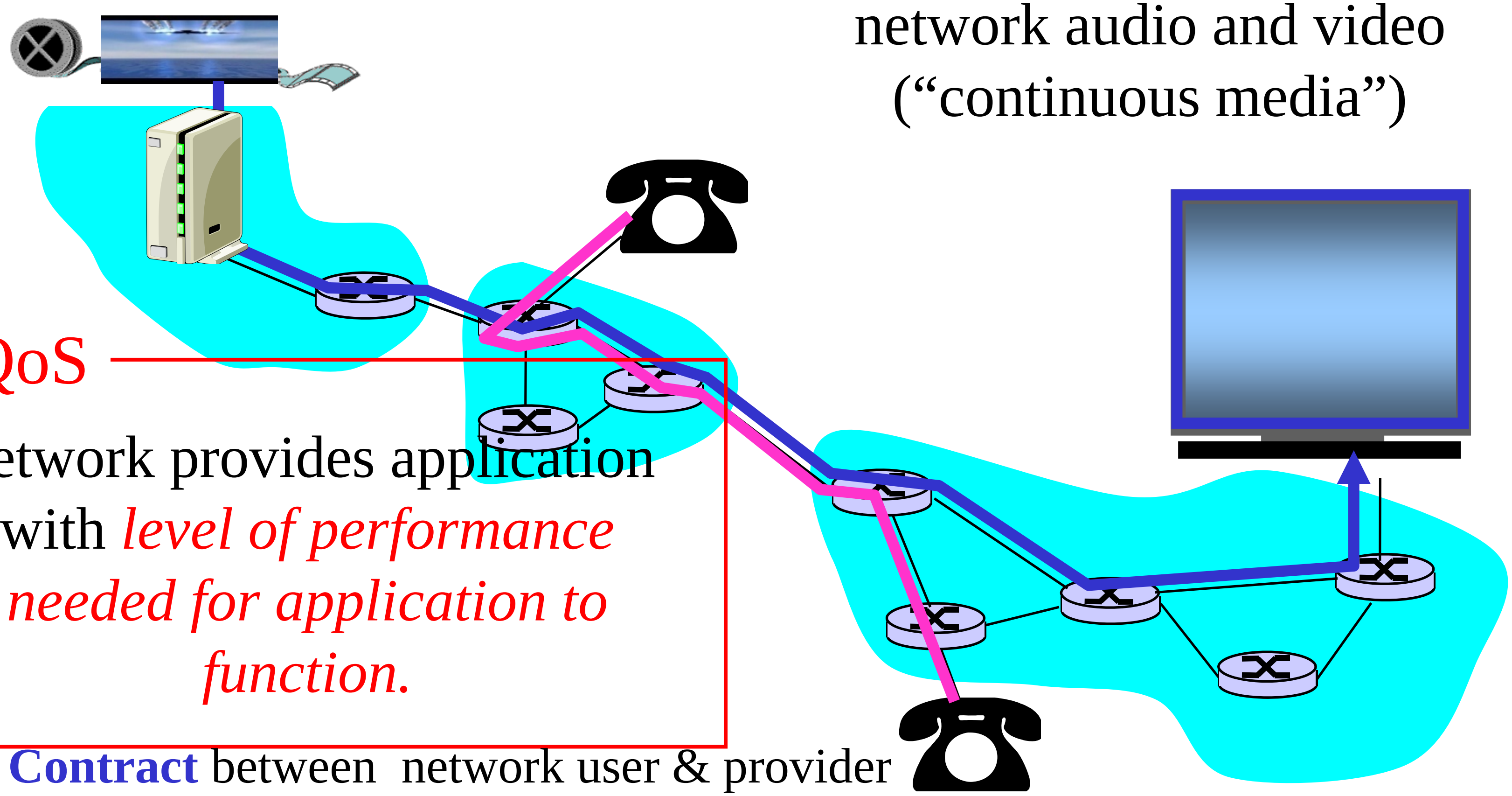
Chapter 7 + ATM/VC networks (3, 4, 5): Multimedia networking, QoS, Congestion control

Course on Computer Communication and Networks, CTH/GU

The slides are adaptation of the slides made available by the authors of the course's main textbook

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.*

i.e. Contract between network user & provider

Agree on

Traffic characteristics (packet rate, sizes, …)

Network service guarantees (delay, jitter, loss rate, …)

MM Networking Applications

Classes of MM applications:

1) stored streaming

2) live streaming

3) interactive, real-time

Fundamental characteristics:

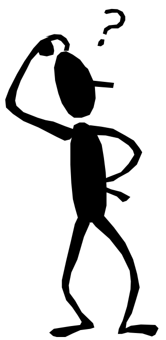
- typically **delay sensitive**
  - end-to-end delay
  - delay jitter
- **loss tolerant**: infrequent losses cause minor glitches
- antithesis with data, which are loss *intolerant* but delay *tolerant*.

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

Multimedia Over Today's Internet

"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

Solution Approaches in IP Networks

- To mitigate impact of "best-effort" protocols:
  - Use UDP to avoid TCP's slow-start phase…
  - Buffer content at client and control playback to remedy jitter
  - Different error control methods
  - Exhaust all uses of caching, proxys, etc
  - Adapt compression level to available bandwidth
  - add more bandwidth

Scalability? May need major change of the protocols (?):

- … to consider resource reservation, traffic classes, service level agreements, … (more on this in a short while…)

Chapter 7: 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

Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

- Improving QoS in Networks (also related with congestion-control)
  - Packet scheduling and policing

- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

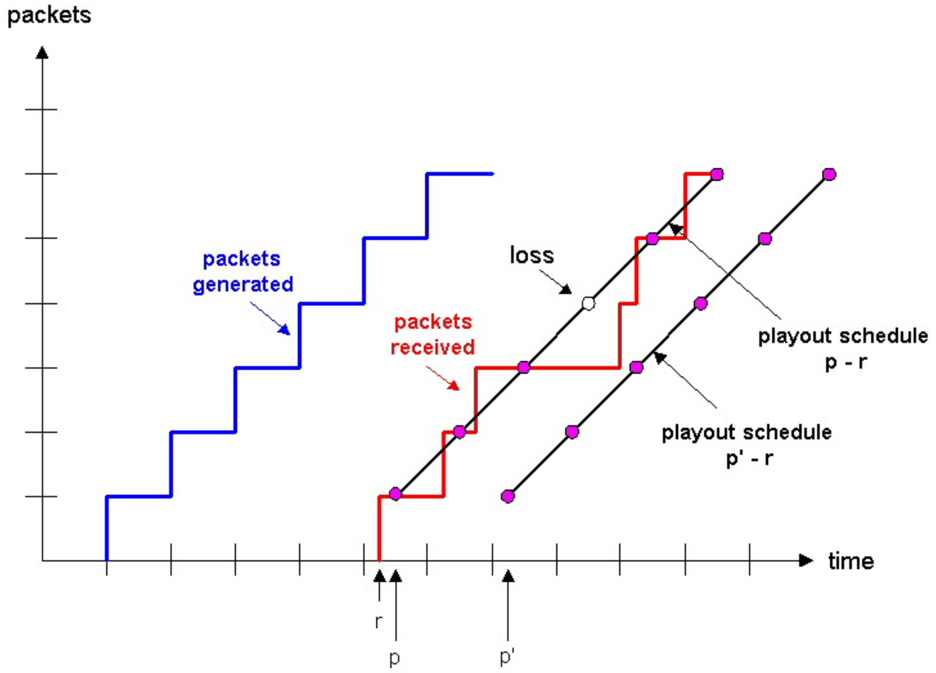
Internet Phone's Playout Delay

Fixed: chunk timestamped t is played out (at the receiver) at time t + q (assuming it arrived)

Observe: delay-loss trade-off

large q: less packet loss

small q: better interactive experience



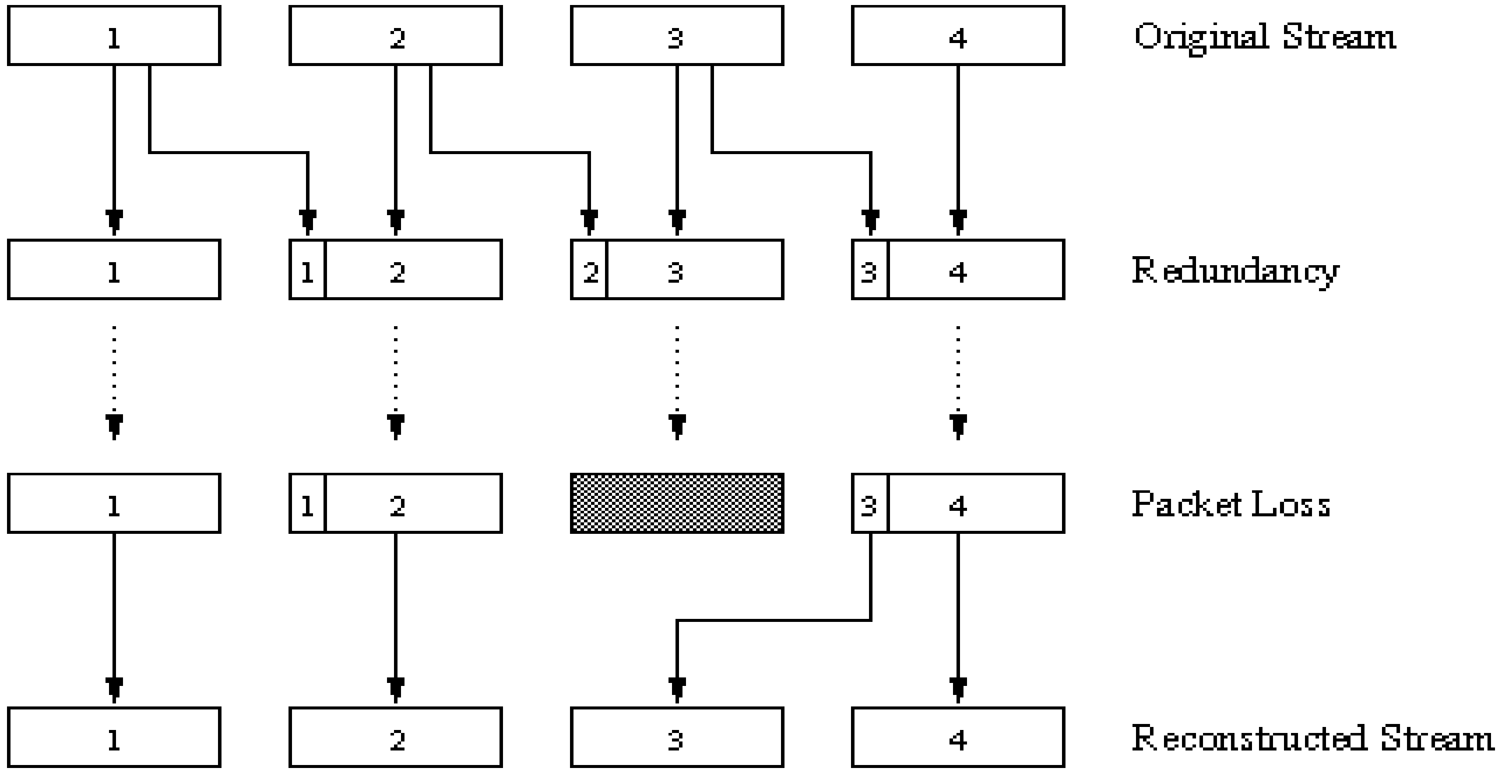
Dynamic:

- estimate network delay +variance (as in TCP) ;
- adjust playout-delay at the beginning of each talkspurt

- will cause silent periods to be compressed and elongated by a small amount; not noticeable in speech

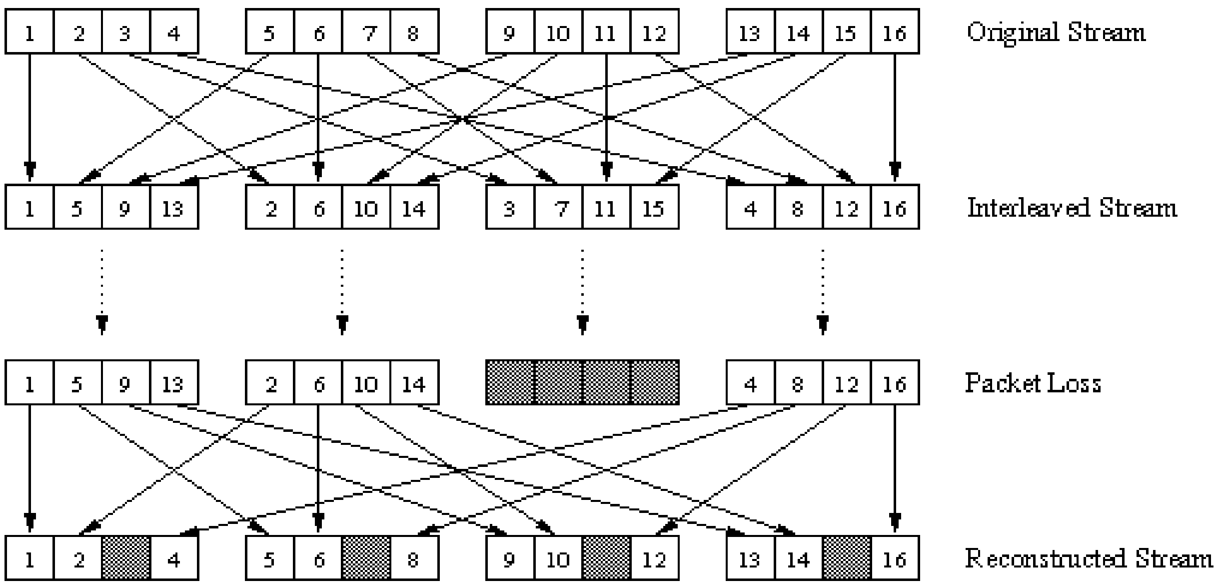
Recovery From Packet Loss (FEC)

Piggybacking Lower Quality Stream



Recovery From Packet Loss/FEC (cont)

3. Interleaving: no redundancy, but can cause delay in playout beyond Real Time requirements
   - Upon loss, have a set of partially filled chunks
   - playout time must adapt to receipt of group
   - Divide 20 msec of audio data into smaller units of 5 msec each and interleave



Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

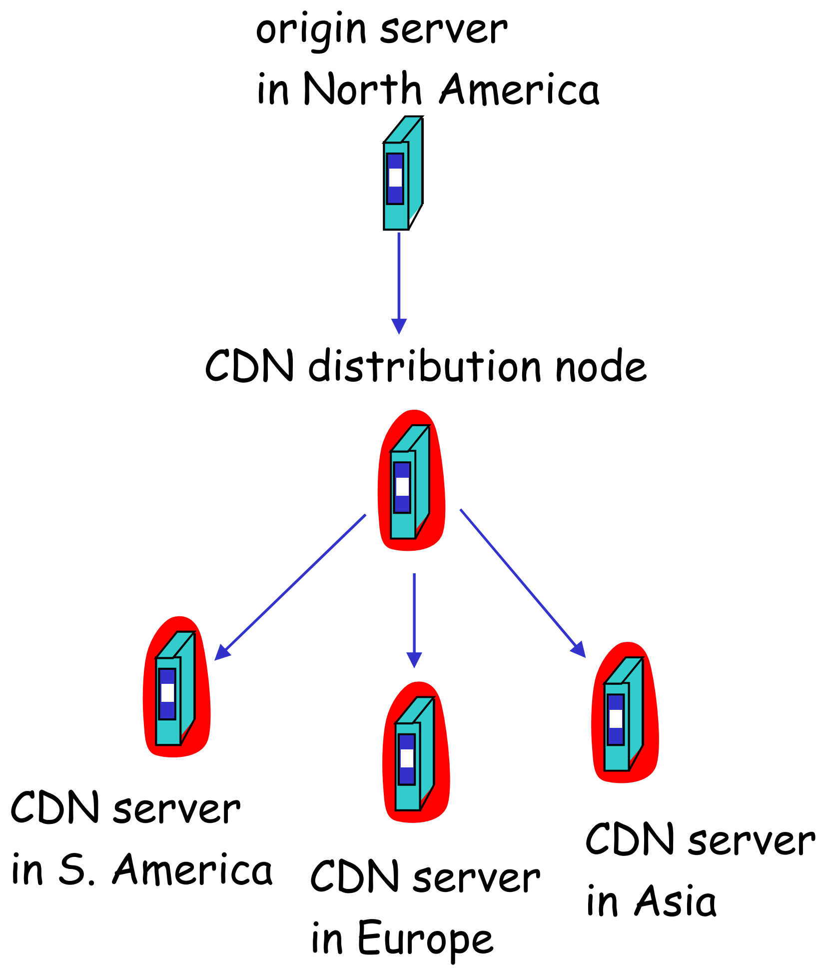
- Improving QoS in Networks (also related with congestion-control)
  - Packet scheduling and policing

- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

Content distribution networks(CDNs)

Content replication

- Challenging to stream large files from single origin server in real time
- Solution: replicate content at several/many servers
  - content downloaded to CDN servers ahead of time
  - content "close" to user avoids impairments (loss, delay) of sending content over long paths
  - CDN server typically in edge/access network
  - Resembles overlay networks in P2P applications

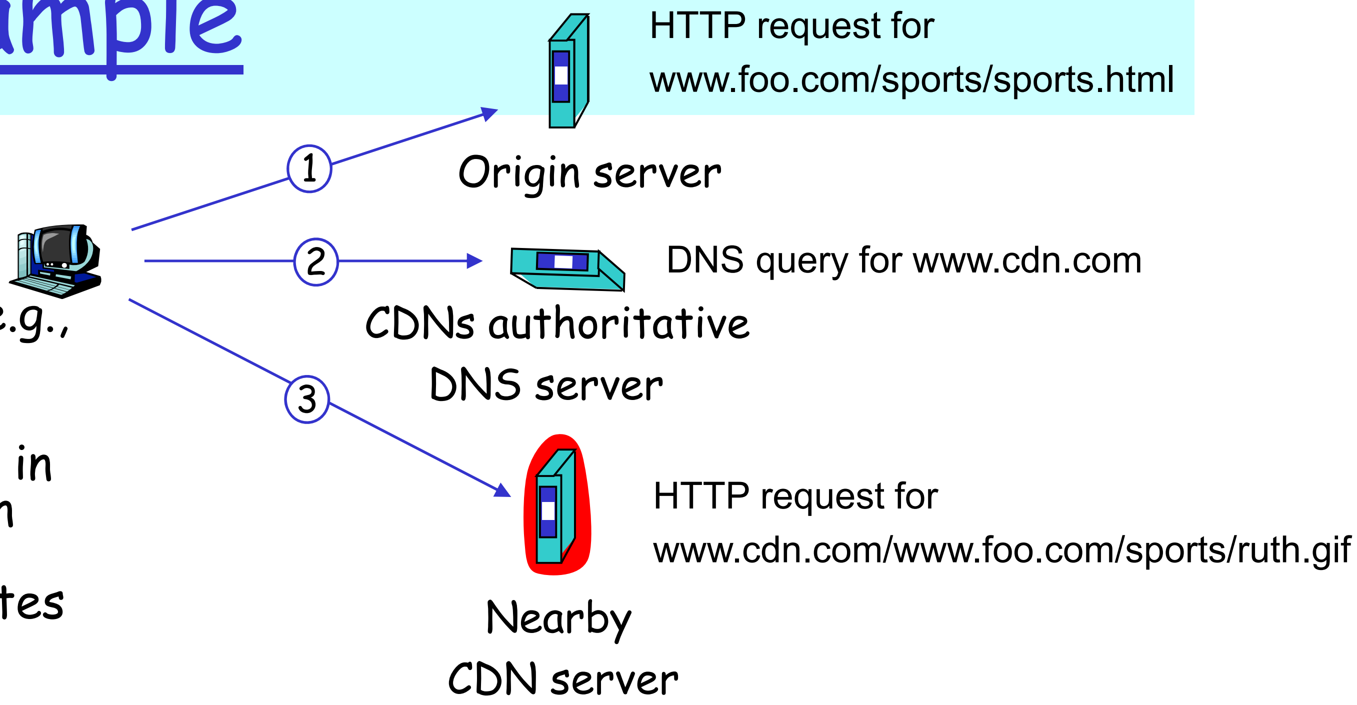


Video link: http://vimeo.com/26469929

CDN example

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



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)

- uses its authoritative DNS server (*always involved*) to redirect requests
  - "map" to determine closest CDN server to requesting ISP

Video link: http://vimeo.com/26469929

Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

- Improving QoS in Networks (also related with congestion-control)
  - Packet scheduling and policing

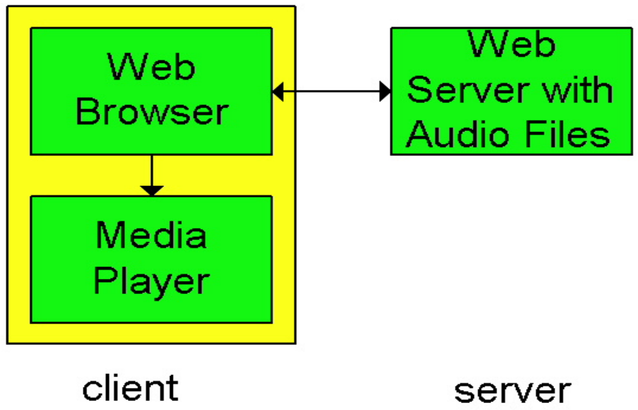
- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

Streaming From Web Servers

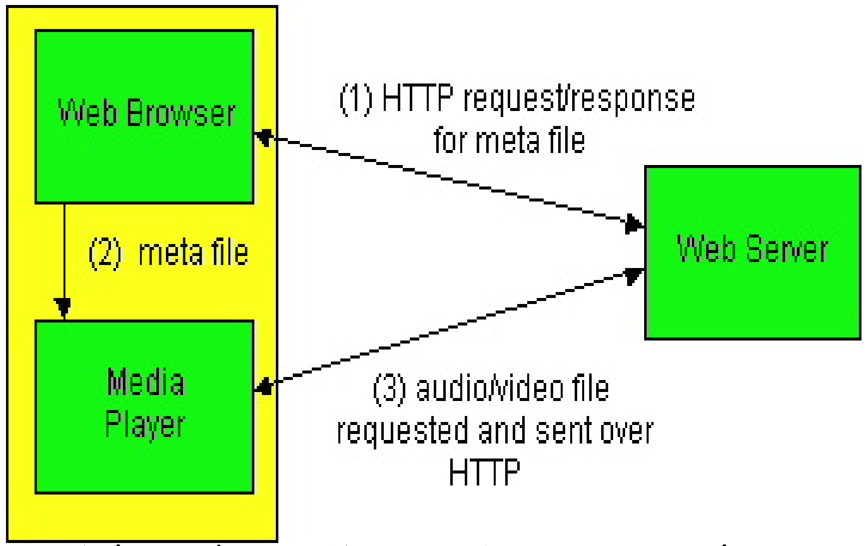
- Audio (in file), Video (interleaved audio+images in 1 file, or 2 separate files + client synchronizes display) sent as HTTP-object
- A simple architecture:

  Browser requests the object(s);

  after reception pass them to the player (no pipelining)

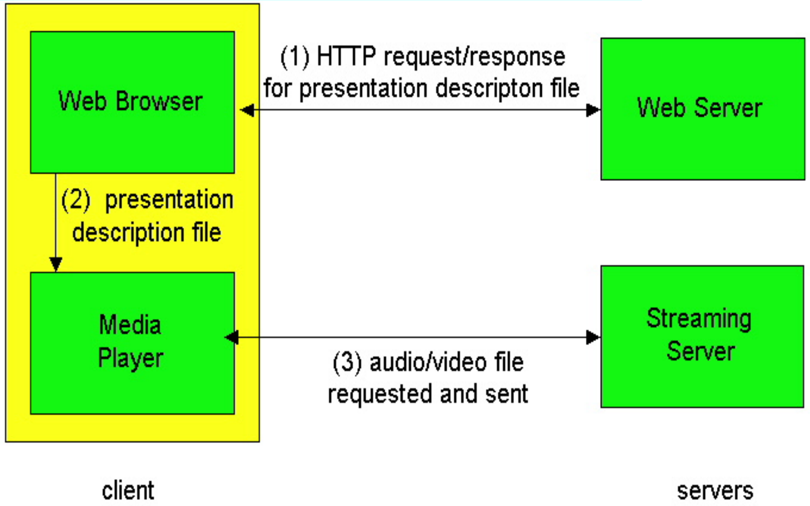


- Alternative:
  - browser requests and receives a **Meta File**
  - Browser launches the appropriate Player and passes it the Meta File;
  - Player sets up a HTTP connection with Web Server and downloads + plays the file



Using a Streaming Server

gets around HTTP = allows a choice of UDP vs. TCP

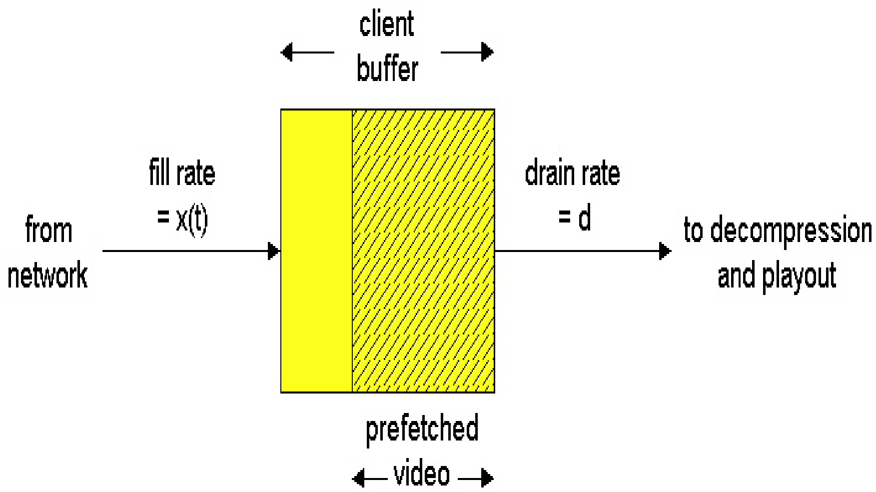


UDP

- sending rate appropriate for client (oblivious to network congestion !) = encoding rate = constant rate
- fill rate = constant rate - packet loss rate
- short playout delay (2-5 seconds) to remove jitter
- error recovery: time permitting

TCP

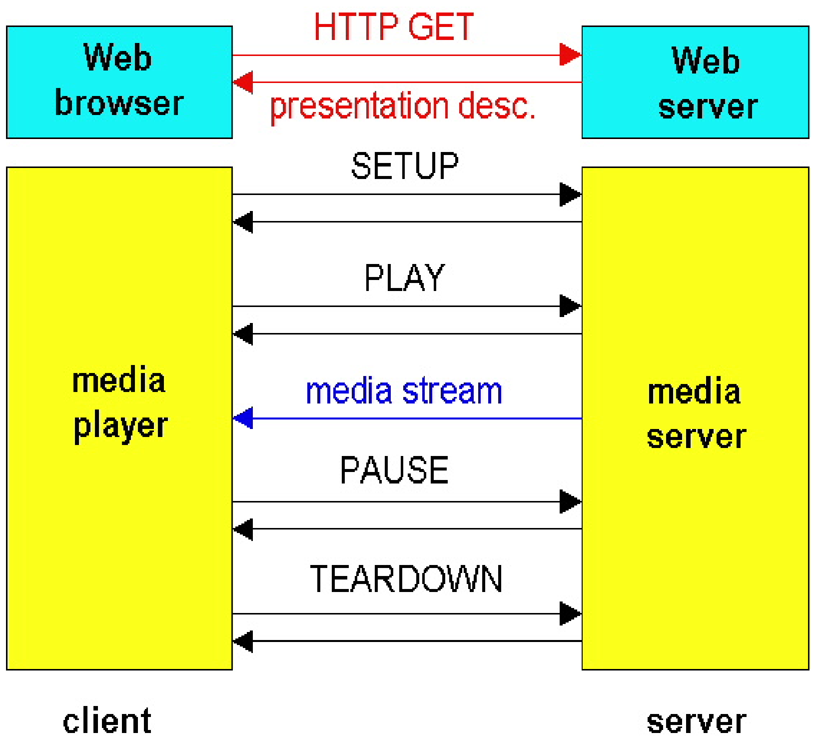
- Send rate as instructed by TCP's flow and congestion ctrl
  - fill rate fluctuates due to TCP congestion control
- larger playout delay: smooth TCP delivery rate
- TCP passes easier through firewalls



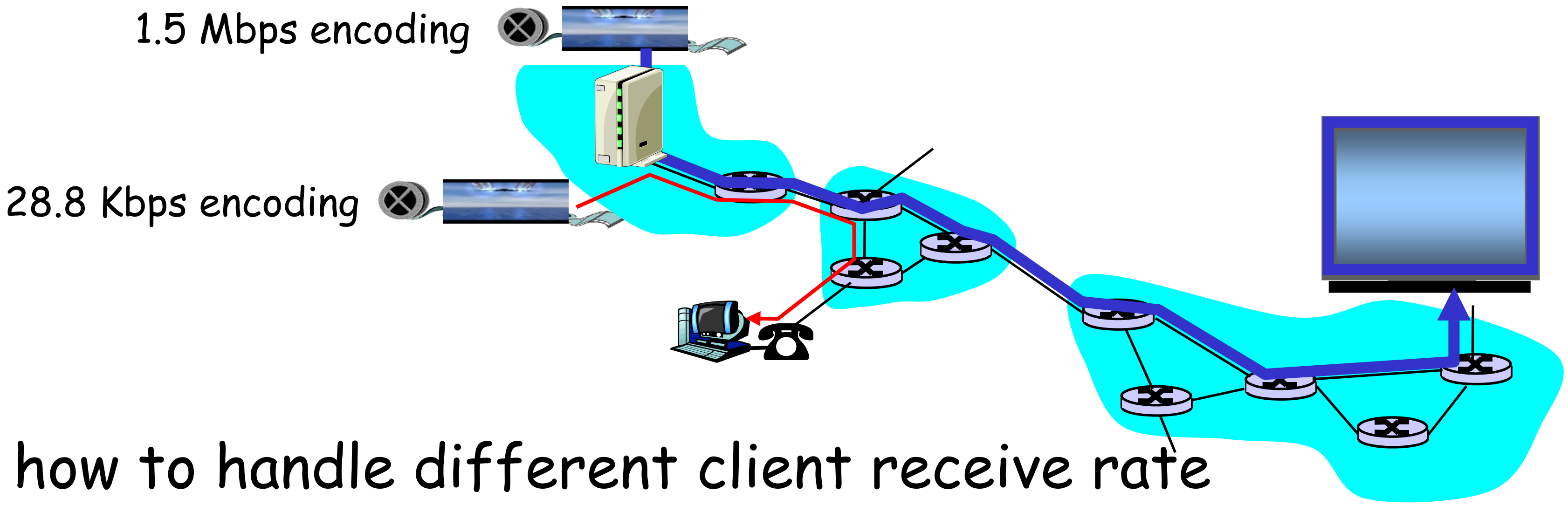
Real Time Streaming Protocol (RTSP)

... replaces http, adds control:

- For user to control display: rewind, fast forward, pause, resume, etc...
- Out-of-band protocol (uses two connections, one for control messages (Port 554) and for media stream)
- RFC 2326 permits use of either TCP or UDP



Streaming Multimedia: client rate(s)



Q: how to handle different client receive rate capabilities?

