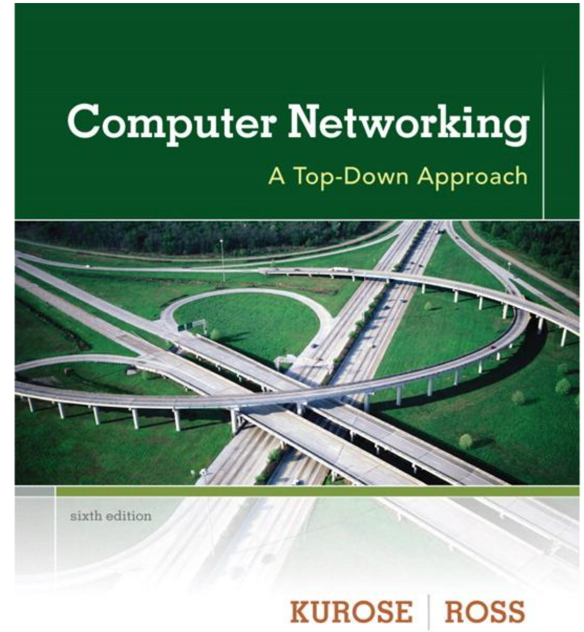
CONTENT: *Part A* "**Wireless and Mobile Networks**" & *Part B* "**Media and real-time Communications; Service-integrated Networks**"

Ackowledgement to:
Jim Kurose & Keith Ross



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

The Future of Internet Traffic I.

[cf. **Study by Cisco** "Global IP traffic forecast and methodology; Approaching the Zettabyte*) Era"; White paper series (2009); → *www.cisco.com*]

- bis 2013 *verdoppelt sich IP-Verkehr* global *alle 2 Jahre* nahezu;
- derzeitig 2-stelliger %-*Anteil von Peer-to-Peer (P2P-)Verkehr schrumpft* im Verhältnis zum gesamten IP-Verkehr, wächst in seinem Volumen jedoch weiterhin;
- *Videodatenströme* (resultierend aus unterschiedlichen Anwendungen wie IPTV, Video-on-Demand/VoD, etc) werden voraussichtlich bis 2013 [in Prognose aus 2011, s.u., ist indes hier vom Jahr 2015 die Rede !] den IP-Verkehr mit einem *Anteil von mehr als 90 %* dominieren;
- *Sprachdatenströme* (insbes. resultierend aus VoIP-Anwendungen) werden voraussichtlich bis 2013 nur noch einen sehr geringen *Anteil < 1 %* an dem gesamten IP-Verkehr ausmachen – trotz weiterhin sehr häufiger Nutzung von Sprachdiensten im Internet.

*) Zettabyte (kurz: ZB) = 10^{21} Byte = 1,000,000,000,000,000,000,000 Byte = 1000^7
[0.75 ZB als Prognose für 2014 für das jährliche globale Verkehrsaufkommen im Internet; ca. 2 Jahre benötigt, um die im Jahr 2014 pro sec übertragenen Videos anzuschauen bzw. 72 Mio. Jahre für die jährl. übertragenen Videos, *vgl.* http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html]

Nota bene: Intensive Forschungsaktivitäten zu IP-Verkehr bei TKRN !

Remark: *Some of the motivation given partially in German language → please excuse !!!*

The Future of Internet Traffic II.

[cf. **Study by Cisco** "Global IP traffic forecast and methodology; Entering the Zettabyte Era"; White paper series (June 1, 2011); → *www.cisco.com*]

June 1, 2011
This forecast is part of the Cisco® Visual Networking Index (VNI), an ongoing initiative to track and forecast the impact of visual networking applications. This document presents the details of the Cisco VNI global IP traffic forecast and the methodology behind it.

Video Highlights

- **Global Internet video traffic surpassed global peer-to-peer (P2P) traffic in 2010, and by 2012 Internet video will account for over 50 percent of consumer Internet traffic**. As anticipated, as of 2010 P2P traffic is no longer the largest Internet traffic type, for the first time in 10 years. Internet video was 40 percent of consumer Internet traffic in 2010 and will reach 50 percent by year-end 2012.

- **It would take over 5 years to watch the amount of video that will cross global IP networks every second in 2015**. Every second, 1 million minutes of video content will cross the network in 2015.

- **Internet video is now 40 percent of consumer Internet traffic, and will reach 62 percent by the end of 2015**, not including the amount of video exchanged through P2P file sharing. The sum of all forms of video (TV, video on demand [VoD], Internet, and P2P) will continue to be approximately 90 percent of global consumer traffic by 2015.

- **Internet video to TV tripled in 2010**. Internet video to TV will continue to grow at a rapid pace, increasing 17-fold by 2015. Internet video to TV will be over 16 percent of consumer Internet video traffic in 2015, up from 7 percent in 2010.

- **Video-on-demand traffic will triple by 2015**. The amount of VoD traffic in 2015 will be equivalent to 3 billion DVDs per month.

- **High-definition video-on-demand will surpass standard definition by the end of 2011**. By 2015, high-definition Internet video will comprise 77 percent of VoD.

Online- und Web-Video kreativ und professionell.

Webvideos und Onlinevideos übernehmen eine wichtige Rolle in der heutigen Kommunikations-Landschaft. *Über die Hälfte der Internet-Nutzer konsumiert regelmäßig im Web Videos* auf Videoplattformen wie Vimeo, YouTube & Co.
Diese *Videoplattformen sind neben Google die größten Suchmaschinen* und werden nicht ohne Grund regelmäßig zum Auffinden und zur Aufnahme von Informationen durch Videos genutzt.

Das Bewegtbild im Netz ist zum Massenmedium geworden.

Wenn Webvideo zum Erreichen von (Kommunikations-)Zielsetzungen genutzt werden soll, sind ansprechendes, ästhetisches Darstellungsmaterial, emotionale Bilder und professionelle Umsetzungen ein wichtiger Bestandteil der Video-Konzeption und -Produktion. Webvideos werden im deutschsprachigen Raum allerdings derzeit erst von etwa 10% aller Unternehmens-Websites genutzt. Das ermöglicht noch immer eine Abhebung vom Wettbewerb durch zeitgemäße (Video-) Kommunikation mit online bereitgestellten Inhalten. Die Einbindung von Webvideos und Filmen unterstützt moderne, zeitgemäße Websites. Neben Präsentationen sind es vor allem Imagefilme oder Firmenvideos, die im Internet die Glaubwürdigkeit des Unternehmens durch Nachvollziehbarkeit steigern.

Im Gegensatz zu längeren Formaten bei anderen Massenmedien (wie TV oder Video-DVD/BD) haben Webvideos im Normalfall kürzere Laufzeiten, die max. 10 bis 15 Minuten nicht überschreiten. Das Hochladen von längeren Videos ist allerdings schon möglich und lässt sich bei YouTube freischalten.
Wann Fernsehsender ins Internet gehen ist nur eine Frage der Zeit. ProsiebenSat1 (SevenOne Media GmbH) beispielsweise hat mit „myvideo“ bereits eine eigene mehrsprachige Plattform geschaffen.

...

