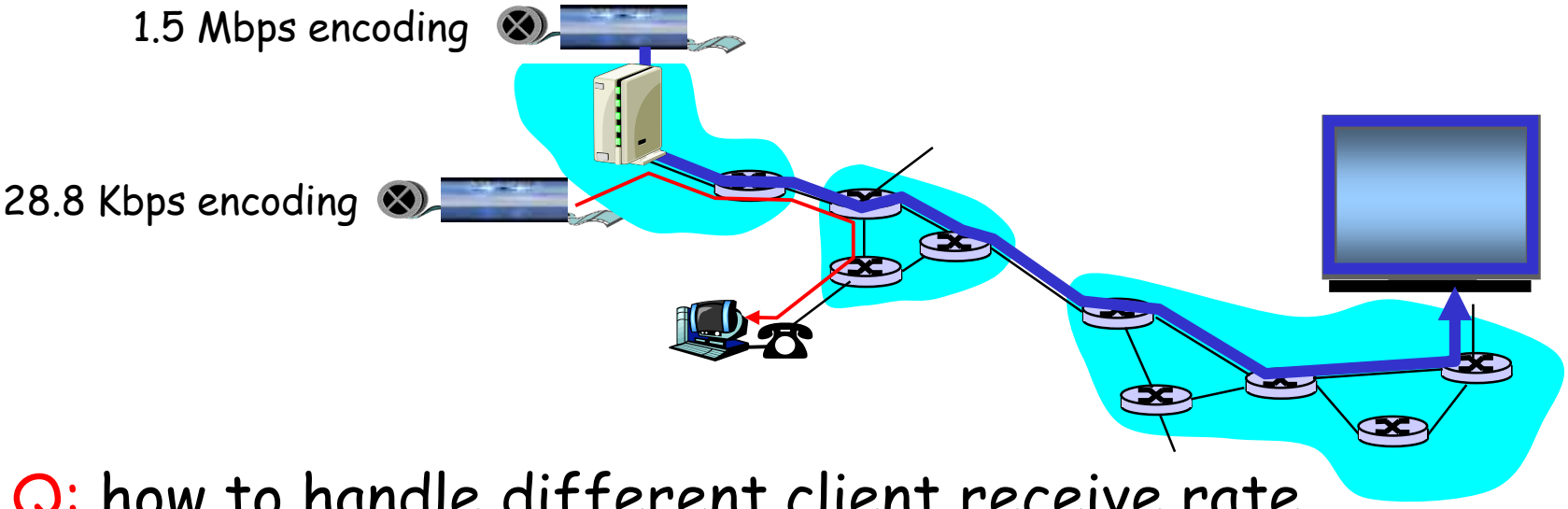
UDP

- server sends at rate appropriate for client (oblivious to network congestion !)
  - often send rate = encoding rate = constant rate
  - then, fill rate = constant rate - packet loss
- short playout delay (2-5 seconds) to remove network jitter
- error recover: time permitting

TCP

- send at maximum possible rate under TCP
- fill rate fluctuates due to TCP congestion control
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

Streaming Multimedia: client rate(s)



Q: how to handle different client receive rate capabilities?

- 28.8 Kbps dialup
- 100 Mbps Ethernet

A: server stores, transmits multiple copies of video, encoded at different rates

User Control of Streaming Media: RTSP

HTTP

- does not target multimedia content
- no commands for fast forward, etc.

RTSP: RFC 2326

- client-server application layer protocol
- user control: rewind, fast forward, pause, resume, repositioning, etc…

What it doesn't do:

- doesn't define how audio/video is encapsulated for streaming over network
- doesn't restrict how streamed media is transported (UDP or TCP possible)
- doesn't specify how media player buffers audio/video

RTSP: out of band control

FTP uses an "out-of-band" control channel:

- file transferred over one TCP connection.
- control info (directory changes, file deletion, rename) sent over separate TCP connection
- "out-of-band", "in-band" channels use different port numbers

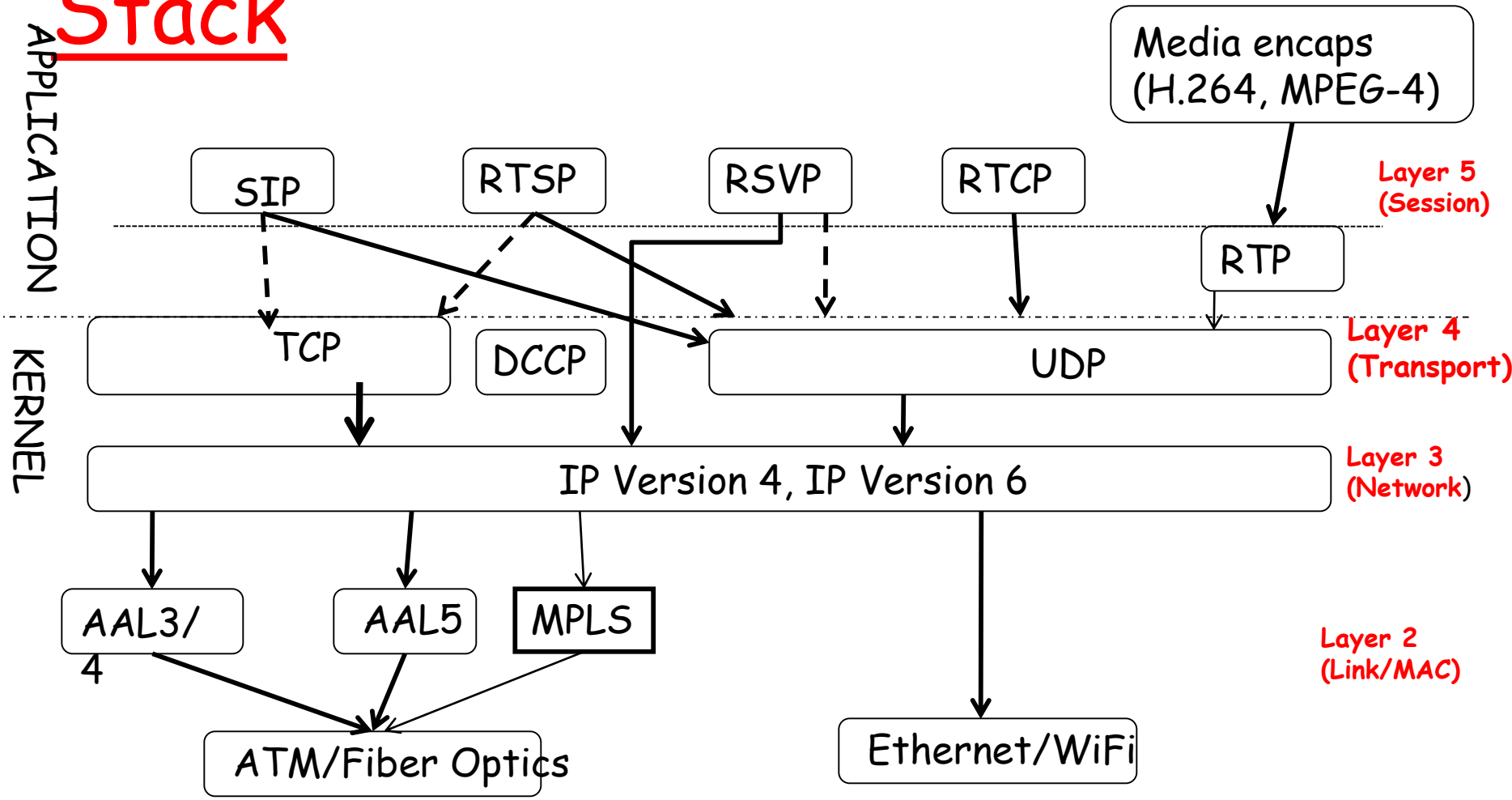
RTSP messages also sent out-of-band:

- RTSP control messages use different port numbers than media stream: out-of-band.
  - port 554
- media stream is considered "in-band".

Outline

- Multimedia Protocols – Standards
  - RTP/UDP/IP – Transmission Protocol
  - RTCP Control/Negotiation Protocol to RTP
  - RTSP – Control VOD Negotiation Protocol

Internet Multimedia Protocol Stack



Service Requirements for Real-time Flows (Voice/Video)

- Sequencing
- Intra-media synchronization
- Inter-media synchronization
- Payload identification
- Frame indication

TCP - Transmission Control Protocol

- TCP provides
  - reliable, serial communication path between processes exchanging a full-duplex stream of bytes.
  - Full-duplex TCP connections
  - sequential delivery (no reordering required);
  - reliable delivery, achieved through retransmission on timeouts and positive acknowledgement on receipt of information.
  - Flow control, based on window technique
- Not suitable for multimedia transmission
  - led to TCP enhancements

Techniques for Going Faster

- Improve protocol implementation
  - Memory management - reduce copying
  - Interrupt handling - clocked interrupts
- Better Lookup Techniques
  - IP must find a route to be able to send an IP packet
    - use caches of frequently used information, find lookup algorithms
  - Caches - maximize hit rate, minimize search and maintenance (conflict)
    - most effective - small caches
    - packets travel in packet trains
  - Lookup Algorithms
    - for transaction processing; hashing using open chaining, where head of each hashed link list keeps a cache of the last accessed control block.

Reducing or Eliminating Checksum Cost

- Computing checksum
  - requires that each byte in the packet be read and added into sum
  - In RISC processors, two instructions per cycle are possible. Include checksum into instructions

```
\load  [%r0],
% r2
\store %r2,
[%r1]
```

```
\load  [%r0],  %r2
\add   %r5, %r2, %r5
\add   %r5, #0, %r5
\store %r2, [%r1]
```

  - ...ld leave out the checksu...
  - Move checksum to the end of t...
  - trailing checksums, which results in...ord being faster, but it doesn't affect receiver.

Prediction

- TCP has many features
  - retransmission, window sizes, urgent data, however these features are expensive to implement.
- TCP behavior is highly predictable
  - one can take advantage by optimizing the frequent path through the TCP code at the sender and receiver.
- Algorithm for TCP receivers
  - header prediction, looks for segment that fits the profile of the segment the receiver expects to receive next.

Sequence Numbers

- Sequence Numbers
  - High delay-bandwidth product has an implication on the TCP window size and sequence space. TCP window size is 64 Kbytes
  - we need possibility to negotiate the window size.
  - Wrap-around counters to put in sequence numbers
- Example
  - In case of 10Mbps
    - IP packet lifetime was designed with 120 seconds and sequence space of 32 bits
    - takes 1700 seconds to send 2^31 bytes with this throughput
  - In case of Gigabits/sec
    - takes 17 seconds to send 2^31 bytes

Flow and Congestion Control

- High delay-bandwidth product causes
  - long time to tell sender to slow down
  - E.g. New York to LA, TCP continues to send packets for about 30ms before it hears request from LA receiver.
- Slow-start algorithm
  - flow and congestion control mechanism
- Probing algorithm
  - requires sender to keep congestion window which is the estimate of how much traffic the network can actually take.
  - Congestion window is managed using 2 part algorithm
    - Sender sends exponentially until segment gets lost
    - Sender sends exponentially up to half of previous window, then the window grows linearly.

User Datagram Protocol

- UDP is an extension of IP
  - supports multiplexing of datagrams
  - supports checksumming
  - Higher level protocols using UDP must provide:
    - retransmission
    - packetization
    - reassembly
    - flow control
    - congestion avoidance
  - UDP by itself is not suitable for MM transmission, but many MM protocols reside on top of UDP.
    - Provides to some degree the real-time transport property.

Internet Services and Protocols

- Internet protocol changed to provide integrated services (differential services)
  - best-effort service, real-time service and controlled link sharing.
- IP provides unreliable delivery of datagrams in a point-to-point fashion.
- IP provides types of services (TOS) which can be used for indication of service quality. TOS specifies:
  - precedence relation
  - services such as minimize delay, maximize throughput, maximize reliability, minimize monetary cost.

Addressing and Routing

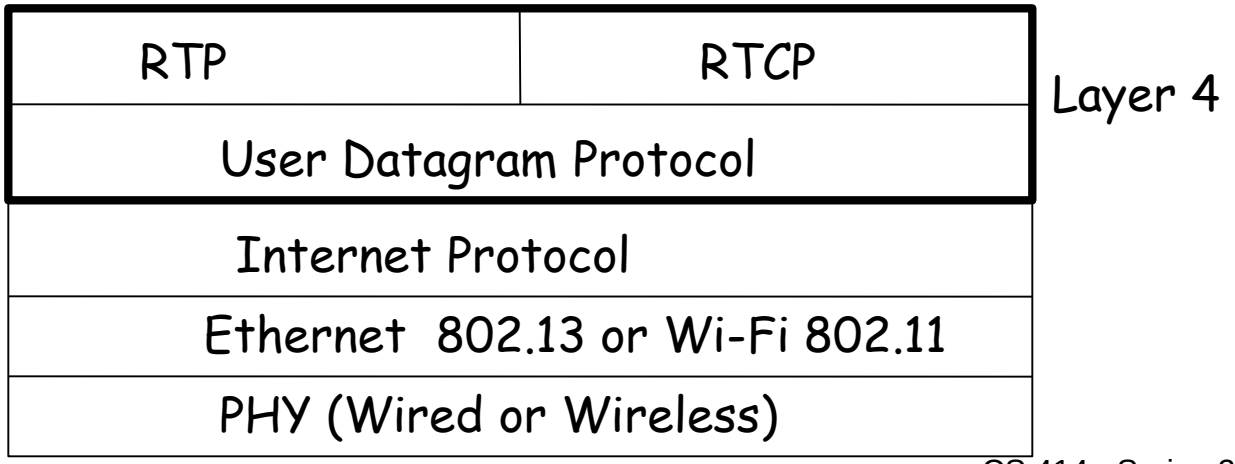
- Five classes of addresses:
  - Class A
    - 24 bits for host addressing and 7 bits for network
  - Class B
    - 16 bits for host addressing and 14 bits for network
  - Class C
    - 8 bits for host addressing and 21 bits for network
  - Class D
    - Multicast Address (multicast tree)
  - Class E
    - for future extensions

Addressing and Routing

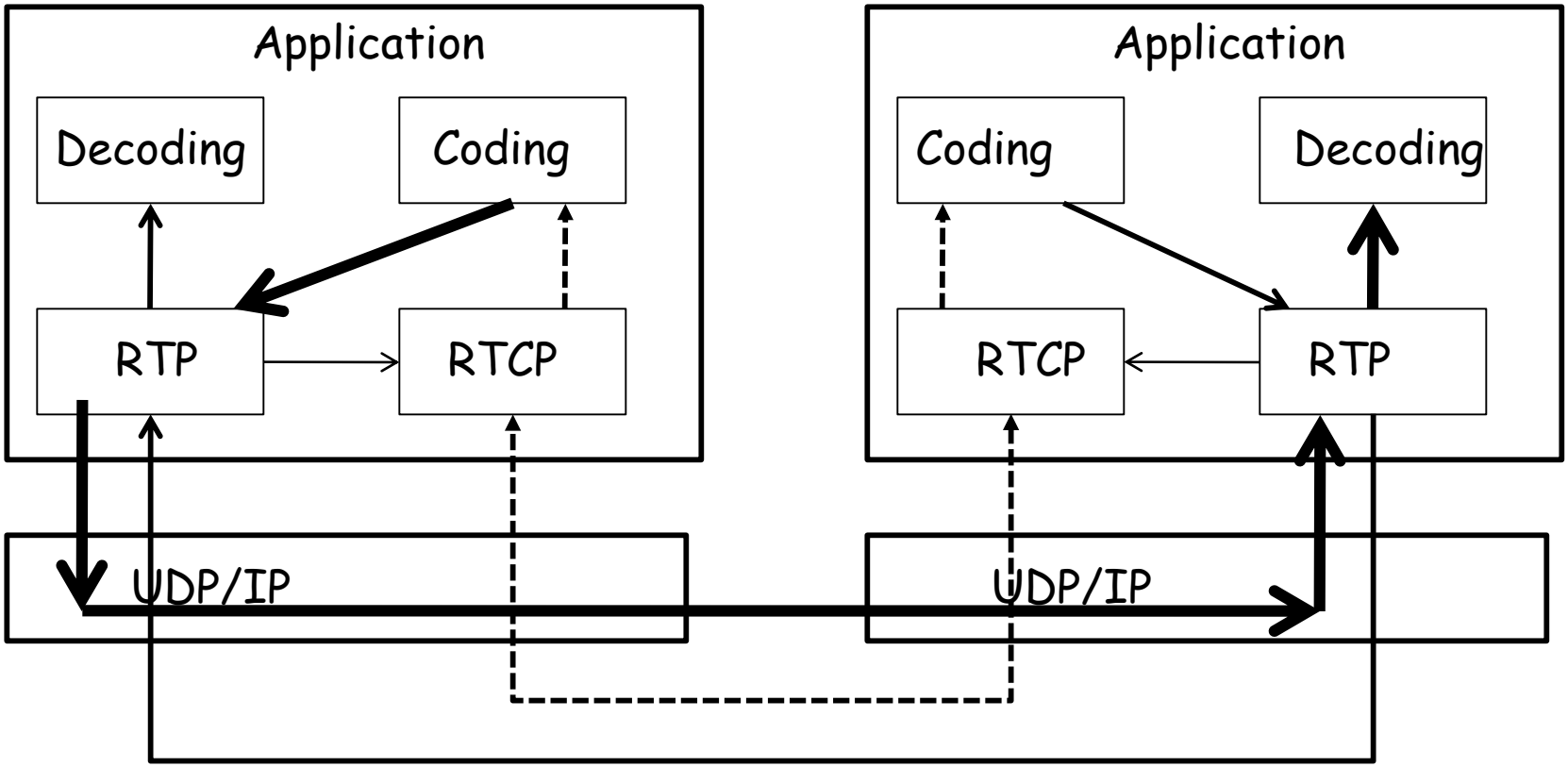
- Interconnectivity of IP and underlying protocols
  - via binding of IP addresses to lower layer network addresses.
  - E.g. ARP (address resolution protocol) maps IP addresses to 48 bit Ethernet addresses.
- Routing - IP uses
  - Interior Gateway Protocol within autonomous systems (Open Shortest Path First)
  - Exterior Gateway Protocol or Border Gateway Protocol among autonomous systems
- For MM communication
  - we will need QoS routing algorithms with new protocols

Real-time Transmission Protocol (RTP)

- RTP provides end-to-end transport functions suitable for real-time audio/video applications over multicast and unicast network services
- RTP companion protocol – Real-time Transport Control Protocol (RTCP)



Relation between RTP and RTCP



RTCP: Control and Management

- Out-of-band control information for RTP flow.
  - Monitors QoS for RTP in the delivery and packaging of multimedia data
  - Used periodically to transmit control packets to participants in a streaming multimedia session.
  - Provides feedback on the quality of service being provided by RTP.
  - Gathers statistics on media connection
    - Bytes sent, packets sent, lost packets, jitter, feedback and round trip delay.
    - Application may use this information to increase the quality of service, perhaps by limiting flow or using a different codec.