- 28.8 Kbps dialup
- 100Mbps Ethernet

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

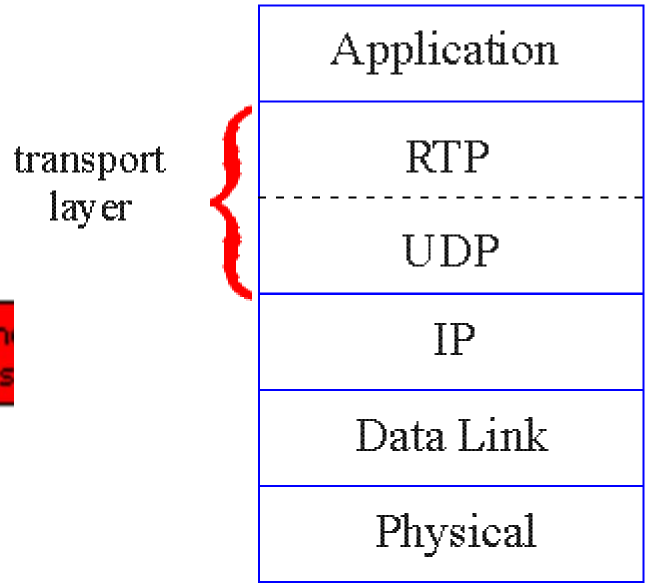
- Info is found in the metafile, RTSP client can GET the preferred one

Real-Time Protocol (RTP) RFC 3550

- RTP specifies packet structure for packets carrying audio, video data
  - payload type (encoding)
  - sequence numbering
  - time stamping

RTP packets encapsulated in UDP segments

- interoperability: if two Internet phone applications run RTP, then they may be able to work together



RTP Header

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
- error recovery (on top of UDP)
  - FEC, interleaving, error concealment
  - Retransmissions only time-permitting
- CDN: bring content closer to clients
- server side matches stream bandwidth to available client-to-server path bandwidth
  - chose among pre-encoded stream rates
  - dynamic server encoding rate

Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

- **Improving QoS in Networks (also related with congestion-control)**
  - Packet scheduling and policing

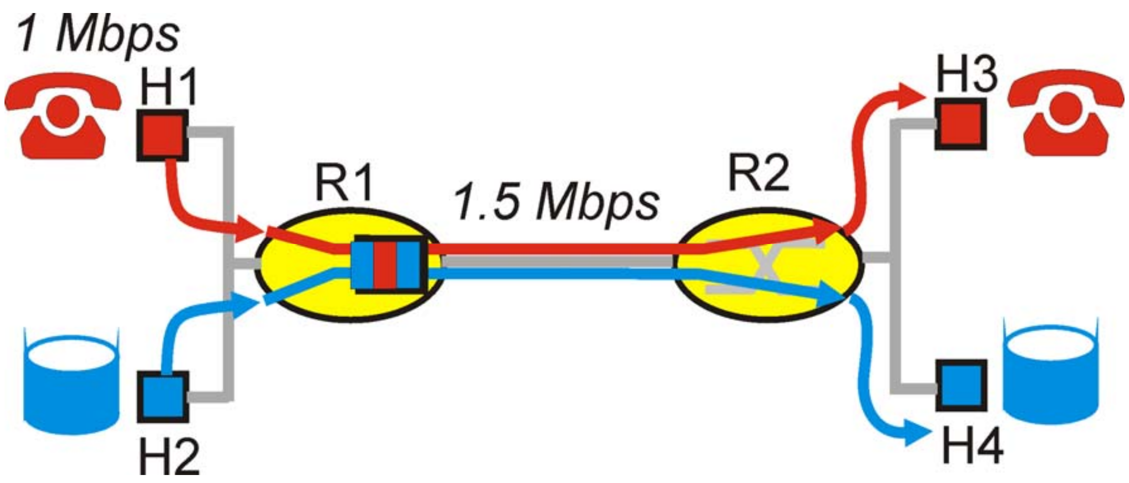
- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

QoS parameters: recall ....

- **Contract** between
  - network user
  - network provider
- ***Agree on***
  - Traffic characteristics (packet rate, sizes, ...)
  - Network service guarantees (delay, jitter, loss rate, ...)

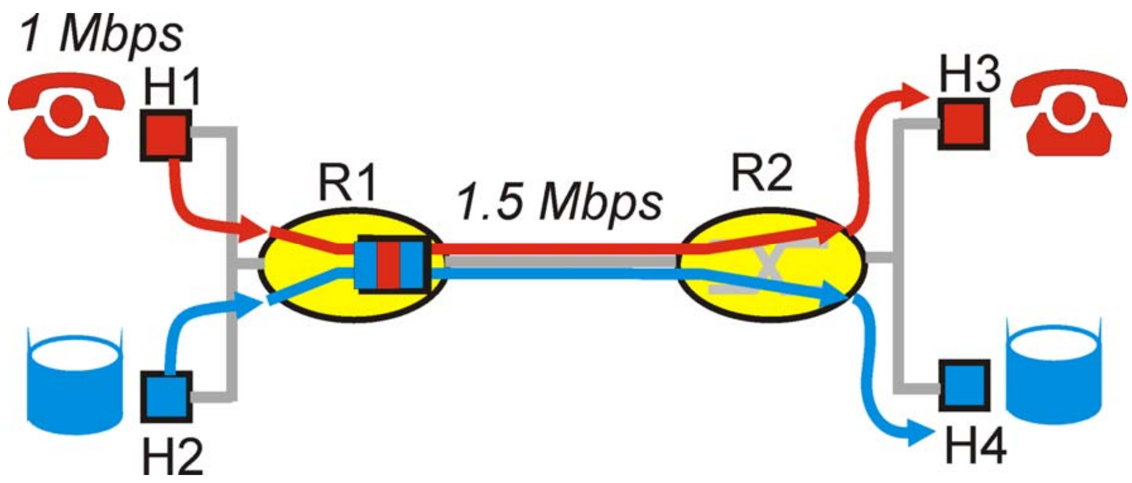
Improving QOS in IP Networks

- IETF groups are working on proposals to provide better QOS control in IP networks, i.e., going beyond best effort
- Simple model for sharing and congestion studies:
- Questions
  - Distinguish traffic?
  - Control offered load? (isolate different "streams"?)
  - Resources? (utilization)
  - Control acceptance of new sessions?



Principles for QoS for networked applications

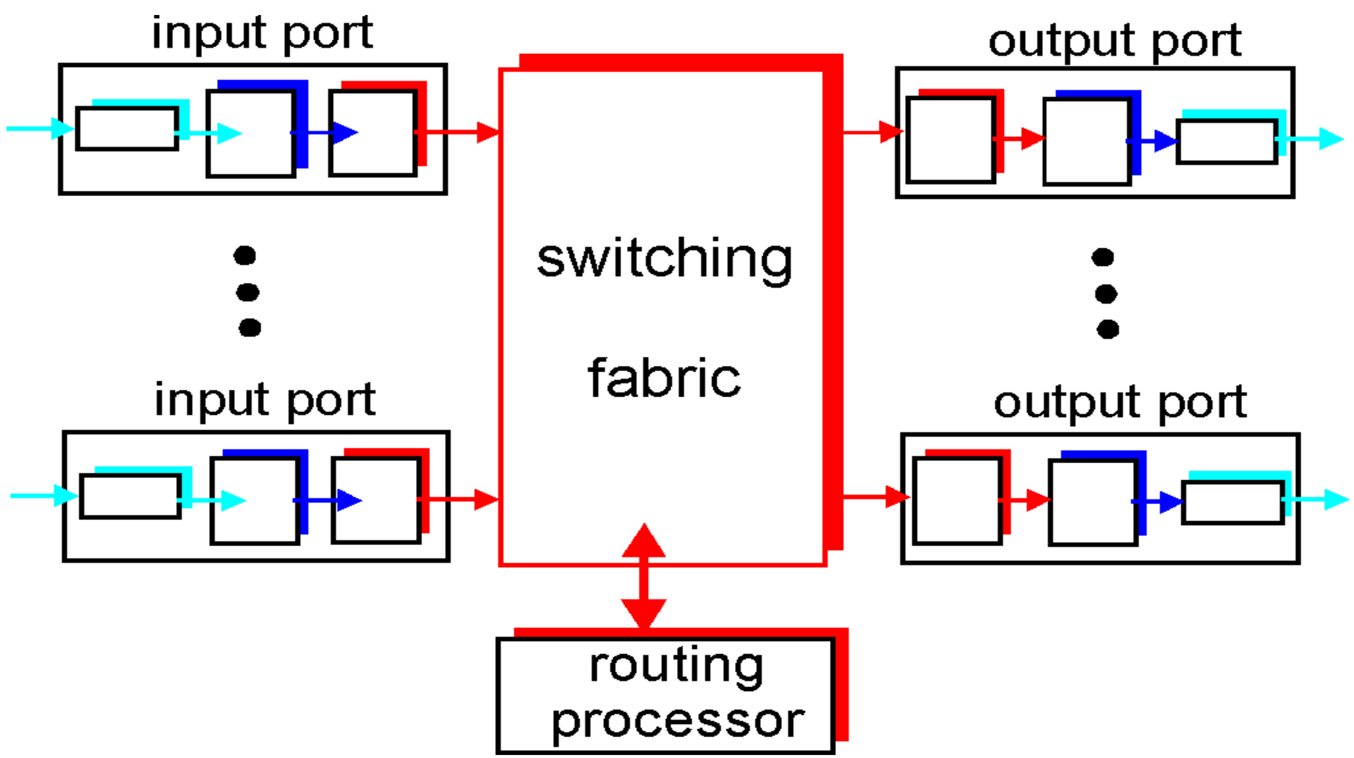
- Packet classification
- Traffic shaping/policing (enforce contract terms)
- Packet scheduling (resource=bandwidth allocation)
- Admission control (will not study here)



Where does this fit in?

Where does this fit in?

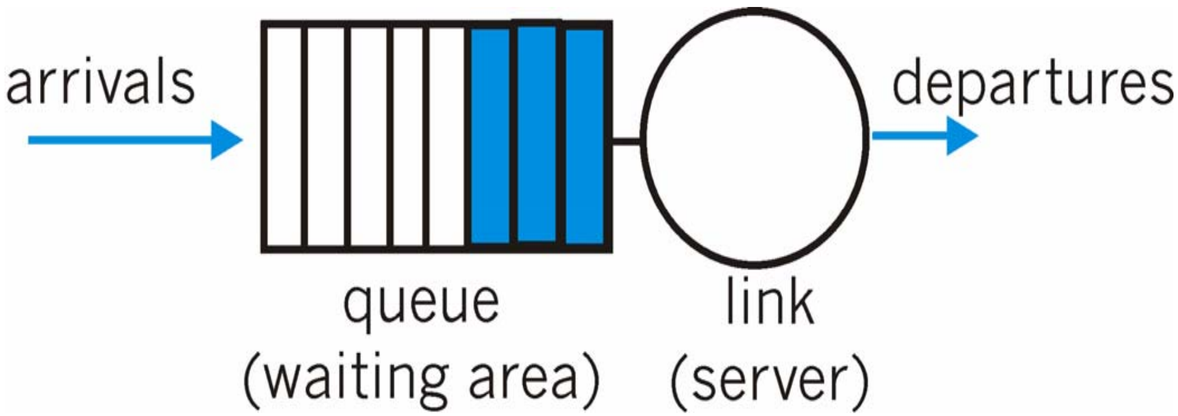
Scheduling = choosing the next packet for transmission on a link (= allocate bandwidth)



Packet Scheduling Policies: FIFO

FIFO: in order of arrival to the queue

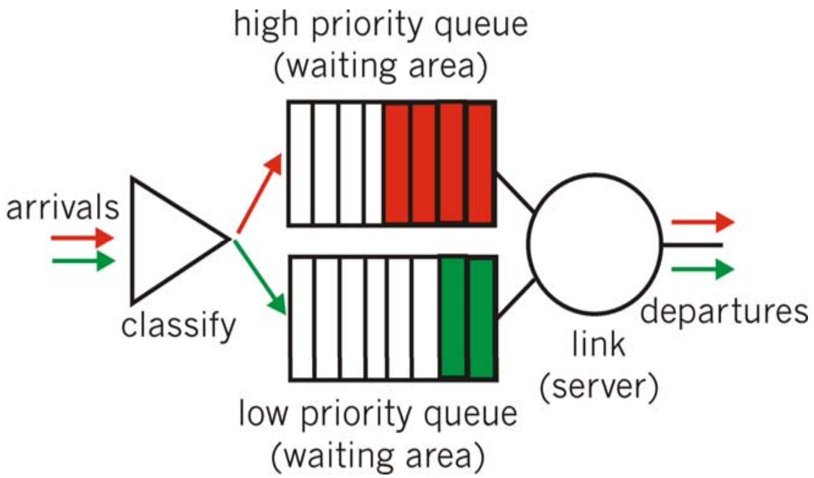
- if buffer full: a discard policy determines which packet to discard among the arrival and those already queued

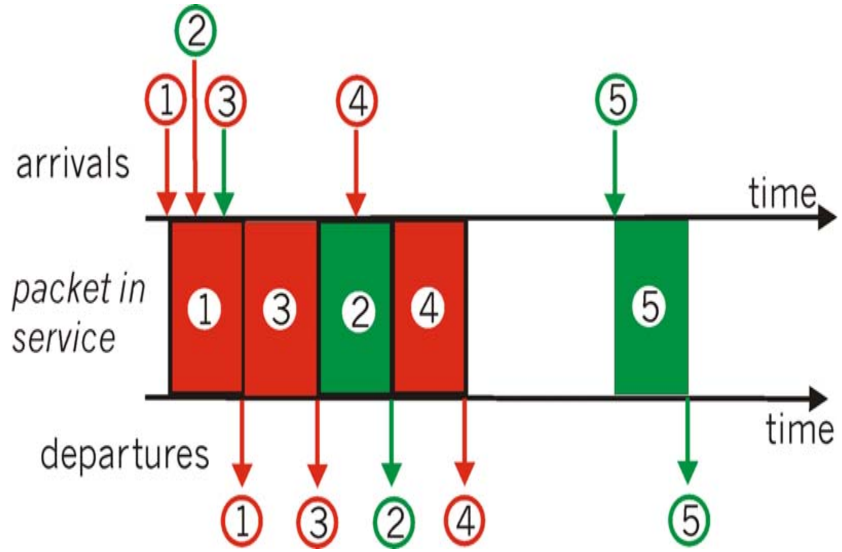


Packet Scheduling Policies: Priority queueing

Priority Queuing: classes have different priorities; priority may depend on explicit marking or other header info, eg IP source or destination, type of packet, etc.