Part B "Media and Real-time Communications; Service-integrated Networks"

References

- I. Akyildiz, et al.: Wireless Sensor Networks – A Survey, Computer Networks, Vol. 38, 2002
- F. Halsall: Multimedia Communications, Addison-Wesley 2001
- P.A. Henning: Taschenbuch Multimedia (4. Aufl.), Hanser-Verlag 2007
- **J.F. Kurose, K.W. Ross: Computer Networking – A Top-Down Approach (4th/5th ed.) *bzw.* Computernetzwerke. Der Top-Down-Ansatz (5. Aufl.), Pearson Education 2008/2010/2012**
- J. Rech: Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail (4. Aufl.), Heise-Verlag 2012
- J. Schiller: Mobilkommunikation (2. Aufl.), Addison-Wesley / Pearson 2003
- R.E. Sheriff, Y.F. Hu, G. Losquadro et al. (eds.): Space / Terrestrial Mobile Networks – Internet Access and QoS Support; J. Wiley 2004
- W. Stallings: Wireless Communications and Networks (2nd ed.), Prentice-Hall 2004
- R. Steinmetz, K. Nahrstedt: Multimedia Systems, Springer 2004
- B. Walke: Mobilfunknetze und ihre Protokolle 1 – Grundlagen, GSM, UMTS und andere zellulare Mobilfunknetze, 3. Aufl., Prentice-Hall 2002
- B.E. Wolfinger, K. David, R. Kraemer (eds.): Proc. zu Workshop „Mobile Computing und Medienkommunikation im Internet", Ulm, Sept. 2004

Part B "**Media and Real-time Communications; Service-integrated Networks**"

Part B “**Media and Real-time Communications; Service-integrated Networks**” *(continued)*

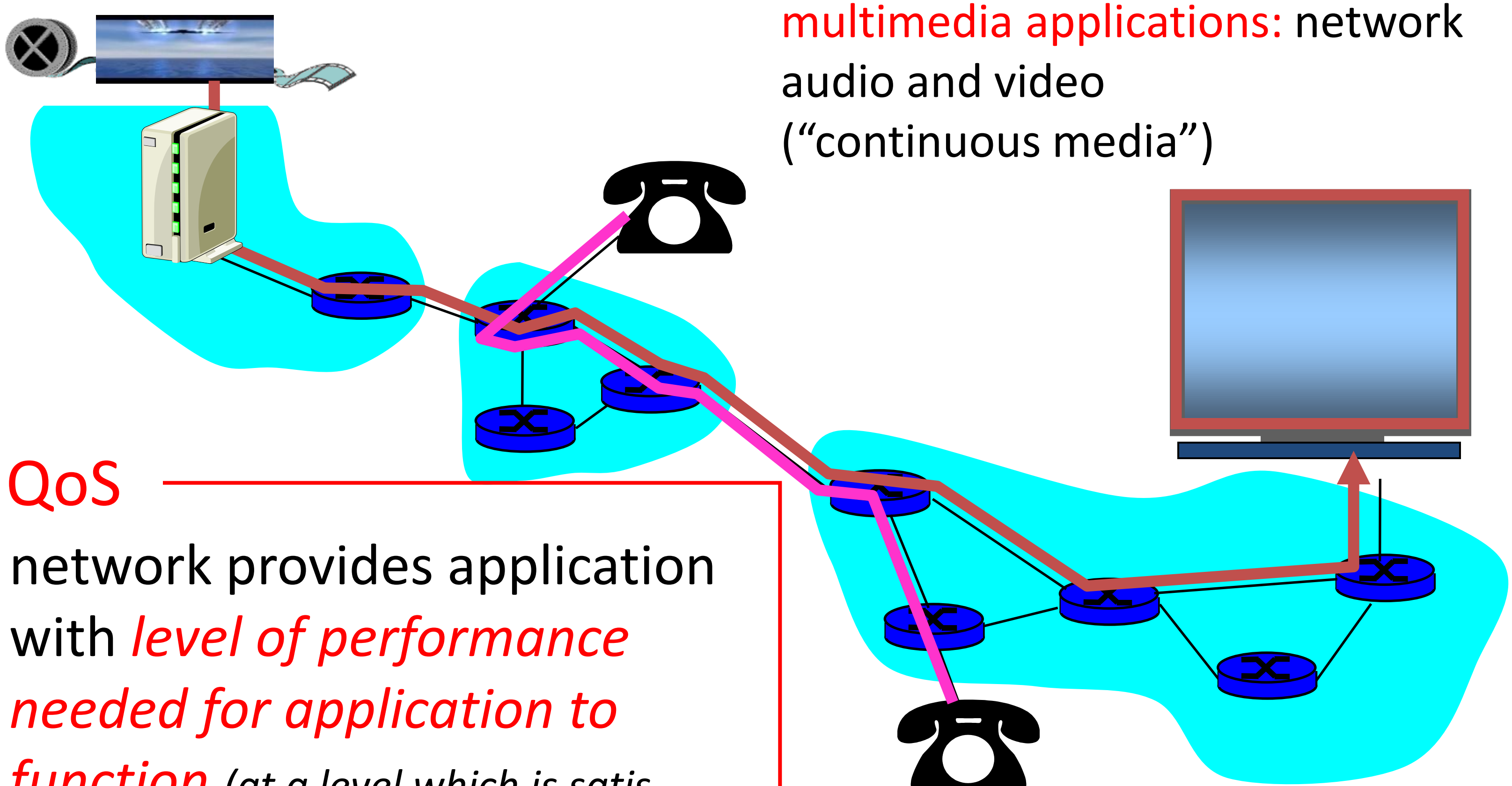
Part B "**Media and Real-time Communications; Service-integrated Networks**"

Part **B**: Media and Real-time Communications; Service-integrated Networks

B1. Multimedia Applications and Resulting Traffic Classes

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* *(at a level which is satis-factory to the user of the application).*

Part B: 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
- service-integration and networks providing QoS guarantees

B1.1 Examples of Multimedia Applications

MM Networking Applications

Classes of MM applications:

1) stored streaming

2) live streaming

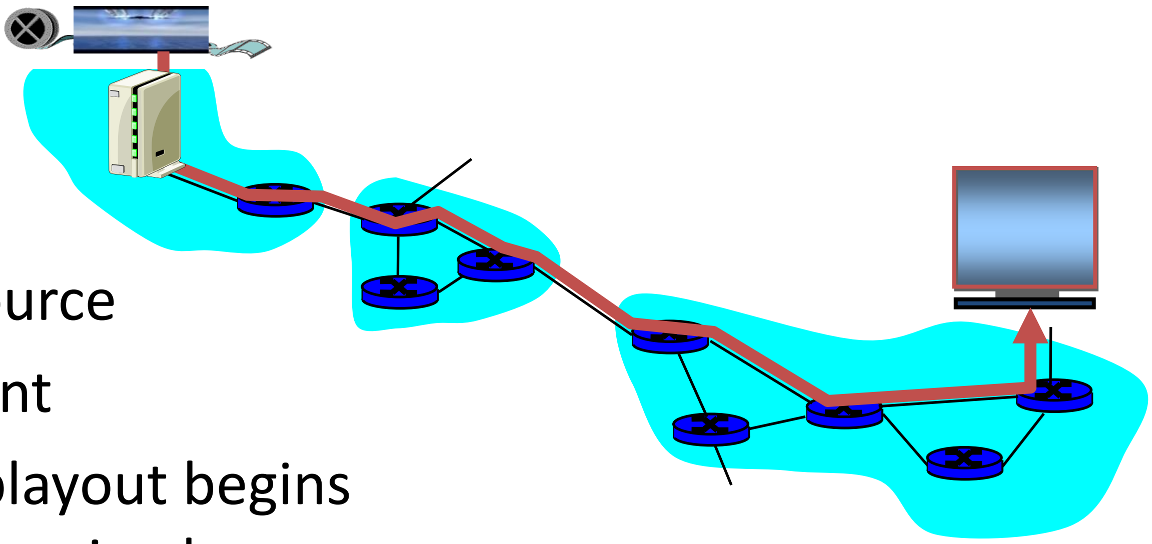
3) interactive, real-time

Fundamental characteristics:

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

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

Streaming Stored Multimedia



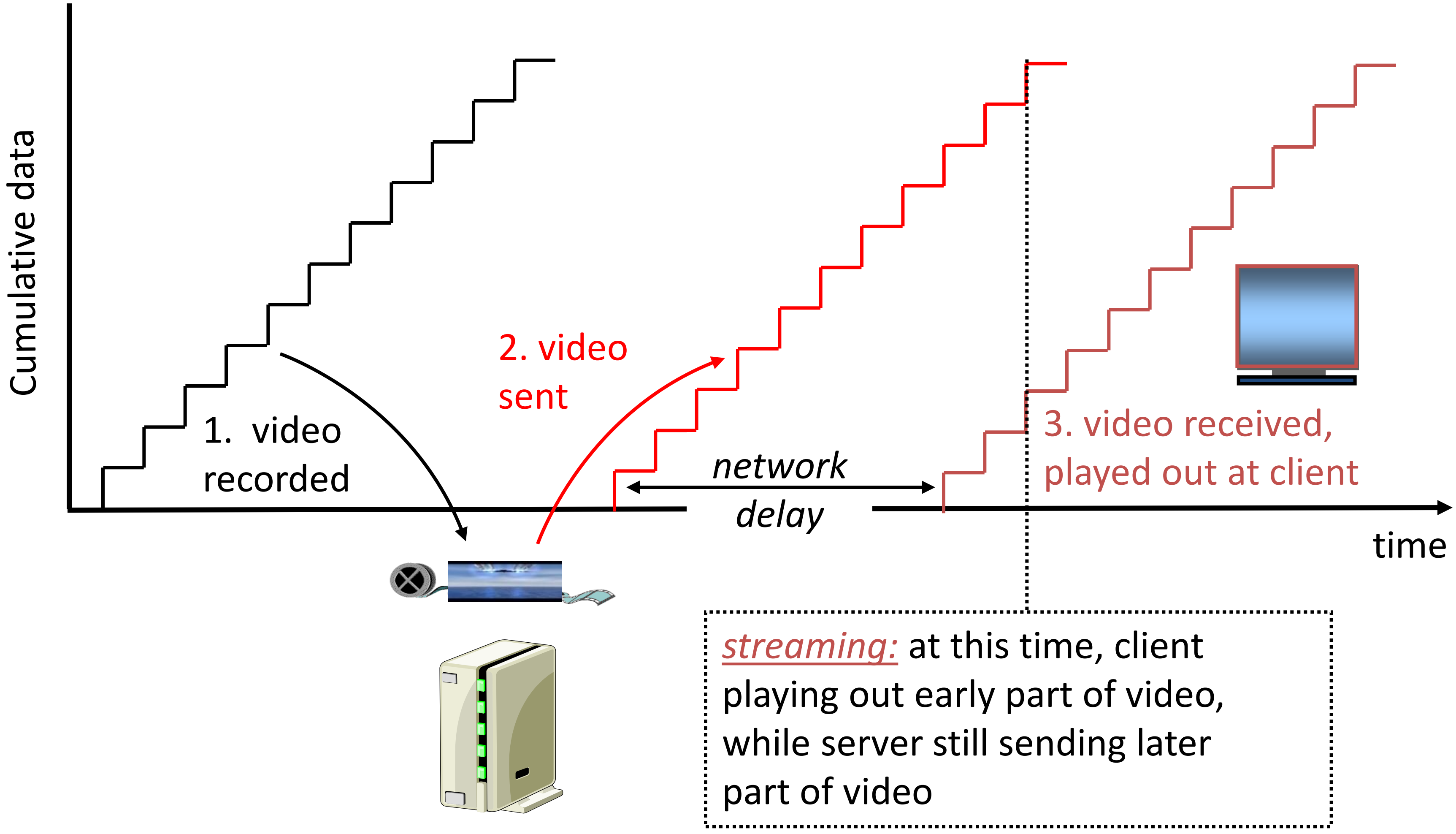
Stored streaming:

- 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

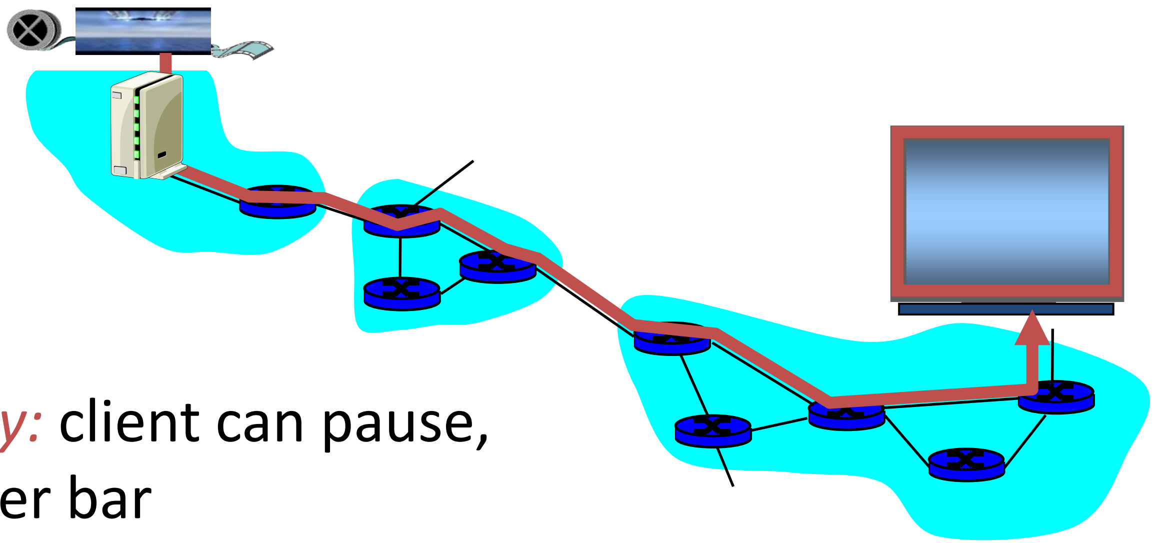
For one of the first proposals for audio/video streaming ,cf.:

B.E. Wolfinger, M. Moran: A Continuous Media Data Transport Service and Protocol for Real-Time Communication in High Speed Networks. Proc. 2nd Int. Workshop on Network and Operating System Support for Digital Audio and Video (Heidelberg, 1991); [ranked number 179 in the *Most Cited Computer Science Articles of the* year *1991*, see *CiteSeerX database as of Febr. 26, 2008*]

Streaming Stored Multimedia: What is it?



