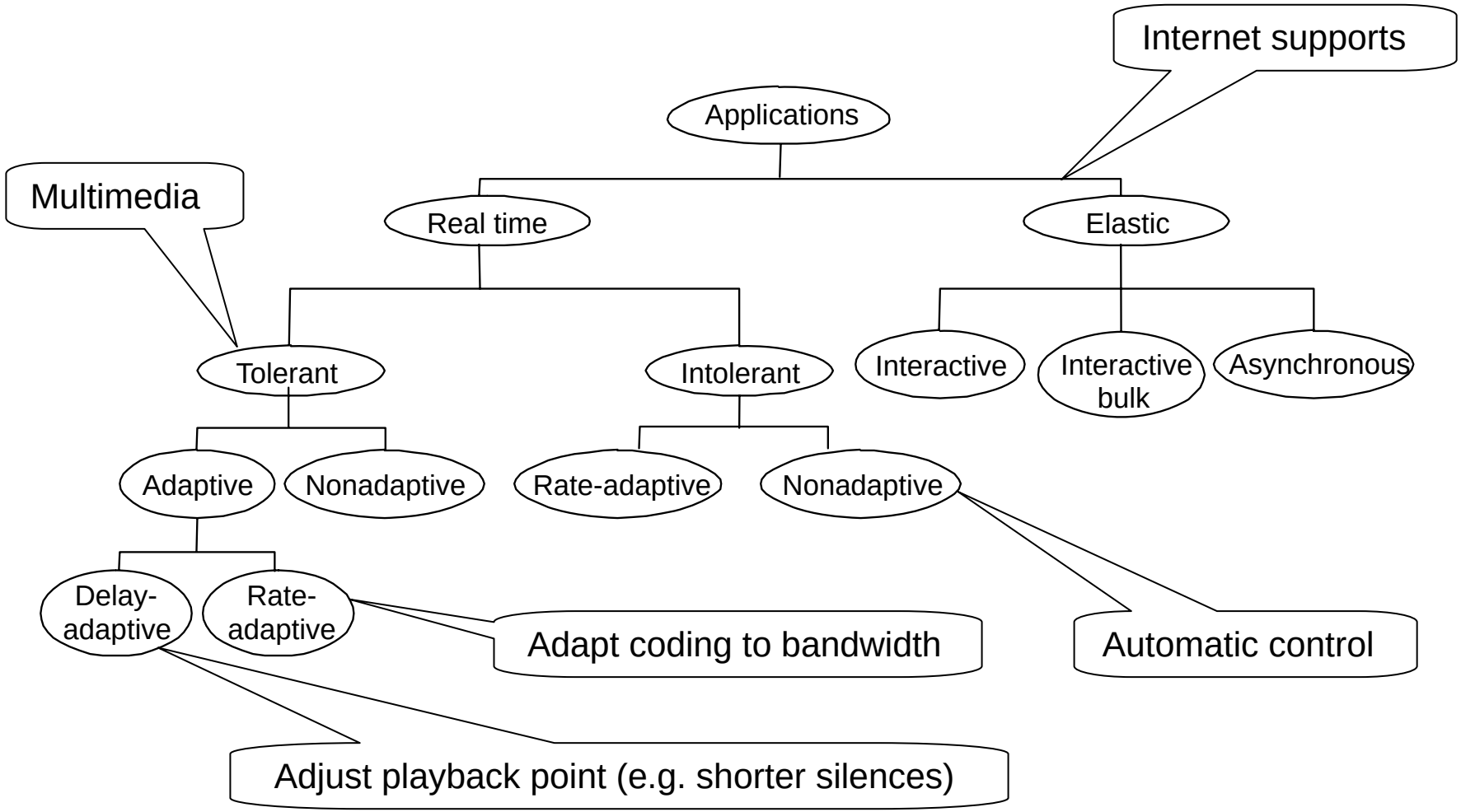
Distributed Multimedia Systems

5. Multimedia Networking

6.1. Quality of Service in Networks

- Raw bandwidth does not suffice
  - E.g. telephone over satellite
    - Enough bandwidth, but respond not immediate
- Non-real-time data, often also called *elastic*
  - Email, ftp, telnet, Web browsing via http
  - Increasing need for timeliness
- Real-time data
  - “Delivery on time” assurances from *inside* the network
  - Huge amount of data and continuous delivery
  - Late data are *useless*
- Fine-grained vs. coarse-grained QoS support
  - E.g. integrated vs. differentiated services

Taxonomy of Applications



MM Networking Applications

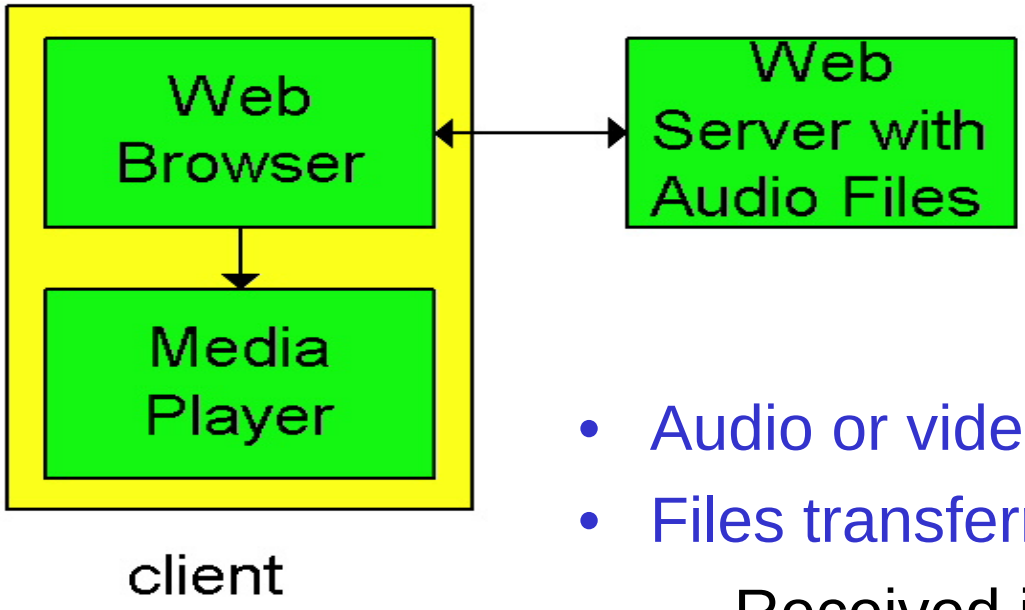
Classes of MM applications:

1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video

Fundamental characteristics:

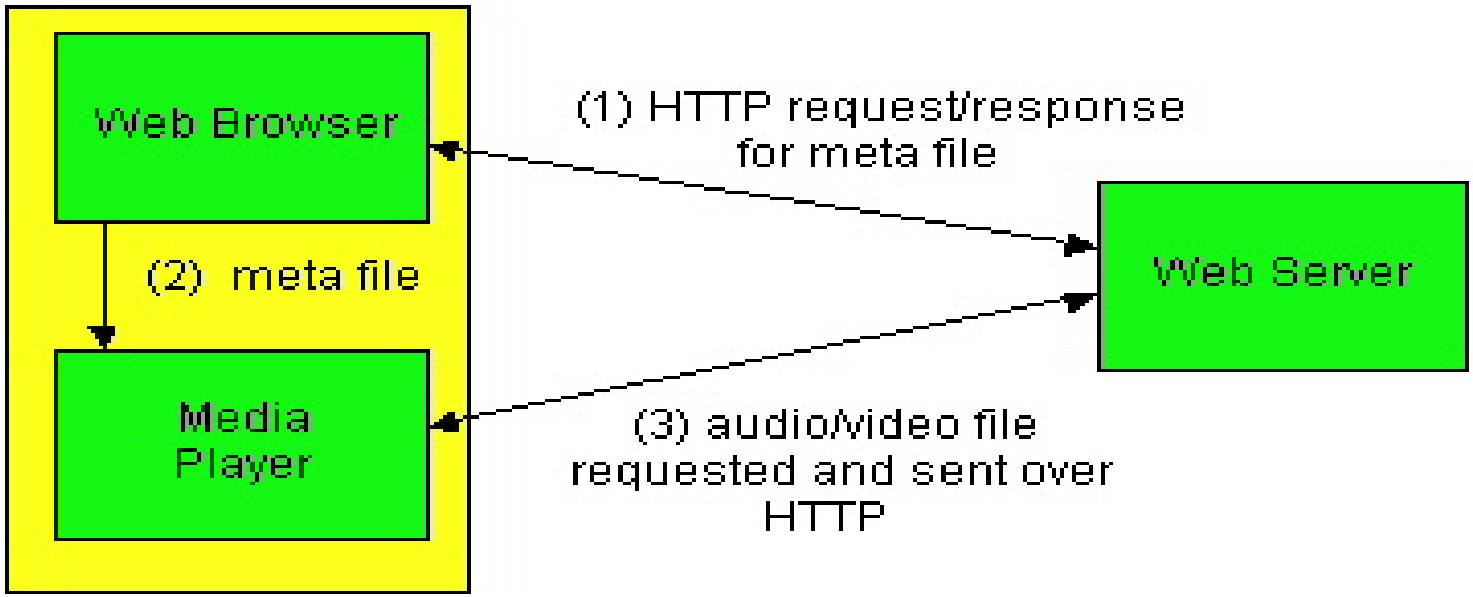
- Typically *delay sensitive*
  - end-to-end delay and *jitter*
- But *loss tolerant*
  - infrequent losses cause minor glitches
- Antithesis of data, which are *loss intolerant* but *delay tolerant.*

Internet multimedia: download and play



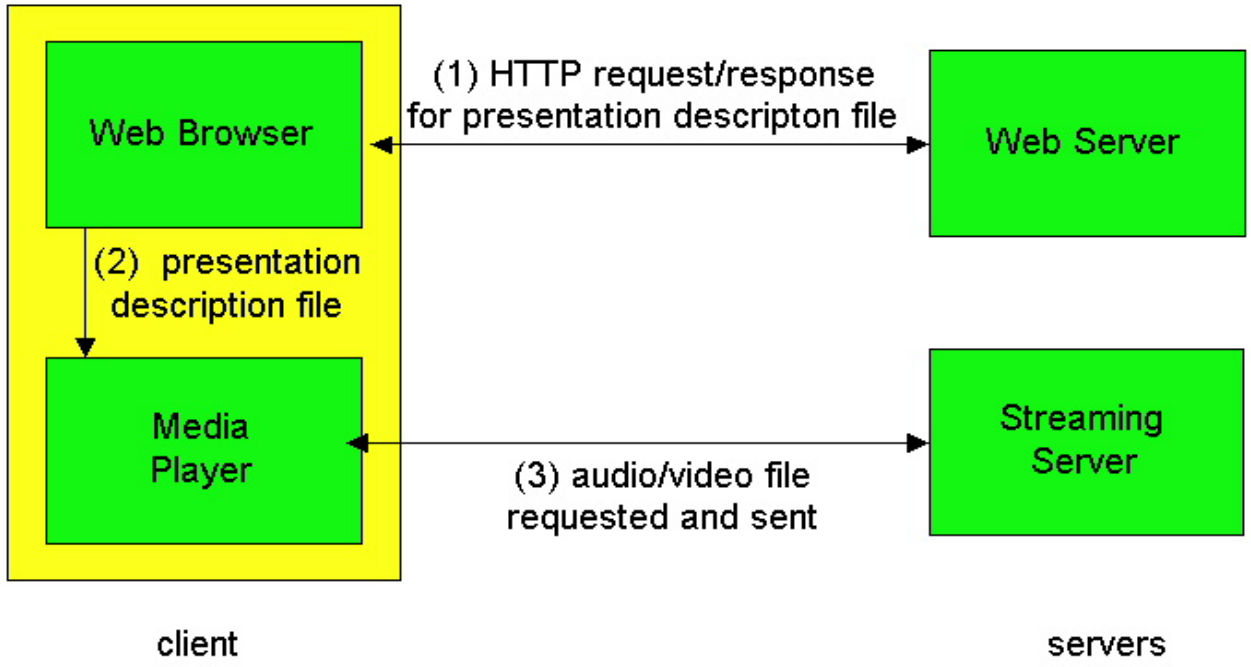
- Audio or video stored in file
- Files transferred as HTTP object
  - Received in entirety at client
  - Then passed to player
- Audio, video not streamed
  - Long start-up delay until playout!

Internet multimedia: progressive download



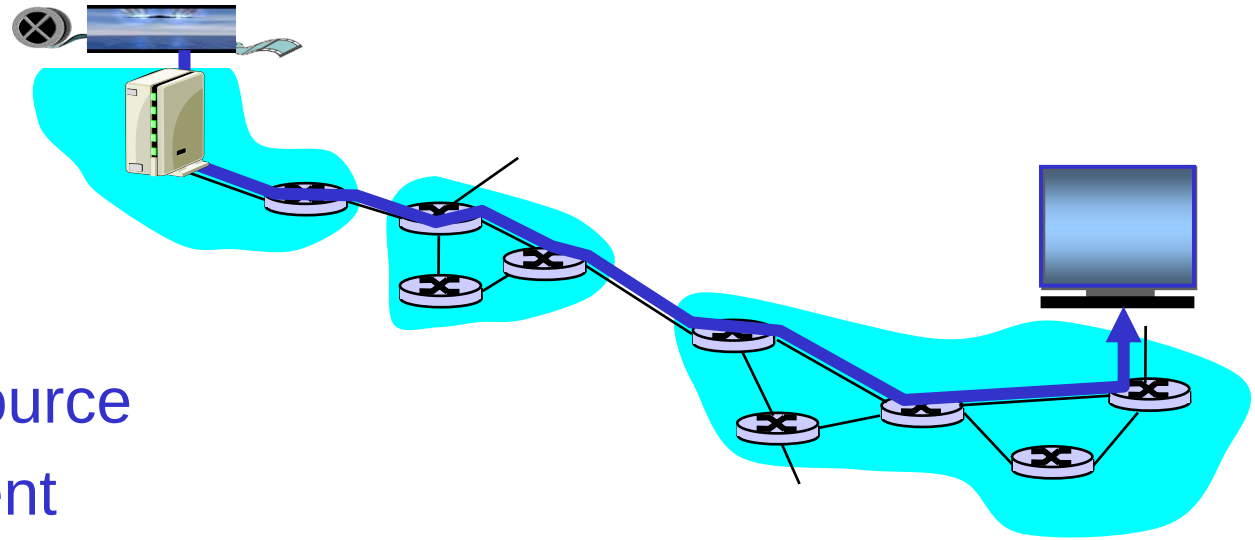
- Browser GETs a *metafile*, containing an URL to the a/v file
- Browser launches player, passing metafile
- Player contacts server via TCP + HTTP
- Player organizes “streaming” via looped HTTP-GETs