- Transmit a packet from the highest priority class with a non-empty queue

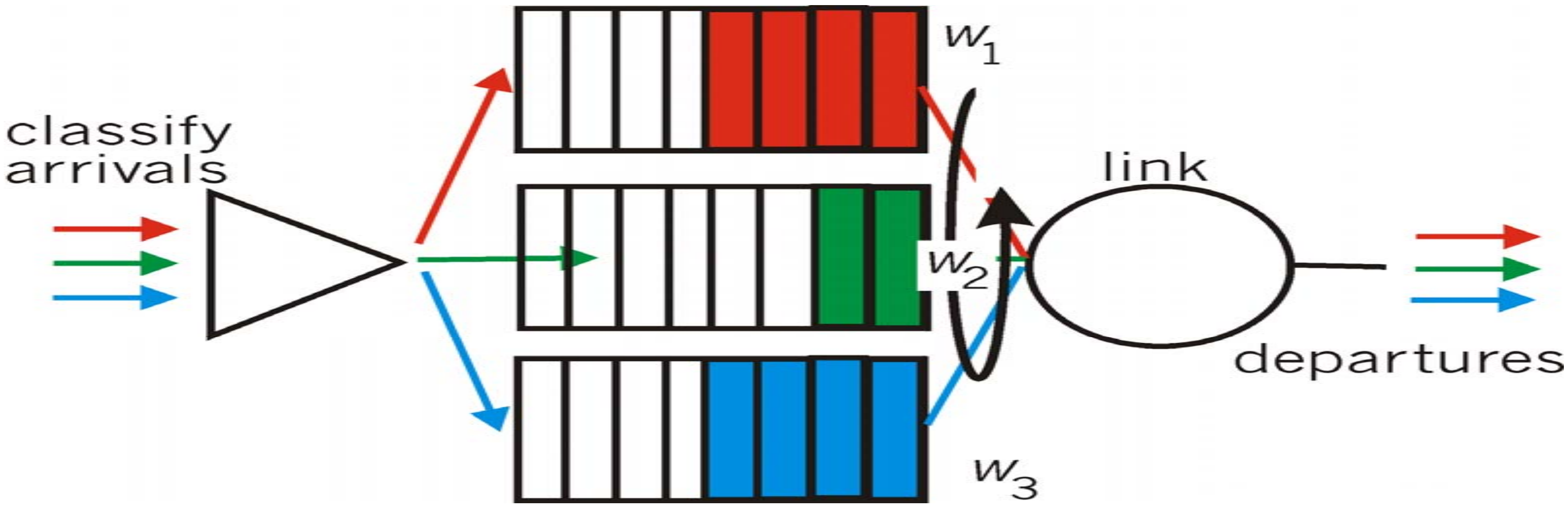




Scheduling Policies: Weighted Fair Queueing

Weighted Fair Queuing: generalized Round Robin, including priorities (weights)

- provide each class with a differentiated amount of service
- class i receives a fraction of service $w_i/\Sigma(w_j)$



- More on packet scheduling: work-conserving policies, delays, …

Policing Mechanisms

Idea: *shape* the packet traffic (the network provider does *traffic policing,* ie monitors/enforces the "shape" agreed).

- Traffic shaping, to limit transmission rates:
  - (Long term) **Average Rate** (100 packets per sec or 6000 packets per min), crucial aspect is the interval length
  - **Peak Rate**: e.g., 6000 p p minute Avg and 1500 p p sec Peak
  - (Max.) **Burst Size**: Max. number of packets sent consecutively, ie over a very short period of time

Policing Mechanisms: Pure *Leaky Bucket*

Idea: eliminates bursts completely; may cause unnecessary packet losses

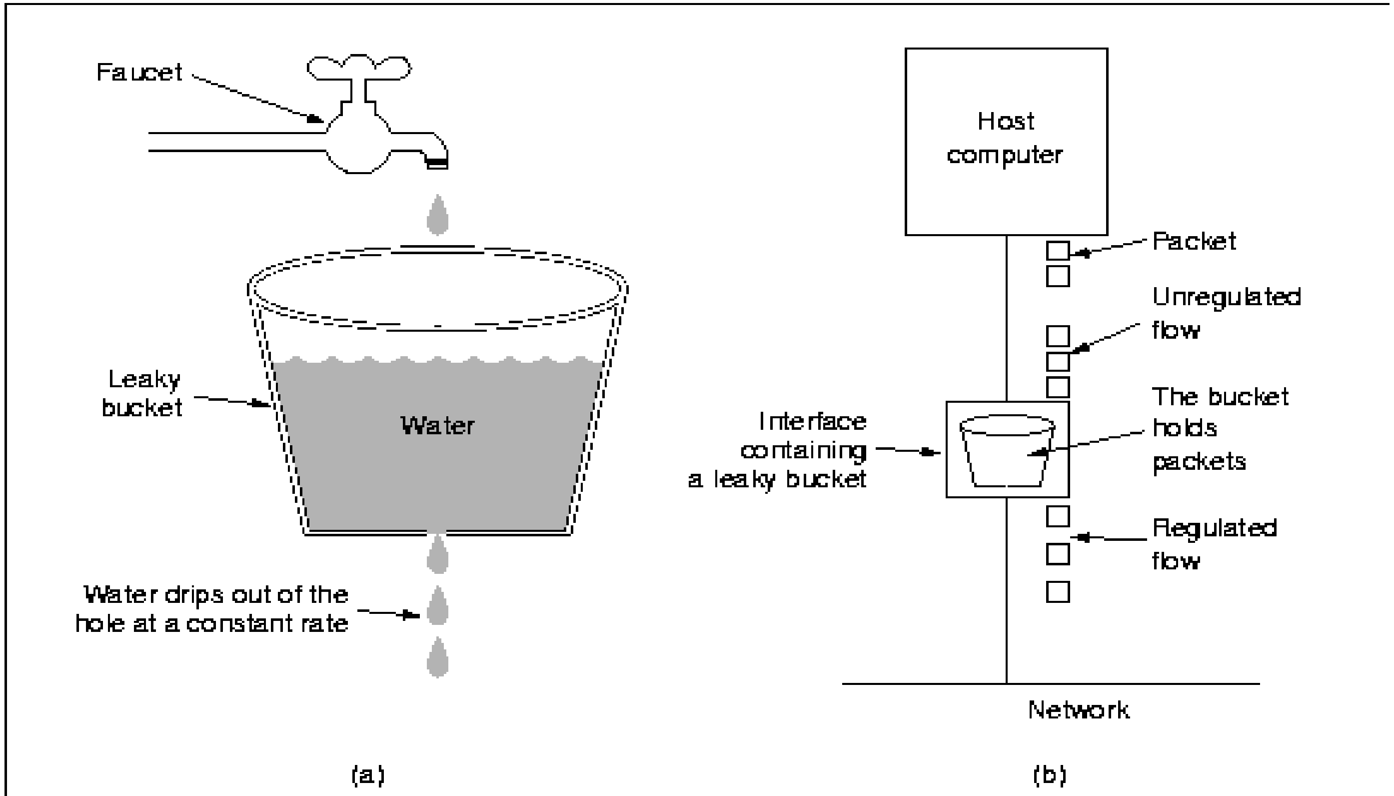


Fig. 5-24. (a) A leaky bucket with water. (b) A leaky bucket with packets.

Policing Mechanisms:Leaky *Token Bucket*

Idea: packets sent by consuming tokens produced at constant rate r

- limit input to specified Burst Size (b= bucket capacity) and Average Rate (max admitted #packets over time period t is b+rt).
- to avoid still much burstiness, put a leaky bucket -with higher rate; *why?*- after the token bucket)

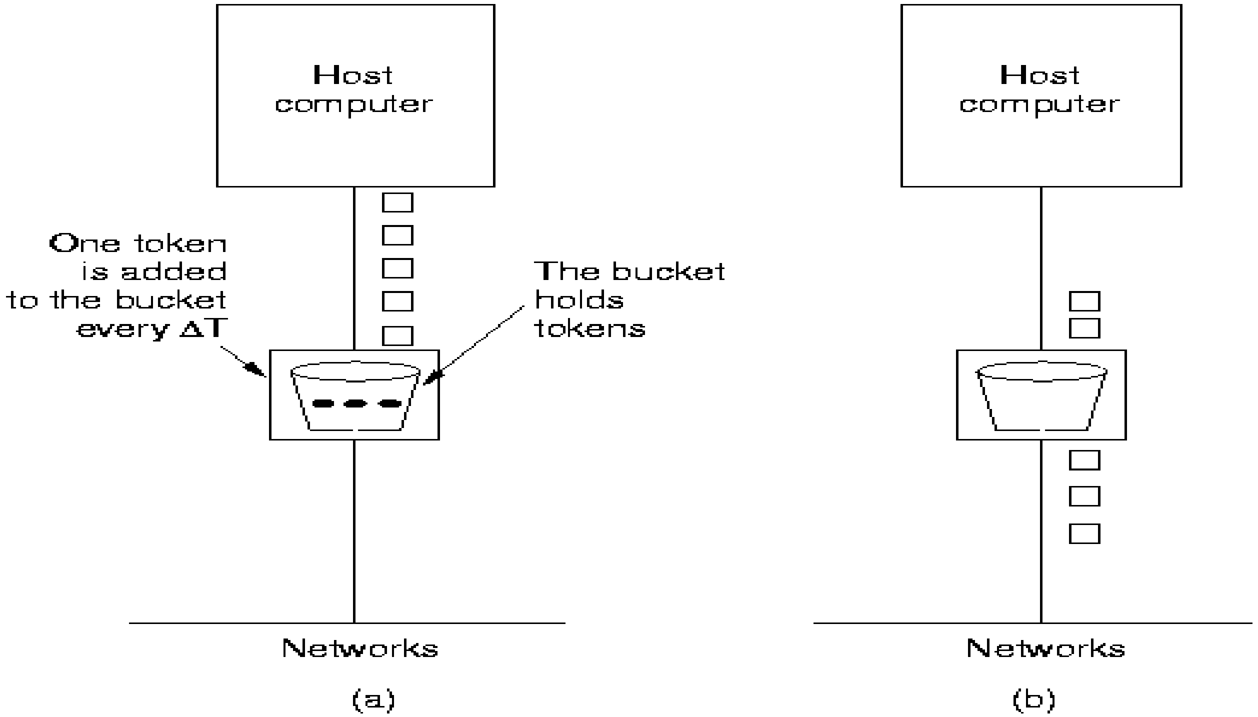
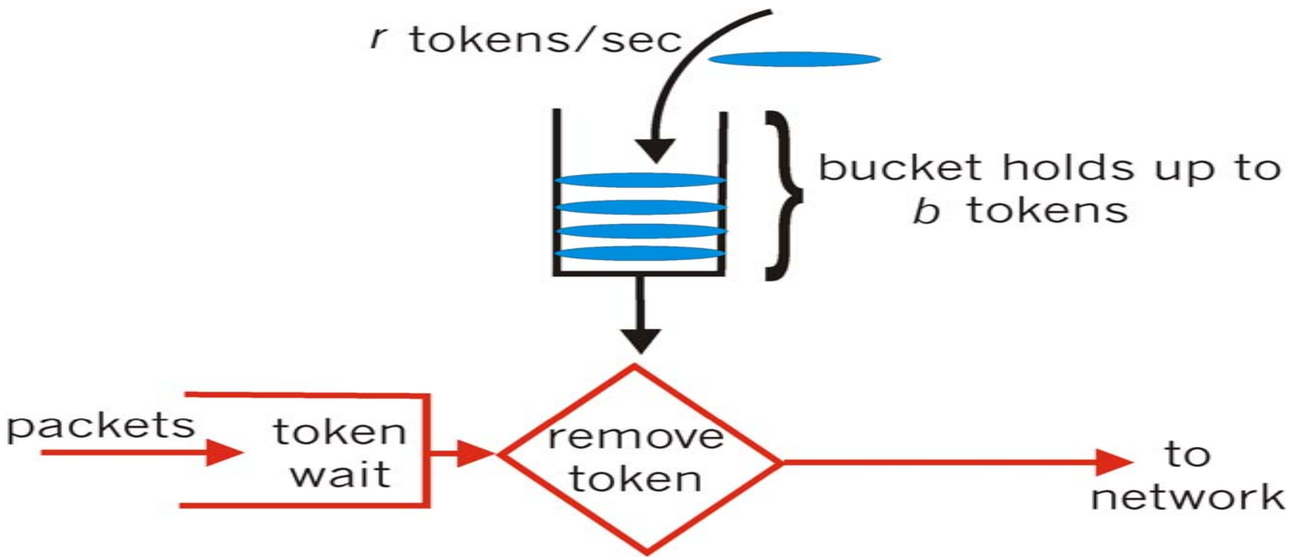


Fig. 5-26. The token bucket algorithm. (a) Before. (b) After.