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

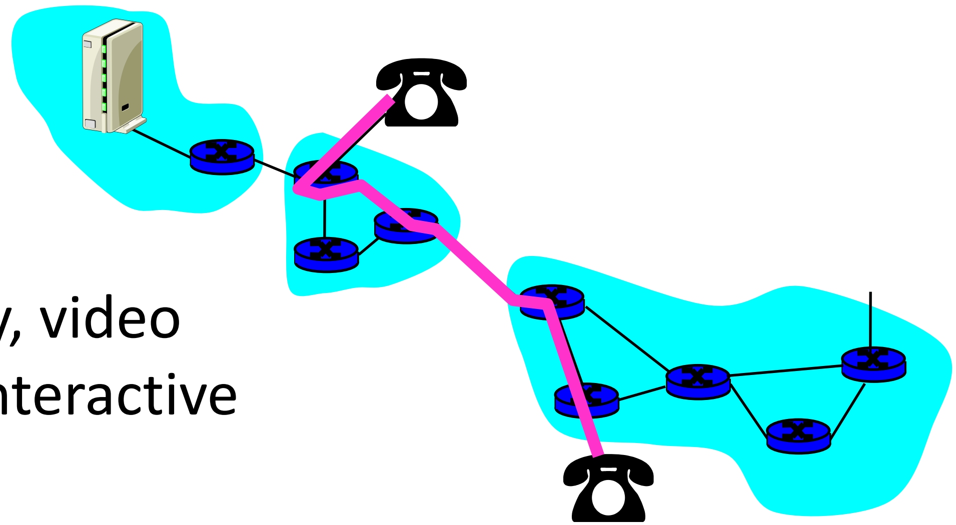
Streaming *Live* Multimedia

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

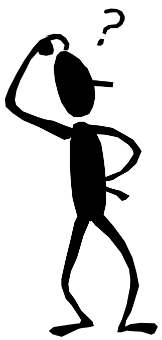


- ❒ applications: IP telephony, video conference, distributed interactive worlds
- end-end delay requirements:
  - audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays
    - higher delays noticeable, impair interactivity
- session initialization
  - how does callee advertise its IP address, port number, encoding algorithms?

B1.2 Hurdles for Multimedia in Today's Internet

TCP/UDP/IP: "best-effort service"

- *no* guarantees on delay, loss



But you said multimedia apps require 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

B1.3 How Should the Internet Evolve to Support Multimedia Better ?

<u>Integrated services philosophy:</u>

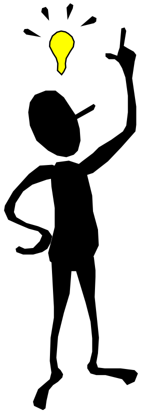
- fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- requires new, complex software in hosts & routers

<u>Laissez-faire</u>

- no major changes
- more bandwidth when needed (e.g., faster links, but also faster routers)
- content distribution, application-layer multicast
  - application layer

<u>Differentiated services philosophy:</u>

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



What's your opinion?

B1.4 Audio and Video Compression

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 (i.e. 8 bit/sample) $\rightarrow$ 64 kbps
- 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