RTCP Functions

- There are several type of RTCP packets:
  - Sender report packet,
  - Receiver report packet,
  - Source Description RTCP Packet,
  - Goodbye RTCP Packet and
  - Application Specific RTCP packets.
- RTCP itself does not provide any flow encryption or authentication means. SRTCP protocol can be used for that purpose.

RTP Services

- Payload Type Identification
  - Determination of media coding
  - Source identification
  - RTP works with Profiles
    - Profile defines a set of payload type codes and their mappings to payload formats
- Sequence numbering
  - Error detection
- Time-stamping
  - Time monitoring, synchronization, jitter calculation
- Delivery monitoring

RTP Message


```
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Ver P X CC M PT Sequence Number

Timestamp

SSRC

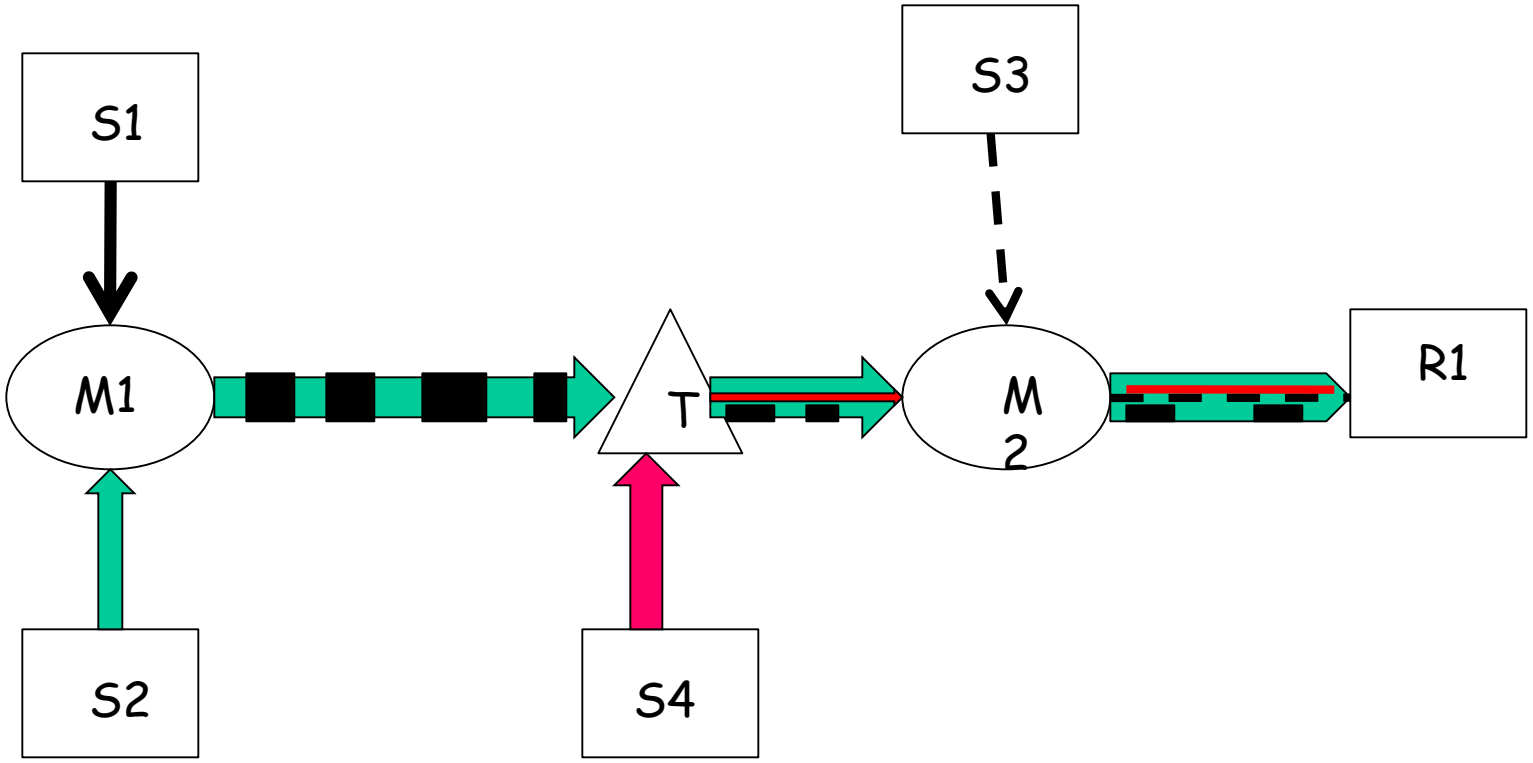
CSRC [0..15] :::

Ver – Version 2
P – Padding
X – Extension, if set, the fixed head is followed by exactly one
header extension
CC – CSRC count
M – Marker – intended to allow significant events such as
frame boundaries to be marked (defined by profile)
PT – Payload type
SSRS – synchronization source, CSRC – contribution source

RTP Services – Support of Heterogeneity

- Mixer service
  - Allows for resynchronization of incoming audio packets
  - Reconstructs constant 20 ms spacing generated by sender
  - Mixes reconstructed audio streams into single stream
  - Translates audio encoding to lower bandwidth
  - Forwards lower bandwidth packet streams
- Translator service
  - Allows for translation between IP and other high speed protocols
  - May change encoding data

Difference between Mixers and Translators

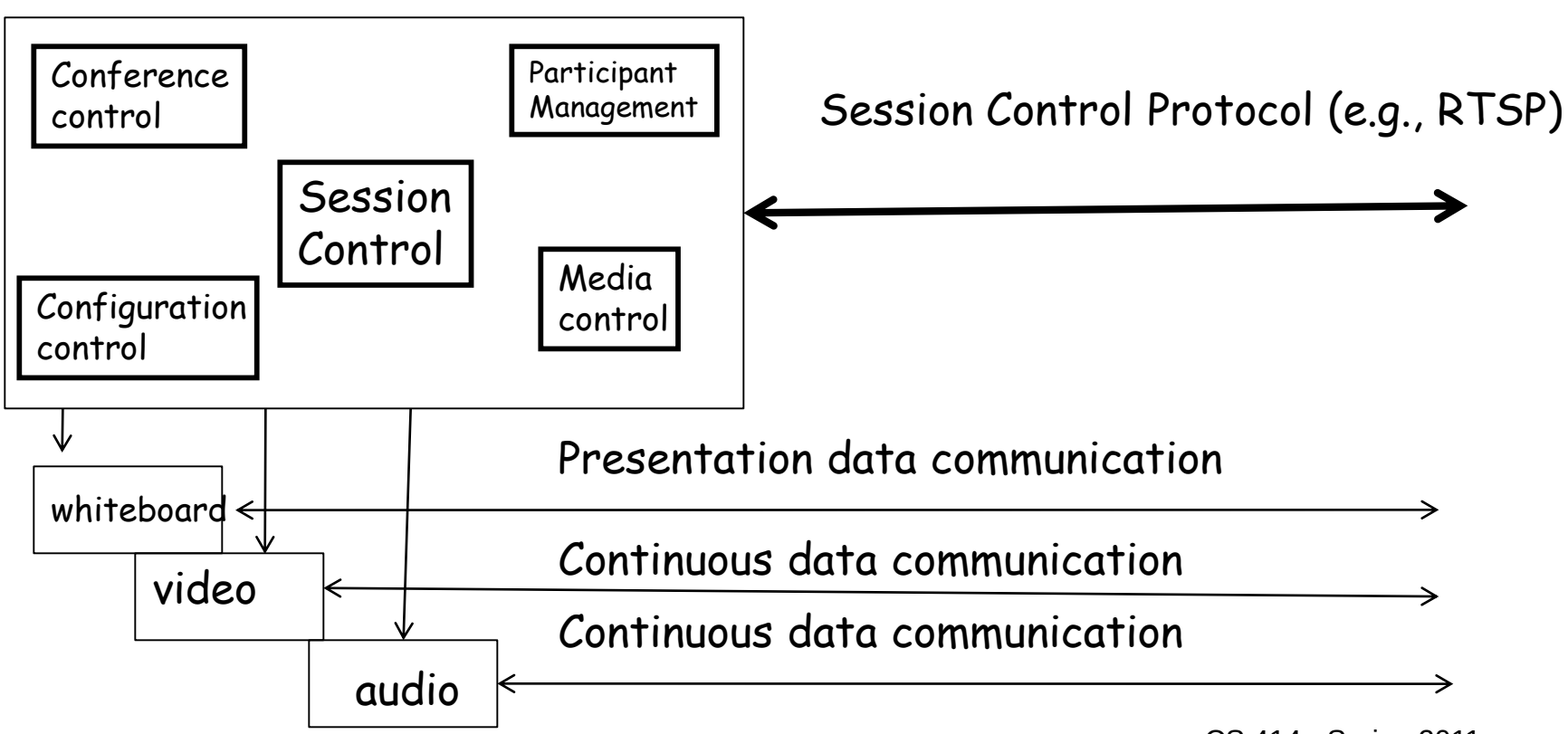


Payload Formats

- Static Payload formats
  - Established in RTP Profile
  - Payload type 0 := μ-law audio codec
- Dynamic Payload formats
  - Applications agree per session on payload format
  - H.263, JPEG, MPEG

Session Management (Layer 5)

- Important part of multimedia communication
- Separates control aspects from transport aspects



Session Manager

- Tasks:
  - Membership control
  - Monitoring of shared workspace
  - Coordination of Media control management
  - Exchange of QoS parameters
  - Conference control management – establishment, modification, termination

Session Control

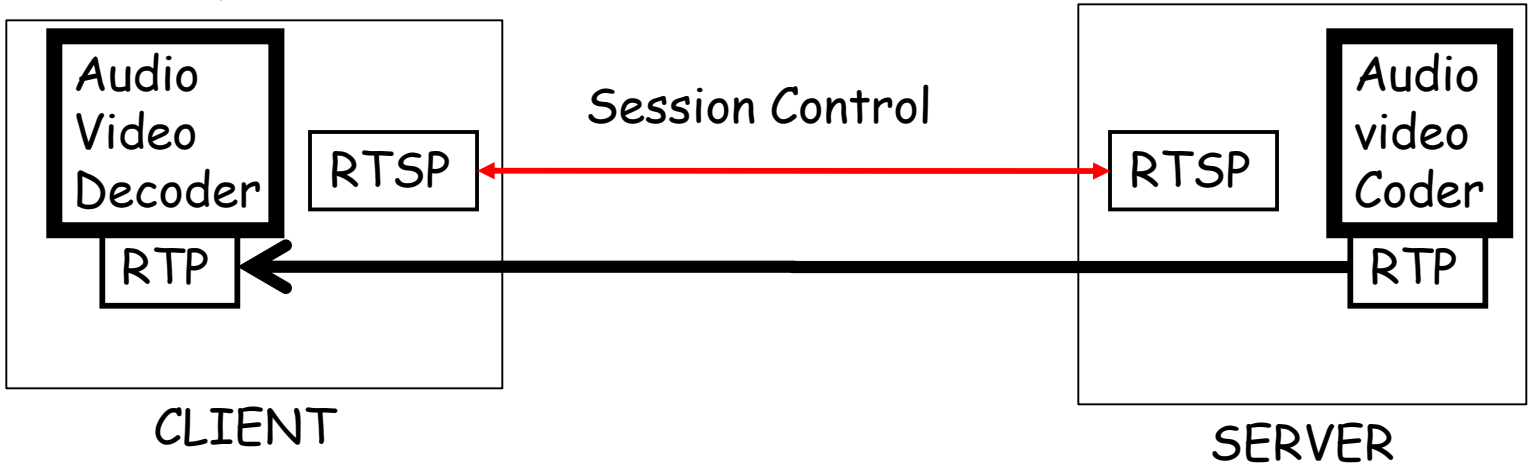
- Session Described by
  - Session state
    - Name of session, start, valid policies
- Session management – two steps for state processing
  - Establishment of session
  - Modification of session

Session Control

- Conference Control
  - Centralized or distributed approach
- Media Control
  - Synchronization
- Configuration Control
  - Negotiation of QoS parameters, admission control and reservation/allocation of resources
- Membership Control
  - Invitation of users; registration of users, change of membership

Real-Time Streaming Protocol (RTSP)

- Application Protocol for Control of multimedia streams
- This is **not** an application data transmission protocol, just **remote control protocol** between client and server



RTSP

- Approved as Internet Draft, February 2, 1998, authors H. Schulzrinne, A. Rao, R. Lanphier
- Enables controlled, on-demand delivery of real-time data such as audio and video
- Intends to control multiple data delivery sessions
- Provides means for choosing delivery channels
  - UDP
  - Multicast UDP,
  - TCP

RTSP Methods

Request	Direction	Description
OPTIONS	S <-> C	Determine capabilities of server (S) or client (C)
DESCRIBE	C -> S	Get description of media stream
ANNOUNCE	S <-> C	Announce new session description
SETUP	C -> S	Create media session
RECORD	C -> S	Start media recording
PLAY	C -> S	Start media delivery
PAUSE	C -> S	Pause media delivery
REDIRECT	S -> C	Use other server
TEARDOWN	C -> S	Destroy media session
SET_PARAMETER	S <-> C	Set server or client parameter
GET_PARAMETER	S <-> C	Read server or client parameter

RTSP Extensions

- Timing
  - RTSP needs to hide latency variations
  - PLAY request may contain information about when request is to be executed
- Three types of timestamps
  - SMPTE (the same as in TV production)
    - Format: hours:minutes:seconds:frames
  - Normal play time
    - Measured relative to beginning of stream and expressed in ours, minutes, seconds and fractions of second
  - Absolute time
    - Wall clock

RTSP Example

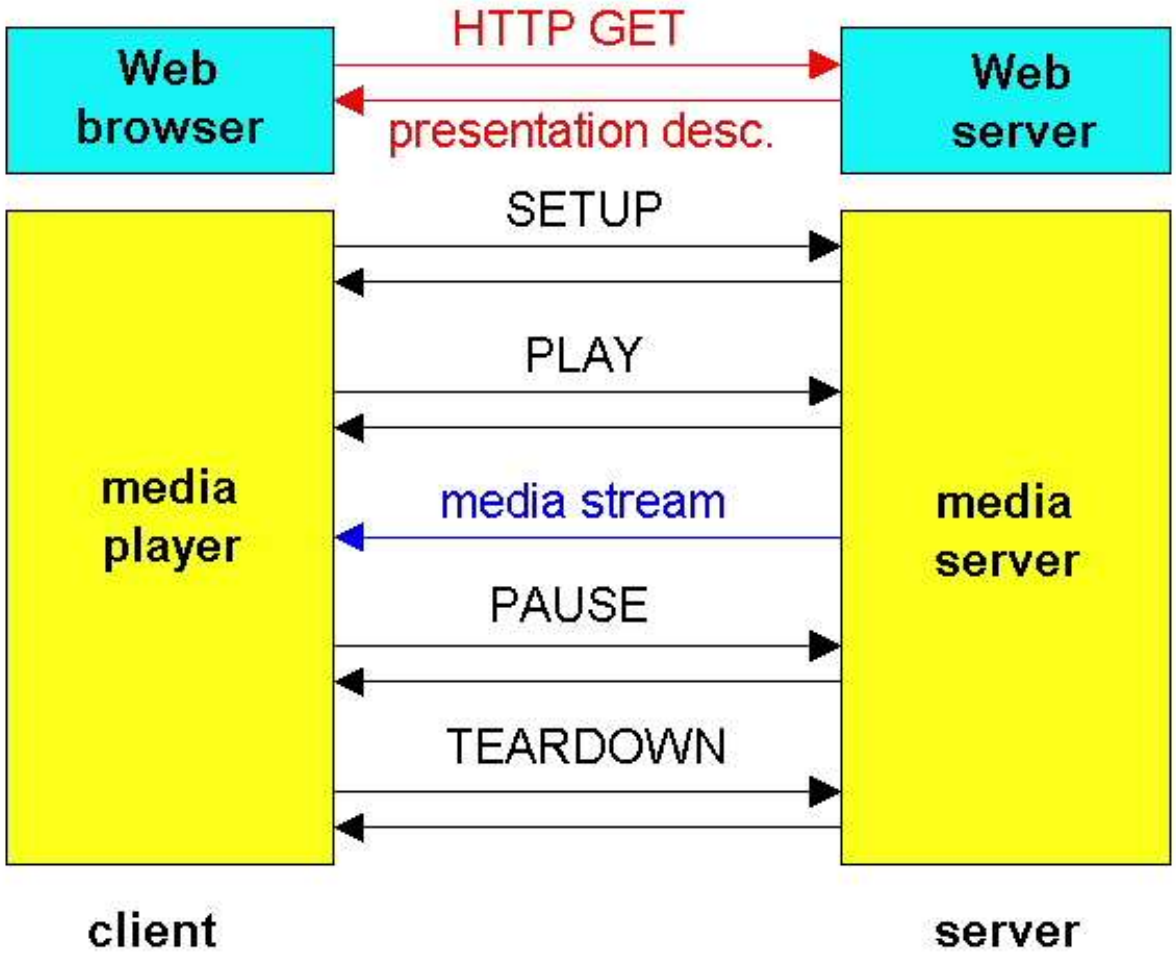
Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

Metafile Example