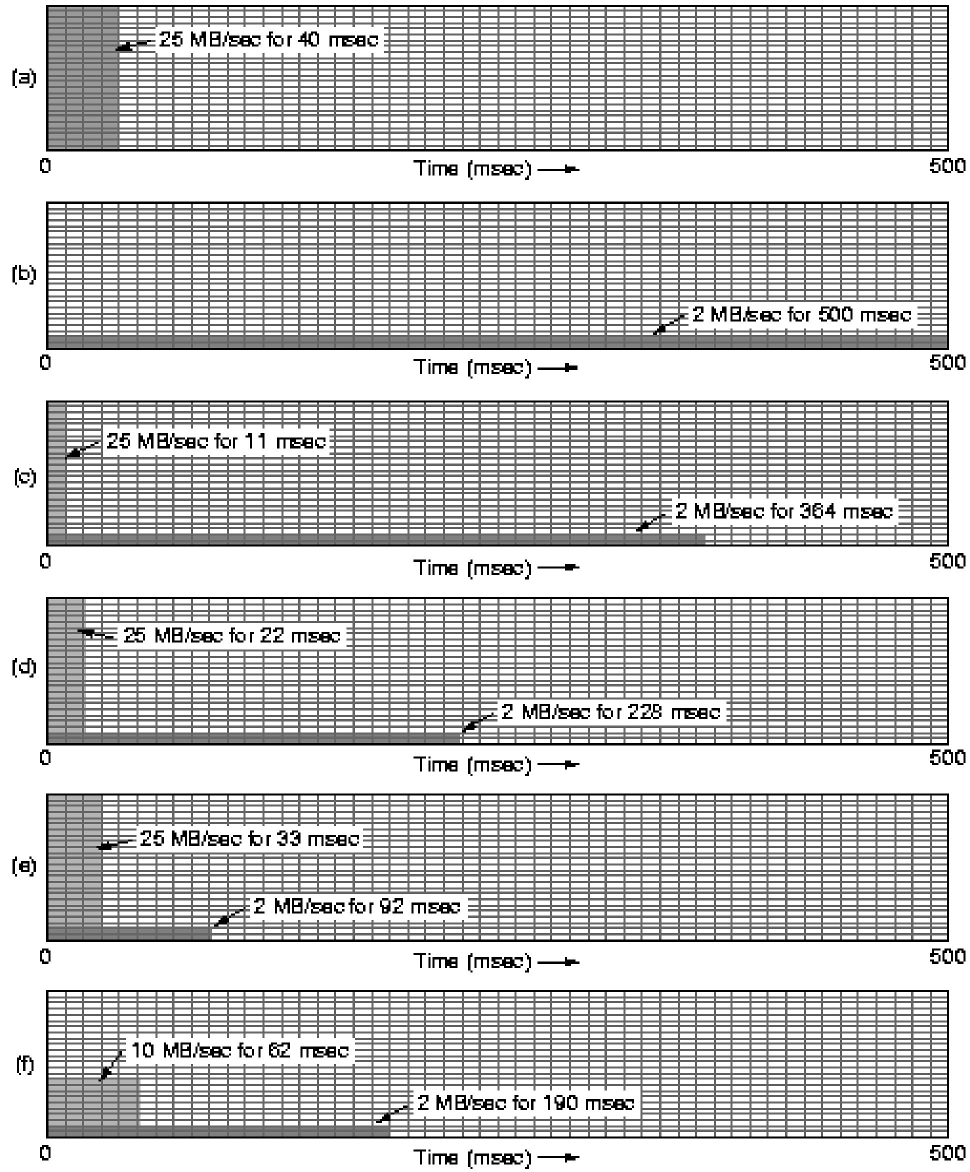
Policing Mechanisms: token bucket

Another way to illustrate token buckets:



Policing: the effect of buckets

- input
- output pure leaky bucket, 2MBps
- output token bucket 250KB, 2MBps
- output token bucket 500KB, 2MBps
- output token bucket 750KB, 2MBps
- output 500KB, 2MBps token bucket feeding 10MBps leaky bucket



Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

- Improving QoS in Networks (also related with congestion-control)
  - Packet scheduling and policing

- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

(VC) ATM: Asynchronous Transfer Mode nets

Internet:

- today's *de facto* standard for global data networking

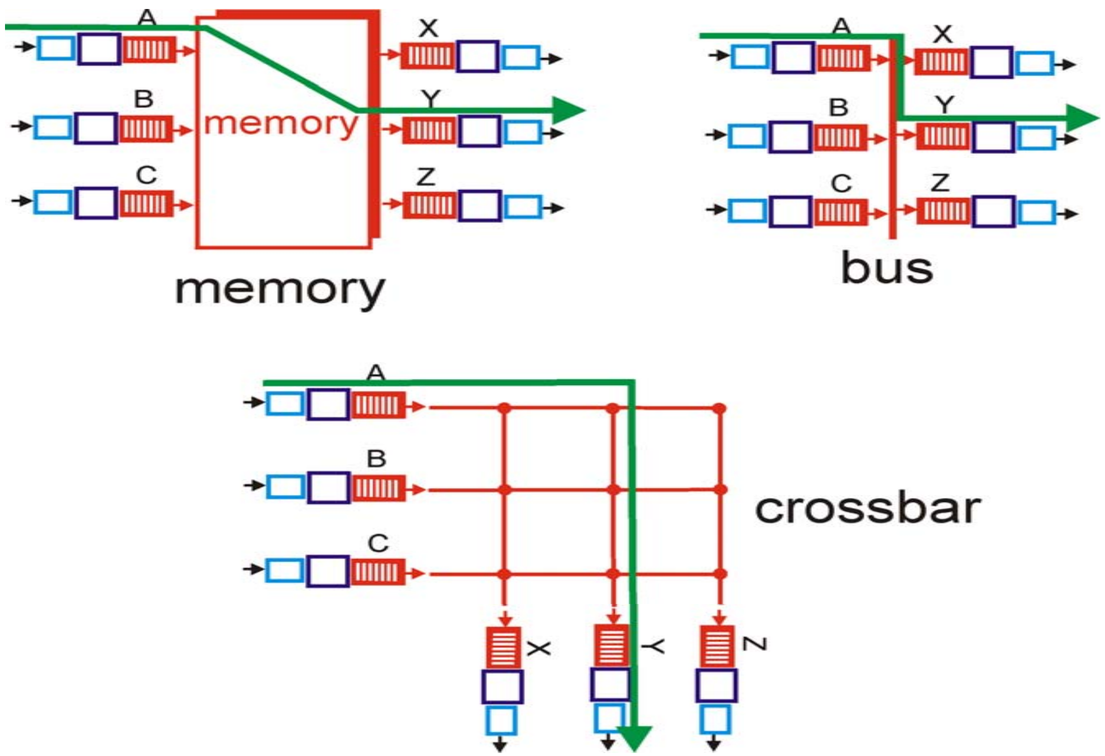
1980's:

- telco's develop ATM: competing network standard for carrying high-speed voice/data

ATM principles:

- virtual-circuit networks: switches maintain state for each "call"
- small (48 byte payload, 5 byte header) fixed length *cells* (like packets)
  - fast switching
  - small size good for voice
- Assume low error-rates, do not perform error control (enhance speed)
- well-defined interface between "network" and "user" (think of telephone company)

Recall: switching fabrics



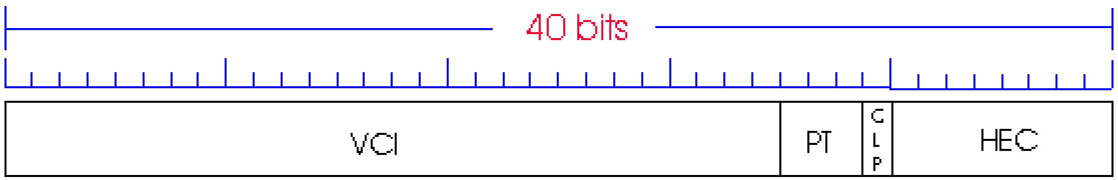
- ATM switches: VC technology
  - Virtual channels, virtual circuits

    Based on Banyan crossbar switches
- ATM routing: as train travelling (hence no state for each ”stream”, but for each ”train”)

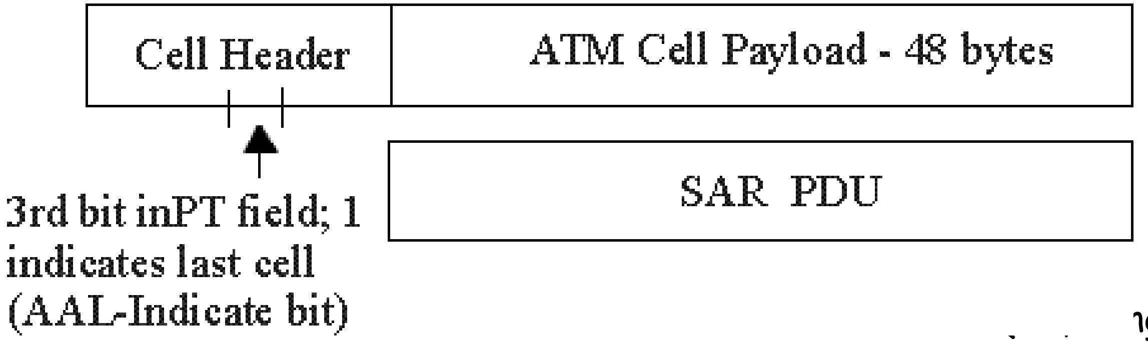
ATM cell (small packet)

- 48-byte payload
  - Why?: small payload -> short cell-creation delay for digitized voice
  - halfway between 32 and 64 (compromise!)
- **Header: 5bytes**
  - **VCI:** virtual channel ID
  - **PT:** Payload type (e.g. Resource Management cell versus data cell)
  - **CLP:** Cell Loss Priority bit
    - CLP = 1 implies low priority cell, can be discarded if congestion
  - **HEC:** Header Error Checksum

Cell header



Cell format

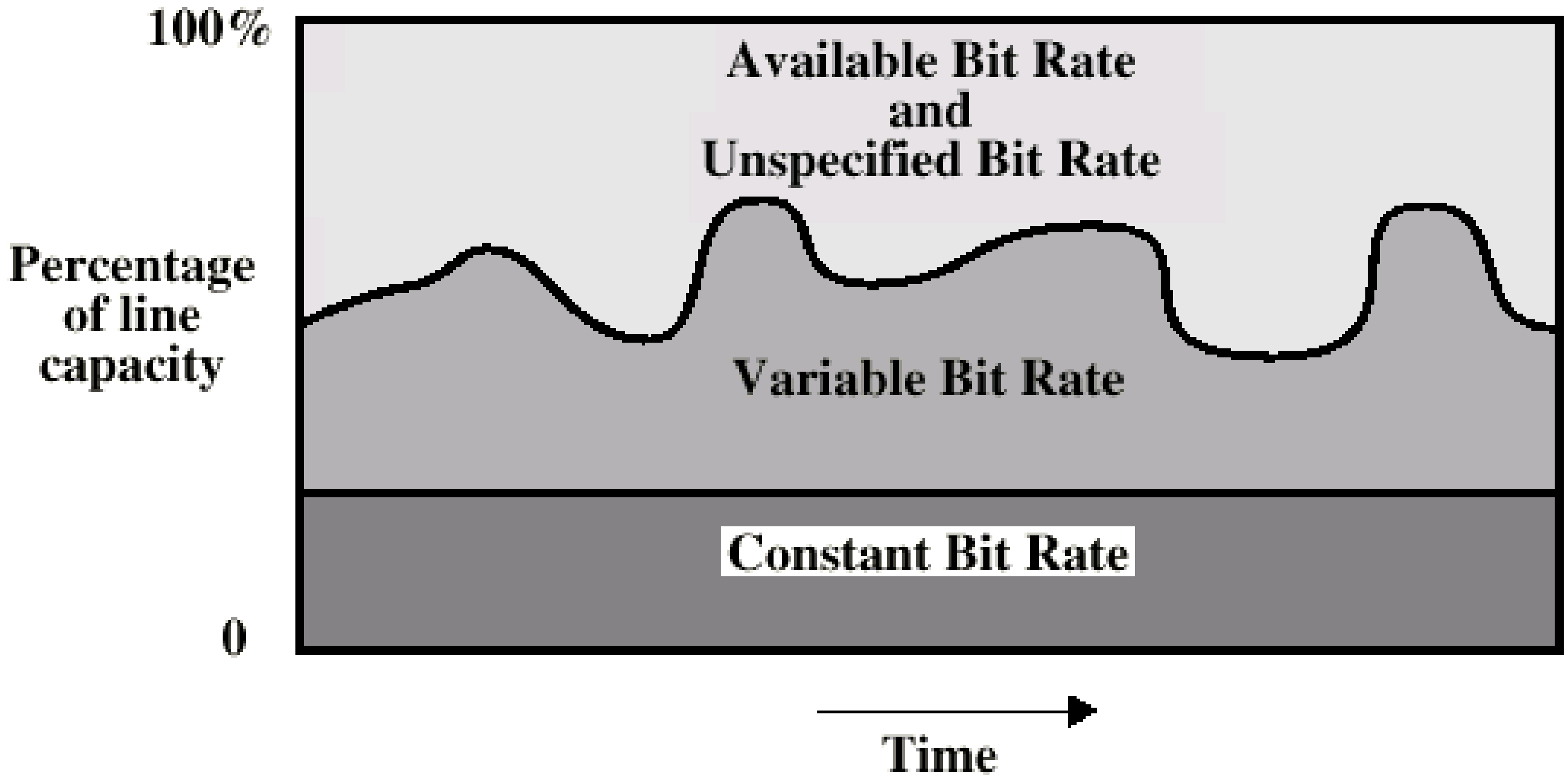


Example VC technology
ATM Network service models:

Service Model	Example	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Constant Bit Rate	voice	constant rate	yes	yes	yes	no congestion
VariableBR (RT/nRT)	Video/ “streaming”	guaranteed rate	yes	yes	yes	no congestion
Available BR	www-browsing	guaranteed minimum	no	yes	no	yes
Undefined BR	Background file transfer	none	no	yes	no	no

- With ABR you can get min guaranteed capacity and better, if possible; with UBR you can get better, but you may be thrown out in the middle ☹

ATM Bit Rate Services



ATM Congestion Control

Several different strategies are used:

- **Admission control and resource reservation**: reserve resources when opening a VC; traffic shaping and policing *(use bucket-like methods)*
- **Rate-based congestion control**: similar to choke packets (method provided in IP (ICMP) also, but not really used in implementations); (especially for ABR traffic)

  idea = give feedback to the sender and intermediate stations on the *min. available (= max. acceptable) rate* on the VC.