Streaming from a streaming server



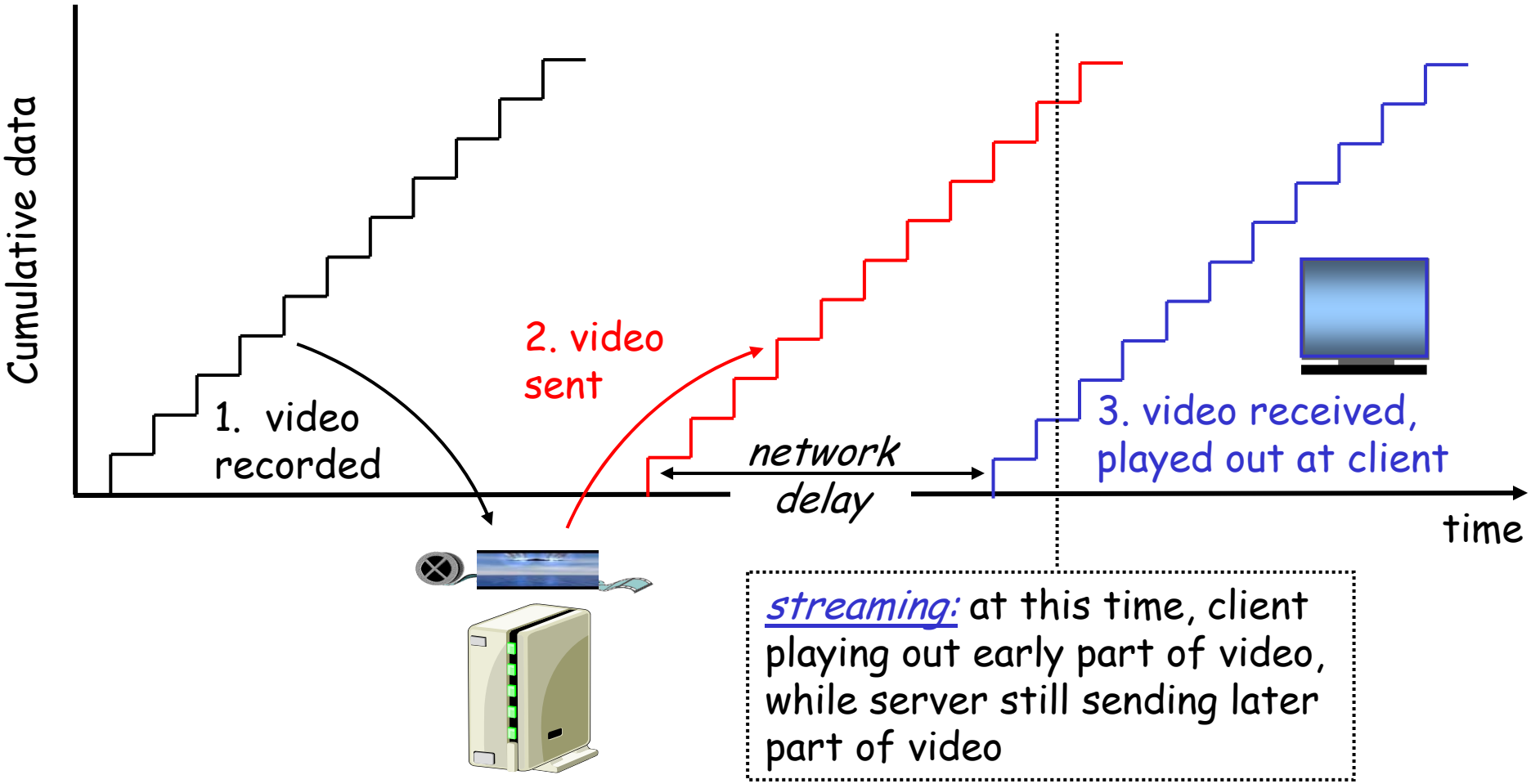
- This architecture allows for non-HTTP protocol between server and media player
- Can use e.g. RTSP+RTP+UDP (see later)

Streaming Stored Multimedia + Interactivity

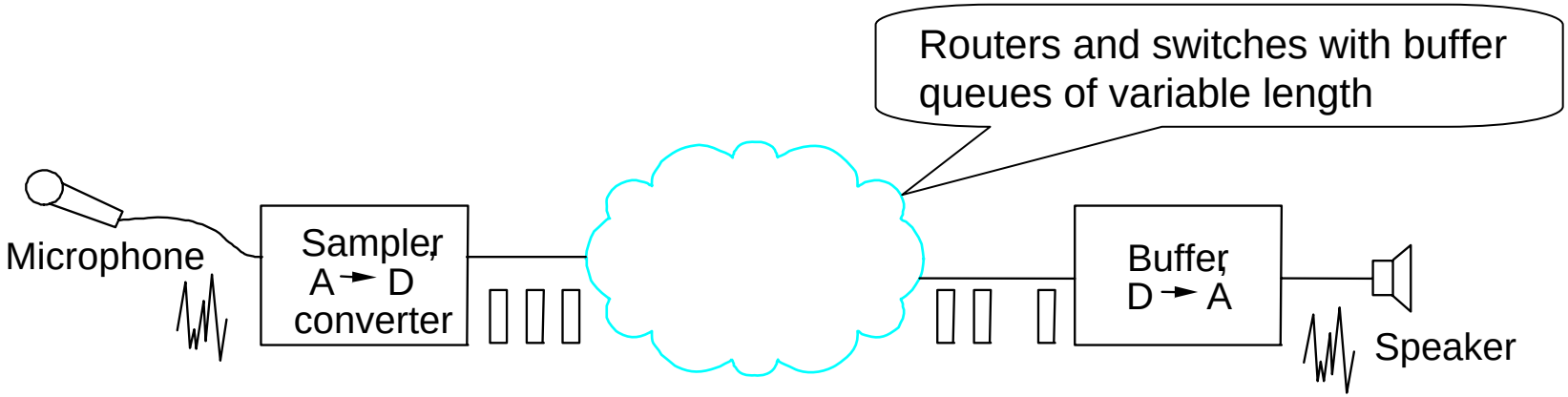


- Media stored at source
- Transmitted to client
- Client playout begins *before* all data has arrived
  - timing constraint: in time for playout
- VCR-like functionality: client can pause, rewind, FF…
  - 10 sec initial delay is generally accepted as OK
  - 1-2 sec until command effect is generally accepted as OK

Phases for Streaming for Stored

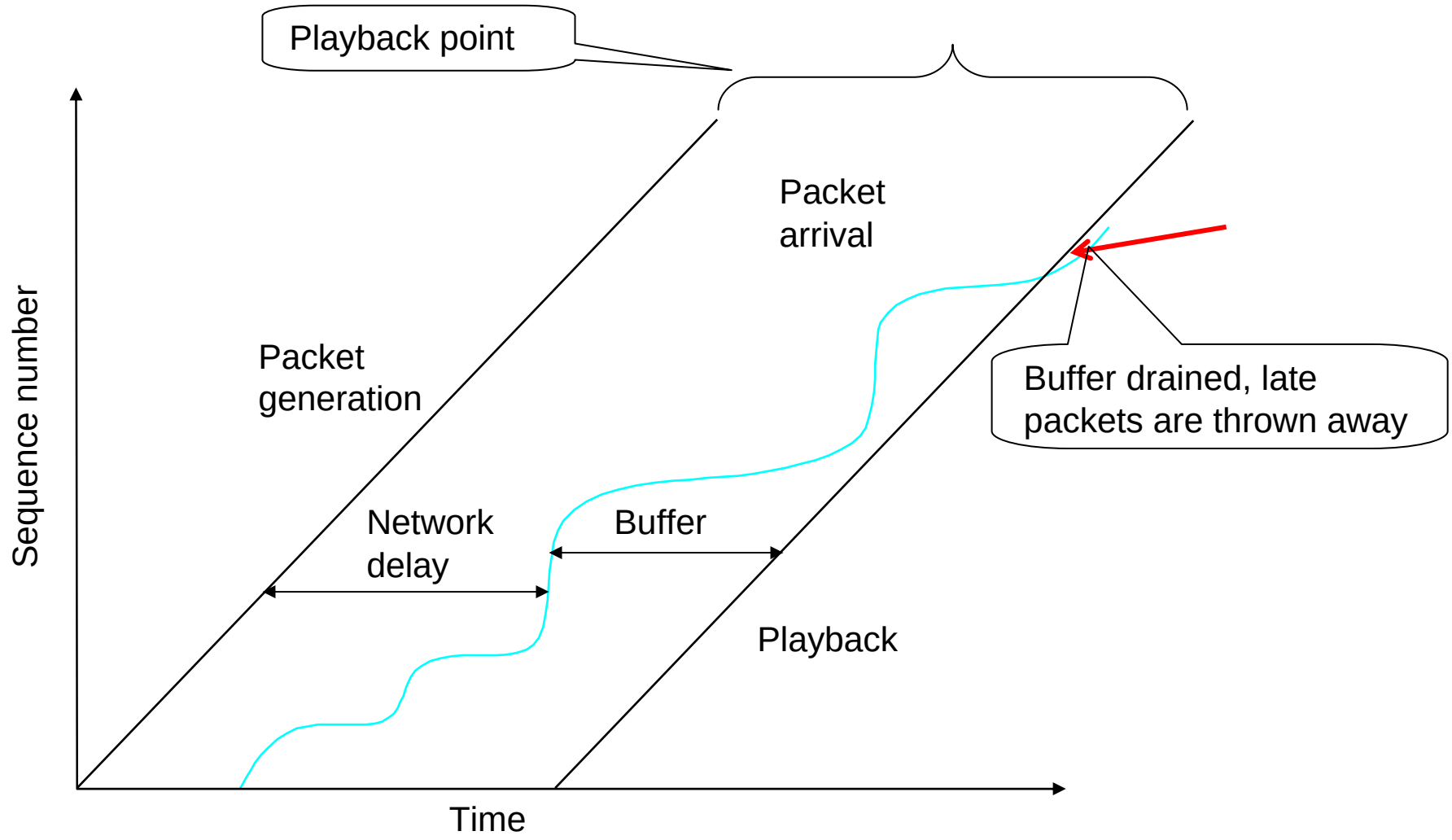


Example audio application

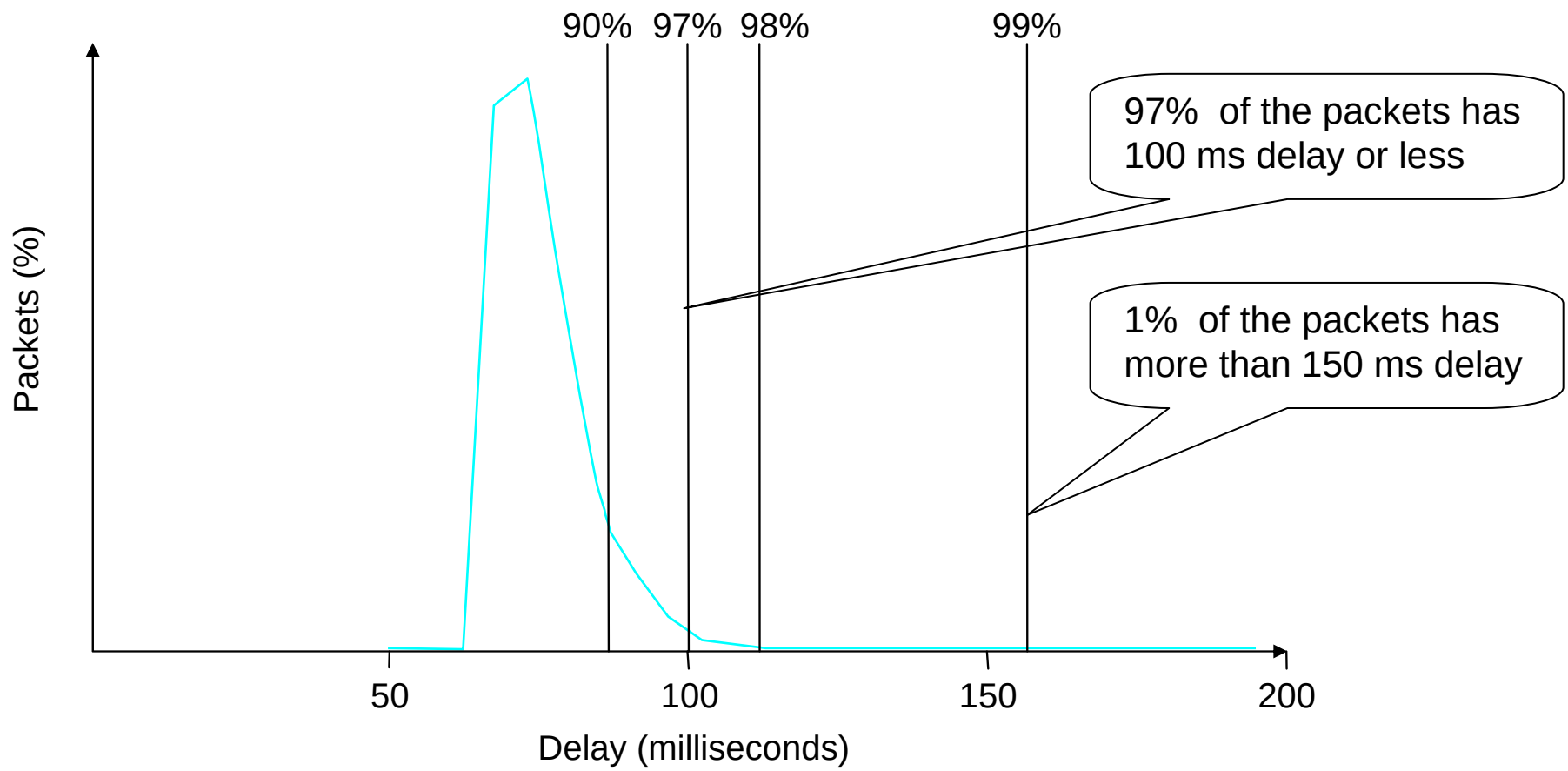


- Sample voice once every 125µs
- Each sample has a *playback time*
- Packets experience variable delay in network
- Add constant “insurance” factor to playback time by buffering data in the receiver: *playback point (interval)*
- For voice, data arrival within 150 ms is good, up to 3-400 ms tolerable

Playback buffer



Distribution of Delays on the Internet



How should Internet support better MM?