```
<title>Twister</title>
<session>
        <group language=en lipsync>
                <switch>
                  <track type=audio
                        e="PCMU/8000/1"
                        src = "rtsp://audio.example.com/twister/audio.en/lofi">
                  <track type=audio
                        e="DVI4/16000/2" pt="90 DVI4/8000/1"
                        src="rtsp://audio.example.com/twister/audio.en/hifi">
                 </switch>
               <track type="video/jpeg"
                        src="rtsp://video.example.com/twister/video">
         </group>
</session>
```

RTSP Operation



RTSP Exchange Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231

S: 200 3 OK

Conclusion

- RTP usage – in several application audio and video tools (vat, vic)
- RTP follows the principle of application level framing and integrated layer processing
- RTP/UDP/IP is being used by the current streaming session protocols such as RTSP
- Session protocols are actually negotiation/session establishment protocols that assist multimedia applications
- Multimedia applications such as QuickTime, Real Player and others use them

Internet Protocols

- Existing Protocols
  - TCP - reliable transport protocol
  - UDP - unreliable transport protocol
  - IP - Internet Network Protocol
  - IMCP - Internet Message Control Protocol
- New Protocols
  - IPng - IP next generation (IP version 6)
  - RTP - Real-time transport Protocol
  - RSVP - resource reservation Protocol
  - RTSP

Multicasting

- Most current networks provide only unicast
  - point-to-point connectivity
  - replicated unicast - partial solution
    - if one wants to reach multiple receivers
  - Multicast - better solution
    - current IP already provides this via MBONE routers that are multicast routers

Internet Group Management Protocol

- Protocol for managing Internet multicast groups
- Host membership query messages
  - sent by multicast routers to refresh their knowledge of memberships present on a network.
- Host membership reports
  - sent by hosts in response to a query. Either individual group or host can respond.
- Queries are sent infrequently
  - to keep IGMP protocol overhead low
- For multimedia
  - IGMP must cooperate with resource management protocols such as RSVP to provide resource reservation.

Real-time Transport Protocol (RTP)

- RTP provides end-to-end transport functions
  - suitable for applications transmitting real-time data over multicast or unicast network services, e.g. multiparty multimedia conferences.
  - Companion protocol - RTCP (Real-time Control protocol)
    - conveys information about participants of a conference
  - RTP functions
    - determination of media encoding, synchronization, framing, error detection, encryption, timing and source identification
  - RTCP function
    - to monitor QoS and convey participant information.
    - QoS monitor used to estimate current QoS, fault diagnosis, long-term statistics.

RTP

- RTP does not do resource reservation or guarantee QoS for real-time applications
  - relies on lower layers for real-time guarantees
  - header carries sequence number for sequencing.
- Uses services of transport protocol - UDP/IP, ST-II etc.
  - Provides application level framing and integrated layer processing
  - RTP works with Profile that
    - defines a set of payload type codes and their mapping to payload formats.
  - Usage of RTP
    - in video and audio tools (vat, nv)
    - nv is a packet video program - supports network I/O for visual interaction in tele-conferencing over the Mbone.

Other RTP Services

- RTP Services
  - Payload Type Identification
  - Sequence Numbering
  - Timestamping
  - Delivery Monitoring
- Other Services
  - Mixer Service
    - allows for resynchronization of incoming audio packets
    - reconstructs the constant 20ms spacing generated by sender
    - mixes the reconstructed audio streams into a single stream
    - translates the audio encoding to a lower bandwidth
    - forwards the lower bandwidth packet stream
  - Translator Service
    - allows for translation between IP and other high-speed protocols (e.g. ST-II and IP)
    - translators may change the encoding of data

IPng - new Internet Protocol

- IPng is IP version 6
  - will replace the current IP version 4
- New Features
  - new addressing and routing
    - large hierarchical addresses (cluster addresses which allow policy route selection)
    - multicast addresses carrying addresses of other Internet protocol suites
  - more options for flow control and security
    - allows for real-time flows, end-to-end security, provider selection
  - host mobility
  - auto-configuration/auto-reconfiguration

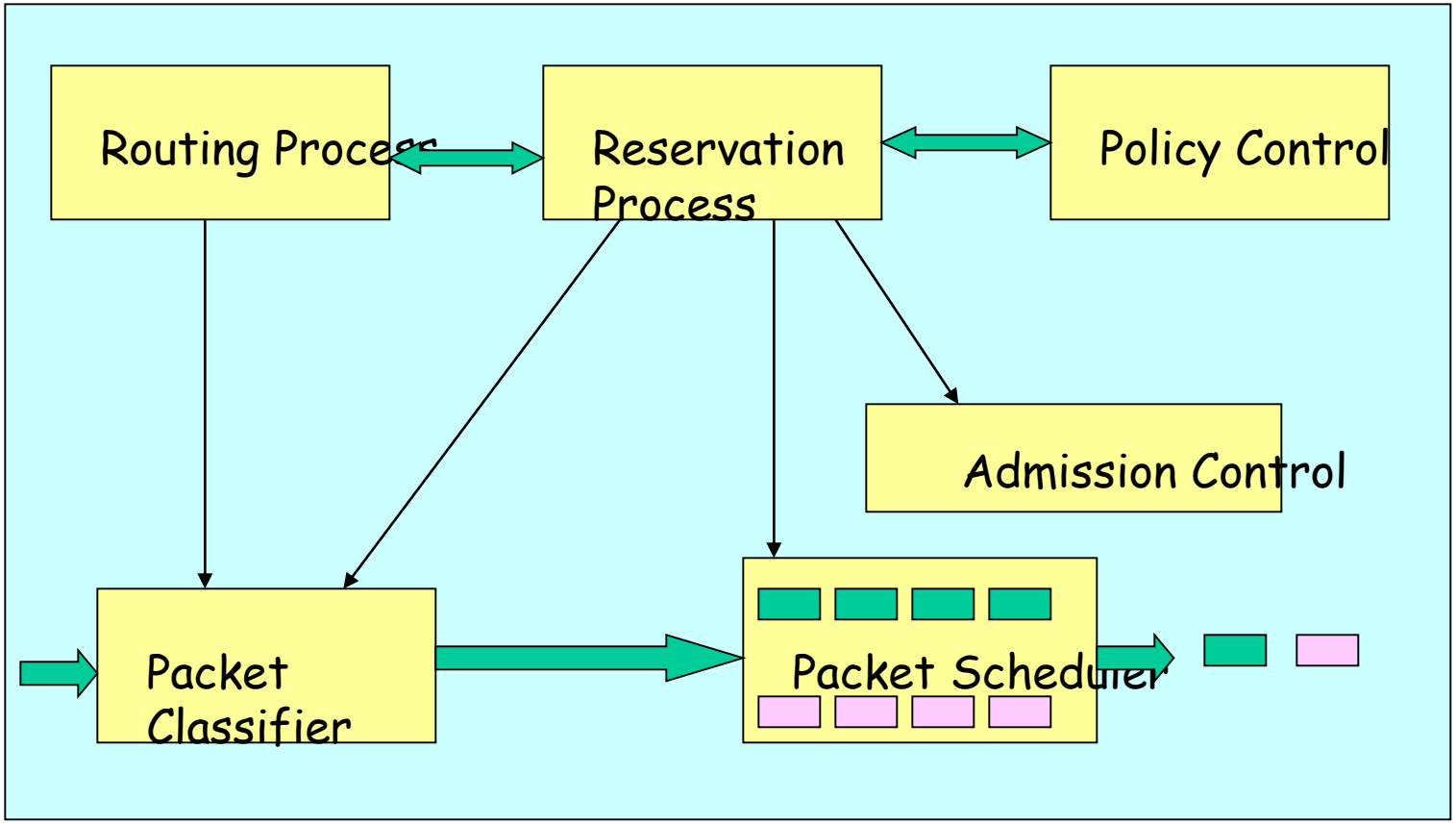
IPng - Routing and Addressing

- Designed to run over high-speed (ATM) networks as well as work with low bandwidth (wireless network)
- Supports clusters, unicast and multicast
  - Multicast addresses must specify scope
  - Cluster addresses specify topological regions rather than individual nodes
- Hierarchical routing tables are required
- Each RTP flow can be labeled with QoS
  - real-time services allow this.

Integrated Services

- Classes of Service
  - Datagram Service
  - Controlled Load Service
    - Performance as good as in an unloaded datagram network. No quantitative assurances.
  - Guaranteed Service
    - Firm bound on delay/throughput provided by every element along path
- Flow Specification
  - Flow Spec = Traffic Spec + QoS Spec
  - Traffic Spec - Peak rate, bucket rate, bucket size, max datagram size, min policed unit
  - QoS Spec (for guaranteed service) - Rate, delay slack

IS Capable Router



RSVP - Resource ReSerVation Protocol

- Reservation
  - specifies the amount of resources to be reserved for all or some subset of packets in a particular session.
  - Implies adding a notion of flow spec (resource quantity) to intermediate nodes - parameterizes packet scheduling
  - Filter spec - specifies the packet subset to receive the resources
- RSVP is a setup protocol of MM flows
  - Senders multicast their data flows
  - Senders periodically transmit *path* that includes a flow specification describing their flows.
  - Receivers tune to appropriate multicast group and look for *path* messages. Based on path messages, receiver decides what part of senders flow to receive. Receiver generates a reservation message, which contains filter and flow specification for each senders flow.

RSVP

- Purpose of filters
  - provide support for heterogeneity - Receivers at the end of slow links can still participate in flows by using a filter to restrict what portion of a flow is passed to it.
  - Dynamic filtering allows receivers to modify flow properties. Also useful when receiver is listening to multiple flows where filter can dynamically change which flows it is listening to.
  - Reduce load and improve bandwidth management
- Three types of filters
  - No filter (wildcard) mode - senders flow is not filtered
  - Fixed filter mode
    - senders flow is filtered according to fixed filter during reservation.
  - Dynamic filter mode (Shared explicit)
    - receiver can change the filter specification during the reservation.

RSVP

- Reservations are Receiver oriented
  - sender starts but receivers perform reservations - support for heterogeneity of receivers
- RSVP uses soft state to maintain information about reservation.
  - Soft state
    - information that is periodically refreshed by interested parties.
  - In RSVP, senders and receivers refresh state at routers.
    - Senders are required to periodically retransmit path messages to allow new receivers to learn of the flow, remind routers that the flow exists, and to adapt to routing changes.
    - Receivers periodically retransmit their reservation messages to remind routers of their reservation.

RTSP (Real-time Streaming Protocol)

- Source -Internet Draft, Feb 1998
- Application level protocol
  - for control over delivery of data with real-time properties
  - Provides extensible framework to enable controlled, on-demand delivery of real-time data such as audio/video