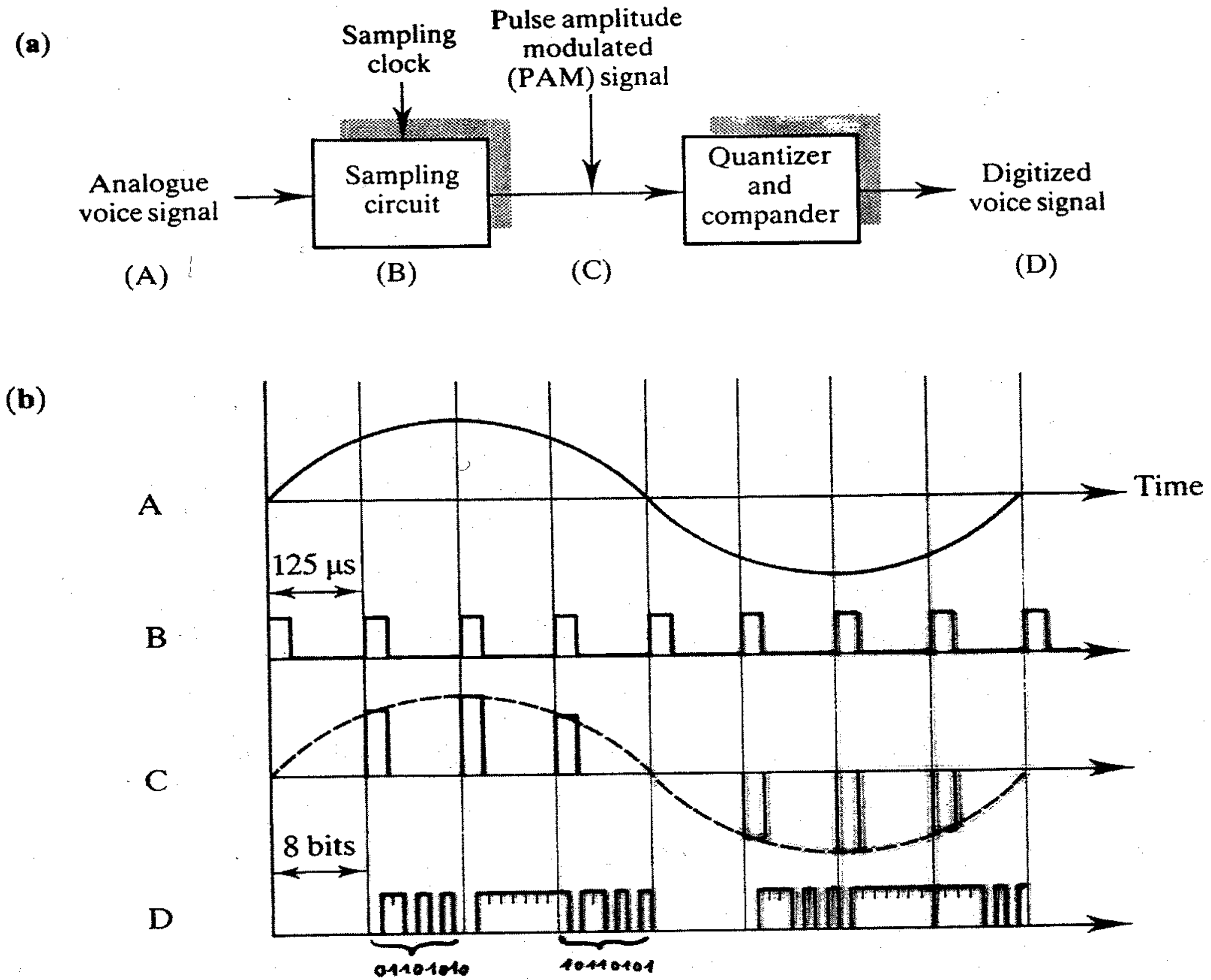
Voice and Audio Compression

Requirements in voice and audio transmissions

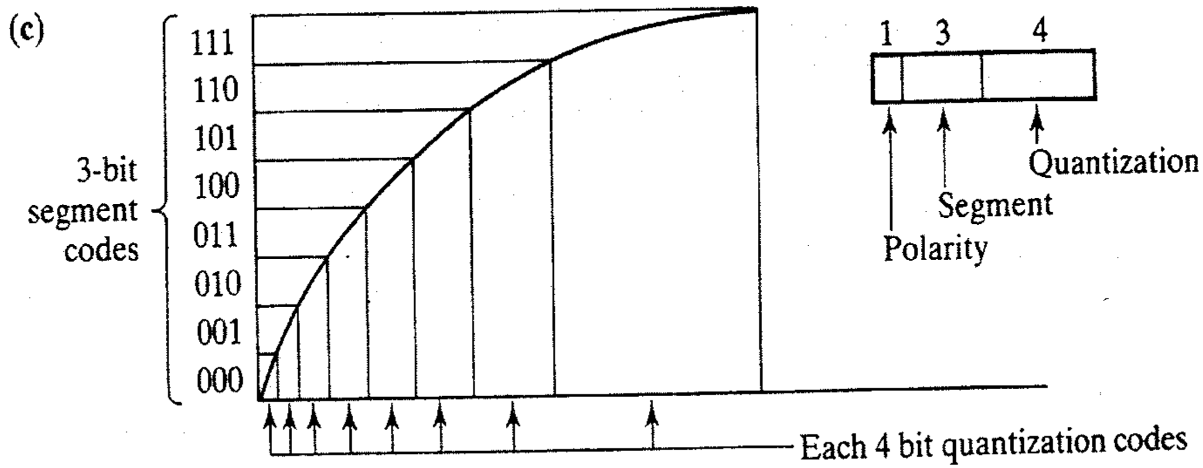
Criteria	Voice	Audio
Quality requirement	Telephone quality	CD quality
Highest frequency considered	about 4.000 Hz	about 20.000 Hz
Primary computer /network resources required	Transmission resources	(Peripheral) Memory resources
Typical signal at the source (i.e. primary signal)	Human voice	Music (mostly recordings in a high quality)
Reproduction /transmission quality typically achieved	Medium (only in wired networks); often weak in mobile/wireless networks	Generally very high

PCM for digital voice transmission

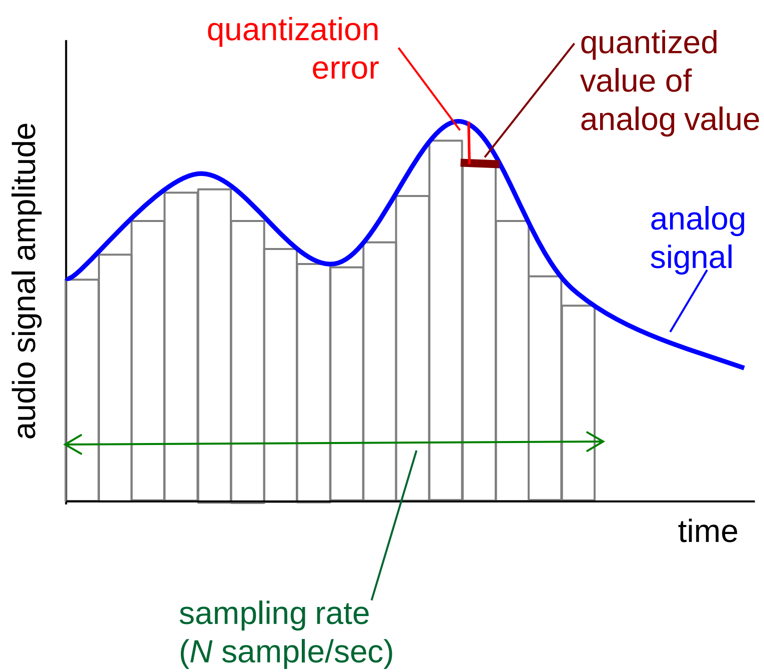
(Example for Analog-Digital-Conversion)



PCM: continuation (encoding)



Summary: PCM as an analog-digital conversion (Analog-Digital-Wandlung)



Resulting data rate in digital voice transmissions

- **Sampling frequency** in [Hz], i.e. number of sampling events per sec: 8000, leading to 8 kHz, because of (Nyquist-)Shannon's sampling theorem :
  → 2 • maximum frequency (i.e. 2 • 4.000 Hz), *"voice channel", e.g.: [300, 3400] Hz*
- **Number of bits used per sample** (to represent the sample value) :
  7 Bit (USA) resp. 8 Bit (Europe)

Therefore:
Resulting data rate:

- USA: 7 Bit • 8.000 Hz = 56.000 [Bit • Hz] = 56.000 [Bit / sec] = 56.000 [bps] = **56 [kbps]**
- Europe: 8 Bit • 8.000 Hz = 64.000 [Bit • Hz] = 64.000 [Bit / sec] = 64.000 [bps] = **64 [kbps]** , cf. data rate of a so-called „B-Channel" in (narrowband-) ISDN

As the data rate does not vary during the „talkspurt periods" (active speaker) of a voice transmission, the resulting traffic is called „***Constant Bit Rate***"(CBR)-traffic.

BUT one should be very careful (!) in using the notion of CBR, because CBR-traffic of, e.g., 10 [kBit/sec] could indeed correspond to very different types of traffic :

- Packets of a size of 100 [Bit], produced after a constant period of 10 [msec],
  or
- Packets of a size of 1 [kBit], produced after a constant period of 100 [msec], and many more !!!

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
- MPEG-2 (DVD) 3-6 Mbps
- MPEG-4 (often used in Internet, < 1 Mbps)

Research:

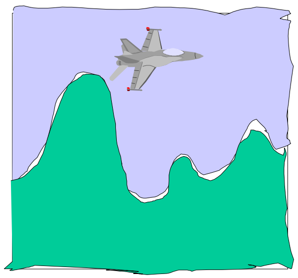
- layered (scalable) video
  - adapt layers to available bandwidth

Details of video compression algorithms

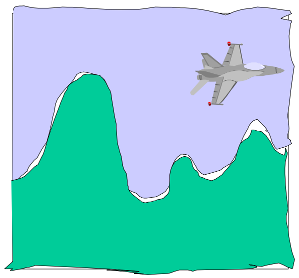
Spacial and temporal redundancy (in a video stream)

- **Temporal redundancy**

  Example: Presenter of news in front of a constant background; slow movement of the camera ($\rightarrow$ motion vector relevant !)



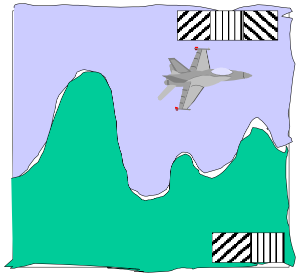
Picture at $t = t_0$



Picture at $t = t_0 + \Delta t$

- **Spacial redundancy**

  Example: “homogeneous“ (similar) areas may be found in a picture.



Picture at $t = t_0$

reduction of redundancy $\rightarrow$ losses becoming more „dangerous“ (error propagation)

- Examples of standards for video compression: **MJPEG**, **MPEG-x**, **H.26x**, ...

The JPEG Standard

- **JPEG** (= Joint Photographic Experts Group)

→ ISO/ITU-Standard [ISO-10918 as well as CCITT-Recommendation T.81]

- ***GOAL of JPEG*** :

  Standard for storage of single images in a compressed format.

- Typical factors of compression achieved :
  - 100:1 with significant loss, and
  - 20:1 without significant loss.