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

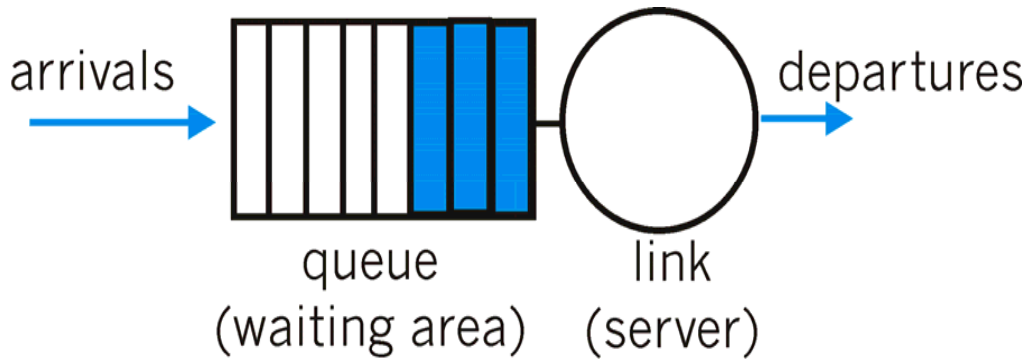
- No major changes
- More bandwidth when needed
- Content distribution, application-layer multicast
  - application layer

Differentiated services philosophy:

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

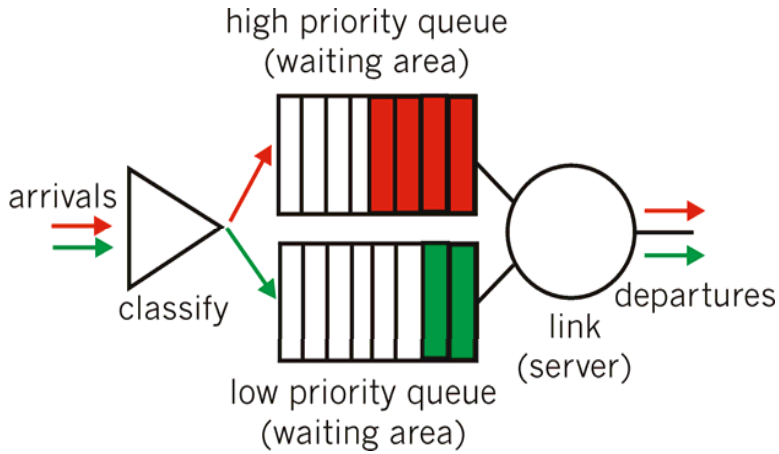
6.1.1. Queuing Disciplines in Routers (1)

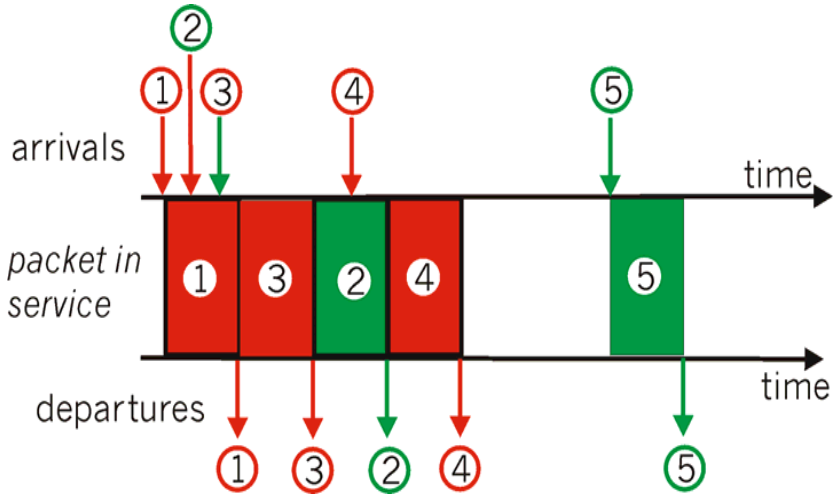
- FIFO (or FCFS) – used in most routers
  - No difference between different traffic sources
  - Can be *unfair* among applications
- Discard policy: If queue full, who to discard?
  - Tail drop: drop arriving packet
  - Priority: drop/ remove on priority basis
  - Random: drop/ remove randomly



Queuing Disciplines in Routers (2)

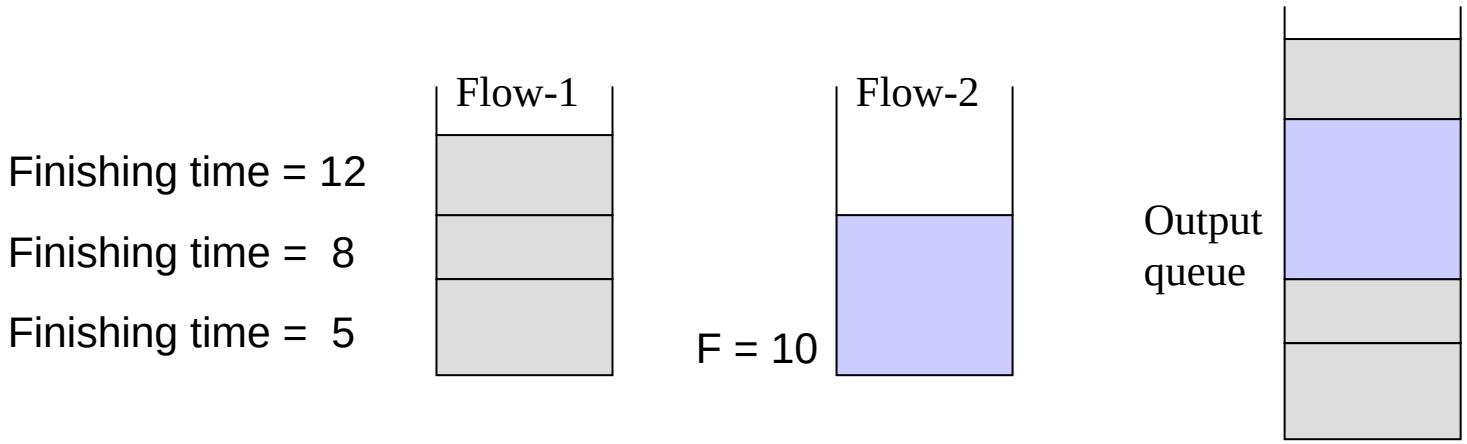
- Priority Queuing
  - Priority class sent e.g. in the TOS (type of service) field
  - Each class has its own FIFO queue
  - Limited usage: starvation danger for low priority classes
  - Especially important packets can be protected
    - E.g. packets for updating the routing tables after a change





Queuing Disciplines in Routers (3)

- Fair Queuing – fair also for many flows
  - Every flow hat its own FIFO queue
  - Simple round robin would be unfair to short packets
  - "Bit-by-bit" round robin among the individual queues
    - Approximation: *Earliest finishing time* sent first + no preemption
  - Work conserving – no waste of bandwidth
  - Guaranteed minimum share of 1/nth of the bandwidth

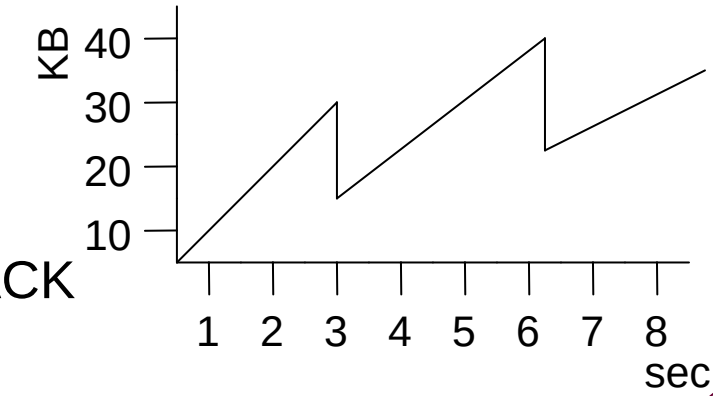


Queuing Disciplines in Routers (4)

- Weighted Fair Queuing
  - A queue with a higher weight gets a higher share
  - $Share_i = R * w_i / (\sum w_j)$ (R: rate of link in packets/sec)
  - E.g. 3 queues, with $W_1 = 2$, $W_2 = 1$, $W_3 = 3$
  - The share of $W_1 = 1/3$, $W_2 = 1/6$, and $W_3 = 1/2$
  - Through the same weight, flows can form a *class*
  - Weights can be assigned statically or signaled e.g. in the TOS field of the IP header (Differentiated Services)
  - Kind of “reservation”
- Separation of policy and mechanism
  - The same mechanism (e.g. WFQ) can be used for different policies

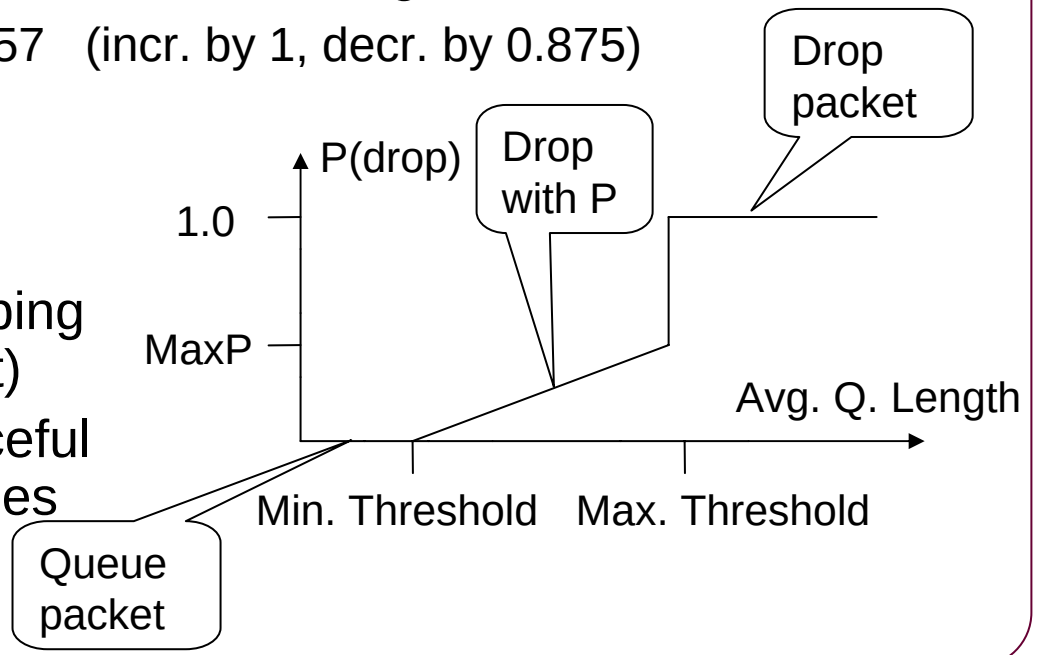
Congestion Control – bad for MM

- No congestion control in the 70ties, early 80ties
  - Congestion → Time-out at hosts → *Retransmission* → Still more congestion → Collapse
- TCP congestion control
  - Congestion → Time-out at hosts → *Slow down* → Recovery from the congestion
  - Exactly the wrong strategy for continuous media
- Additive Increase / Multiplicative Decrease
  - The congestion window (cw)
    - Must be “learned”, e.g:
    - Halved on every packet loss (min 1)
    - Enlarged in additive steps on each ACK
  - Saw tooth pattern



Congestion Avoidance

- Tries to avoid congestion instead of detecting it
- DECbit (destination echos bit back to source)
  - Routers set the congestion bit, if the queue length ≥ 1
  - This bit returns to the sender with the ACK packet
  - If the sender receives < 50% ACKs with congestion bit set:
    - cw++, otherwise cw -= 0.857 (incr. by 1, decr. by 0.875)
- Random Early Detection (RED) – for TCP use
  - Implicit notification by dropping a packet (enforced time-out)
  - Drop probability grows graceful between two threshold values
- Still bad for a/v streams

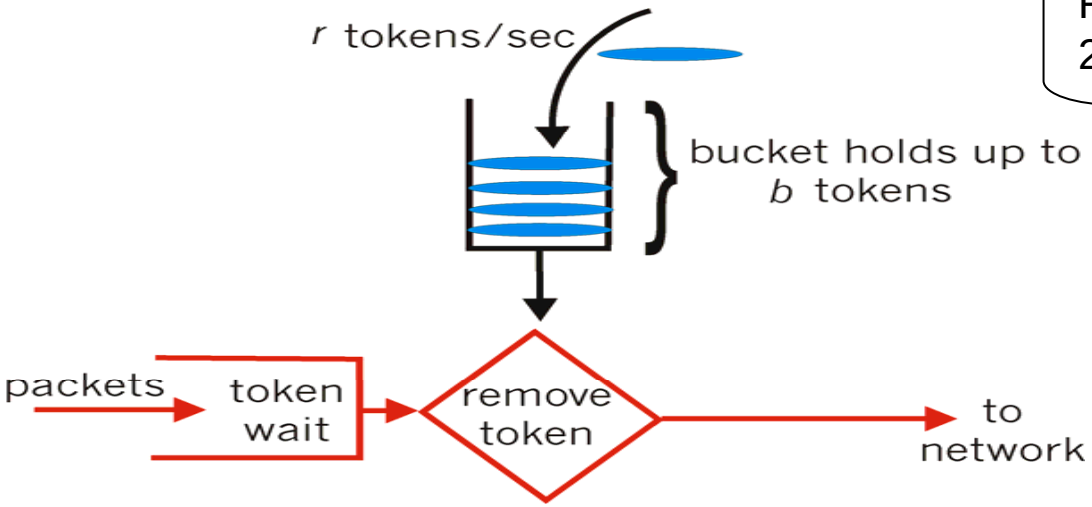


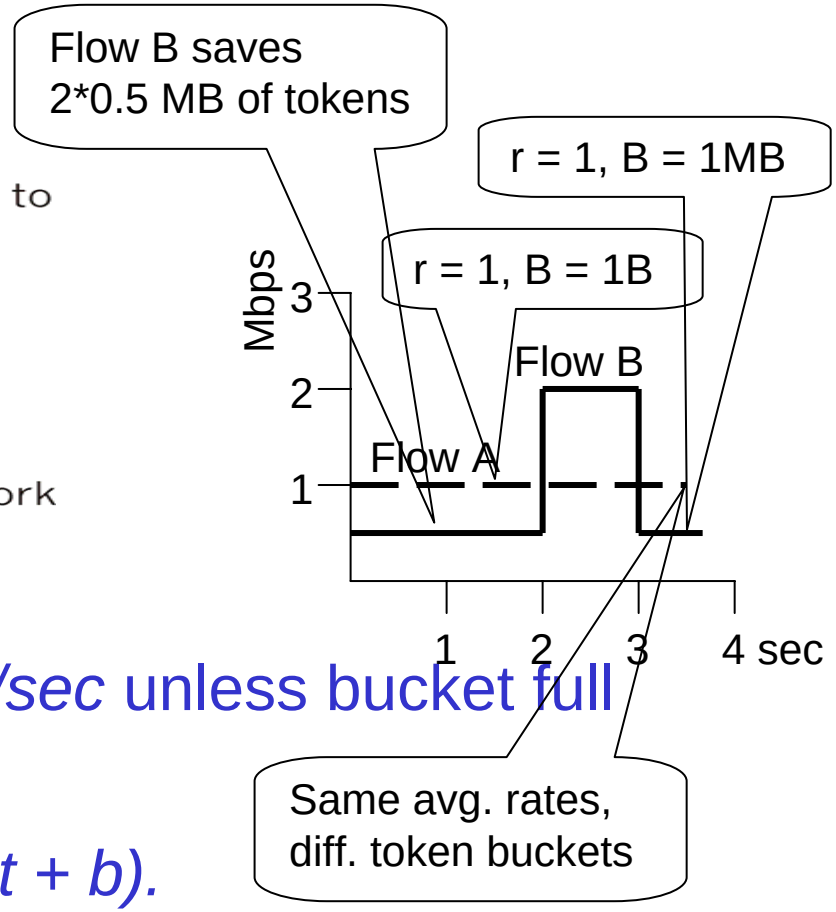
Policing Mechanisms

- Goal: limit traffic to not exceed declared parameters
- Three common-used criteria:
  - Average Rate (long term): how many packets can be sent per unit time (in the long run)
    - Crucial question: What is the interval length?
    - 100 packets / sec or 6000 packets / min have same average!
    - 100 packets / sec is a stronger constraint!
  - Peak Rate
    - E.g., 6000 packets / min. (ppm) avg.; 1500 pps peak rate
  - (Max.) Burst Size
    - Max. number of pkts sent consecutively (with no intervening idle)