    - establishes and controls either a single or several time-synchronized streams of continuous media
  - Sources - both live data feeds and stored clips
  - can control multiple data delivery sessions
    - provide means for choosing delivery channels such as UDP, multicast UDP and TCP and provide a means for choosing delivery mechanisms
  - does not deliver media, acts a network remote control for servers.

RTSP

- RTSP Session
  - No notion of a RTSP connection server maintains a session labeled by an identifier that is in no way connected to a transport level connection such as a TCP connection.
  - During an RTSP session, an RTSP client may open and close many connections to the server to issue RTSP requests.
- RTSP protocol
  - works between an RTSP server and client
    - server maintains state by default in almost all cases.
    - Both RTSP server and client can issue a request.

RTSP Operations

- Basic RTSP Control Requests
  - SETUP, PLAY, RECORD, PAUSE and TEARDOWN
  - Retrieval of media from media server
    - Client requests a presentation description. If presentation is being multicast, presentation description contains the multicast addresses and ports for continuous media. For unicast, client provides the destination for security reasons.
  - Invitation of a media server to a conference
    - media server invited to join an existing conference to playback media into the presentation or to record all or a subset of the media in a presentation.
  - Addition of media to an existing presentation
    - Useful for live presentation where server can tell the client about additional media becoming available

Outline

- Multimedia networking applications
- Requirements for Multimedia Communication
  - User and application requirements
  - Processing and protocol constraints
  - Mapping to OSI layers

- Network QoS and Resource Management
  - Providing multiple classes of service
  - Negotiation, Translation, Admission
  - Traffic Shaping, Rate Control, Error Control
  - Monitoring, Adaptation
- MM over Internet
  - Protocols for real-time interactive applications (RTP,RTCP,SIP)
- Other Case Studies
  - Fast Ethernet, FDDI, DQDB, ATM

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

Chapter 7 outline

7.1 multimedia networking applications

7.2 streaming stored audio and video

7.3 making the best out of best effort service

7.4 protocols for real-time interactive applications

RTP,RTCP,SIP

7.5 providing multiple classes of service

7.6 providing QoS guarantees

Real-time interactive applications

- PC-2-PC phone
  - Skype
- PC-2-phone
  - Dialpad
  - Net2phone
  - Skype
- videoconference with webcams
  - Skype
  - Polycom

Going to now look at a PC-2-PC Internet phone example in detail

Interactive Multimedia: Internet Phone

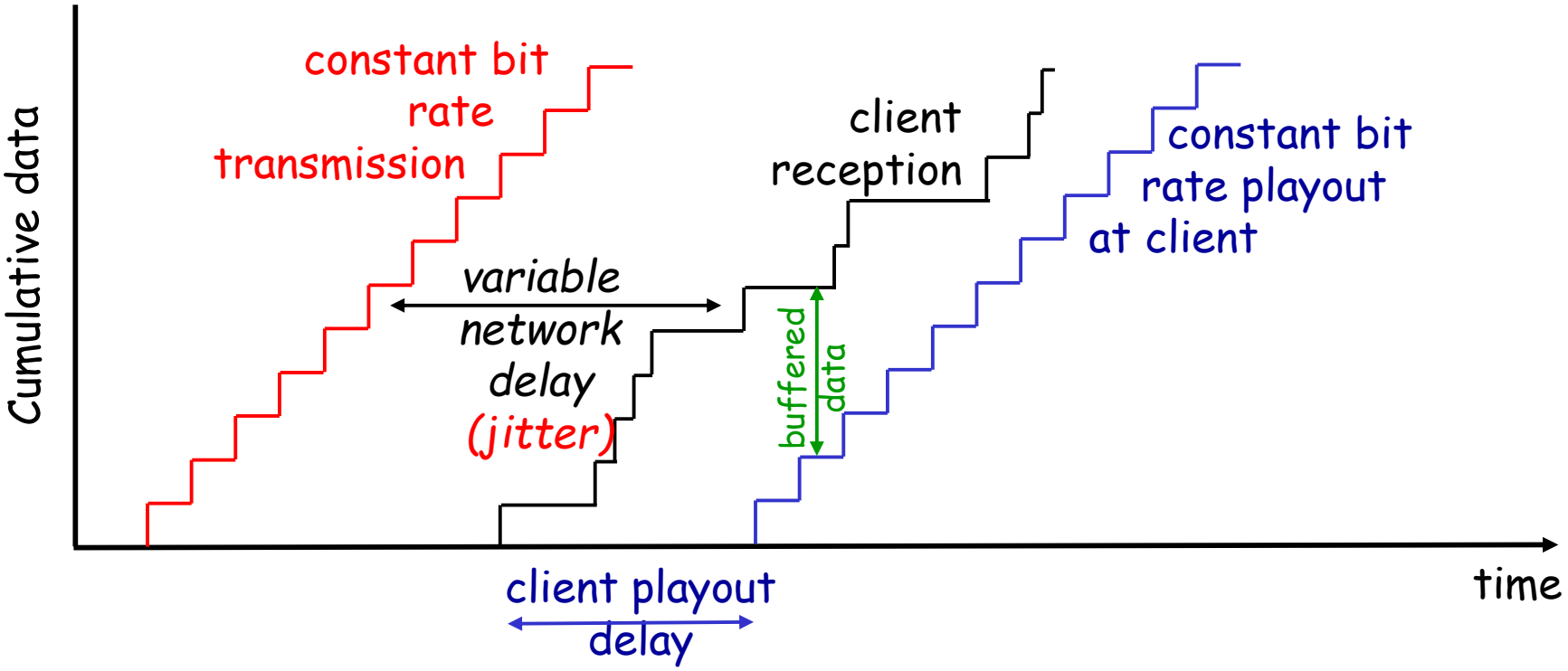
Introduce Internet Phone by way of an example

- speaker's audio: alternating talk spurts, silent periods.
  - 64 kbps during talk spurt
  - pkts generated only during talk spurts
  - 20 msec chunks at 8 Kbytes/sec: 160 bytes data
- application-layer header added to each chunk.
- chunk+header encapsulated into UDP segment.
- application sends UDP segment into socket every 20 msec during talkspurt

Internet Phone: Packet Loss and Delay

- network loss: IP datagram lost due to network congestion (router buffer overflow)
- delay loss: IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- loss tolerance: depending on voice encoding, losses concealed, packet loss rates between 1% and 10% can be tolerated.

Delay Jitter



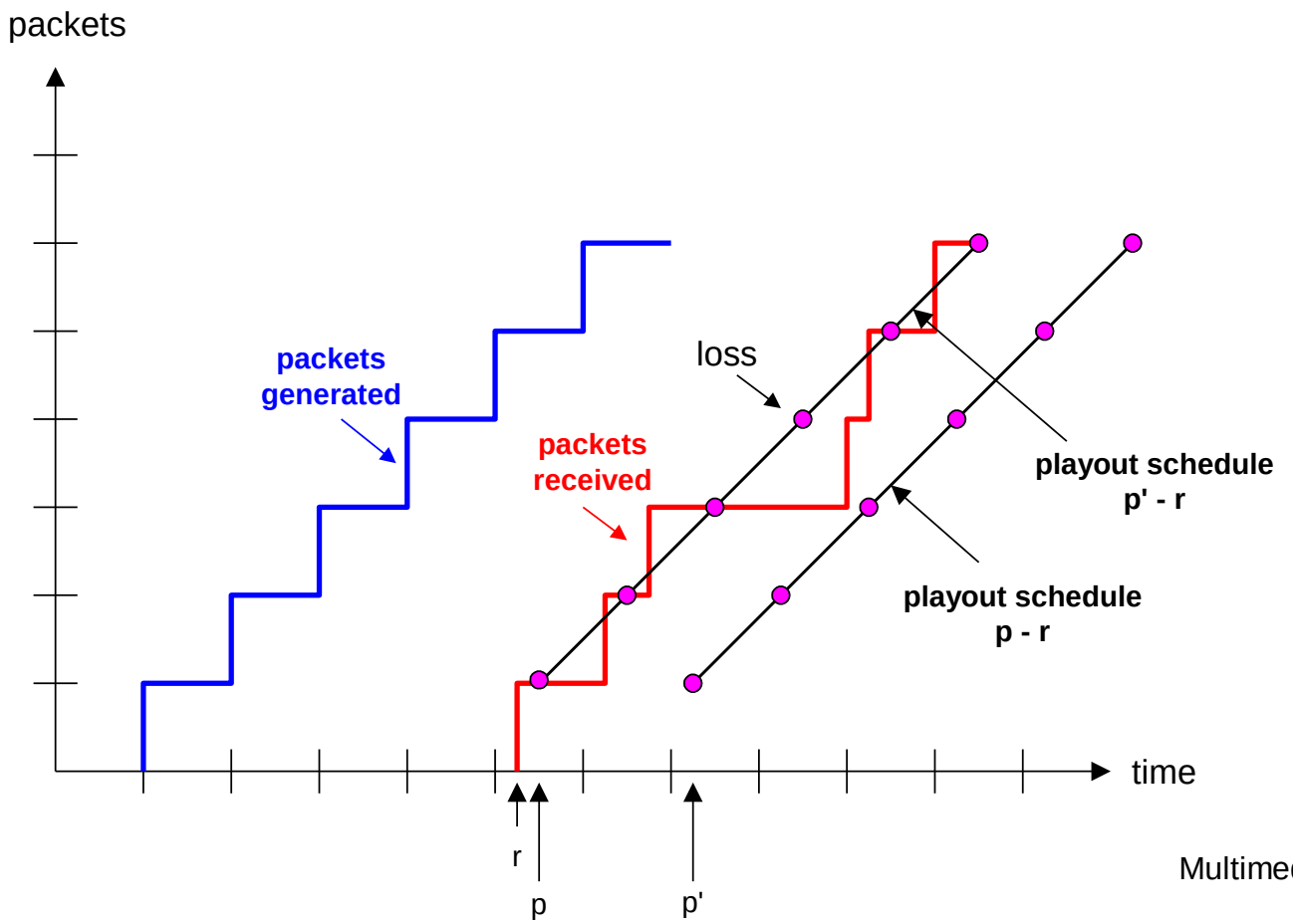
- consider end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

Internet Phone: Fixed Playout Delay

- receiver attempts to playout each chunk exactly q msecs after chunk was generated.
  - chunk has time stamp t: play out chunk at t+q .
  - chunk arrives after t+q: data arrives too late for playout, data "lost"
- tradeoff in choosing q:
  - *large q:* less packet loss
  - *small q:* better interactive experience

Fixed Playout Delay

- sender generates packets every 20 msec during talk spurt.
- first packet received at time r
- first playout schedule: begins at p
- second playout schedule: begins at p'



Adaptive Playout Delay (1)

- Goal: minimize playout delay, keeping late loss rate low
- Approach: adaptive playout delay adjustment:
  - estimate network delay, adjust playout delay at beginning of each talk spurt.
  - silent periods compressed and elongated.
  - chunks still played out every 20 msec during talk spurt.

t_i = timestamp of the ith packet

r_i = the time packet i is received by receiver

p_i = the time packet i is played at receiver

$r_i - t_i$ = network delay for ith packet

d_i = estimate of average network delay after receiving ith packet

dynamic estimate of average delay at receiver:

$$d_i = (1-u)d_{i-1} + u(r_i - t_i)$$

where u is a fixed constant (e.g., u = .01).

Adaptive playout delay (2)

- also useful to estimate average deviation of delay, v_i :

$$v_i = (1-u)v_{i-1} + u\,|\,r_i - t_i - d_i\,|$$

- estimates d_i , v_i calculated for every received packet
- (but used only at start of talk spurt

- for first packet in talk spurt, playout time is:

$$p_i = t_i + d_i + Kv_i$$

where K is positive constant

- remaining packets in talkspurt are played out periodically

Adaptive Playout (3)

Q: How does receiver determine whether packet is first in a talkspurt?

- if no loss, receiver looks at successive timestamps.
  - difference of successive stamps > 20 msec -->talk spurt begins.
- with loss possible, receiver must look at both time stamps and sequence numbers.
  - difference of successive stamps > 20 msec and sequence numbers without gaps --> talk spurt begins.

Recovery from packet loss (1)

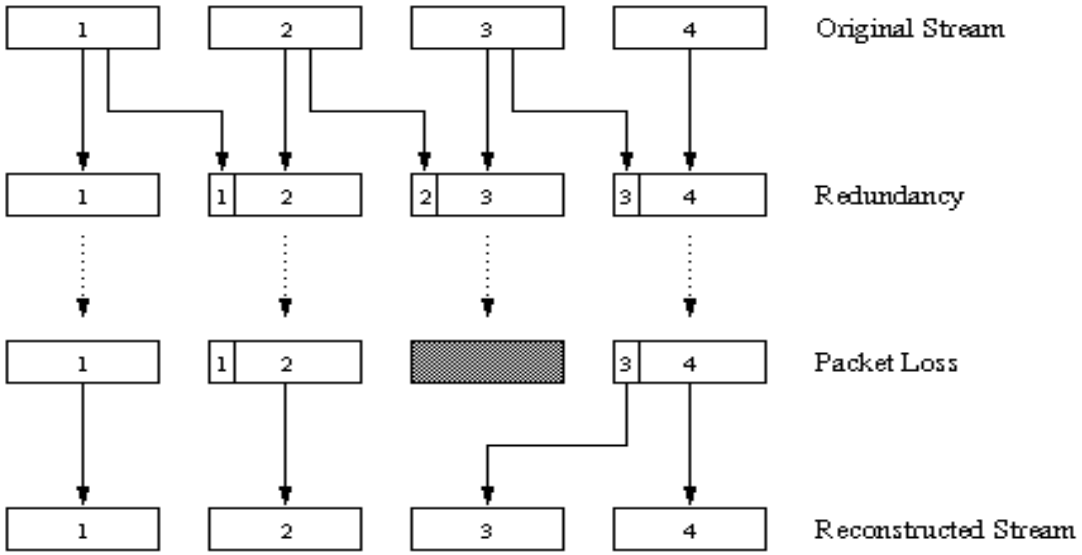
Forward Error Correction (FEC): simple scheme

- for every group of *n* chunks create redundant chunk by exclusive OR-ing *n* original chunks
- send out *n+1* chunks, increasing bandwidth by factor *1/n*.
- can reconstruct original *n* chunks if at most one lost chunk from *n+1* chunks
- playout delay: enough time to receive all n+1 packets
- tradeoff:
  - increase n, less bandwidth waste
  - increase n, longer playout delay
  - increase n, higher probability that 2 or more chunks will be lost

Recovery from packet loss (2)

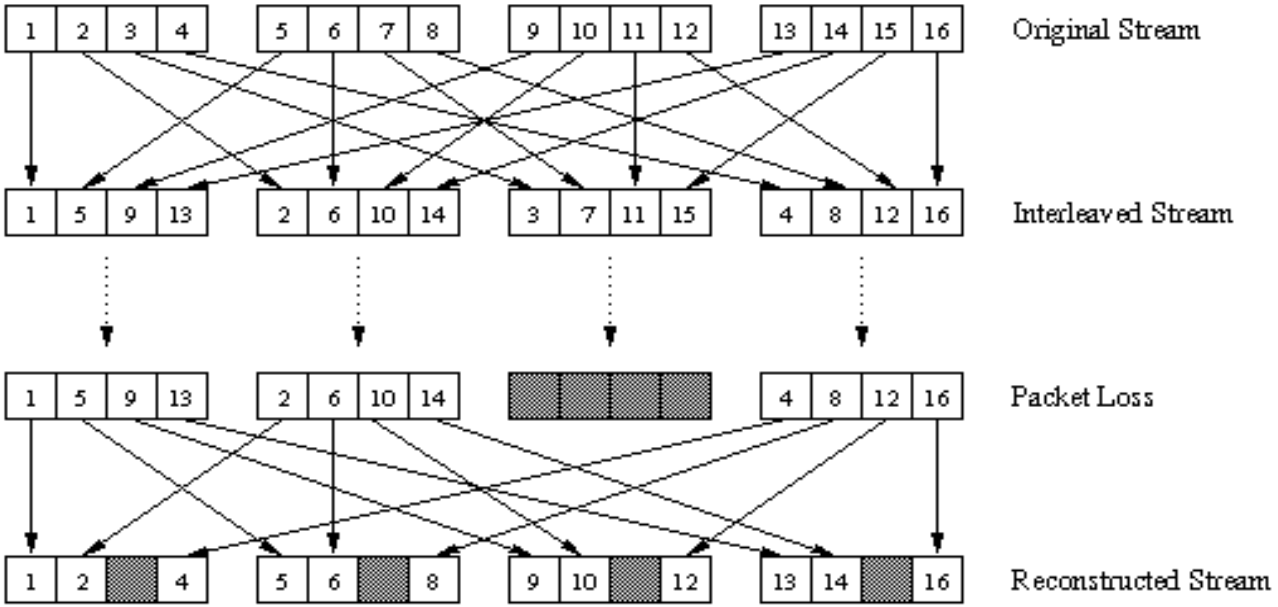
2nd FEC scheme

- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
- e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps.



- whenever there is non-consecutive loss, receiver can conceal the loss.
- can also append (n-1)st and (n-2)nd low-bit rate chunk

Recovery from packet loss (3)



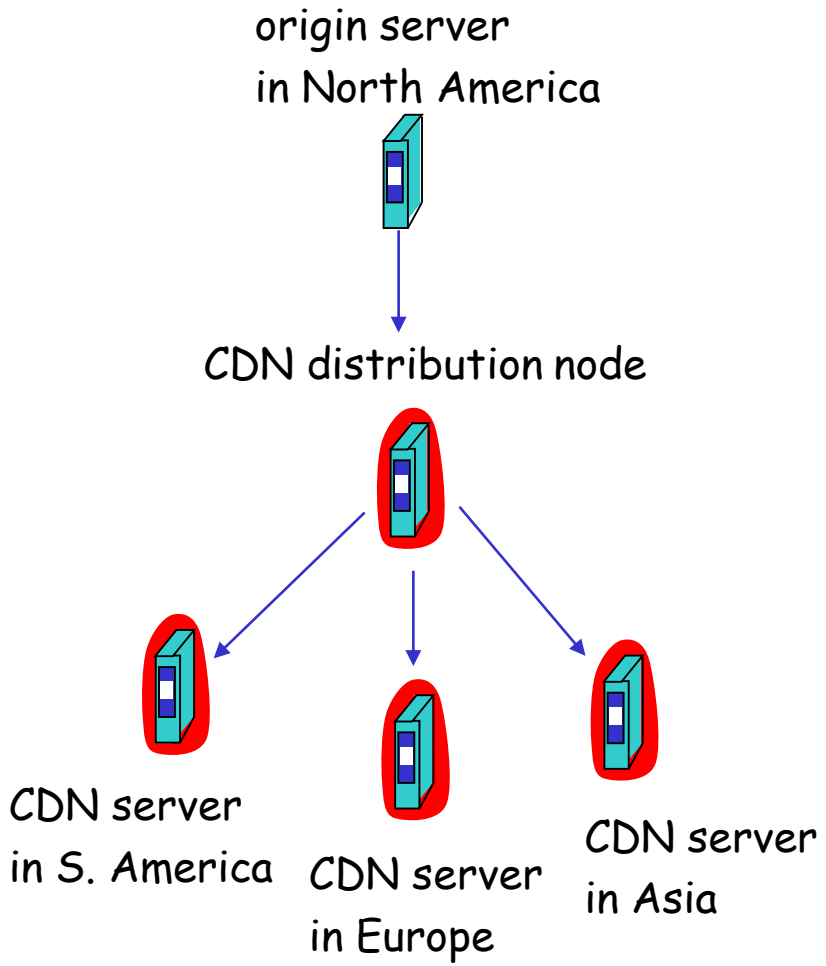
Interleaving

- chunks divided into smaller units
- for example, four 5 msec units per chunk
- packet contains small units from different chunks
- if packet lost, still have most of every chunk
- no redundancy overhead, but increases playout delay

Content distribution networks (CDNs)

Content replication

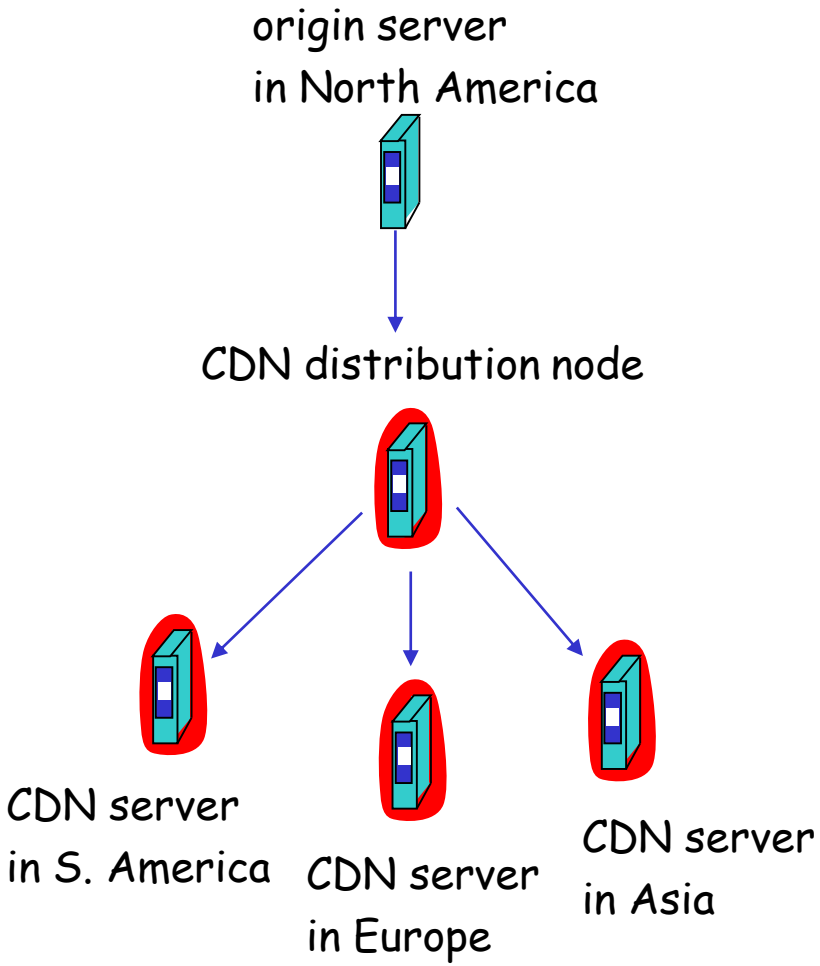
- challenging to stream large files (e.g., video) from single origin server in real time
- *solution:* replicate content at hundreds of servers throughout Internet
  - content downloaded to CDN servers ahead of time
  - *placing content "close" to user avoids impairments (loss, delay) of sending content over long paths*
  - CDN server typically in edge/access network



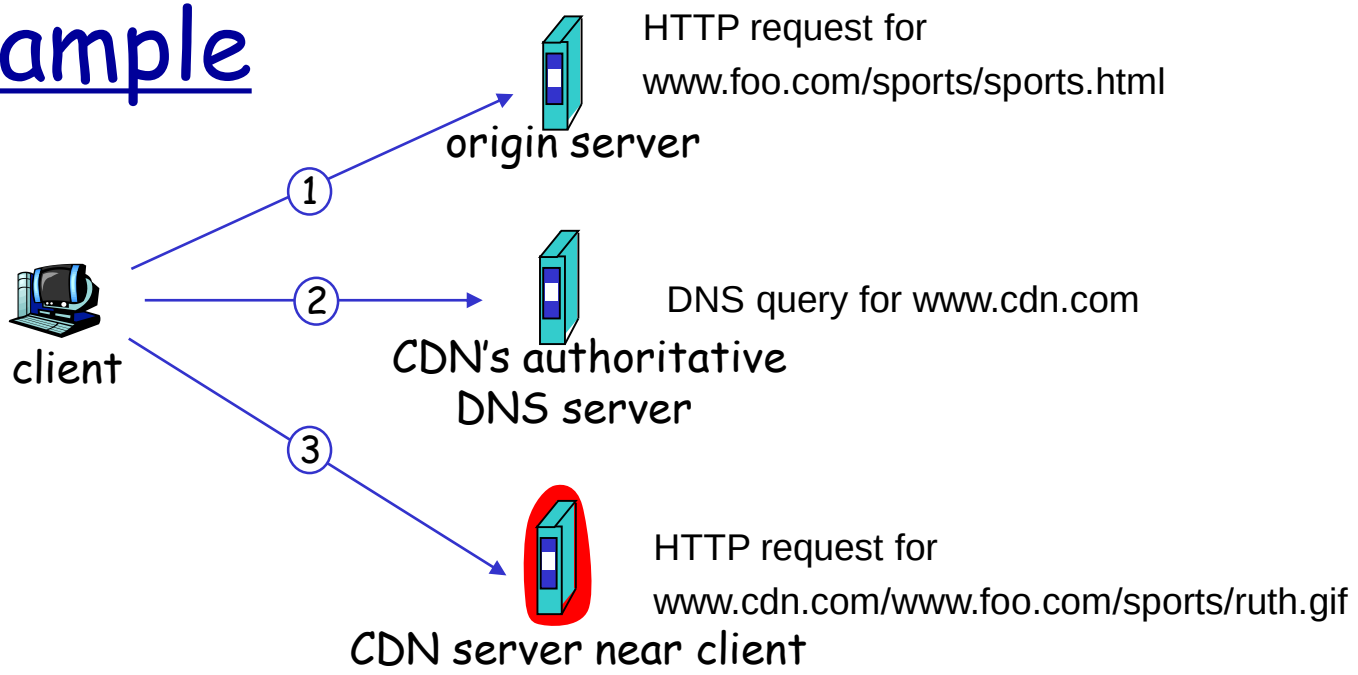
Content distribution networks (CDNs)

Content replication

- CDN (e.g., Akamai) customer is the content provider (e.g., CNN)
- CDN replicates customers' content in CDN servers.
- when provider updates content, CDN updates servers



CDN example



origin server (www.foo.com)

- distributes HTML
- replaces:

  http://www.foo.com/sports.ruth.gif

  with

  http://www.cdn.com/www.foo.com/sports/ruth.gif

CDN company (cdn.com)

- distributes gif files
- uses its authoritative DNS server to route redirect requests

More about CDNs

routing requests

- CDN creates a "map", indicating distances from leaf ISPs and CDN nodes