Multimedia Applications, Services, Needs, …

- Application Classes, QoS, challenges
- Today's representative technology
  - recovery from jitter and loss (eg IP telephony)
  - (Overlays) CDN: content distribution networks
  - Streaming protocols

- Improving QoS in Networks (also related with congestion-control)
  - Packet scheduling and policing

- Two generally different approaches
  - The VC (ATM) approach (incl. material from Ch 3, 4, 5)
  - Internet approach: Int-serv + RSVP, Diff-serv

Recall:
Solution Approaches in IP Networks

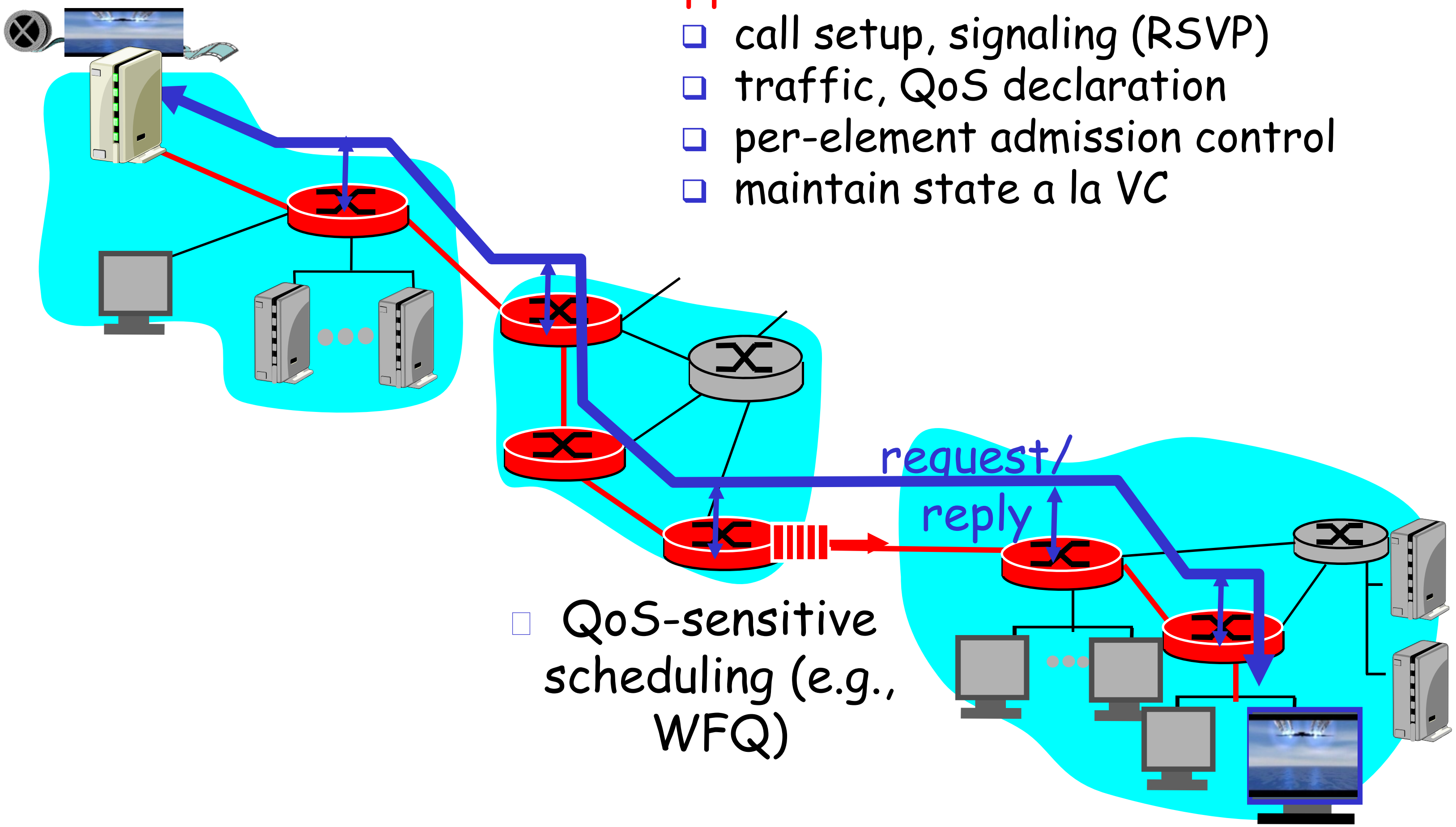
- To mitigate impact of "best-effort" protocols:
  - Use UDP to avoid TCP's slow-start phase…
  - Buffer content at client and control playback to remedy jitter
  - Adapt compression level to available bandwidth
  - Exhaust all uses of caching, proxys, etc
  - add more bandwidth

Scalability? May need major change of the protocols (?):

- Incorporate resource reservation (bandwidth, processing, buffering), and new scheduling policies
- Use traffic classes for packets and differentiate service accordingly
- Set up service level agreements with applications, monitor and enforce the agreements, charge accordingly

Intserv: QoS guarantee scenario

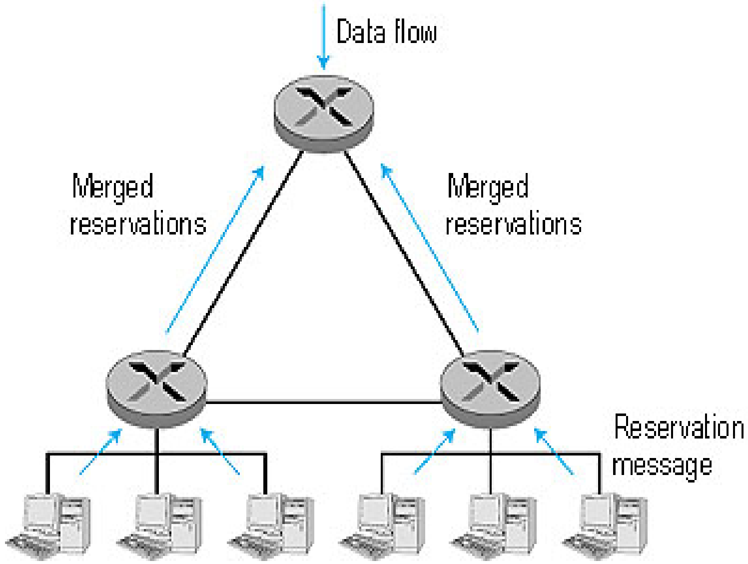
- Resource reservation per individual application session
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control
  - maintain state a la VC



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

RSVP: resource reservation protocol

- RSVP: a leading candidate for signaling protocol
  - allows reservations for bandwidth in multicast trees
  - is receiver-oriented (the receiver of a data flow initiates and maintains the resource reservation for the flow).
  - Maintains soft-state
    - receivers renew interest regularly
  - does not specify how the network provides the reserved bandwidth, only allows the applications to reserve it.
  - is not a routing protocol; it depends on an underlying routing protocol to determine the routes for the flows; when a route changes, RSVP re-reserves resources.
  - does not define the admission test, but it assumes that the routers perform such a test and that RSVP can interact with the test.



Back to Internet QoS support:alternatively?

Concerns with Intserv:

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows

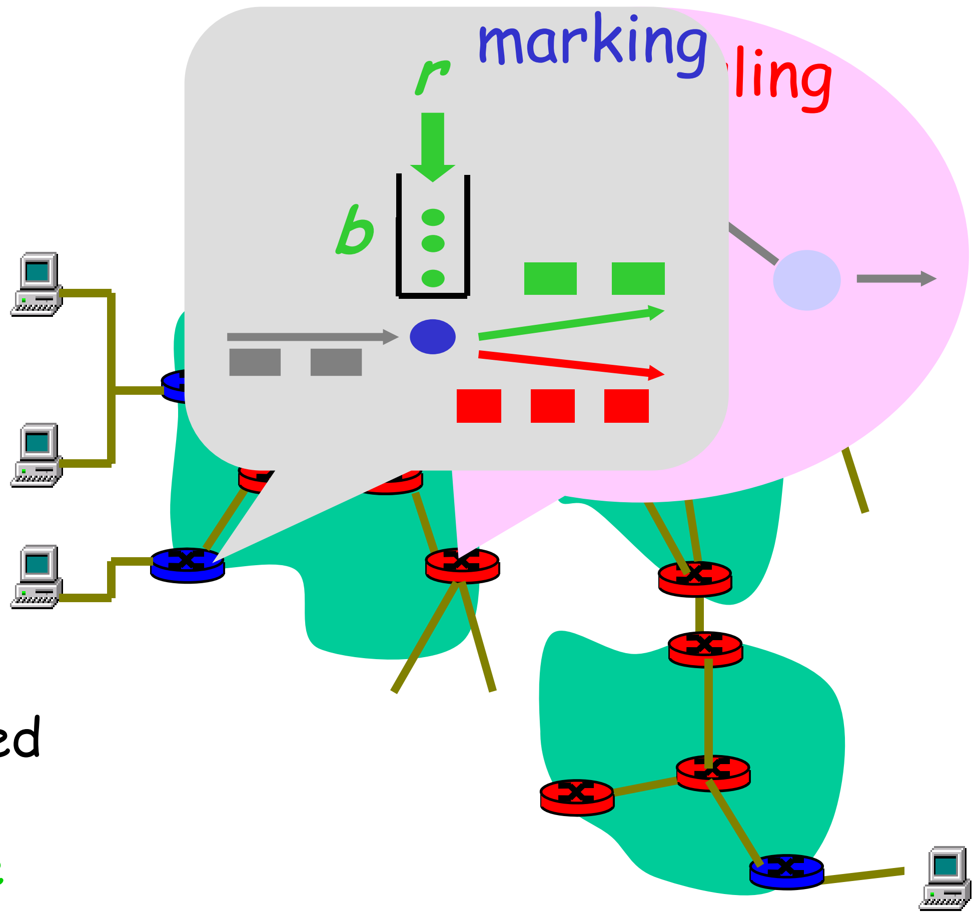
Diffserv approach:

- Don't define service classes, provide functional components to build service classes
  - Network core: stateless, simple
  - Combine flows into aggregated flows
  - Classification, shaping, admission at the network edge

Diffserv Architecture

Edge router:

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

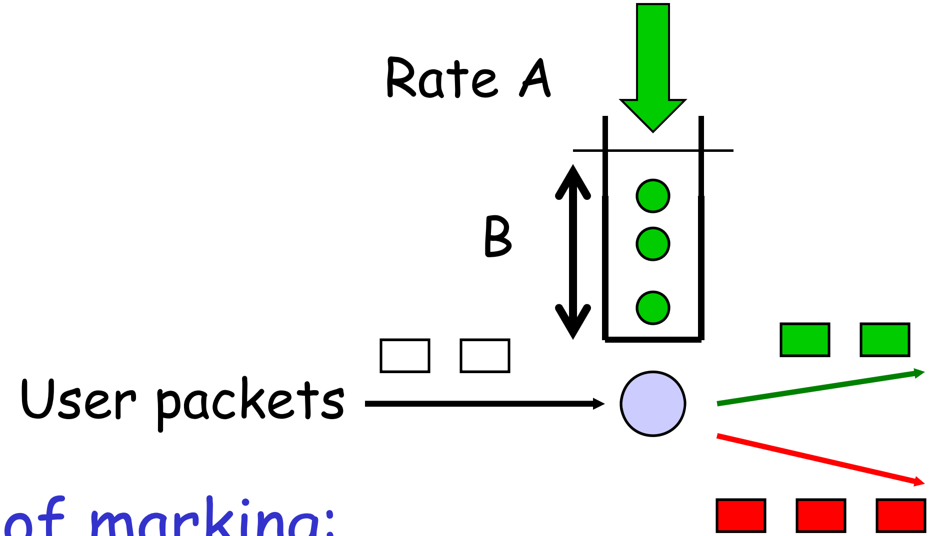


Core router:

- 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
- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6

DiffServ Core Functions

- **Forwarding**: according to "Per-Hop-Behavior" (PHB) specified for the particular packet class; PHB is strictly based on classification marking
  - 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 before packets from class B

- BIG ADVANTAGE:

  No state info to be maintained by routers!

Summary: How should the Internet evolve to better support multimedia?

Integrated services philosophy:

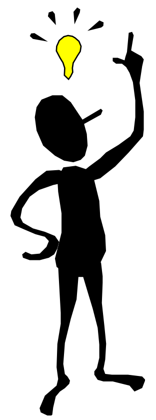
- Fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- Requires new, complex software in hosts & routers

Laissez-faire

- no major changes
- more bandwidth when needed
- Let application layer solve the problems

Differentiated services philosophy:

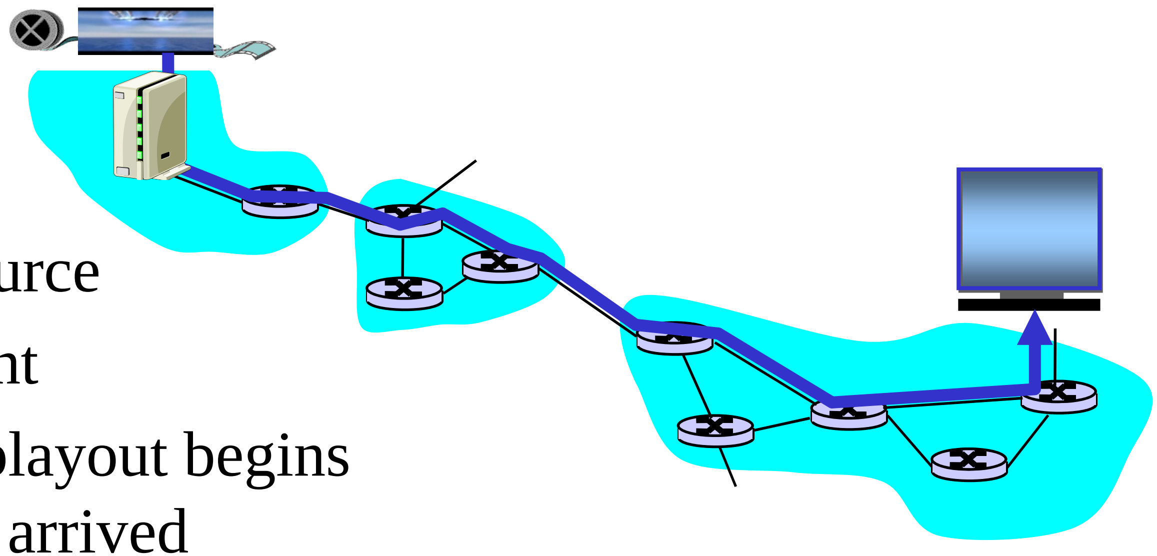
- Fewer changes to Internet infrastructure, yet provide variable class service.



Opinions?