Leaky Token Bucket

- Limit input to specified Burst Size (b) and Average Rate (r)

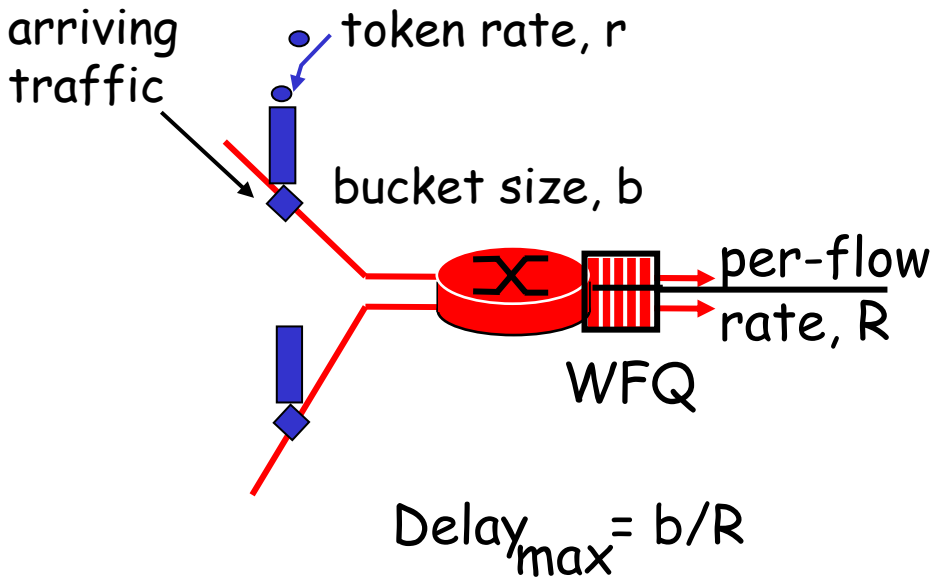


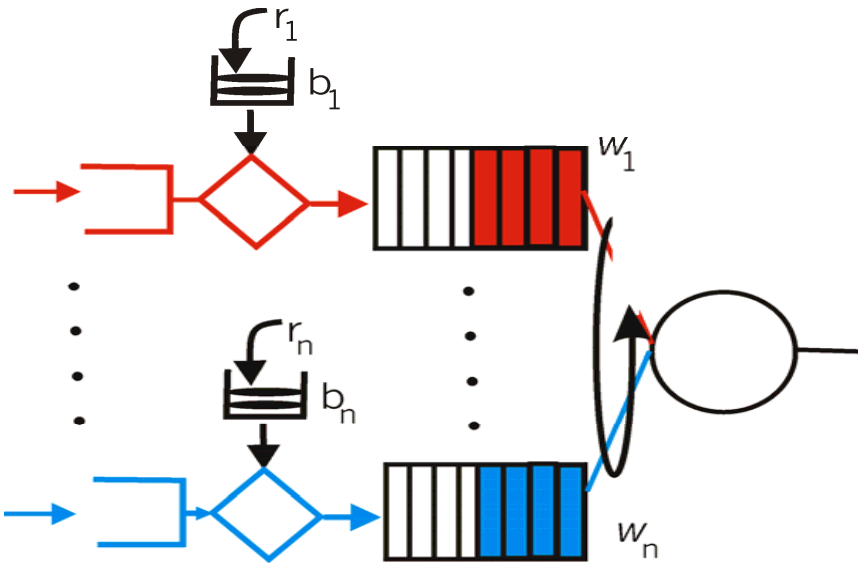


- Bucket can hold b tokens
- *Tokens* generated at rate *r token/sec* unless bucket full
- *Over an interval of length t:*
  number of packets admitted ≤ (r t + b).

Token bucket + WFQ

- Combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee*!





6.1.2. Integrated Services (IntServ)

- IETF (Internet Engineering Task Force) standard
- Many service classes possible, most important:

1. Guaranteed service – for intolerant applications
   - No packet arrives after its play back time
   - Early packets must be buffered
2. Controlled load service – for adaptive app.s
   - Under reasonable load “illusion” of reserved channels
3. Best effort service
   - Available per default

- Admission control is needed for 1 and 2
- WFQ is needed for 1

Integrated Services Mechanisms

- Flowspec
  - Specifies the requirements of an application's flow
- Admission control
- Resource reservation or signaling protocol (ATM)
  - Exchange information such as request for service, flowspecs, admission control decisions
  - Achieved by RSVP
- Policing
  - Users must be controlled for keeping the rules
- Packet scheduling
  - Routers and switches must properly schedule packets

Flowspec

- RSpec (Reserve required characteristics of the nw.)
  - Best effort
    - No additional parameters
  - Controlled-load
    - No additional parameters
  - Guaranteed
    - Delay bound as parameter
- TSpec (Traffic characteristics of the flow)
  - Average bandwidth + burstiness
    - Token rate *r, b*ucket depth *B*
  - To avoid “traffic jams”

Per-Router Mechanisms

- Admission Control – per flow
  - Uses TSpec + RSpec
  - Answer depends on service class and the queuing discipline used in the router
    - For *guaranteed* hard in general, easier with WFQ
    - For *controlled load,* good prediction is not too hard
- Packet Processing
  - Classification
    - Associate each packet with the appropriate reservation
  - Scheduling
    - Manage queues so each packet receives the requested service
  - Policing
    - Decide on a per-packet basis, whether client conforms to TSpec

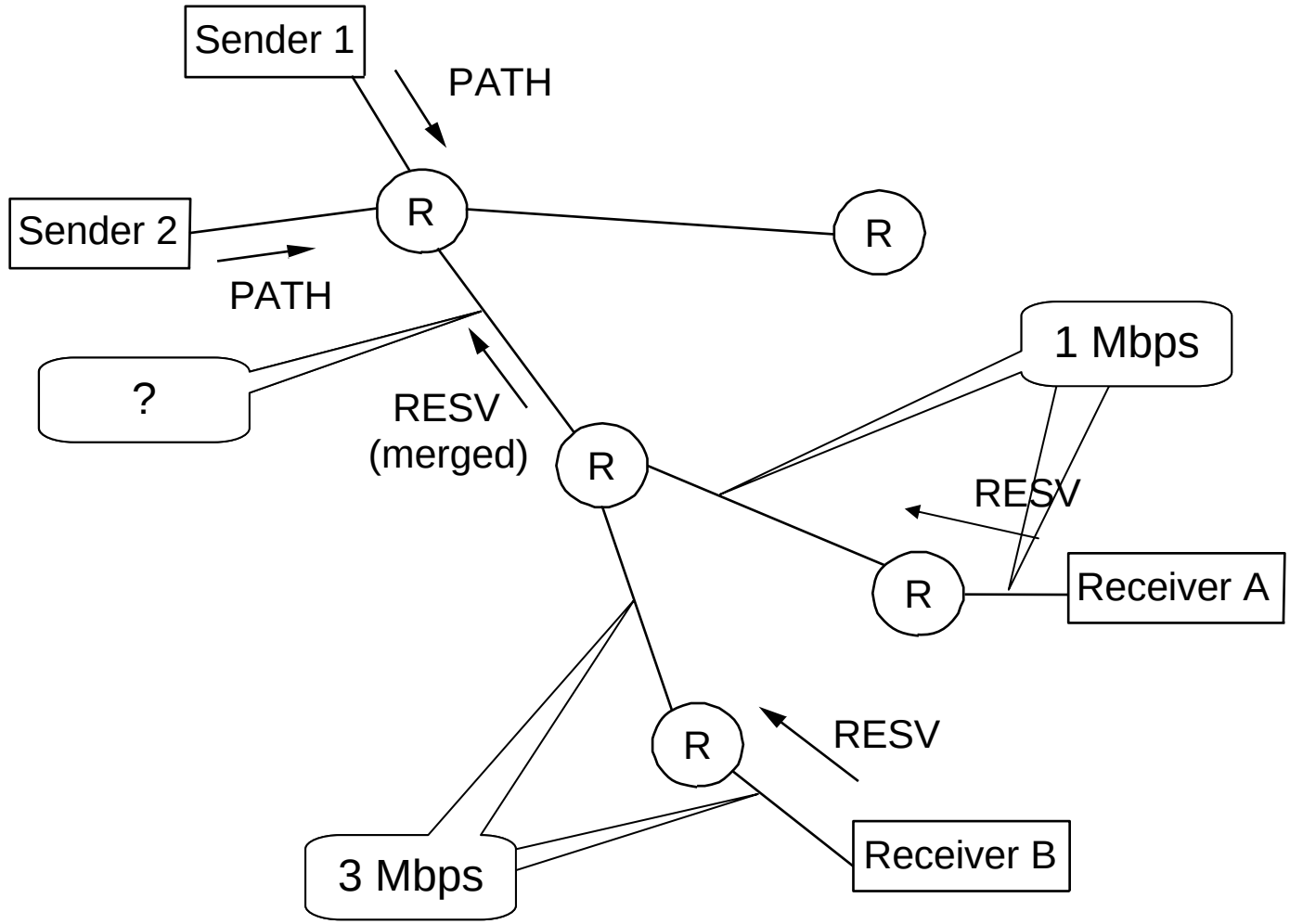
Reservation Protocol – (1)

- Called “signaling” in ATM
- Internet standard: Resource Reservation Protocol (RSVP)
- Connectionless model
  - Robust, because needs no (few) states
- Soft states (approximation for stateless)
  - States time out automatically (after e.g. 1 minute)
  - Can be refreshed periodically
- Multicast support
- Receiver-oriented
  - No need for the senders to keep track of many receivers

Reservation Protocol – (2)

- Two messages: PATH and RESV
- Source transmits PATH every 30 seconds
  - PATH conveys TSpec of the source
  - Routers figure out the reverse path
- Destination responds with RESV message (also periodic)
  - Conveys TSpec and RSpec of the receiver
  - Routers on the path try to make appropriate reservations
- In case of router or link failure
  - New route will be automatically established – repeated PATHs
  - The reservation will be also automatically renewed on the new path
- Try to merge requirements of receivers in case of multicast
- In case of several senders all TSpecs must be considered
  - TSpecs are *merged*, not simply added; maybe application dependent
  - E.g. in an audio conference more than 2 “speakers” are unnecessary

RSVP Example



Poor Scalability of Integrated Services

- Best-effort
  - Requires (almost) no state about individual flows
  - Scales well, the only things that have to grow are
    - Routing tables
    - Throughput
- Integrated Services
  - Needs (soft) states about all individual flows
  - E.g. on an optical link (2.5 Gbps) we can multiplex
    $(2.5 * 10^6) / (64 * 10^3) = 39.000$ ISDN (64Kbps) flows
  - For a per-flow management this requires huge memory and CPU resources

6.1.2. QoS in ATM Networks

- 5 service classes (ATM is connection oriented)
  - Constant bit rate (CBR)
    - Source sends at constant rate (a kind of guaranteed service)
  - Variable Bit Rate – real-time (VBR-rt)
    - Similar to guaranteed service in IP Integrated Services
  - Variable Bit Rate – non-real-time (VBR-nrt)
    - Similar to controlled load service in IP Integrated Services
  - Available Bit Rate – real-time (ABR) – no IP counterpart
    - Loss possible, but ordering is correct
    - Minimum cell transmission rate (MCR) is guaranteed
    - Specifies congestion control as well
  - Unspecified Bit Rate – (UBR)
    - Best effort, however at VC setup helpful infos may be sent

RSVP versus ATM (Q.2931)

- RSVP
  - Receiver generates reservation
  - Soft state (refresh/timeout)
  - Separate from route establishment
  - QoS can change dynamically
  - Receiver heterogeneity
- ATM
  - Sender generates reservation at connection request
  - Hard state (explicit delete necessary)
  - Concurrent with route establishment
  - QoS is static for life of connection
  - Uniform QoS to all receivers

6.1.3. Differentiated Services (DiffServ)

- Flexible service model
  - No predefined service classes (rather *“behavior aggregates”*)
  - Functional components to build classes and *relations*
    - E.g. class A receives better service than B (1. and 2. class)
- Scalability
  - No per-flow reservation,
  - Packets of several flows may belong to the same class
- Edge functions
  - Packet classification and traffic conditioning
  - Mark packets at administrative boundary
    - E.g. at the service provider edge, based on payment
    - Maybe mark only up to a limit
- Core function: forwarding
  - Packets are handled corresponding to their class