- Fundamental algorithm used as a basis for JPEG :

  DCT (= *Discrete Cosine Transformation*)

  DCT is transforming lightness (*Helligkeitswerte*) into the frequency domain, cf. Fourier Transformation mapping from time into frequency domain.

Newer JPEG-Standard :

JPEG 2000 (since Jan. 2001) [specified in : ISO/IEC 15444-1 & 2 as well as in ITU-T Recomm. T.800 & T.801]

Essential Steps of JPEG Algorithm (only a coarse description)

- **STEP 1 :** (Optional) *Change of the color model* (‘*Farbmodells*‘) from (mostly) RGB-color space (‘Farbraum‘) [Read-Green-Blue] to YCbCr-color model.

- **STEP 2 :** *low pass filtering and subsampling* of the signals of color deviation (‘Farbabweichungssignale‘) Cb and Cr (with loss)

  $\rightarrow$ consequence: decrease of resolution of color channels.

- **STEP 3 :** For each subpicture :

  Partitioning into macro-blocks with 8x8 pixels each.

- **STEP 4 :** For each macro-block :
  DCT of the‘Helligkeitswert‘ f(i,j) in the frequency domain, in particular :

$$F(u,v) = \tfrac{1}{4} \sum_{i,j=0,\ldots,7} f(i,j) \cdot h_u \cdot h_v \cdot \cos((2i+1) \cdot u \cdot \pi/16) \cdot \cos((2j+1) \cdot v \cdot \pi/16)$$

$h_u = 1/\sqrt{2}$ if $u=0$, $h_u = 1$ else.

Essential steps of JPEG Algorithm

- Details w.r.t. **Step 1 :** (Optional) *Change of the color model (‚Farbmodell')* from (typically) RGB-color space [Red-Green-Blue] into ***YCbCr-color model***.

Original image

Y-component (brightness / Helligkeit, luma / Luminanz)

Cb-component (colorfulness / Farbigkeit, chroma / Chrominanz): Deviation „grey“ in direction of „blue“ / „yellow“

Cr-component (colorfulness, chroma): Deviation „grey“ in direction of „red“ / „turquoise“

Essential steps of JPEG Algorithm *(continuation)*

- **STEP 5 :** *Quantization*, i.e. division of F(u,v) by a number q(u,v) and rounding of the results.

  Thus :

  $$F'(u,v) = \text{int}\,(\,F(u,v) \,/\, q(u,v)\,)$$

  $\rightarrow$ possibly this step may be rather lossy *(‘verlustbehaftet‘)*.

- **STEP 6 :** *Re-sorting of the values F‘(u,v)*

  $\rightarrow$ Transmission according to a so-called zig-zag-serialization (*‘Zick-Zack-Serialisierung‘)*.

- **STEP 7 :** *Run-length-encoding* and subsequently *Entropy-encoding* (e.g. Huffman- or arithmetic encoding)

nota bene :

Details cf., e.g., P.A. Henning „Taschenbuch Multimedia“ (Fachbuchverlag Leipzig, 2. Auflage, 2001).

The Family of MPEG-Algorithms

- **MPEG** (= Moving Pictures Experts Group)

  → Series of standards

- ***1993***: **MPEG-1** → - Usage, e.g., for video CDs;
  - MPEG-1 Layer 3 (MP3 for short) is defined in the audio section of MPEG-1.
- ***1994/95***: **MPEG-2** → - Video- and sound formats in TV quality;
  - Usage, e.g., for video DVDs.
- *only planned, but never published*: **MPEG-3**
- ***1998-2001***: **MPEG-4** → - E.g., specification of a container format (similar to QuickTime);
  - 3D-language similar to VRML and, in particular, improved video compression (as compared to MPEG predecessor standards).
- ***2002***: **MPEG-7** → - System to describe multimedia contents.

Some basic principles underlying the MPEG standards

- **Principle :** ***Partitioning of individual (video-) frames into smaller data units,*** such as :
  - ***Slices*** (“**Bildscheiben**“)
  - ***Macro Blocks*** (“**Makroblöcke**“)
  - ***Blocks*** (“**Blöcke**“).

→ ADVANTAGE :

Smaller data units, e.g. macro blocks, can repeatedly appear (in an approximately identical manner) within the same frame or in one or even more subsequent frames.

Therefore: Minor changes will require only little information (during encoding), because just the macro block referenced and the areas having changed are required by the receiver, in order to completely calculate the new macro block.

Basic principles underlying MPEG (continued)

- **Principle : *Different degrees of dependencies between subsequent frames within a video stream***

  → e.g. the following 3 types of frames exist :

  - **I-Frames** (“***I**ntraframes*“) : encoded completely independently of other frames.
  - **P-Frames** (“***P**redicted Frames*“) : produced by means of motion prediction and encoding of differences w.r.t. preceding frames.
  - **B-Frames** (“***B**idirectionally Predicted Frames*“) : produced by means of motion prediction and encoding of differences w.r.t. the preceding and the subsequent I- or P-Frame.

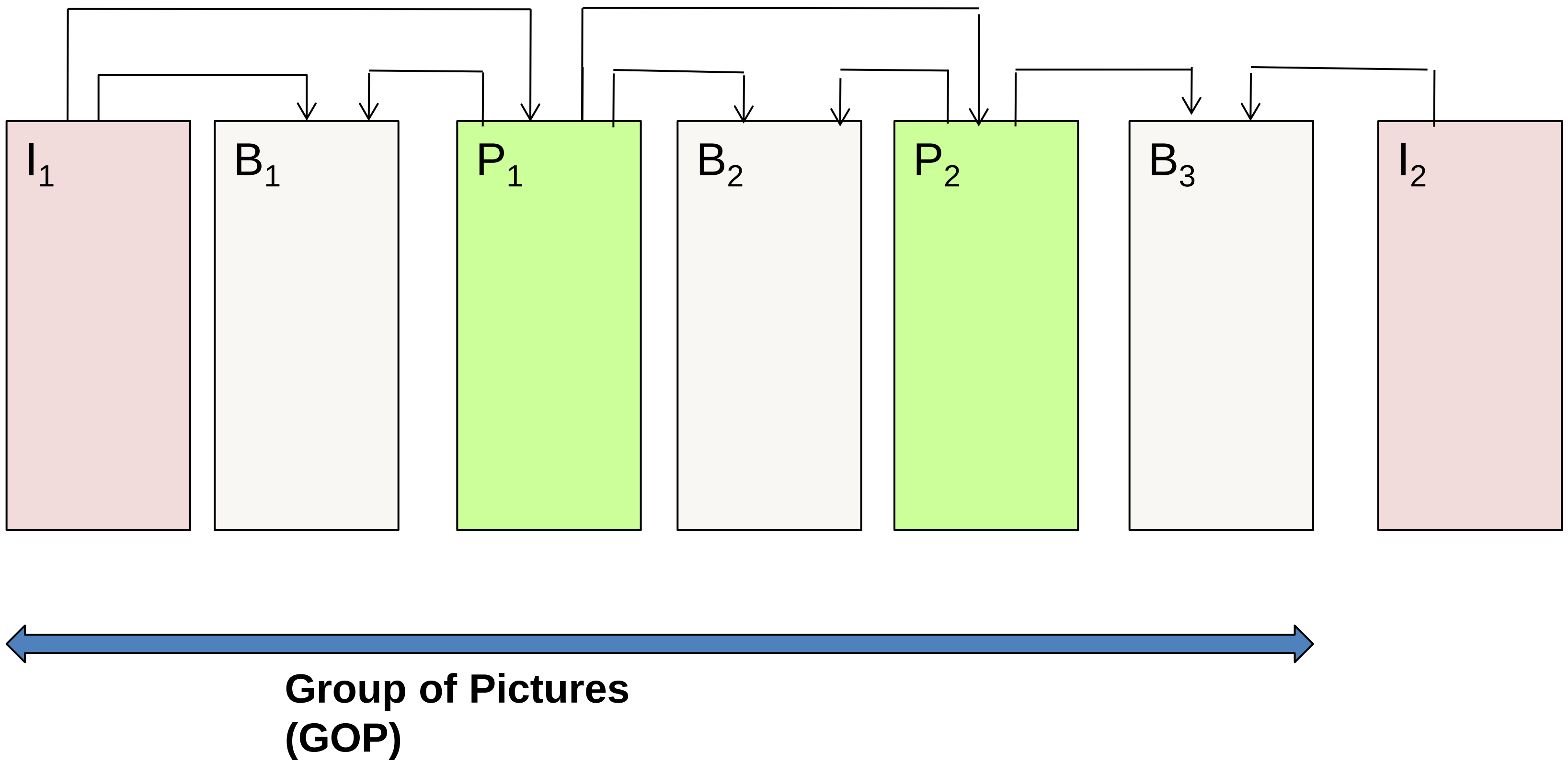
To be noted:

– I-Frames typically largest and B-Frames smallest frames.

– Strong error propagation if I-Frames are damaged or lost, less negative impact of erroneous/lost P-Frames, and B-Frame only renders impossible decoding of just this B-Frame.

– Predictive and interpolating encoding is also applicable for smaller units, such as macro blocks (also used in some MPEG standards).

Mutual dependencies and error propagation in case of MPEG



Meaningful sequence of transmission:

– I_1, P_1, B_1, P_2, B_2, I_2, B_3 .

Multimedia: video (Summary I)

- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- coding: use redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)

frame *i*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

frame *i+1*

Multimedia: video
(Summary II)

- CBR: (constant bit rate): video encoding rate fixed
- VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
- examples:
  - MPEG 1 (CD-ROM) 1.5 Mbps
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (often used in Internet, < 1 Mbps)

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)

frame *i*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

frame *i+1*

Multimedia networking: 3 application types

- *streaming, stored* audio, video
  - *streaming:* can begin playout before downloading entire file
  - *stored (at server):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *conversational* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype
- *streaming live* audio, video
  - e.g., live sporting event (football), live IPTV

Part B “**Media and Real-time Communications; Service-integrated Networks**”

B2. Quality of Service (QoS): Measures and Assessment Methods

B2.1 Typical QoS Requirements of Multimedia Applications

Which **general requirements** important for multimedia applications /end-users ?

- Transmission of different types of traffic with their specific QoS requirements, being dependent on
  - ***Class of traffic*** → Example : voice vs. video *and*
  - ***Type of application*** → Example : video games vs. medical applications
- Synchronisation of streams (e.g. voice and video in video-telephony)
- Tapping security *(‘Abhörsicherheit‘)*
- Multicast functionality
- Low („reasonable“) costs, etc., etc.

- Which **QoS requirements** are important for end-users (dependent on classes of traffic) ?
  - **Data** communication
    - (End-to-end) Delay, in general ≤ 1-10 sec (approximately)
    - Throughput, in general strongly variable
    - Reliability, in general very high
  - **Voice** communication
    - (End-to-end) Delay, in general ≤ 100-500 ms (approx.)
    - Throughput, in general relatively low (e.g. 64 kb/s)
    - Reliability, in general quite low (e.g. packet loss probability in digital voice transmission acceptable if < 5 %)
  - **Audio/Video** communication
    - (End-to-end) Delay, about 10-100 msec, … , 1 sec in VoD services
    - Delay jitter, in general to be kept quite low (rather strict limit)
    - Throughput, in general relatively high (e.g. 1-100 Mb/s), possibly compression for a throughput reduction
    - Reliability, in general relatively low (dependent on compression factor)
  - **Fax** communication (or transmission of **single images**)
    - Throughput, in general relatively high (but only for short period)

- **Problems for the service provider resulting from end-user oriented requirements**
  - Requirements typically
    - only of *qualitative nature*
    - (possibly very) *subjective*
    - *difficult to specify* them precisely (in general: quite "fuzzy")
  - From service providers view: *Requirements may be contradictory*
    - high throughput and, at the same time, low end-to-end delay
    - high quality and, at the same time, low costs (*therefore :* let users pay for high quality services to control their behaviour !)
  - *Transformation of qualitative requirements onto quantifiable* ones
  - *Transformation of end-user oriented requirements into those ones* which are *meaningful for* typical interfaces in communication *networks* (cf. so-called "QoS-Mapping")

→ **Example** :

What is the implication (for the communication network) of requiring a high-quality transmission of medical video sequences or requiring lip synchronization in audio/video communications (e.g. in video-telephony) ?

B2.2 QoS Measures

- **Assessment/ judgement of QoS**
  - at a concrete interface between service provider and user
  - from point of view of service user (in comparison to the QoS requirements as stated by the user)
  - possibly by means of measurements
  - necessarily considering the load as offered by the user resp. the sequence of requests handed over to be served (e.g. the sequence of transmission requests)

- Typical requirements towards communication / computer networks :

  "Garantees" regarding
  - *Delay* $\rightarrow$ not measurable at the sender
  - *Delay Jitter* $\rightarrow$ precise definition ?!
  - *Throughput* , NOT : *"bandwidth"* !
    $\rightarrow$ Transmission requests successfully served per sec, e.g. packets/s, bit/s, ...
  - *Loss rate* , e.g. for packets, video frames, ...
    $\rightarrow$ problem in defining throughput and loss rates :
    observation interval chosen indispensable for the definition !

QoS-Guarantees

- **What is ″guaranteed″ ?**
  - Maximum
  - Minimum
  - Mean / average
  - Variance

  . . .

- **What does ″garanteed″ mean ?**
  - (nearly) with probability 1
  - with very high probability
    (possibly determined by means of the relative frequency, e.g. in 95% of all cases, ***BUT*** again : in which observation interval ?)

- *Example* for an important class of communication networks with QoS-guarantees : cf. ATM architecture, -services and -protocols

QoS Parameters I
- Delay Jitter -

Numerous **problems**:

- delay measured at what interfaces (at sending *and* receiving site)?
- variety of different, mutual contradictory definitions existing
- synchronized clocks at sender and receiver may be necessary.

Examples of possible **definitions for "Delay Jitter "** :

Let $d_1, d_2, d_3, \ldots, d_n$ be the (measured) delays of a sequence of transmitted packets $P_1, P_2, \ldots, P_n$.