- when query arrives at authoritative DNS server:
  - server determines ISP from which query originates
  - uses "map" to determine best CDN server
- CDN nodes create application-layer overlay network

Summary: Internet Multimedia: bag of tricks

- use UDP to avoid TCP congestion control (delays) for time-sensitive traffic
- client-side adaptive playout delay: to compensate for delay
- server side matches stream bandwidth to available client-to-server path bandwidth
  - chose among pre-encoded stream rates
  - dynamic server encoding rate
- error recovery (on top of UDP)
  - FEC, interleaving, error concealment
  - retransmissions, time permitting
- CDN: bring content closer to clients

Chapter 7 outline

7.1 multimedia networking applications

7.2 streaming stored audio and video

7.3 making the best out of best effort service

7.4 protocols for real-time interactive applications

RTP, RTCP, SIP

7.5 providing multiple classes of service

7.6 providing QoS guarantees

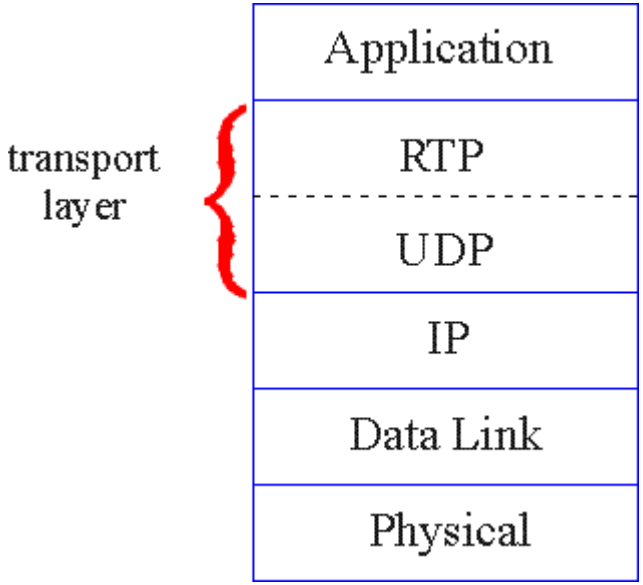
Real-Time Protocol (RTP)

- RTP specifies packet structure for packets carrying audio, video data
- RFC 3550
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - time stamping
- RTP runs in end systems
- RTP packets encapsulated in UDP segments
- interoperability: if two Internet phone applications run RTP, then they may be able to work together

RTP runs on top of UDP

RTP libraries provide transport-layer interface that extends UDP:

- port numbers, IP addresses
- payload type identification
- packet sequence numbering
- time-stamping



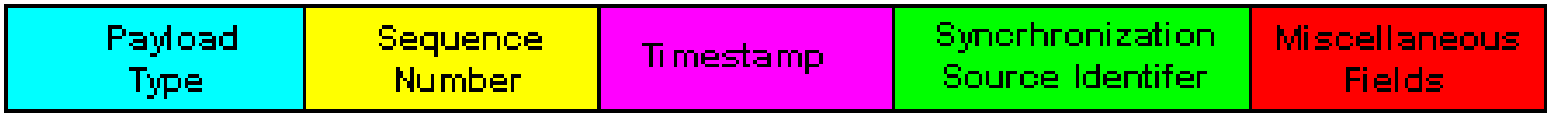
RTP Example

- consider sending 64 kbps PCM-encoded voice over RTP.
- application collects encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk.
- audio chunk + RTP header form RTP packet, which is encapsulated in UDP segment
- RTP header indicates type of audio encoding in each packet
  - sender can change encoding during conference.
- RTP header also contains sequence numbers, timestamps.

RTP and QoS

- RTP does **not** provide any mechanism to ensure timely data delivery or other QoS guarantees.
- RTP encapsulation is only seen at end systems (not) by intermediate routers.
  - routers providing best-effort service, making no special effort to ensure that RTP packets arrive at destination in timely matter.

RTP Header



RTP Header

Payload Type (7 bits): Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs receiver via payload type field.

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3, GSM, 13 kbps
- Payload type 7, LPC, 2.4 kbps
- Payload type 26, Motion JPEG
- Payload type 31. H.261
- Payload type 33, MPEG2 video

Sequence Number (16 bits): Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence.

RTP Header (2)

- *Timestamp field (32 bytes long):* sampling instant of first byte in this RTP data packet
  - for audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for 8 KHz sampling clock)
  - if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.

- *SSRC field (32 bits long):* identifies source of t RTP stream. Each stream in RTP session should have distinct SSRC.

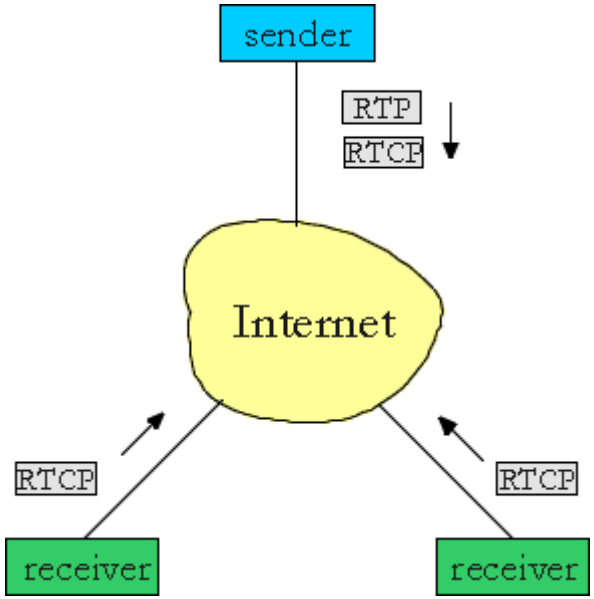
RTSP/RTP Programming Assignment

- build a server that encapsulates stored video frames into RTP packets
  - grab video frame, add RTP headers, create UDP segments, send segments to UDP socket
  - include seq numbers and time stamps
  - client RTP provided for you
- also write client side of RTSP
  - issue play/pause commands
  - server RTSP provided for you

Real-Time Control Protocol (RTCP)

- works in conjunction with RTP.
- each participant in RTP session periodically transmits RTCP control packets to all other participants.
- each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application: # packets sent, # packets lost, interarrival jitter, etc.
- feedback can be used to control performance
  - sender may modify its transmissions based on feedback

RTCP - Continued



- each RTP session: typically a single multicast address; all RTP /RTCP packets belonging to session use multicast address.
- RTP, RTCP packets distinguished from each other via distinct port numbers.
- to limit traffic, each participant reduces RTCP traffic as number of conference participants increases

RTCP Packets

Receiver report packets:

- fraction of packets lost, last sequence number, average interarrival jitter

Sender report packets:

- SSRC of RTP stream, current time, number of packets sent, number of bytes sent

Source description packets:

- e-mail address of sender, sender's name, SSRC of associated RTP stream
- provide mapping between the SSRC and the user/host name

Synchronization of Streams

- RTCP can synchronize different media streams within a RTP session
- consider videoconferencing app for which each sender generates one RTP stream for video, one for audio.
- timestamps in RTP packets tied to the video, audio sampling clocks
  - *not* tied to wall-clock time
- each RTCP sender-report packet contains (for most recently generated packet in associated RTP stream):
  - timestamp of RTP packet
  - wall-clock time for when packet was created.
- receivers uses association to synchronize playout of audio, video

RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of session bandwidth.

Example

- Suppose one sender, sending video at 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
- RTCP gives 75% of rate to receivers; remaining 25% to sender
- 75 kbps is equally shared among receivers:
  - with R receivers, each receiver gets to send RTCP traffic at 75/R kbps.
- sender gets to send RTCP traffic at 25 kbps.
- participant determines RTCP packet transmission period by calculating avg RTCP packet size (across entire session) and dividing by allocated rate

SIP: Session Initiation Protocol [RFC 3261]

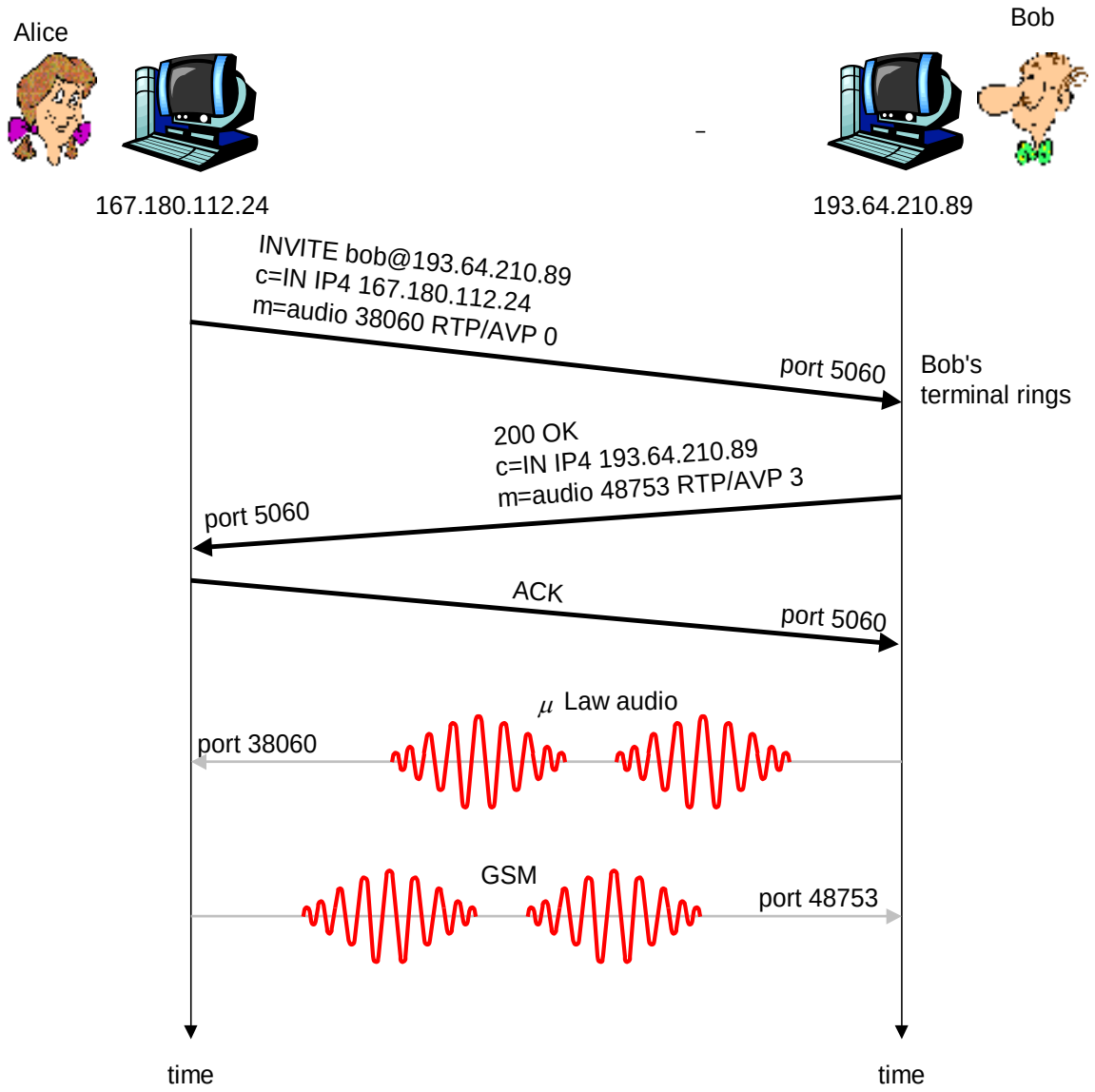
SIP long-term vision:

- all telephone calls, video conference calls take place over Internet
- people are identified by names or e-mail addresses, rather than by phone numbers
- you can reach callee, no matter where callee roams, no matter what IP device callee is currently using

SIP Services

- Setting up a call, SIP provides mechanisms ..
  - for caller to let callee know she wants to establish a call
  - so caller, callee can agree on media type, encoding
  - to end call
- determine current IP address of callee:
  - maps mnemonic identifier to current IP address
- call management:
  - add new media streams during call
  - change encoding during call
  - invite others
  - transfer, hold calls

Setting up a call to known IP address



- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- default SIP port number is 5060.

Setting up a call (more)

- codec negotiation:
  - suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply, listing his encoders Alice can then send new INVITE message, advertising different encoder
- rejecting a call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden"
- media can be sent over RTP or some other protocol

Example of SIP message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- HTTP message syntax
- sdp = session description protocol
- Call-ID is unique for every call.

- Here we don't know Bob's IP address.
  - intermediate SIP servers needed.
- Alice sends, receives SIP messages using SIP default port 506
- Alice specifies in header that SIP client sends, receives SIP messages over UDP

Name translation and user location

- caller wants to call callee, but only has callee's name or e-mail address.
- need to get IP address of callee's current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, PDA, car device)
- result can be based on:
  - time of day (work, home)
  - caller (don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

Service provided by SIP servers:

- SIP registrar server
- SIP proxy server

SIP Registrar

- when Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server
  (similar function needed by Instant Messaging)

Register Message:

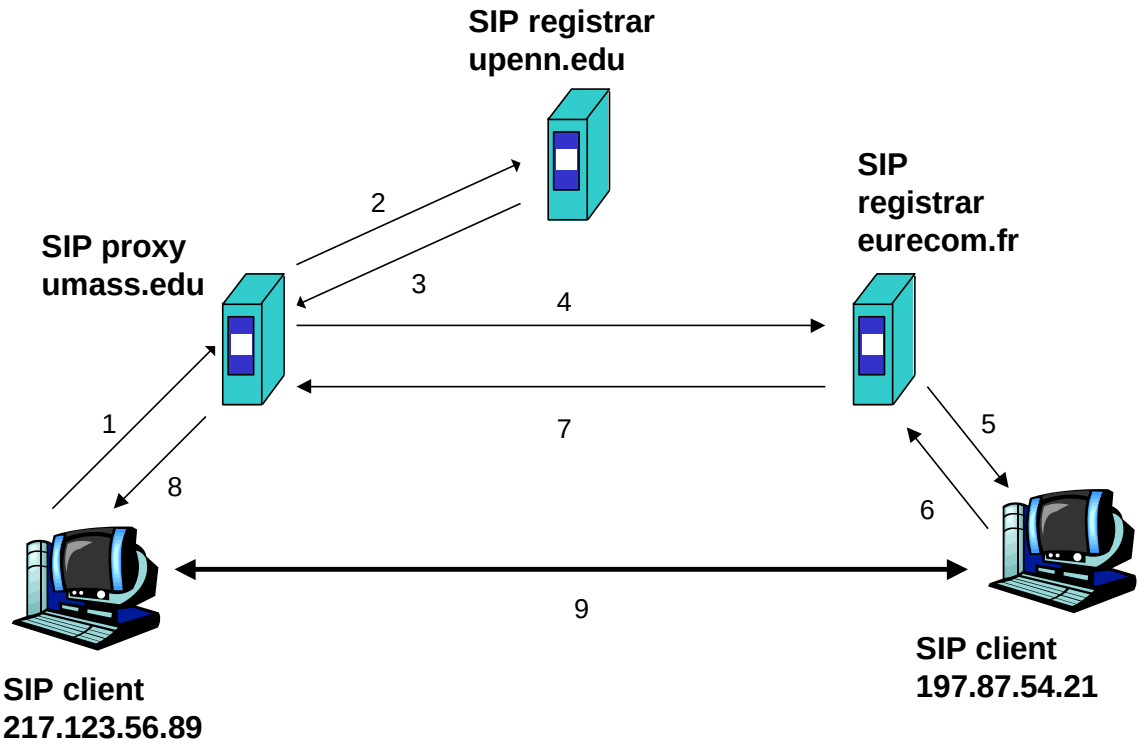
```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