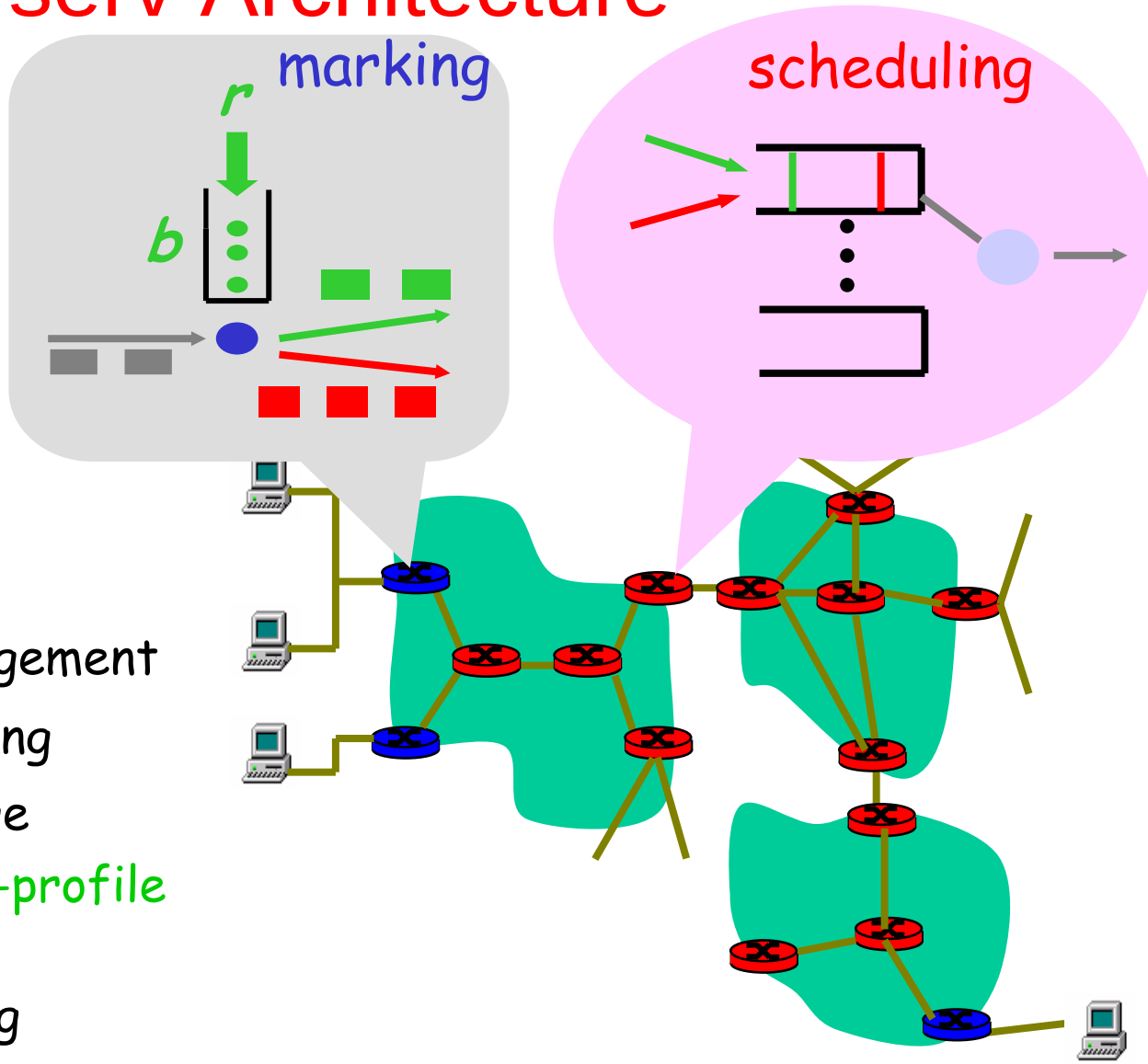
Diffserv Architecture

Edge router:

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

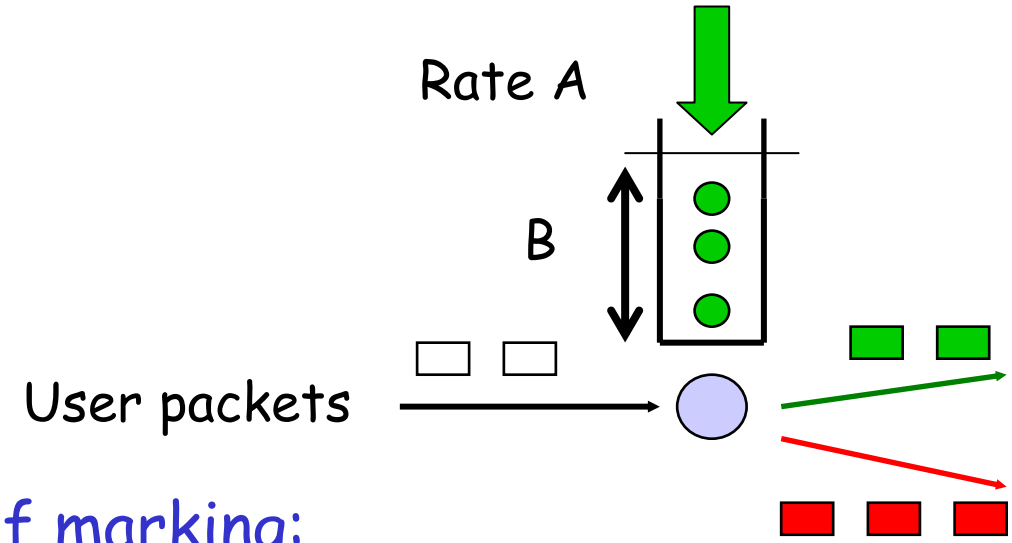
Core router:

- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets
- e.g. assured forwarding



Edge-router Packet Marking

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

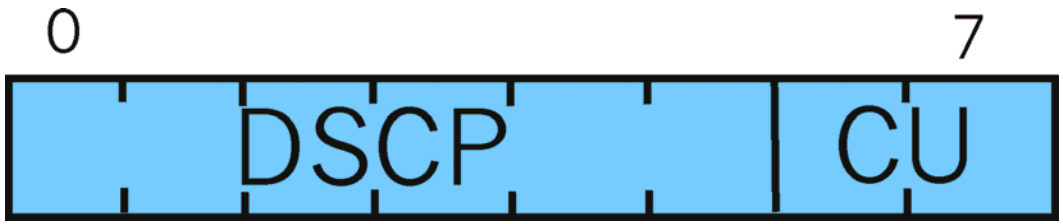


Possible usage of marking:

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

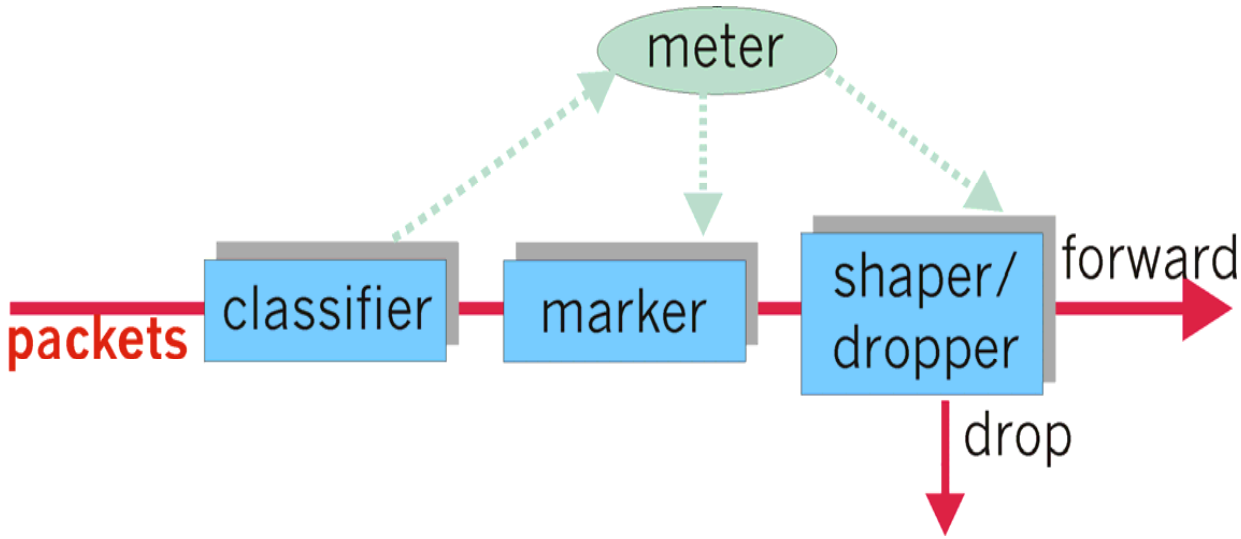
Classification and Conditioning (1)

- 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 (2)

- May be desirable to limit traffic injection rate of some class:
  - user declares traffic profile (e.g., rate, burst size)
  - traffic metered, shaped, or remarked if non-conforming



Forwarding (PHB)

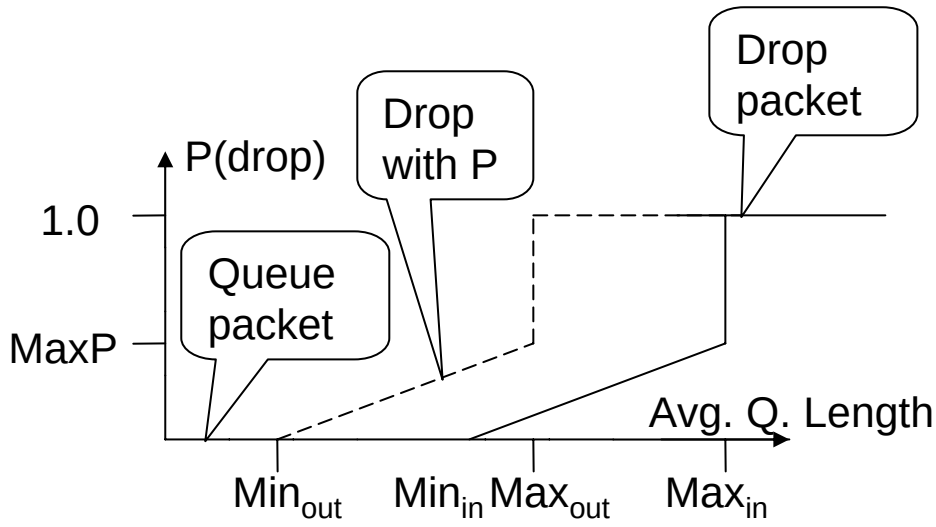
- PHB (per-hop behavior) results in a different observable (measurable) forwarding performance behavior
- PHB does *not* specify what mechanisms to use
- Examples:
  - Class A gets x% of outgoing link bandwidth over a time interval
    - Can be easily realized by WFQ
  - Class A packets leave first before packets from class B (prioQ)
- Currently standardized
  - Expedited forwarding and assured forwarding
- Expedited Forwarding
  - Packet departure rate of a class equals or exceeds specified rate
    - Logical link with a minimum guaranteed rate
  - Used to implement 1st and 2nd class
  - Even if the 2nd class is overwhelmed, 1st class can work

Assured forwarding (1)

- 4 classes of traffic with 3 levels each
  - Each class has a guaranteed minimum amount of
    - bandwidth
    - buffering
  - 3 drop preference categories within each class, if congestion occurs
  - Can implement classes as gold, silver, bronze …
  - Is gold always better than silver? Example:
    - Gold has bandwidth x, silver x/2
    - Silver has packet rate r, gold 100 * r
    - Is gold better than silver?
  - Classification and resource dimensioning must correlate
  - A cost model may help

Assured forwarding (2)

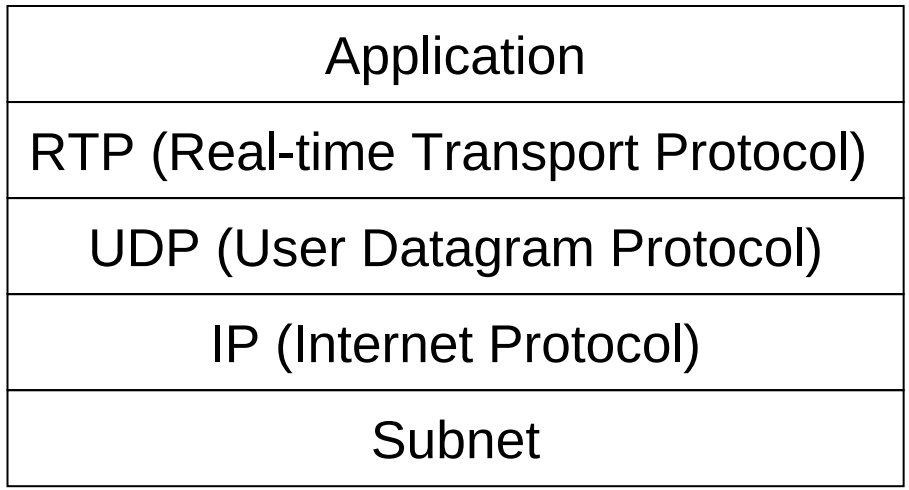
- RIO (RED with In and Out)
  - Provider and customer agree on profiles
    - E.g. customer is allowed to send up to *x* Mbps of assured traffic
  - Edge routers: tag packets
    - *in* bit: the packet is in the profile
    - *out* bit: the packet is out of profile – no assurance needed
  - Tries not to drop *in* packets
- WRED (weighted RED)
  - Generalization of RIO
  - More probability curves
  - Assignment of proper weights is important



6.2. Real-time Transport

- Real-time Transport Protocol (RTP)
- Origins at the *vat* audio conferencing tool's *application protocol*
- RTP is a "transport protocol", running over the usual transport protocols, typically over UDP

- Protocol stack for multimedia application using RTP:



Requirements – (1)

- Interoperability between different applications
  - Between conferencing and streaming applications
  - Between e.g. two different audio conference systems
- Negotiation about coding issues
  - Agree on media type, compression method etc.
- Timing for proper playback (in a single stream)
- Synchronization (among multiple streams)
  - E.g. between video and corresponding audio
- Indication of packet loss, thus also congestion
  - RTP runs typically over non-reliable transport (UDP)
  - This enables applications to do something (e.g. adapt)