Extra slides/notes

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

- *VCR-like functionality:* client can pause, rewind, FF, push slider bar
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK

Streaming *Live* Multimedia

Examples:

- Internet radio talk show
- live sporting event

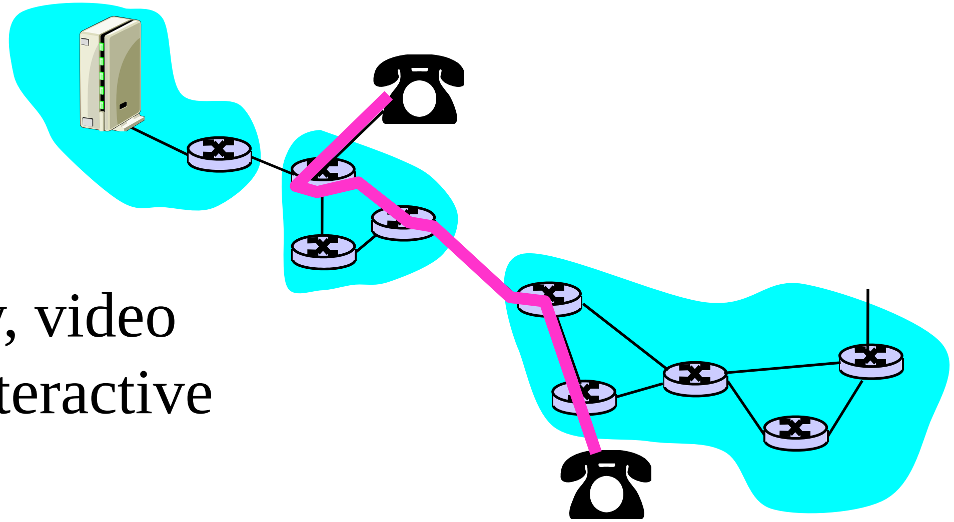
Streaming (as with streaming *stored* multimedia)

- playback buffer (to be explained soon)

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

Real-Time (Phone) Over IP's Best-Effort

- Delay jitter is handled by using timestamps, sequence numbers, and delaying playout at receivers either a fixed or a variable amount

- Forward Error Control: to fix errors, make up losses

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

Streaming

- Audio/Video file is segmented and sent over TCP or UDP;
- User interactive control provided, e.g. **Real Time Streaming Protocol (RTSP)**
- **Helper Application:** displays content, (typically requested via a Web browser); e.g. RealPlayer; typical functions:
  - Decompression
  - Jitter removal
  - Error correction: use redundant packets to be used for reconstruction of original stream
  - GUI for user control

RTSP 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 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

Real-Time Protocol (RTP) & RT Control Protocol (RTCP)

- standard packet format for real-time application
  - **Payload Type**: 7 bits: 128 possible types of encoding; eg PCM, MPEG2 video, GSM, etc. (sender can change in the middle of session)
  - **Sequence Number**: to detect packet loss
  - **Timestamp**: sampling instant of first byte in packet; to remove jitter introduced by the network
  - **Synchronization Source identifier (SSRC)**: id for the source of a stream; assigned randomly by the source

RTP Header

- Real-Time Control Protocol (RTCP): specifies report packets exchanged between sources and destinations, with statistics (# packets sent/lost, inter-arrival jitter
  - Can be used to modify sender transmission rates

SIP Service Initiation Protocol

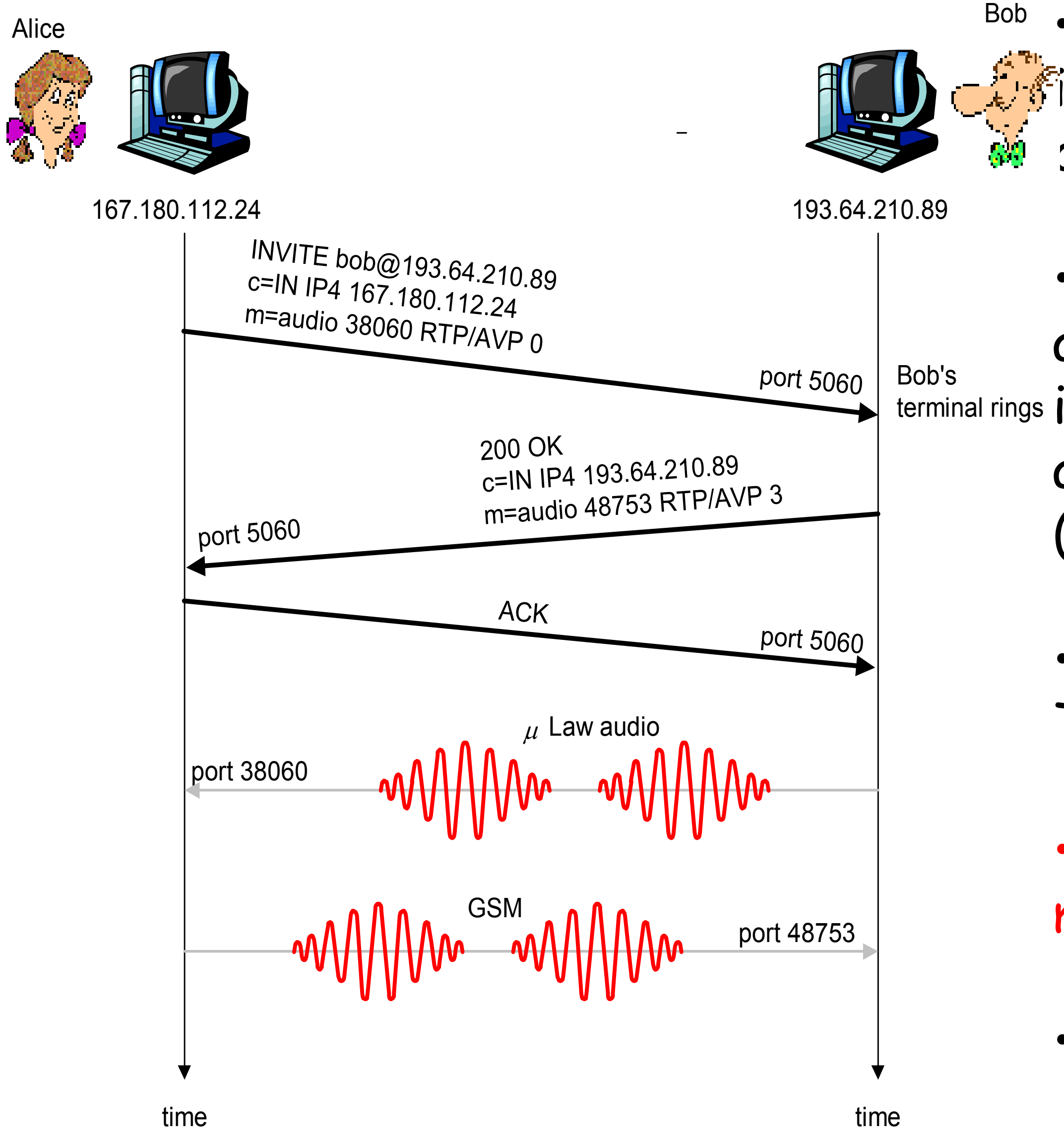
SIP long-term vision

- All phone/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.

What does it do:

- Determine current IP address of callee.
  - Maps mnemonic identifier to current IP address
- Setting up/ending a call
  - Provides also mechanisms so that caller and callee can agree on media type and encoding.
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

Setting up a call to known IP address



- Alice's SIP invite message indicates her port number & IP address+encoding

- Bob's 200 OK message (could also reject, say "busy", etc) indicates his port number, IP address & preferred encoding (GSM)

- SIP messages can be sent over TCP or UDP; here over RTP/UDP.

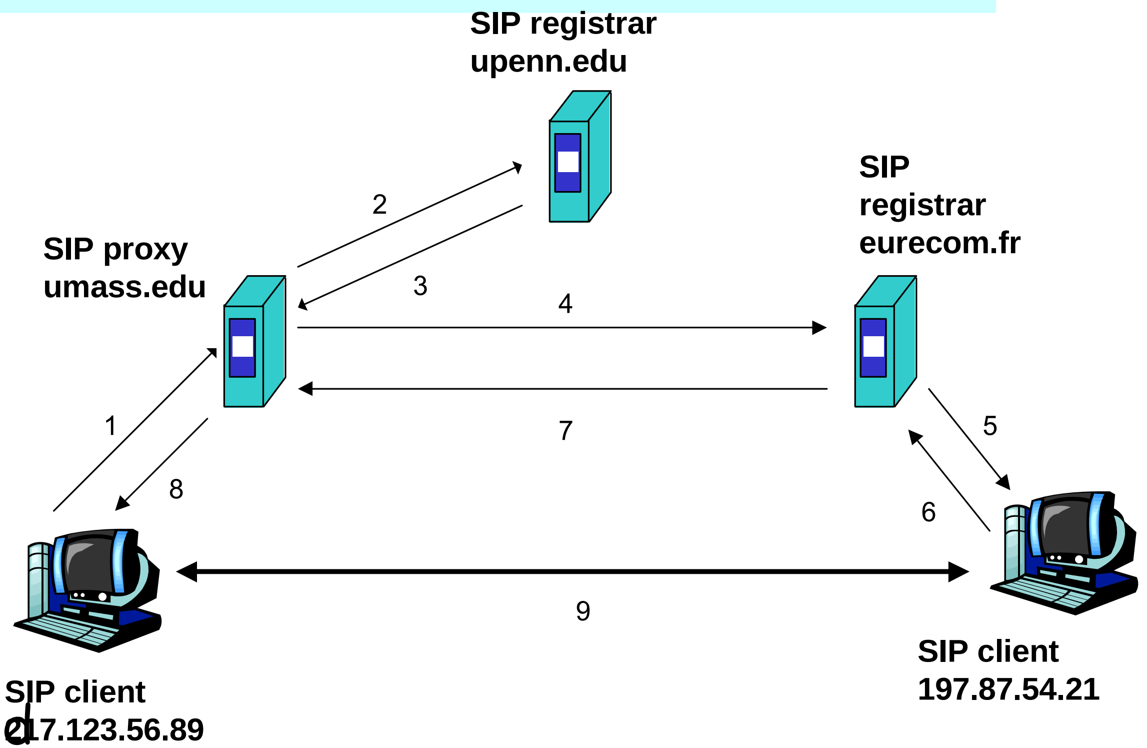
- HTTP message syntax (but SIP maintains state)

- Default SIP port number: 5060.

Example name translation, user location

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

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



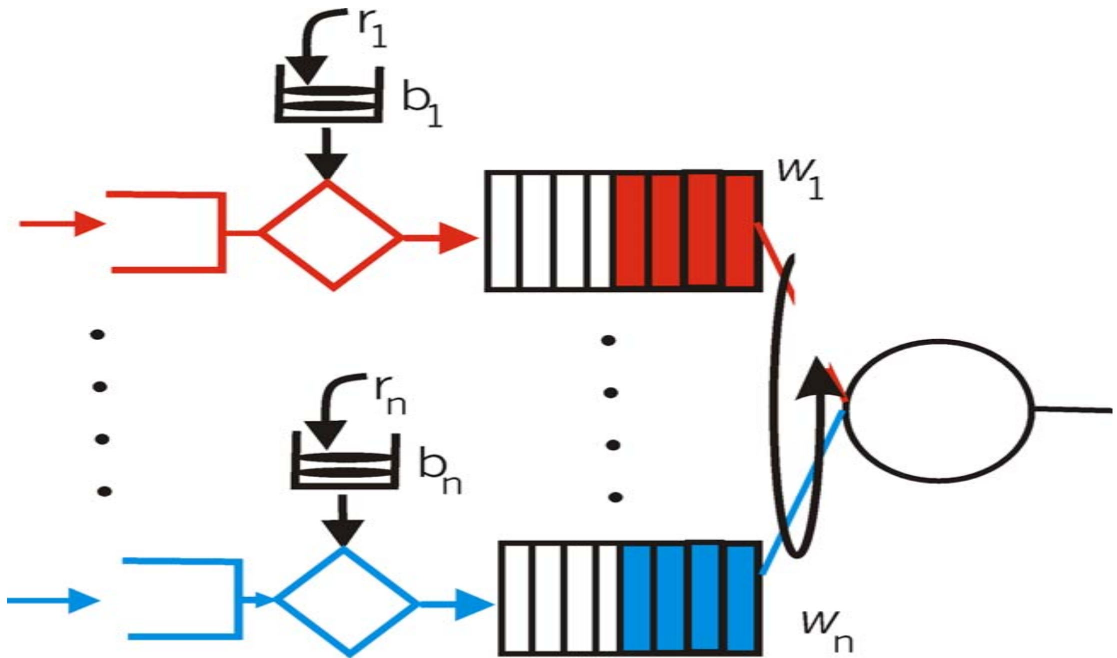
(4) umass proxy sends INVITE to eurecom registrar.
(5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client.
(6-8) SIP response sent back
(9) media sent directly between clients.
(follows pretty much the DNS inquiry structure)

Token bucket + WFQ…

…can be combined to provide upper bound on packet delay in queue:

- b_i packets in queue, packets are serviced at a rate of at least $R \cdot w_i/\sum(wj)$ packets per second, then the time until the last packet is transmitted is at most

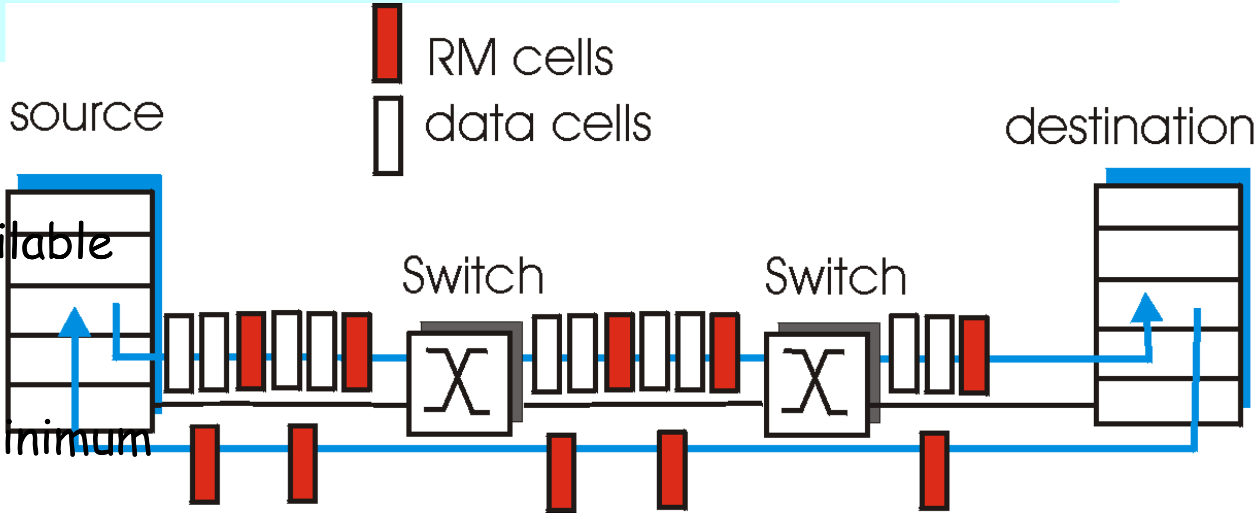
$$b_i / (R \cdot w_i/\sum(wj))$$



ATM ABR congestion control

ABR: available bit rate:

- "elastic service"
- if path "underloaded":
  - sender should use available bandwidth
- if path congested:
  - sender throttled to minimum guaranteed rate



RM (resource management) cells:

- interspersed with data cells
- bits in RM cell set by switches ("*network-assisted*")
  - NI bit: no increase in rate (mild congestion)
  - CI bit: congestion indication two-byte ER (explicit rate) field in RM cell
  - congested switch may lower ER value in cell
  - sender' send rate thus minimum supportable rate on path

Traffic Shaping and Policing in ATM

Enforce the QoS parameters:
check if *Peak Cell Rate (PCR)* and *Cell Delay Variation* (CDVT are within the negotiated limits:

Generic Cell Rate Algo: introduce

expected next time for a successive cell, based on T = 1/PCR

border time L (= CDVT) < T in which next transmission may start (but never before T-L)

A nonconforming cell may be discarded, or its *Cell Loss Priority* bit be set, so it may be discarded in case of congestion

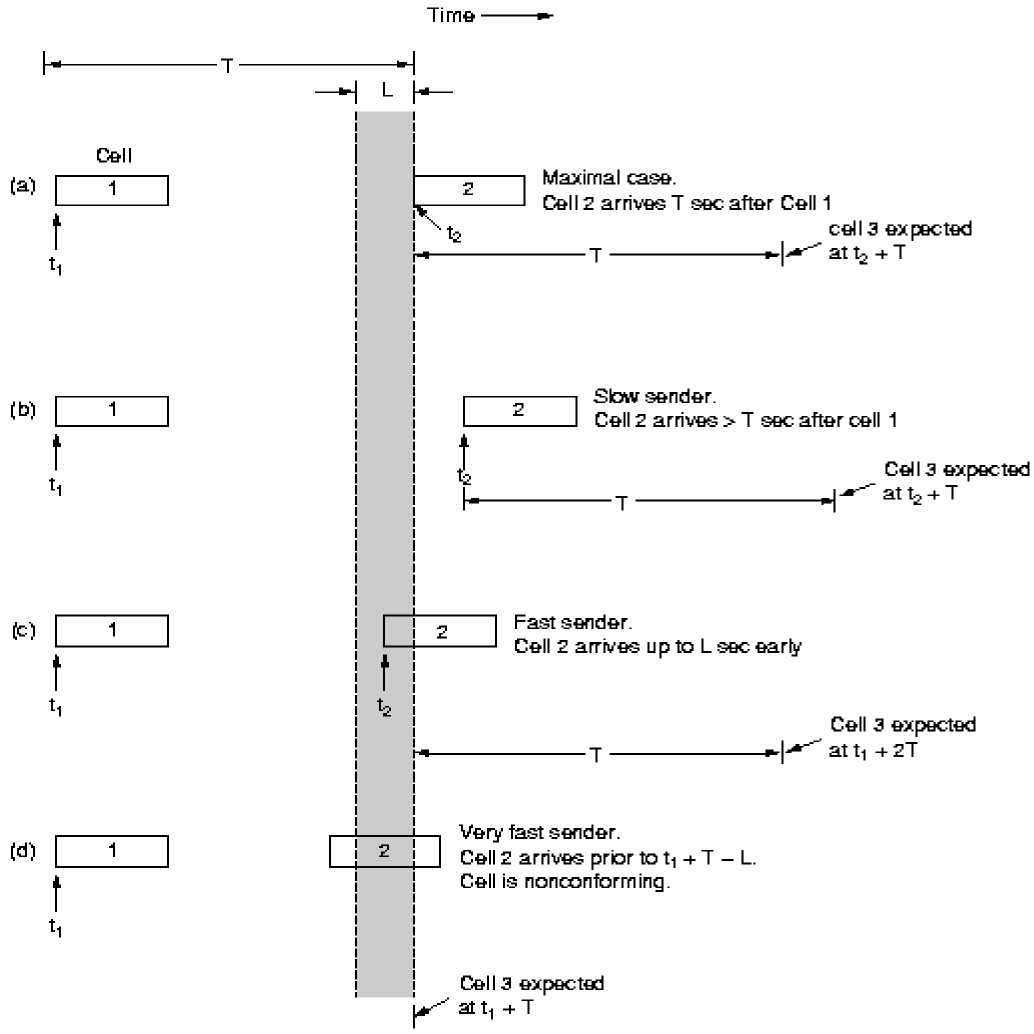


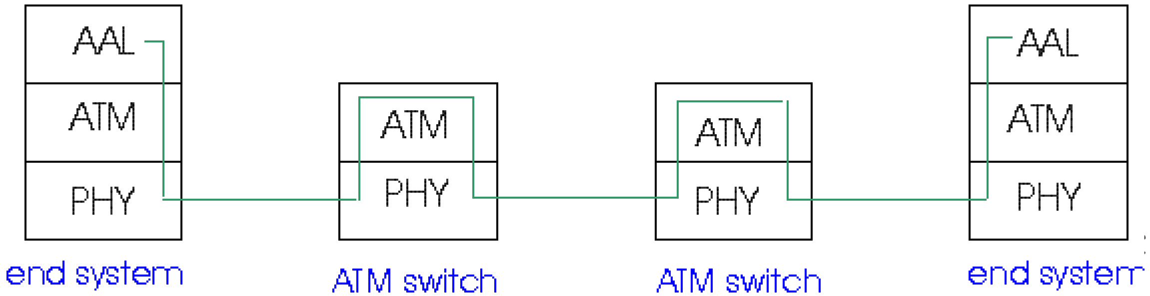
Fig. 5-73. The generic cell rate algorithm.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

ATM Adaptation (Transport) Layer: AAL

Basic idea: cell-based VCs need to be "complemented "to be supportive for applications.

- Several ATM Adaptation Layer (AALx) protocols defined, suitable for different classes of applications
  - AAL1: for CBR (Constant Bit Rate) services, e.g. circuit emulation
  - AAL2: for VBR (Variable Bit Rate) services, e.g., MPEG video
  - .....
- "suitability" has not been very successful
- computer science community introduced AAL5, (simple, elementary protocol), to make the whole ATM stack usable as switching technology for data communication under IP!



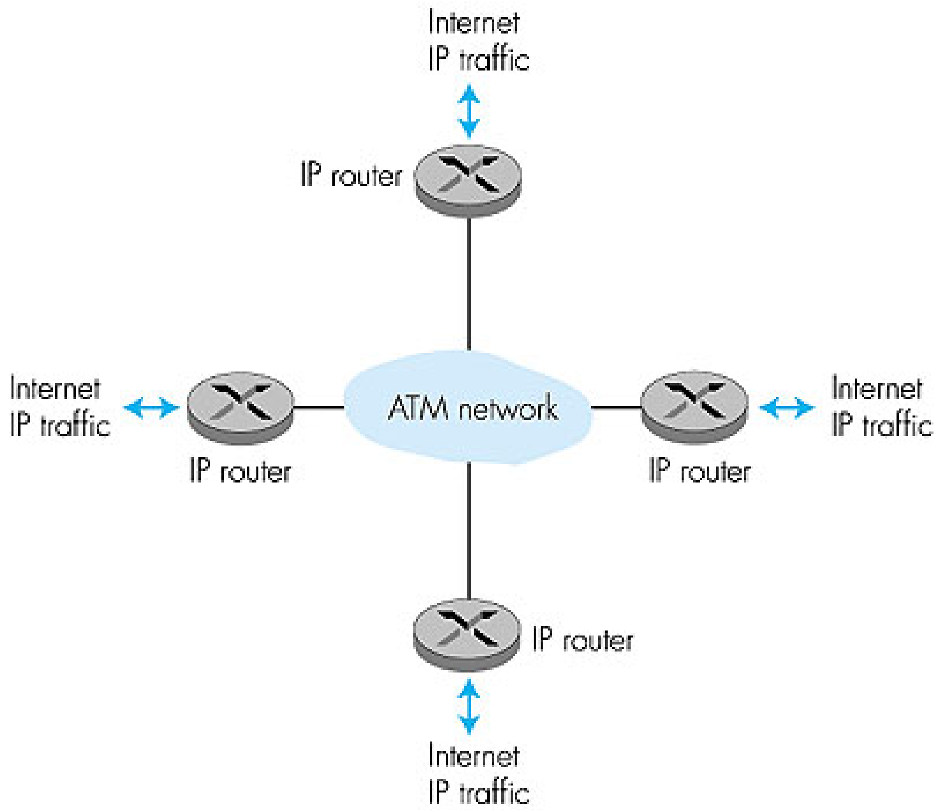
ATM: network or link layer?

Vision: end-to-end transport: "ATM from desktop to desktop"

- ATM *is* a network technology

Reality: used to connect IP backbone routers

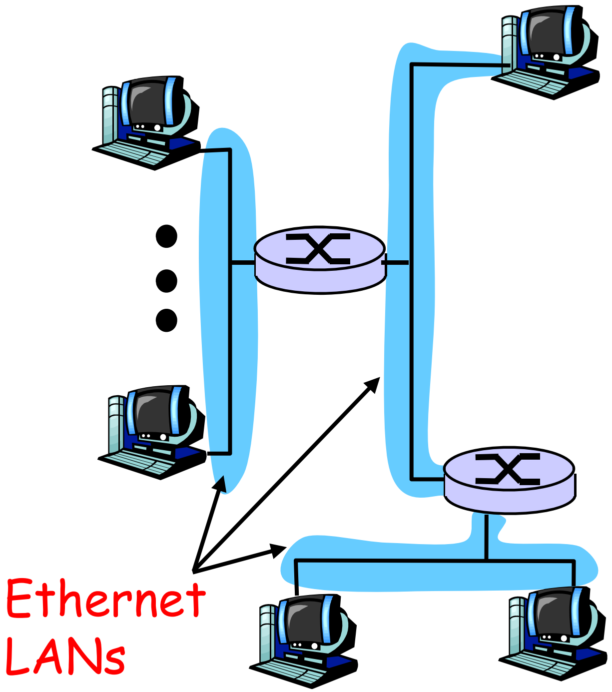
- "IP over ATM"
- ATM as switched link layer, connecting IP routers

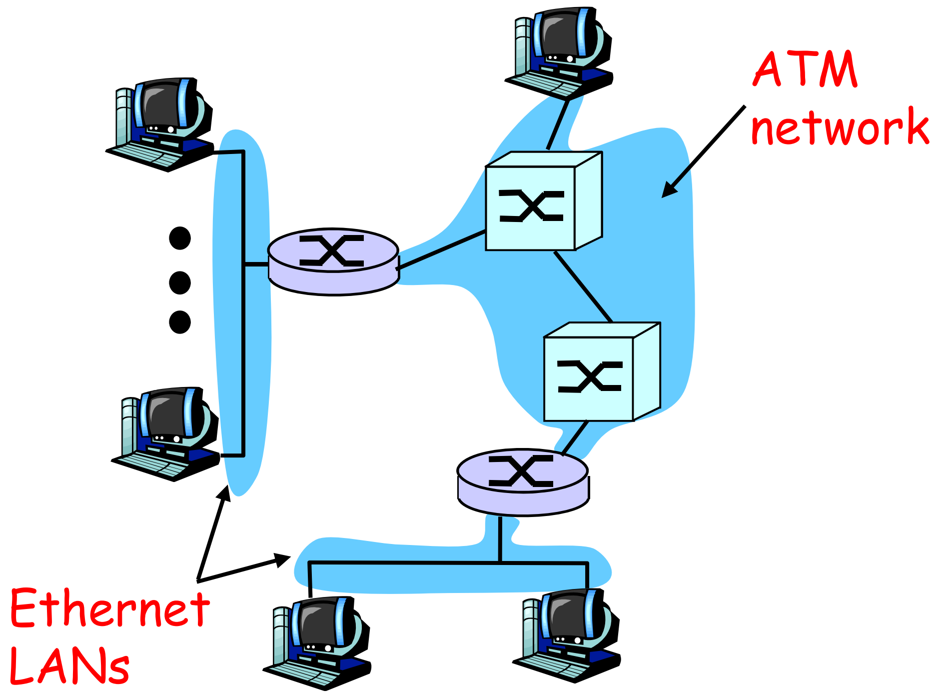


IP-Over-ATM

Classic IP only
- 3 "networks" (e.g., LAN segments)
- MAC (802.3) and IP addresses

IP over ATM
- replace "network" (e.g., LAN segment) with ATM network
- ATM addresses, IP addresses

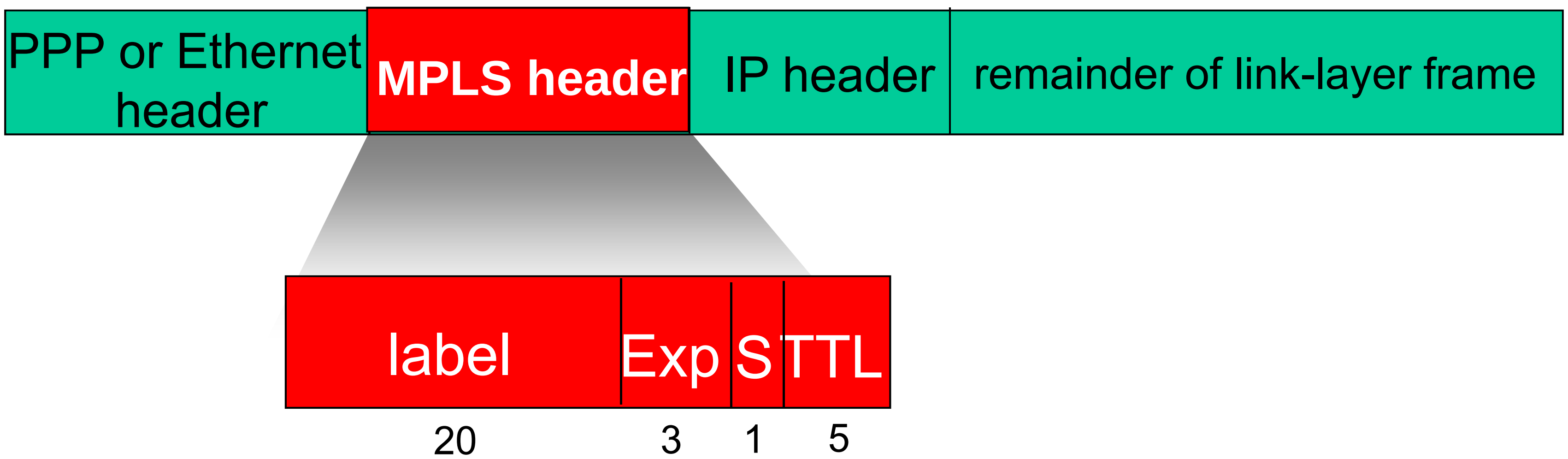




To be continued...

A parallel story: Evolution from ATM/VC related approach: Multiprotocol label switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



MPLS capable routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE (extension for "traffic-engineering", use MPLS)
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
- must co-exist with IP-only routers