SIP Proxy

- Alice sends invite message to her proxy server
  - contains address sip:bob@domain.com
- proxy responsible for routing SIP messages to callee
  - possibly through multiple proxies.
- callee sends response back through the same set of proxies.
- proxy returns SIP response message to Alice
  - contains Bob's IP address
- proxy analogous to local DNS server

Example

Caller jim@umass.edu
with places a
call to keith@upenn.edu



(1) Jim sends INVITE message to umass SIP proxy. (2) Proxy forwards request to upenn registrar server.
(3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr

(4) umass proxy sends INVITE to eurecom registrar. (5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client. (6-8) SIP response sent back (9) media sent directly between clients.

Note: also a SIP ack message, which is not shown.

Comparison with H.323

- H.323 is another signaling protocol for real-time, interactive
- H.323 is a complete, vertically integrated suite of protocols for multimedia conferencing: signaling, registration, admission control, transport, codecs
- SIP is a single component. Works with RTP, but does not mandate it. Can be combined with other protocols, services
- H.323 comes from the ITU (telephony).
- SIP comes from IETF: Borrows much of its concepts from HTTP
  - SIP has Web flavor, whereas H.323 has telephony flavor.
- SIP uses the KISS principle: Keep it simple stupid.

Chapter 7 outline

7.1 multimedia networking applications

7.2 streaming stored audio and video

7.3 making the best out of best effort service

7.4 protocols for real-time interactive applications

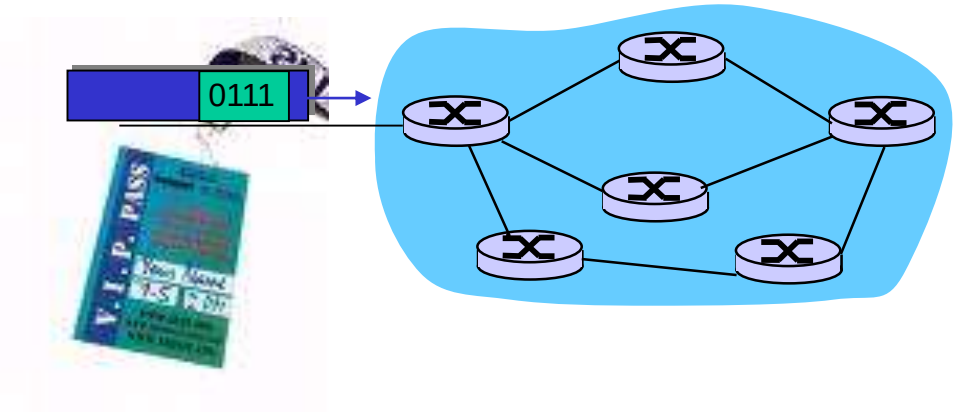
RTP, RTCP, SIP

7.5 providing multiple classes of service

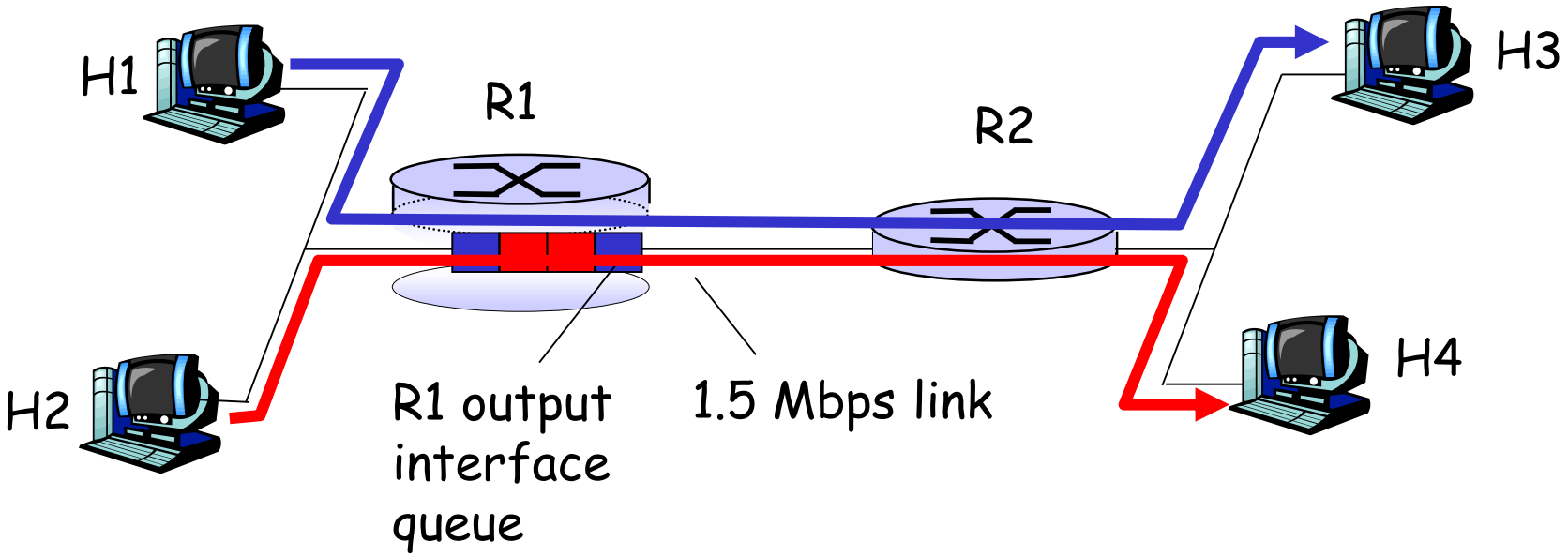
7.6 providing QoS guarantees

Providing Multiple Classes of Service

- thus far: making the best of best effort service
  - one-size fits all service model
- alternative: multiple classes of service
  - partition traffic into classes
  - network treats different classes of traffic differently (analogy: VIP service vs regular service)
- granularity: differential service among multiple classes, not among individual connections
- history: ToS bits

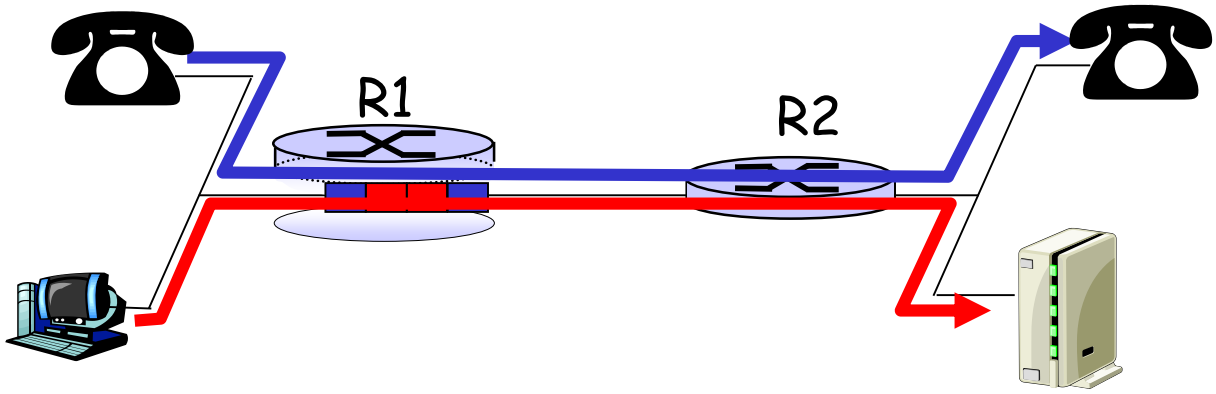


Multiple classes of service: scenario



Scenario 1: mixed FTP and audio

- Example: 1Mbps IP phone, FTP share 1.5 Mbps link.
  - bursts of FTP can congest router, cause audio loss
  - want to give priority to audio over FTP

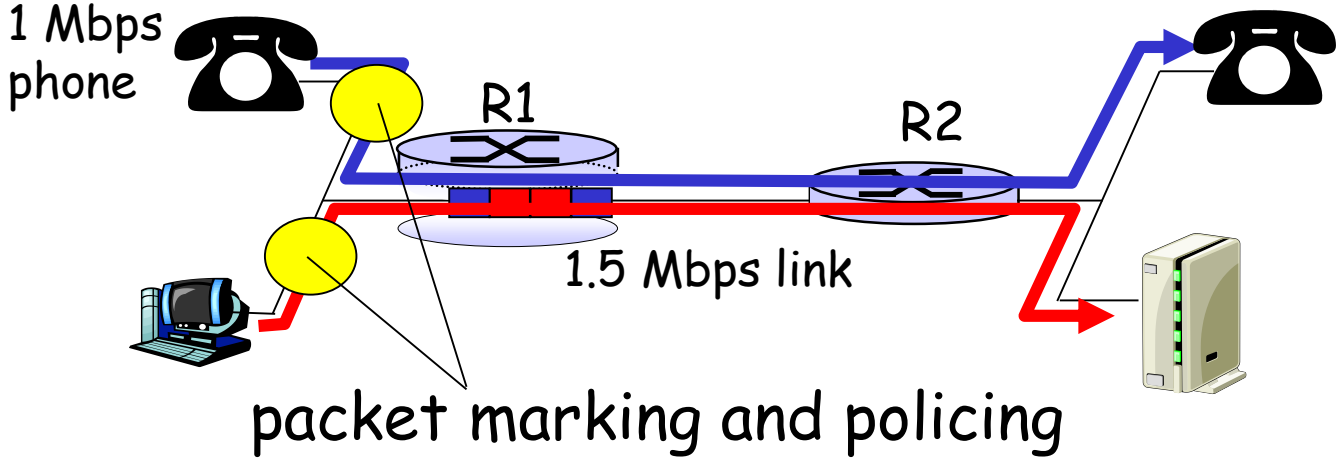


Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

Principles for QOS Guarantees (more)

- what if applications misbehave (audio sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- marking and policing at network edge:
  - similar to ATM UNI (User Network Interface)

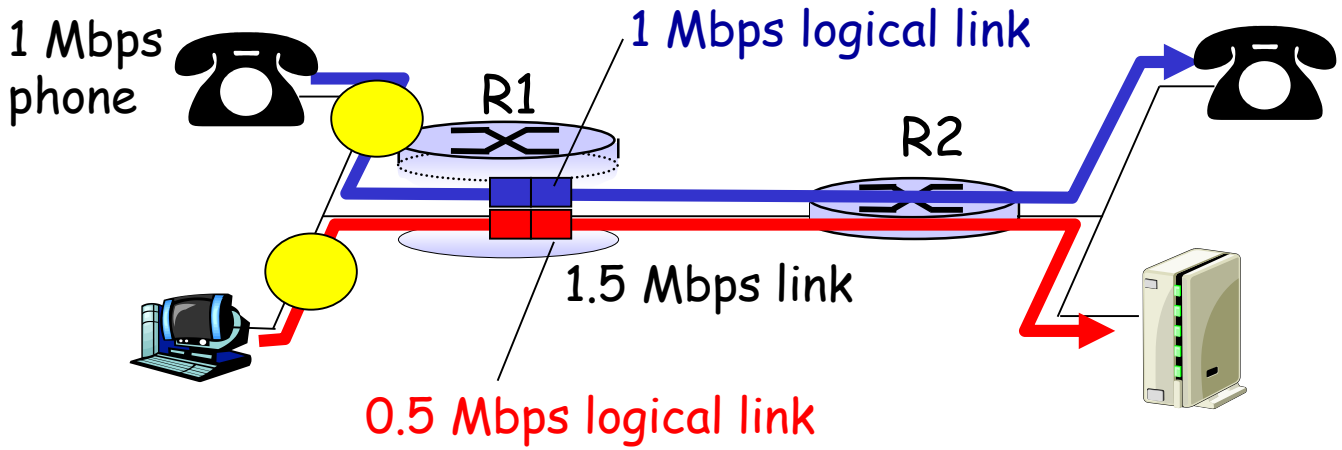


Principle 2

provide protection (*isolation*) for one class from others

Principles for QOS Guarantees (more)

- Allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flows doesn't use its allocation

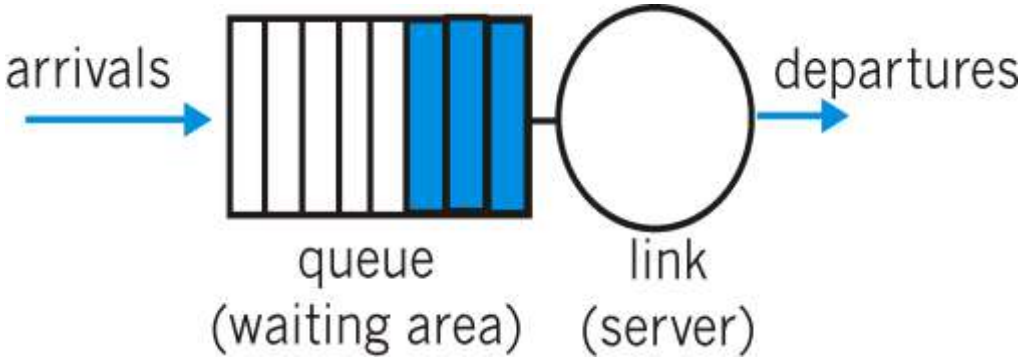


Principle 3

While providing isolation, it is desirable to use resources as efficiently as possible

Scheduling And Policing Mechanisms

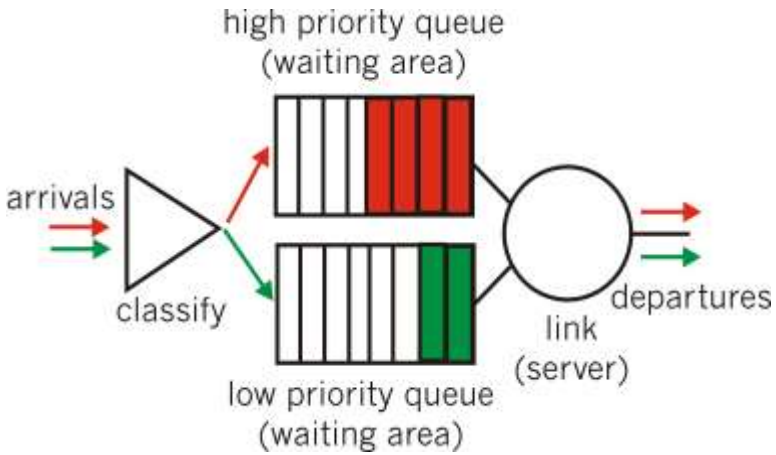
- scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - real-world example?
  - discard policy: if packet arrives to full queue: who to discard?
    - Tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly

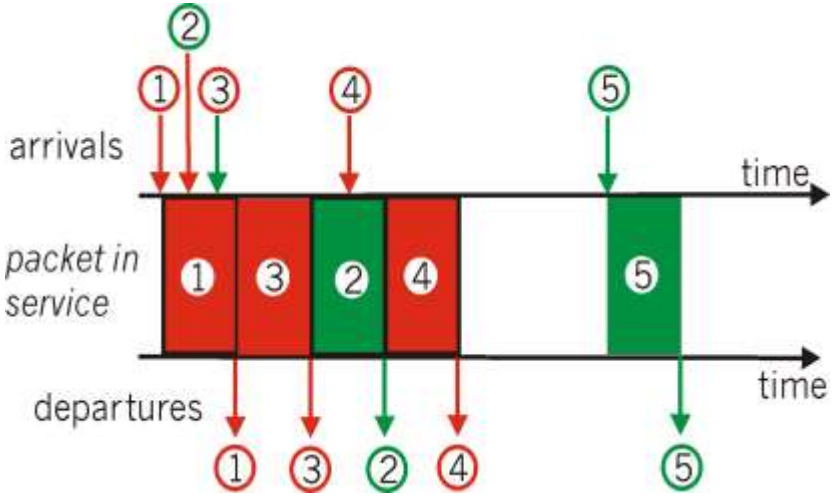


Scheduling Policies: more

Priority scheduling: transmit highest priority queued packet

- multiple *classes*, with different priorities
  - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..
  - Real world example?

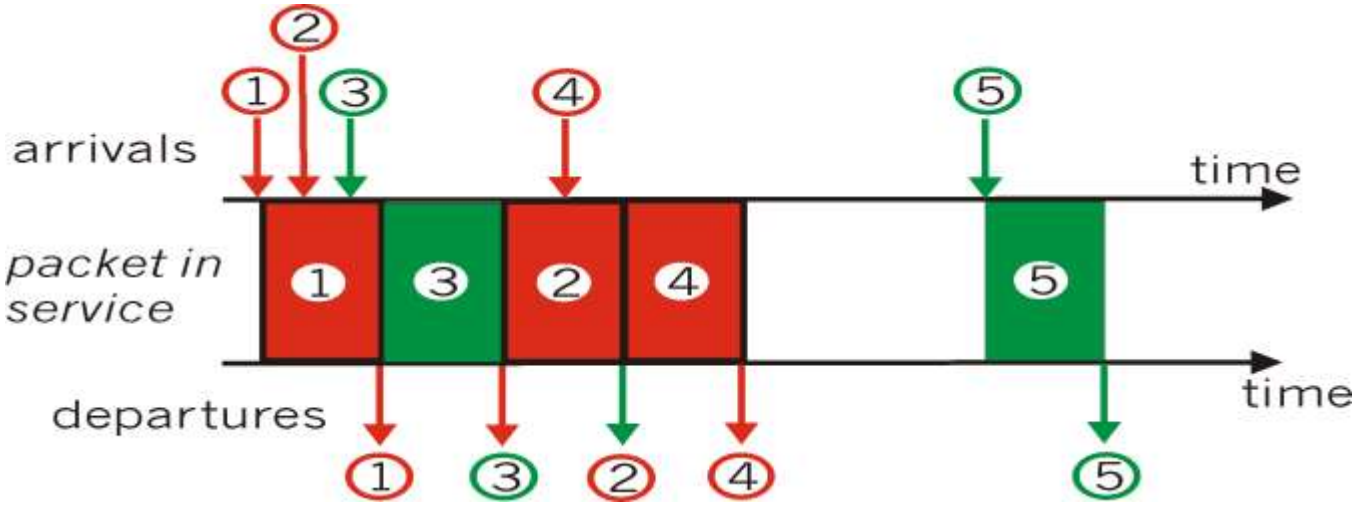




Scheduling Policies: still more

round robin scheduling:

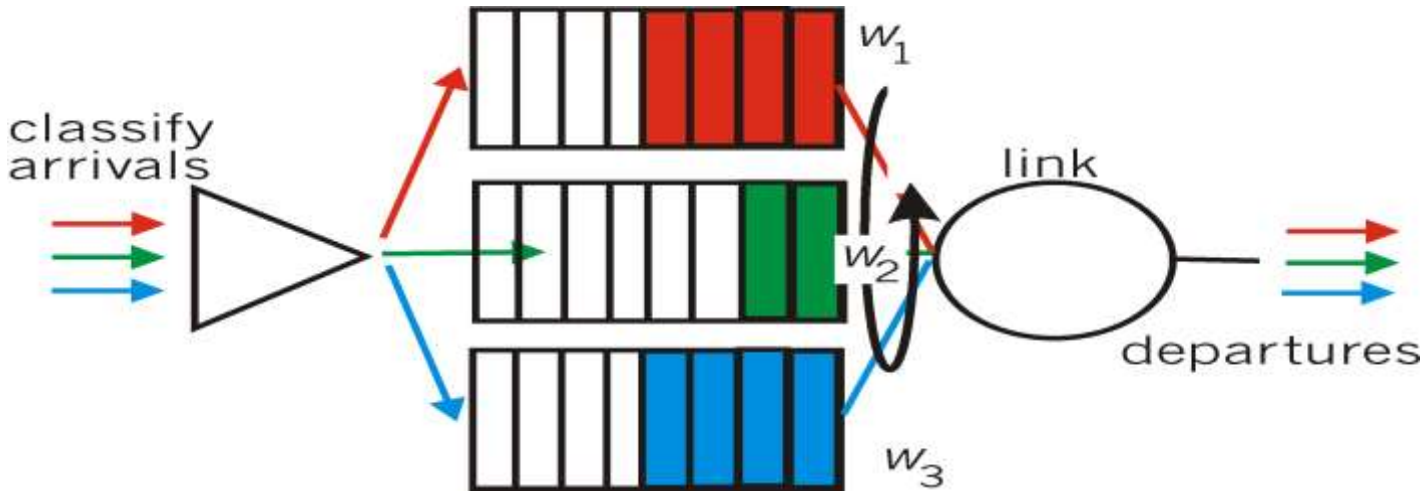
- multiple classes
- cyclically scan class queues, serving one from each class (if available)
- real world example?



Scheduling Policies: still more

Weighted Fair Queuing:

- generalized Round Robin
- each class gets weighted amount of service in each cycle
- real-world example?



Policing Mechanisms

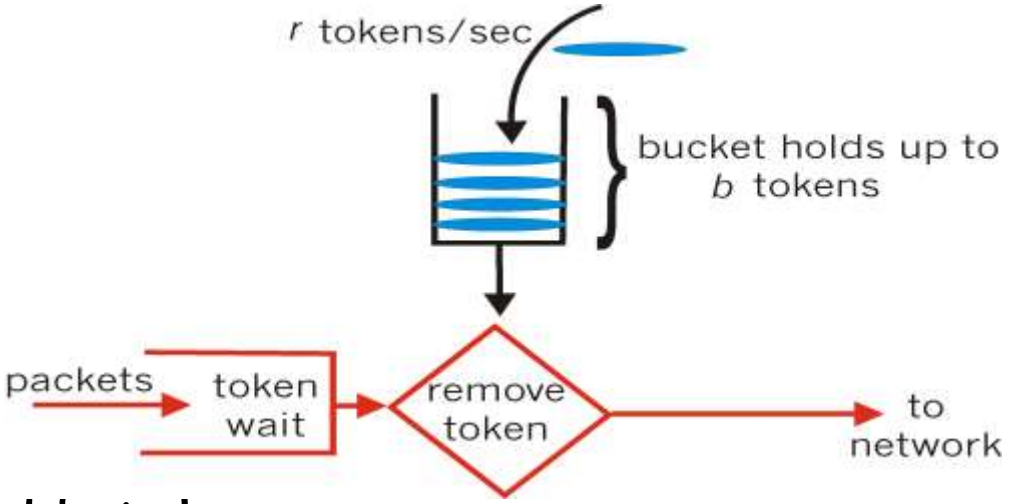
Goal: limit traffic to not exceed declared parameters

Three common-used criteria:

- *(Long term) Average Rate:* how many pkts can be sent per unit time (in the long run)
  - crucial question: what is the interval length: 100 packets per sec or 6000 packets per min have same average!
- *Peak Rate:* e.g., 6000 pkts per min. (ppm) avg.; 1500 ppm peak rate
- *(Max.) Burst Size:* max. number of pkts sent consecutively (with no intervening idle)

Policing Mechanisms

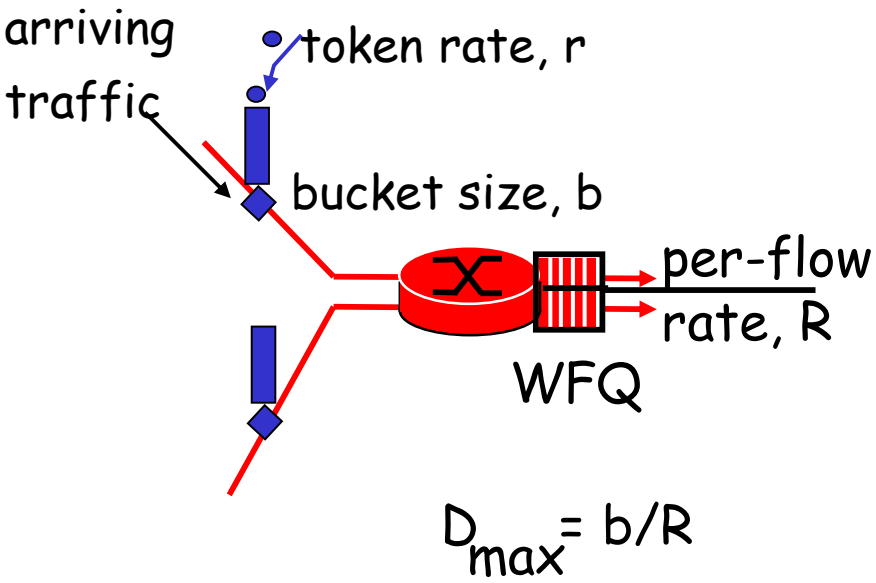
Token Bucket: limit input to specified Burst Size and Average Rate.

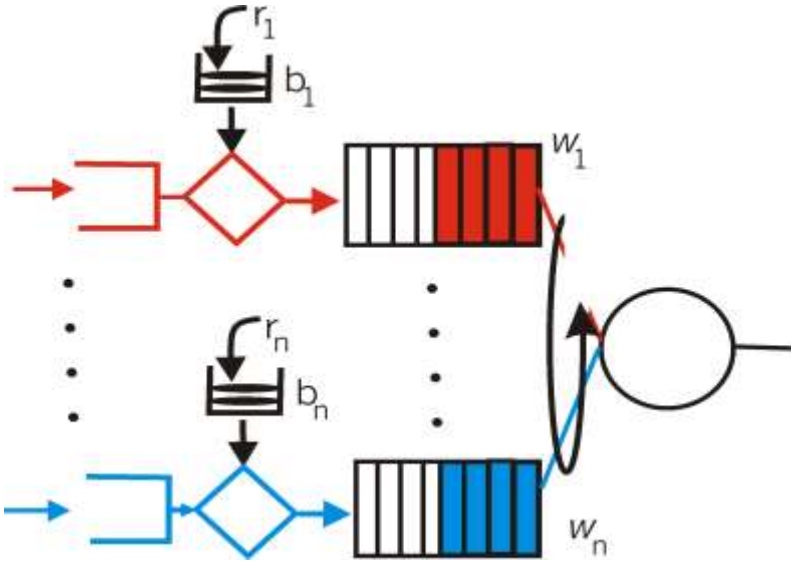


- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- *over interval of length t: number of packets admitted less than or equal to (r t + b).*

Policing Mechanisms (more)

- token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee*!