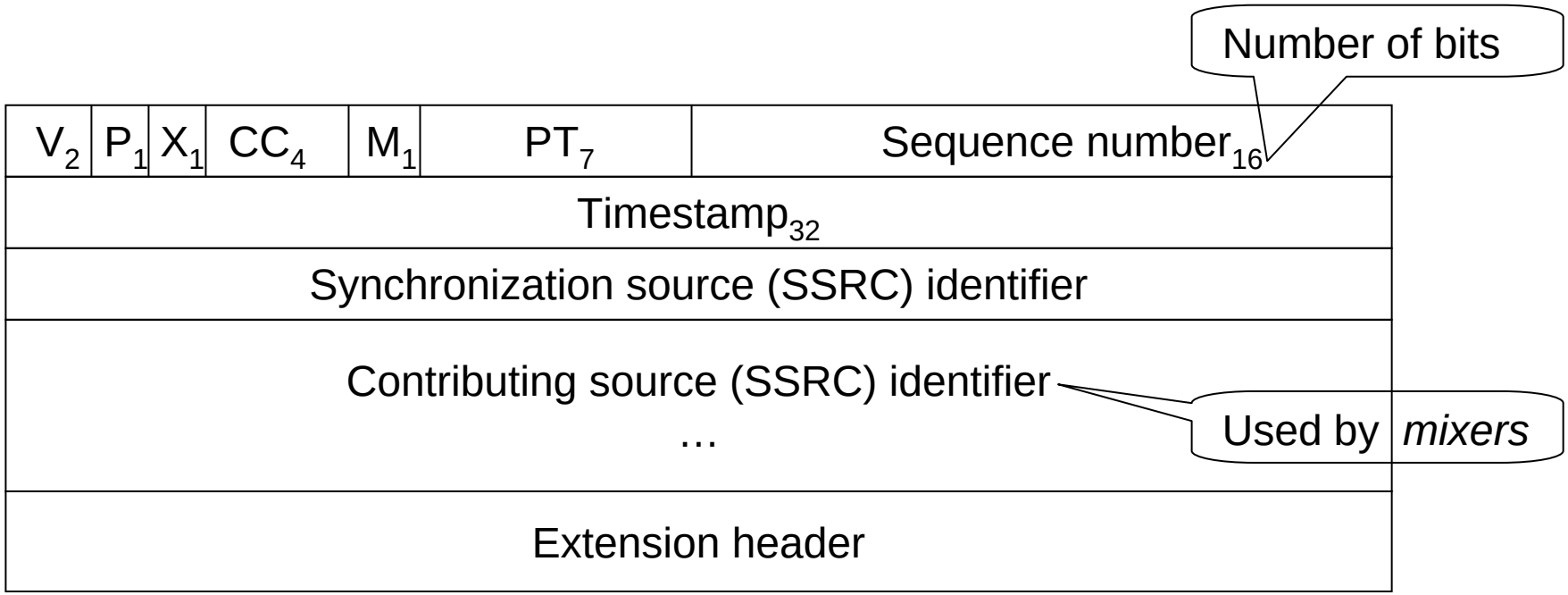
Requirements – (2)

- Framing
  - Enable applications to mark start and end of frames
  - E.g. mark the beginning of a “talkspurt”: the application may shorten or lengthen silences
- Sender identification
  - The IP address is not extremely user-friendly
- Efficiency
  - No long headers are acceptable
  - E.g. audio data packets are typically short, a great overhead is undesirable
- Uni- and multicast RTP sessions

Real-time Transfer Protocol (RTP)

- A twin-standard by IETF (with RTCP)
  - RTP for the exchange of multimedia data
  - RTCP for sending periodically control information
  - They use consecutive even-odd port numbers
- Application Level Framing (ALF)
  - Applications understand their needs best
  - Flexibility is needed to allow new applications
  - Profile
    - Defines the meaning of certain fields in the header
  - Formats
    - Interpretation of data following the header, such as
    - A stream of bytes of audio samples
    - Some more complex structure (MPEG)

RTP header format – (1)

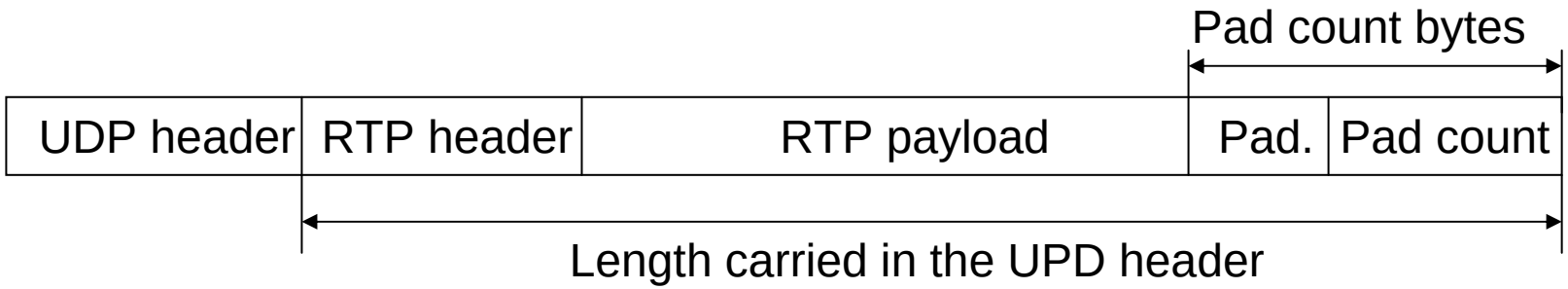


- V: version number – 2 bits might be few, later extensions possible (subversion)
- P: Padding is used
- X: Extension header exists (rare, the payload *format* description should suffice)
- CC: Contributing sources – is needed only if the RTP streams are “mixed”
- M: Marks the packet, e.g. start of frame – the application uses it at wish
- PT: Payload type

RTP header format – (2)

- Payload type
  - E.g: GSM, H.261, MPEG Audio, MPEG-1/2 video etc.
  - Generally not used as de-multiplexing key; transport mechanisms are used e.g. diff. UDP ports for each stream
- Sequence number
  - The sender just increments it – handling by application
    - E.g. replay the last frame for a lost video frame
- Time stamp
  - Tick is defined by the application
    - E.g. 125µs / audio sample (8KHz)
    - If an RTP packets contains 160 samples: RTP-TS increases by 160
- Synchronization source (SSRC)
  - Random number identifying a single source of a stream
  - In a *conference*: ∀ sender 1 (needs conflict resolution)

RTP packet format



- The length of the RTP data is coded in the UDP header
- If RTP.payload < UDP.length: padding is used
  - UDP.length maybe fixed, e.g. due to an encryption alg.
- Advantage
  - The RTP header remains short: 1 bit suffices
  - The Pad bytes are used only if this place is unused by the payload anyway

Real-time Transport Control Protocol

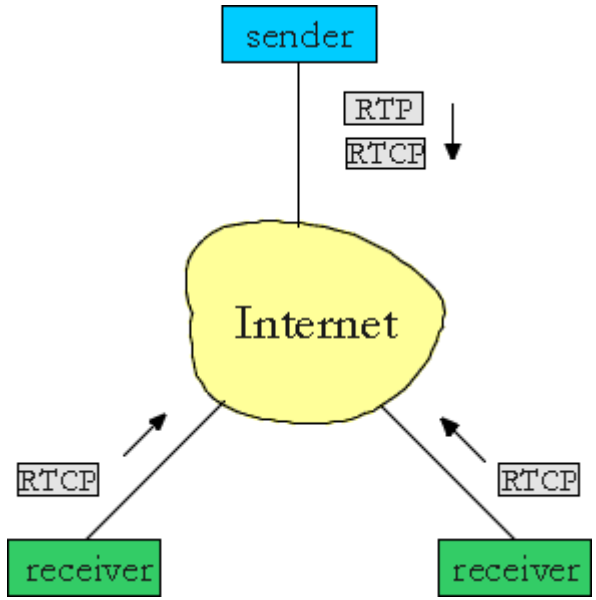
- RTCP provides control stream associated with the data stream, with the functions:
  1. Performance feedback
     - Can be used e.g. by adaptive applications
  2. Correlate and synchronize media streams
     - A sender (a presentation) may have many SSRC values, from different nodes; SSRC collisions must be resolved
     - Canonical name (CNAME) assigned to a sender
       - General form: user@full_domain_name_of_host
       - Serves as a kind of *scope*
     - Different clocks must be synchronized
  3. Convey the identity of the sender
     - E.g. to list the partners in an audio conference

RTCP Packet Types

- Receiver reports
  - SSRC (synchronization source) identifier
  - Statistics of lost data packets from this source
  - Highest sequence number from this source
  - Estimated inter arrival jitter for this source
  - Last actual timestamp received via RTP for this sender
  - Delay since last sender report received via RTP for this sender
- Sender reports additionally
  - Timestamp and “wall clock” (usual) time of the most recently generated RTP packet
    - This enables synchronization (time of day ↔ RTP stamp)
  - Cumulative packet and byte counts since transmission begin
- Source descriptions
  - CNAME, SSRC and maybe other sender description information
- Application-specific control packets

RTCP Operation

- RTCP traffic is limited to ca. 5% of RTP traffic
  - The report generation slows down if necessary
- Recipients and sender may react to the reports
  - Recipient may require resource reservation noticing that other recipients have better QoS
  - Sender may reduce rate if too many packets are lost
  - RTP timestamp + time of day enable synchronization of streams, even with different clock granularity
  - CNAME enables the identification of the media stream with several (maybe even changing) SSRC values

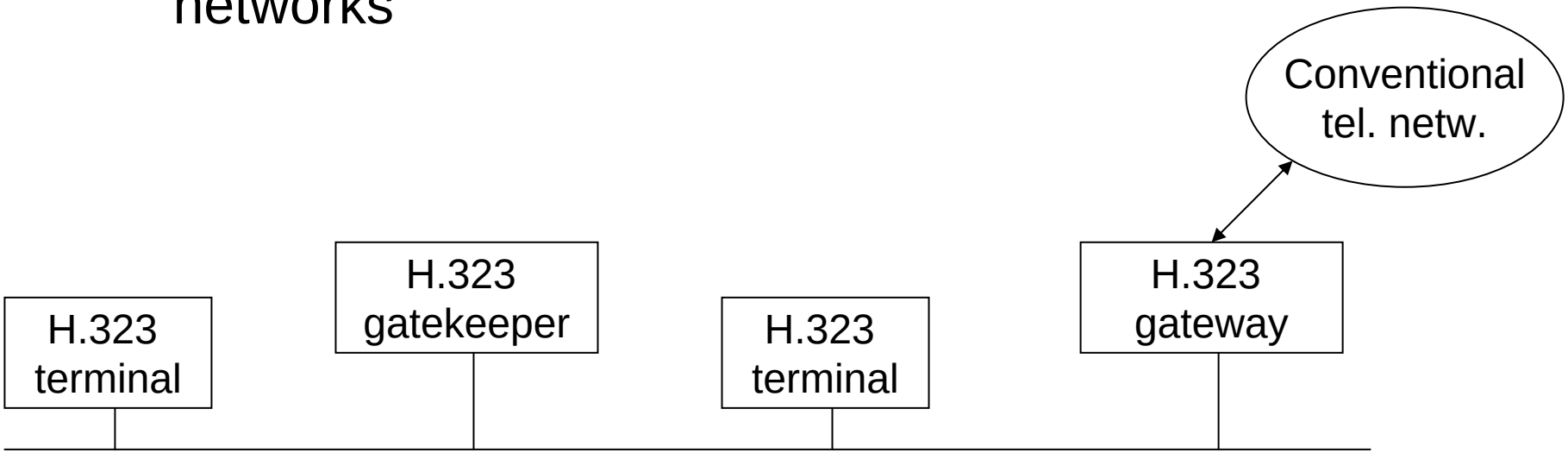


Session and Call Control

- Conferencing needs also session control, IETF
  - Session Description Protocol (SDP)
    - To exchange codecs and protocols
  - Session Announcement Protocol (SAP)
    - Initiates multicast (multimedia) sessions
    - Sends periodic announcements to the multicast address
  - Simple Conference Control Protocol (SCCP)
    - Protocol for tightly coupled conferences
    - Management of the set of members
    - Management of the set of media/application sessions that constitute the conference;
    - Floor control
  - Session Initiation Protocol (SIP)

Internet telephony

- Internet telephony (H.323 by ITU, also SIP)
  - H.245 call control to negotiate properties
  - Gatekeepers control and convert the calls
  - Gateways connect to conventional telephone networks



SDP – an example

Simple, human readable ASCII coding, with single-character codes

```
V=0                                          % version number
O=larry  s-id  … IN IP4 10.01.5              % origin of the session
S=DMMS                                       % session name
I=Distributed Multimedia Systems             % session description
U=laszlo@itec.uni-klu.ac.at                  % session URI
C= IN IP4 224.2.17.12/127          % multicast IP address of the session
T=start-time end-time              % integers (network time protocol)
M=audio port# RTP/AVP 0            % uses RTP with profile AVP
                                      encoding: 0 (8KHZ 8-bit sampling)
M=video port# RTP/AVP 31           % uses RTP with profile AVP
                                      encoding: 31 (H.261 encoding)
M=application port# udp wb         % uses UDP directly, encoding:
                                                specific to the wb application
```

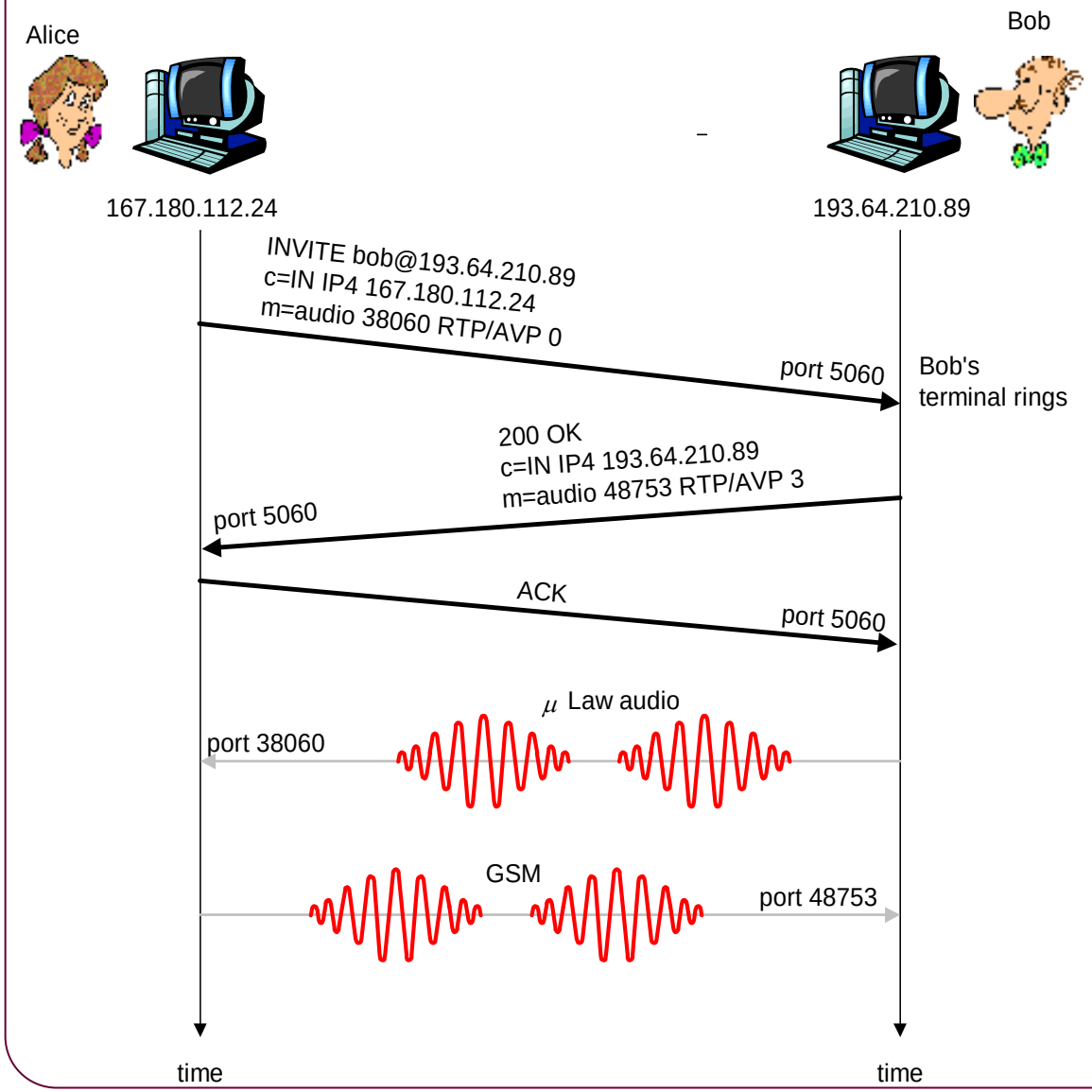
SIP

- Session Initiation Protocol
- Comes from IETF
- SIP long-term vision
  - All telephone calls and video conference calls take place over the Internet
  - People are identified by names or e-mail addresses, rather than by phone numbers
  - You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using

SIP Services

- Setting up a call
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms 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 and hold calls

Setting up a call to a known IP address



- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers

- Bob's 200 OK message indicates his port number, IP address & preferred encoding

- 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 the required encoder.
  - Bob will instead reply with 606 Not Acceptable Reply and list encoders he can use.
  - Alice can then send a new INVITE message, advertising an appropriate encoder.
- Rejecting the call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden".
- Media can be sent over RTP or some other protocol.