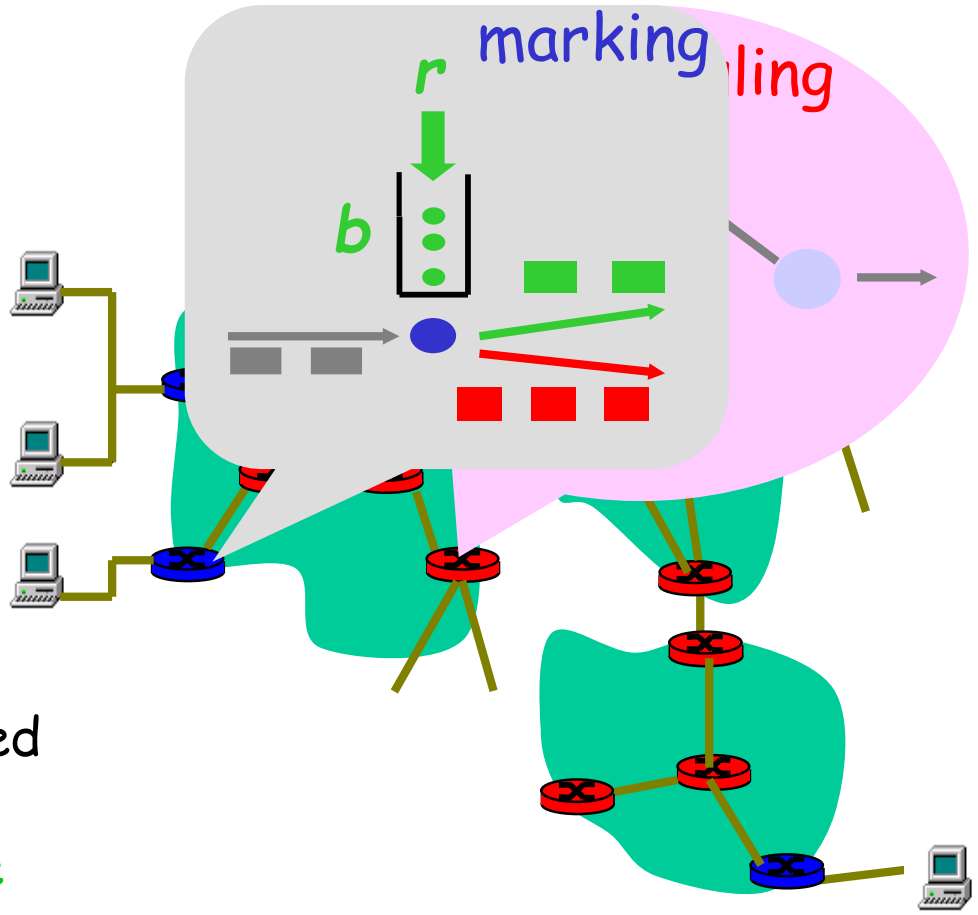
IETF Differentiated Services

- want "qualitative" service classes
  - "behaves like a wire"
  - relative service distinction: Platinum, Gold, Silver
- *scalability:* simple functions in network core, relatively complex functions at edge routers (or hosts)
  - signaling, maintaining per-flow router state difficult with large number of flows
- don't define define service classes, provide functional components to build service classes

Diffserv Architecture

Edge router:

- per-flow traffic management
- marks packets as in-profile and out-profile

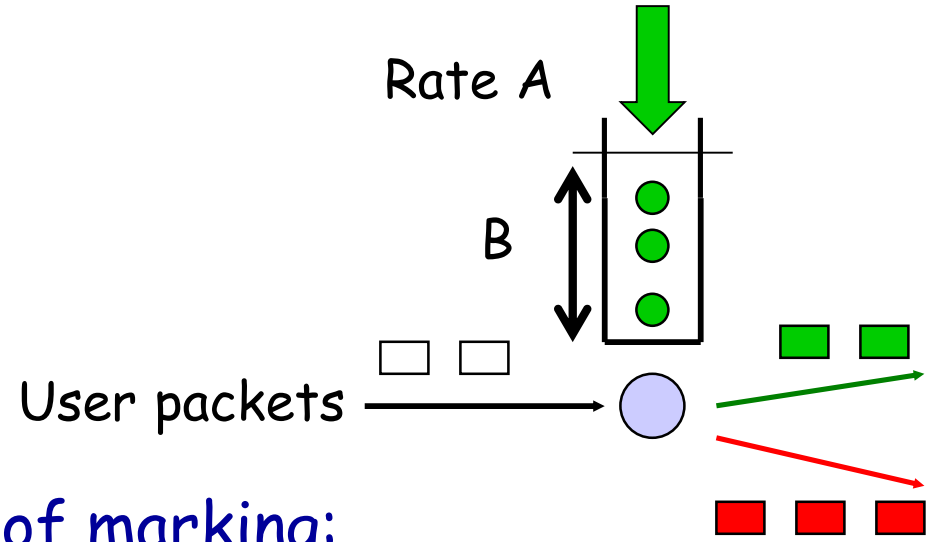


Core router:

- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets

Edge-router Packet Marking

- profile: pre-negotiated rate A, bucket size B
- packet marking at edge based on per-flow profile



Possible usage of marking:

- class-based marking: packets of different classes marked differently
- intra-class marking: conforming portion of flow marked differently than non-conforming one

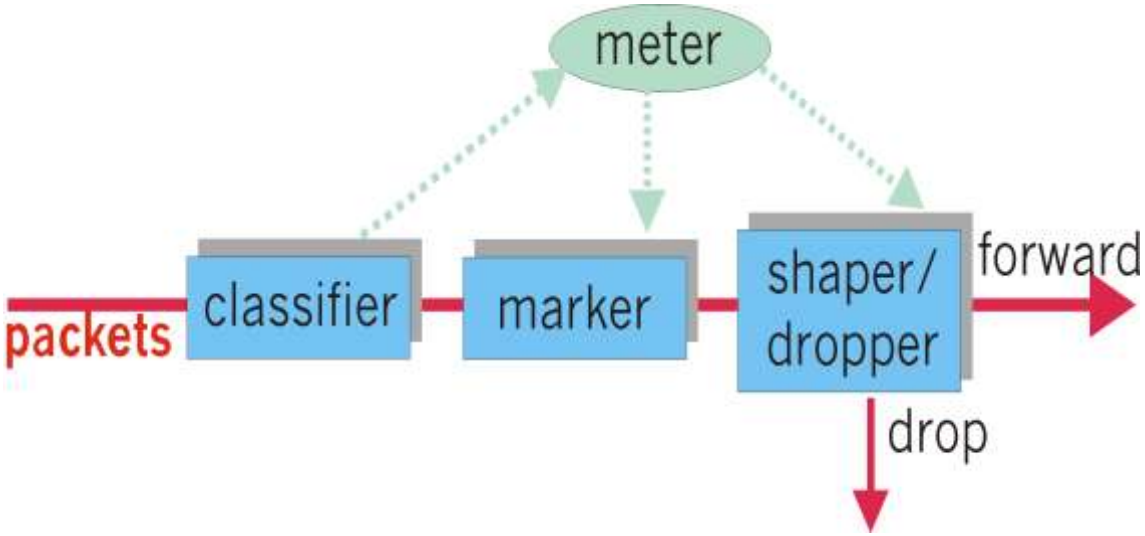
Classification and Conditioning

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused

Classification and Conditioning

may be desirable to limit traffic injection rate of some class:

- user declares traffic profile (e.g., rate, burst size)
- traffic metered, shaped if non-conforming



Forwarding (PHB)

- PHB result in a different observable (measurable) forwarding performance behavior
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B

Forwarding (PHB)

PHBs being developed:

- Expedited Forwarding: pkt departure rate of a class equals or exceeds specified rate
  - logical link with a minimum guaranteed rate
- Assured Forwarding: 4 classes of traffic
  - each guaranteed minimum amount of bandwidth
  - each with three drop preference partitions

Chapter 7 outline

7.1 multimedia networking applications

7.2 streaming stored audio and video

7.3 making the best out of best effort service

7.4 protocols for real-time interactive applications

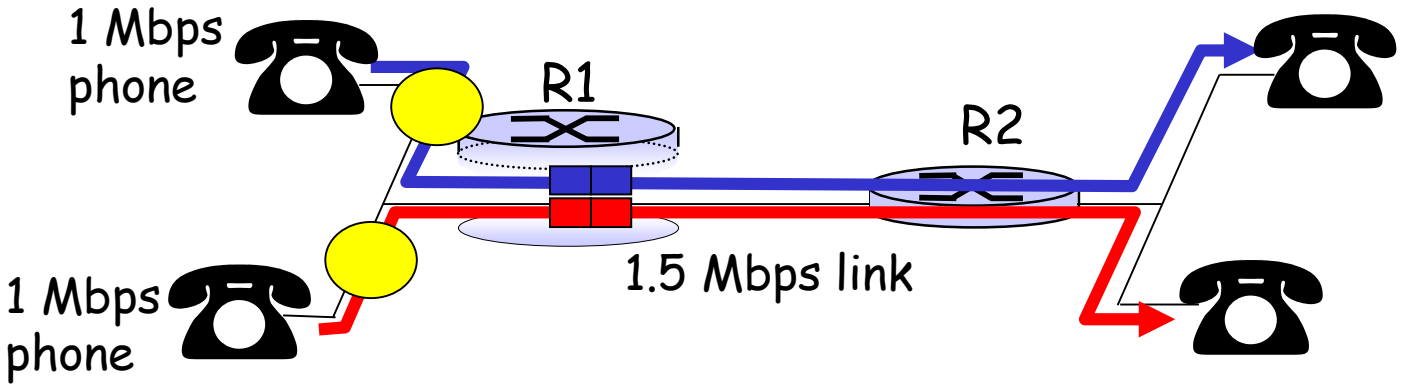
RTP, RTCP, SIP

7.5 providing multiple classes of service

7.6 providing QoS guarantees

Principles for QOS Guarantees (more)

- *Basic fact of life:* can not support traffic demands beyond link capacity

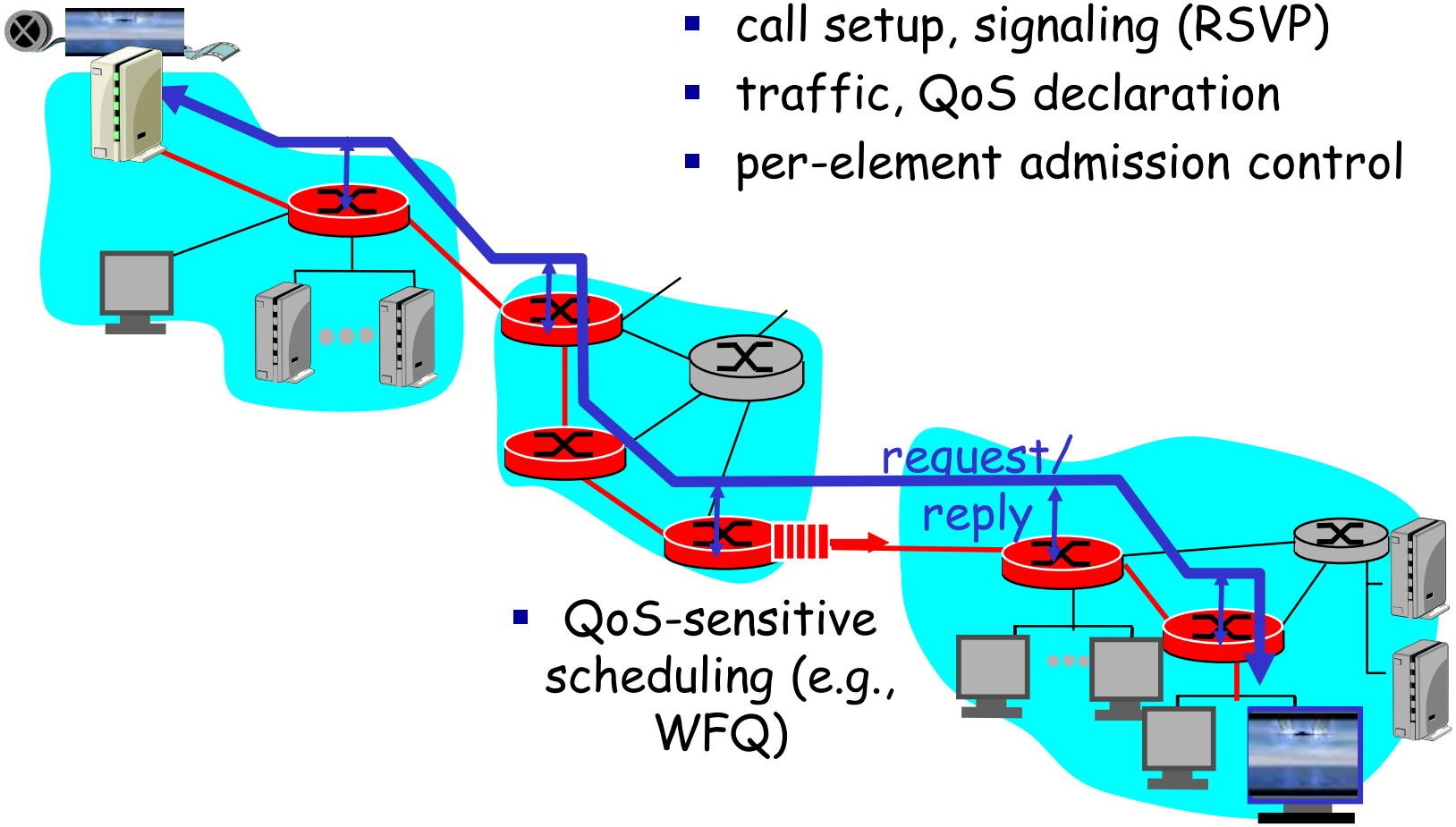


Principle 4

Call Admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

QoS guarantee scenario

- Resource reservation
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control



- QoS-sensitive scheduling (e.g., WFQ)

IETF Integrated Services

- architecture for providing QOS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info (a la VC) of allocated resources, QoS req's
- admit/deny new call setup requests:

Question: can newly arriving flow be admitted with performance guarantees while not violated QoS guarantees made to already admitted flows?

Call Admission

Arriving session must :

- declare its QOS requirement
  - R-spec: defines the QOS being requested
- characterize traffic it will send into network
  - T-spec: defines traffic characteristics
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP

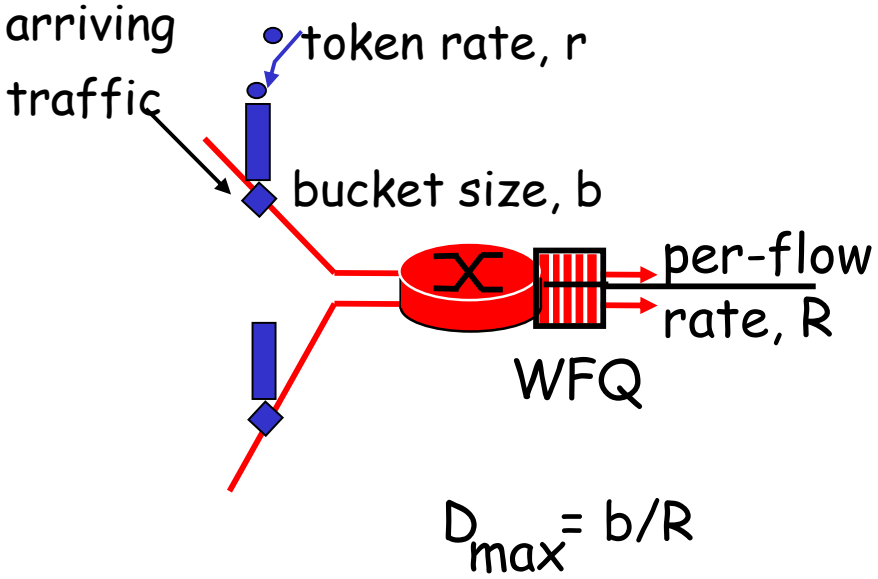
Intserv QoS: Service models [rfc2211, rfc 2212]

Guaranteed service:

- worst case traffic arrival: leaky-bucket-policed source
- simple (mathematically provable) *bound* on delay [Parekh 1992, Cruz 1988]

Controlled load service:

- "a quality of service closely approximating the QoS that same flow would receive from an unloaded network element."



Signaling in the Internet

connectionless (stateless) forwarding by IP routers + best effort service = no network signaling protocols in initial IP design

- New requirement: reserve resources along end-to-end path (end system, routers) for QoS for multimedia applications
- RSVP: Resource Reservation Protocol [RFC 2205]
  - " … allow users to communicate requirements to network in robust and efficient way." i.e., signaling !
- earlier Internet Signaling protocol: ST-II [RFC 1819]

RSVP Design Goals

1. accommodate heterogeneous receivers (different bandwidth along paths)
2. accommodate different applications with different resource requirements
3. make multicast a first class service, with adaptation to multicast group membership
4. leverage existing multicast/unicast routing, with adaptation to changes in underlying unicast, multicast routes
5. control protocol overhead to grow (at worst) linear in # receivers
6. modular design for heterogeneous underlying technologies

RSVP: does not…

- specify how resources are to be reserved
  - rather: a mechanism for communicating needs
- determine routes packets will take
  - that's the job of routing protocols
  - signaling decoupled from routing
- interact with forwarding of packets
  - separation of control (signaling) and data (forwarding) planes

RSVP: overview of operation

- senders, receiver join a multicast group
  - done outside of RSVP
  - senders need not join group
- sender-to-network signaling
  - *path message:* make sender presence known to routers
  - path teardown: delete sender's path state from routers
- receiver-to-network signaling
  - *reservation message:* reserve resources from sender(s) to receiver
  - reservation teardown: remove receiver reservations
- network-to-end-system signaling
  - path error
  - reservation error

Chapter 7: Summary

Principles

- classify multimedia applications
- identify network services applications need
- making the best of best effort service

Protocols and Architectures

- specific protocols for best-effort
- mechanisms for providing QoS
- architectures for QoS
  - multiple classes of service
  - QoS guarantees, admission control