Name translation and user locataion

- 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
```

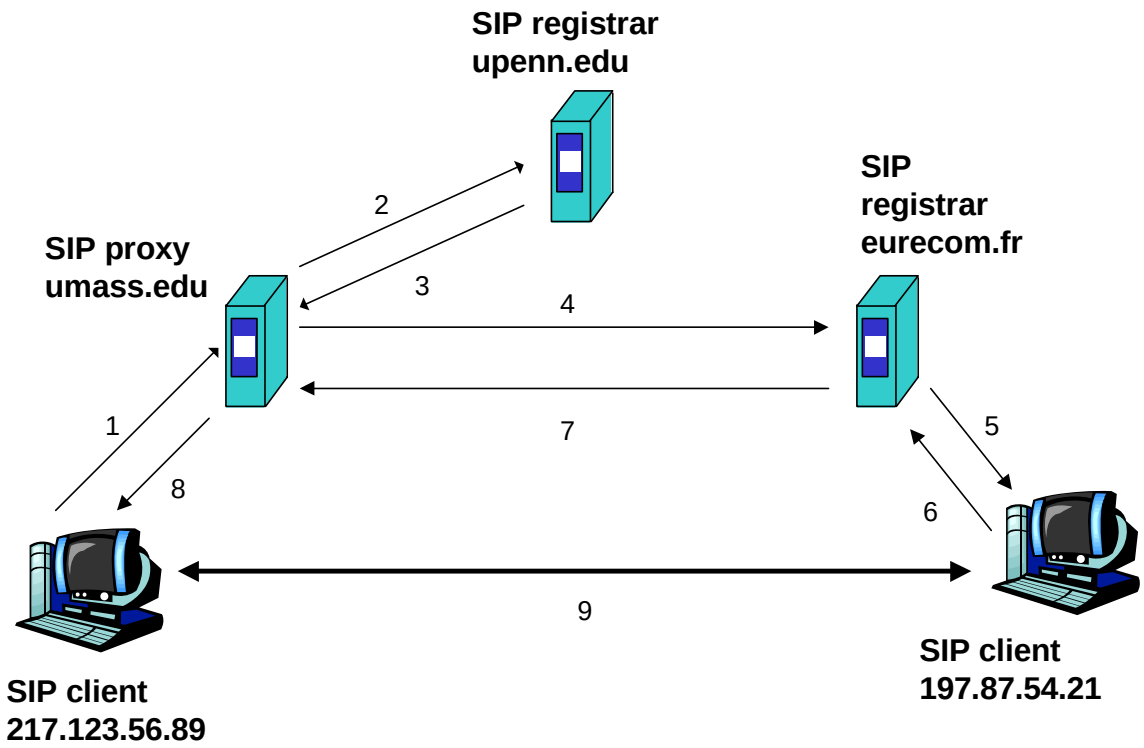
SIP Proxy

- 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

- Note: proxy is 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.

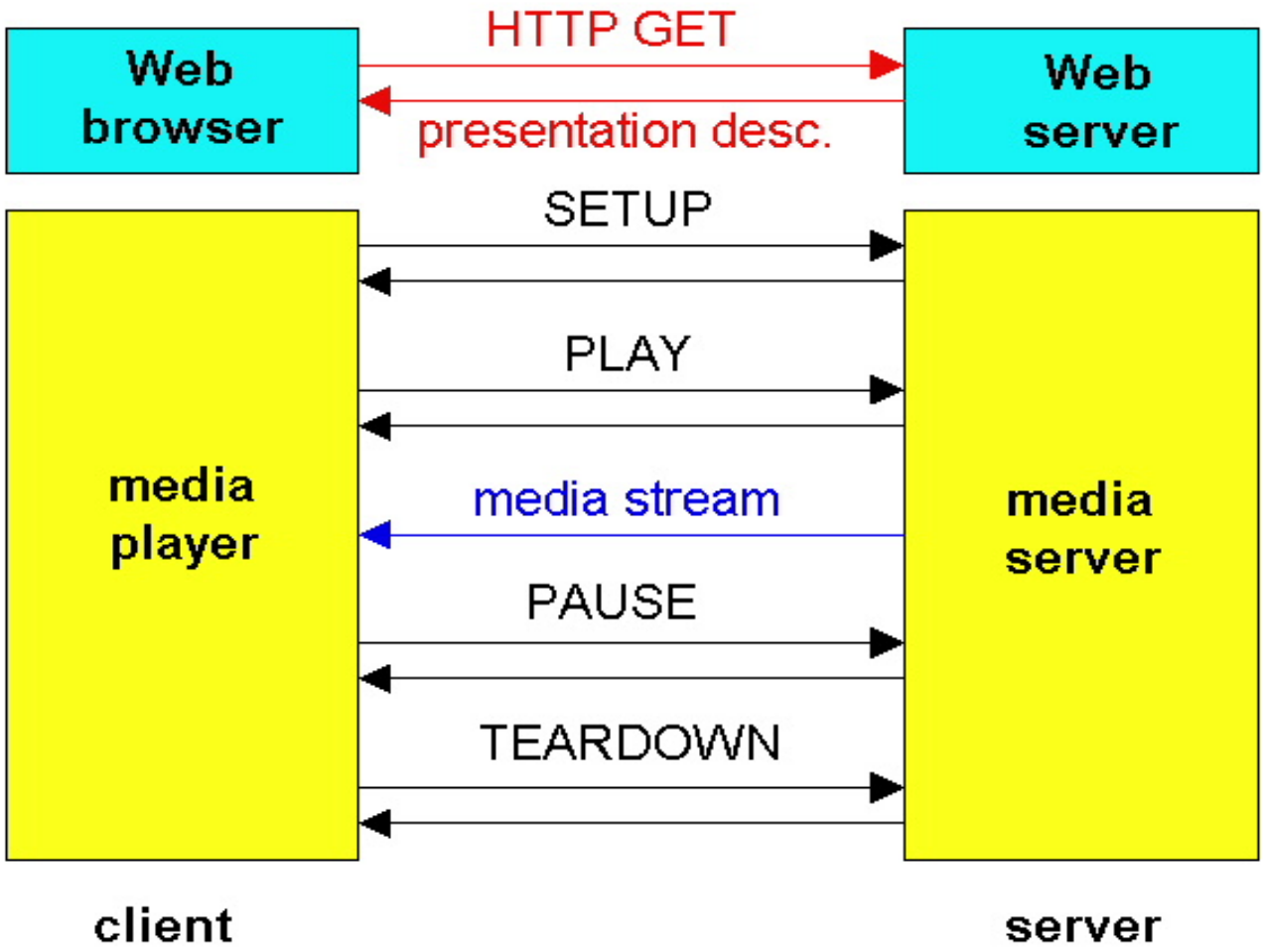
Comparison with H.323

- 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 and codecs
- SIP is a single component. Works with RTP, but does not mandate it. Can be combined with other protocols and services
- H.323 comes from the ITU (telephony).
- SIP comes from IETF: Borrows much of its concepts from HTTP
- SIP has a Web flavor, whereas H.323 has a telephony flavor
- SIP uses the KISS principle: Keep it simple stupid

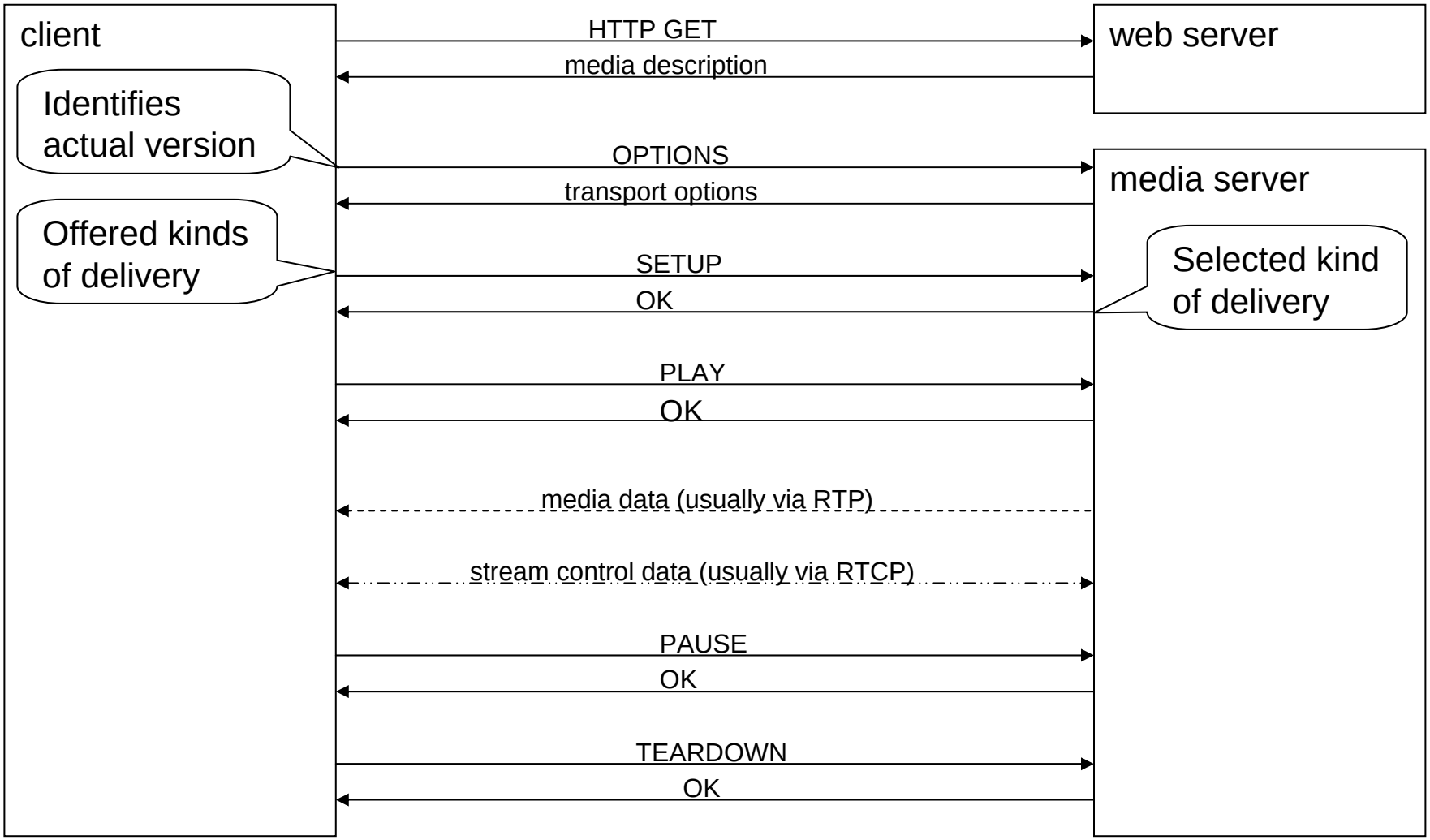
Real-time Streaming Protocol (RTSP)

- User-level protocol
- No assumption about the transport level
- No explicit QoS mechanisms
- Delivers only control data, no payload
- Syntax similar to HTTP, but the sever *has* states
- Extensible by new methods and/or parameters
- Supported operations
  - Stream of media data from a media server
  - Invitation of a media server in a conference
  - Adding of media to an existing presentation

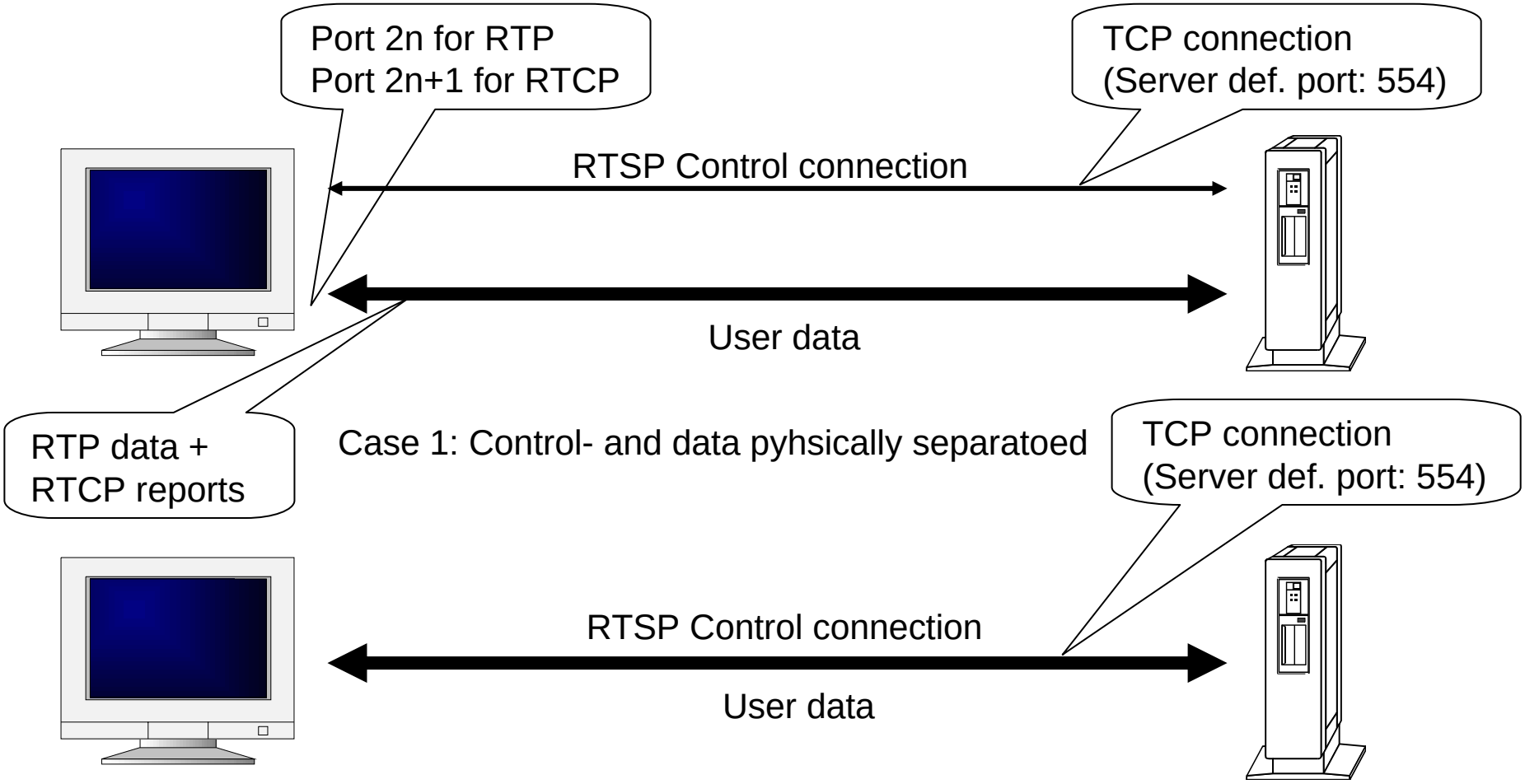
RTSP Streaming Example



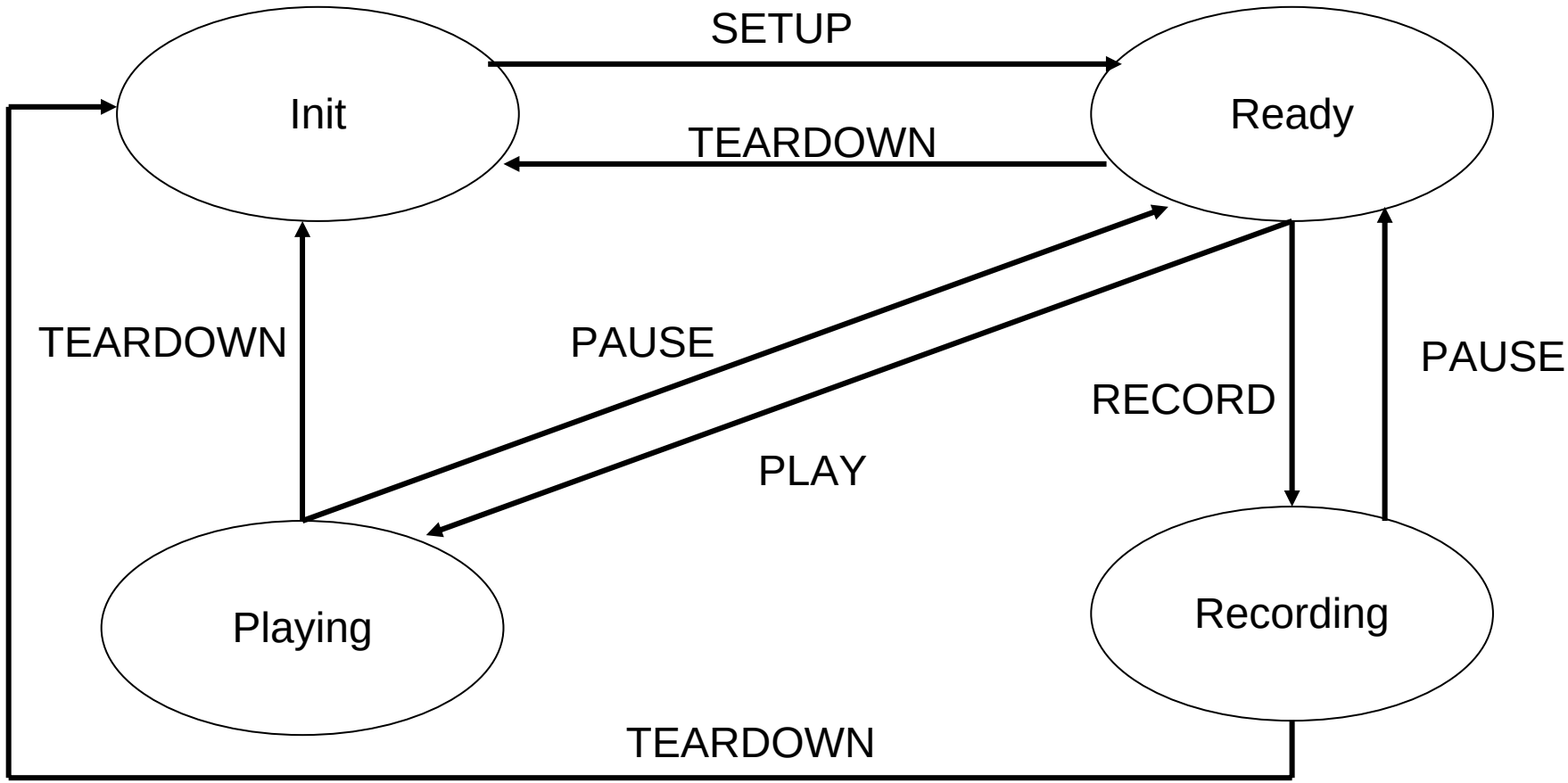
RTSP session protocol



Control and user data



RTSP state diagram



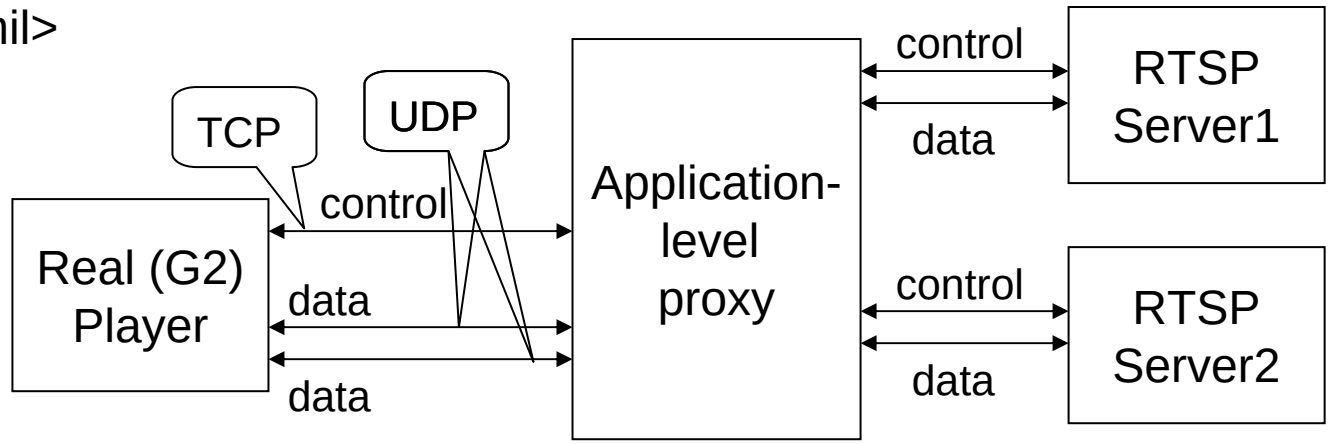
RTSP – methods

Method	Direction	Object	Availability
DESCRIBE	C -> S	P, S	suggested
ANNOUNCE	C < - > S	P, S	optional
GET_PARAMETER	C < - > S	P, S	optional
OPTIONS	C < - > S	P, S	mandatory
			(S - > C: optional)
PAUSE	C - > S	P, S	suggested
PLAY	C - > S	P, S	mandatory
RECORD	C - > S	P, S	optional
REDIRECT	S - > C	P, S	optional
SETUP	C - > S	S	mandatory
SET_PARAMETER	C < - > S	P, S	optional
TEARDOWN	C - > S	P, S	mandatory

Use of RTSP from SMIL

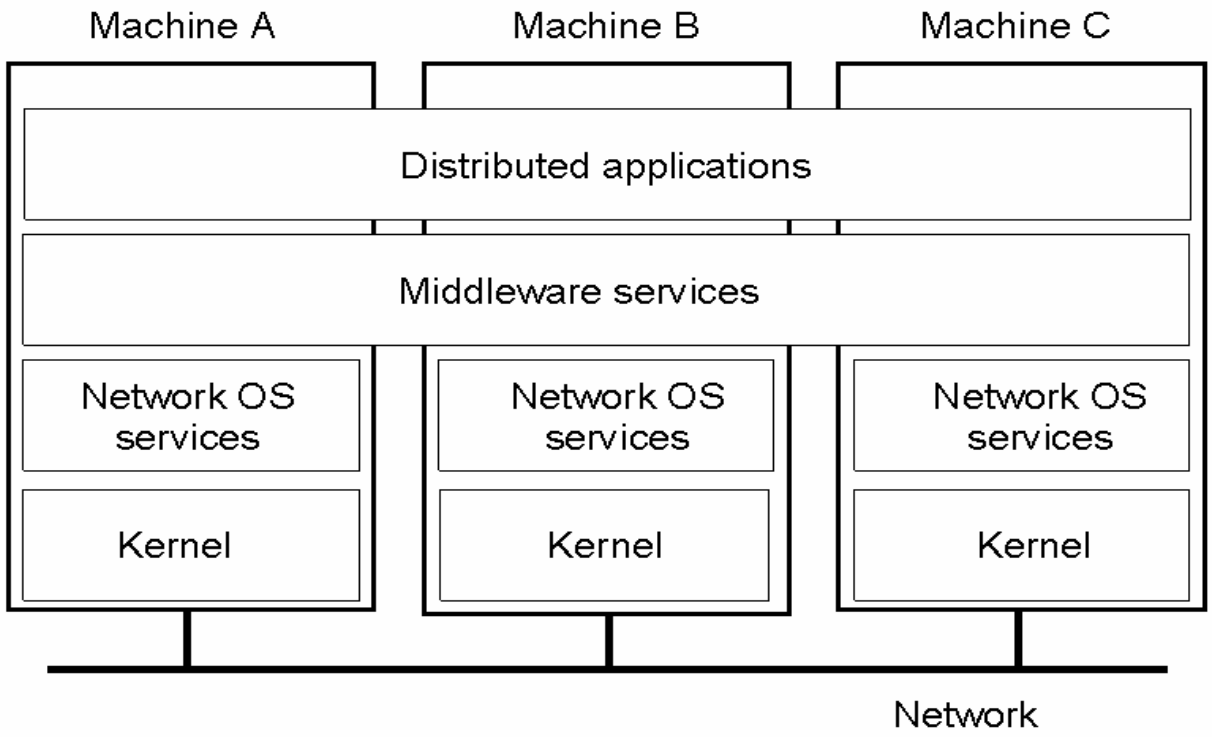
- Synchronized Multimedia Integration Language
  - Standard by the World Wide Web Consortium (W3C)
  - Presentations can be specified, e.g.:

```
<smil>
    <body>
        < audio src="rtsp://realserver1.company.com/one.rm" />
        < audio src="rtsp://realserver2.company.com/two.rm" />
    </body>
</smil>
```



Middleware

- Middleware is the practical compromise among “true” distributed and network file system



Multimedia Middleware

- Middleware in general
  - A distributed software layer above the operating systems and under the applications
  - Enables unified communication over the network
  - Hides the differences between platforms (HW, OS)
  - Maybe even language independent (as CORBA)
  - Main issue: *interoperability*
- Additional views
  - Upperware, middleware, underware
  - Application domain specific middleware
  - Customized middleware

Requirements on MM middleware

- Programming abstractions to represent multimedia
  - E.g.continuous interaction, not just RPC or RMI
- Real-time synchronization mechanisms
  - Intra- and inter-media (e.g. lip synchronization)
- Static and dynamic QoS management
  - As integral part of the middleware platform
  - Specification, supervising and controlling
- See also chapter “Frameworks”
- Current middleware gives only limited support

Some relevant research systems

- Real-time middleware
  - Real-time CORBA / TAO, Dynamic TAO
- Extended middleware platforms
  - Sumo, Dimma, ReTINA (Jonathan)
- Adaptation and QoS management
  - QoO, Agilos, Quasar, DJINN, Multe, Adapt
- Service architectures
  - IMA MSS (and PREMO), CORBA A/V Streams
- Component frameworks
  - JMF, TOAST, DirectShow, Gibbs framework, VuSystem, CMT, Mash

Basic Multimedia Services

- Resource management and adaptation
- Transcoding, mixing, filtering, stream scaling ...
- Presentation management
- Timing/synchronization service
- Session and group management
- Data models and standards
  - MHEG, MPEG-4, Quicktime, SMIL, VRML, ScriptX ...
- Multimedia metadata
- Persistence, consistency, transaction, placement, recovery, replication, caching, paying, security ...