- **Def. 1:** The **delay jitter** *d_j* is defined as

$$d\_j := \frac{1}{n}\sum_{i=1}^{n} | d_i - \overline{d} | \text{, where } \overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \quad (i.e.\ \overline{d} = avg.\ delay)$$

- **Def. 2:** The **delay jitter** *d_j* is defined as

$$d\_j := \max_i(d_i) - \min_i(d_i)$$

- **Def. 3:** The **delay jitter** *d_j* is defined as

$$d\_j := \frac{1}{n}\sqrt{\sum_{i=1}^{n} (d_i - \overline{d})^2}, \text{ where } \overline{d} = \text{avg. delay, cf. Def.1}$$

QoS Parameters II
- Packet Loss Probability -

Numerous **Problems**:

- only frequencies but not probabilities can be measured
- in which observation interval to observe ? (not too small and not too large !)
- (when) can a receiver decide that a packet is definitively lost ?
  ... sequence n$^{\underline{0}}$, max. time to live
- burstiness of losses should be taken into a account ($\rightarrow$ negative impact on QoS)

How to test whether a given requirement w.r.t. packet loss probability has been satisfied by a service provider ?

1. Choose an adequate time (resp. observation) interval T <u>OR</u> an adequate number N of successive packet transmissions

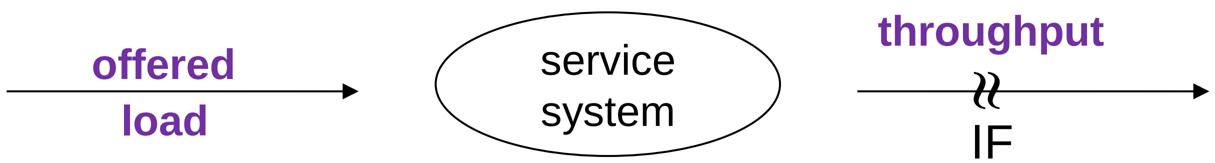
2. Test: *p_loss* (T) < *p_l** ?
   OR *p_loss* (N) < *p_l** ?

   where *p_l** = upper bound required for still acceptable packet loss probability
   p_loss (T) resp. *p_loss* (N) = packet loss frequency observed during T resp. for the last N packet transmissions

QoS Parameters III
- Minimum Throughput -

Numerous **problems**:

- throughput measured at what interface (IF) ? including/excluding control information ?
- in which observation interval to observe? (not too small and not too large ?)
- throughput always less or equal than "offered load" in adequately chosen observation intervals
  $\rightarrow$ test of min. throughput requirement only meaningful, if a sufficient amount of load has been offered



- fluctuations in throughput provisioning should be taken into account ($\rightarrow$ negative impact on QoS)

Def.

Throughput *t_d* of data (units) passing an interface IF during a time interval T = [t_1, t_2], $t_2 > t_1$, is defined as

$$\boldsymbol{t\_d\ (T, IF)} = \frac{\text{quantity of data passing at IF during T (in Bits, Bytes, packets, frames, ...)}}{(t_2 - t_1)}$$

B2.3 Methods to Assess QoS in Networks and at User Level

Quality Measures at Network Level

- Sufficiently high ***throughput***
- Sufficiently low ***delay***
- Sufficiently low ***(delay) jitter***
- Sufficiently low ***(packet) loss probability***
- Sufficiently high ***availability of network services***
- Sufficiently high degree of ***security of network services***

Quality Measures at User Level

A. *Objective* **Measures**

PSNR (peak signal to noise ratio)

*GIVEN: Two m x n monochrome images **I** (original) and **K** („noisy“ approximation)*

Def.:

$$\boldsymbol{PSNR = 10 \bullet \log_{10} (MAX_I^{\,2} / MSE) = 20 \bullet \log_{10} (MAX_I / \sqrt{MSE})}$$

where:

$MSE = 1 / (m \bullet n) \sum_{i=0,\dots,m-1} \sum_{j=0,\dots,n-1} \| I(i,j) - K(i,j) \|^2$

and

MAX_I = maximum possible pixel value of the image;
e.g., when pixels are represented using 8 Bit per sample then $MAX_I = 255$ resp. (more generally) $MAX_I = 2^B-1$, when B Bit are used per sample

Typical values for PSNR:

- *with lossy image compression : PSNR $\in$ [30 dB, 50 dB]*
- *still acceptable quality in lossy wireless networks : PSNR $\in$ [20 dB, 25 dB]*

Quality Measures at User Level

B. Subjective Measures

→ ***Quality-of-Experience (QoE)*** testing

Basis: ***MOS (Mean Opinion Score)***, *cf.: http://www.itu.int/rec/T-REC-P.800-199608-I/en ;* ***5 levels of quality****, in particular* :

Excellent = 5; Good = 4; Fair = 3; Poor = 2; Bad = 1

sufficient nº of test persons consulted – result averaged – MOS used, e.g., in :

- ***PESQ (Perceptual Evaluation of Speech Quality)***

  Def.: cf. ITU-T *P.862 : Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,* [*http://www.itu.int/rec/T-REC-P.862/en*](http://www.itu.int/rec/T-REC-P.862/en)

- ***PEVQ (Perceptual Evaluation of Video Quality)***

  Def.: cf. ITU-T *J.247 : Objective perceptual multimedia video quality measurement in the presence of a full reference,* [*http://www.itu.int/rec/T-REC-J.247/en*](http://www.itu.int/rec/T-REC-J.247/en)

… and : ***PEAQ*** *(… for Audio Quality)*

Part B "**Media and Real-time Communications; Service-integrated Networks**"

B3. Streaming Stored Audio and Video

Streaming Stored Multimedia

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

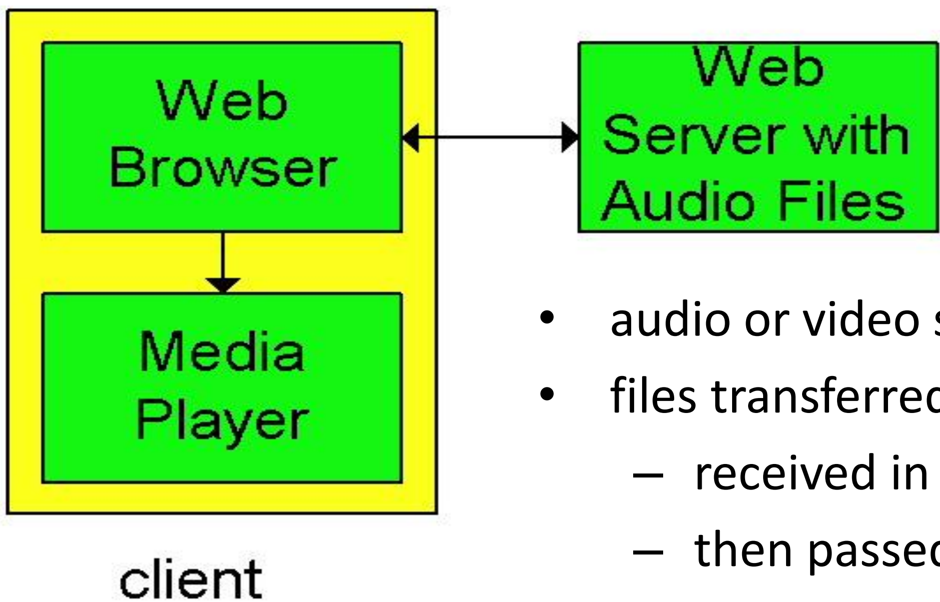
- client-side buffering
- use of UDP versus TCP
- multiple encodings of multimedia

Media Player

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

B3.1 Accessing Audio and Video Through a Web Server

Internet multimedia: simplest approach



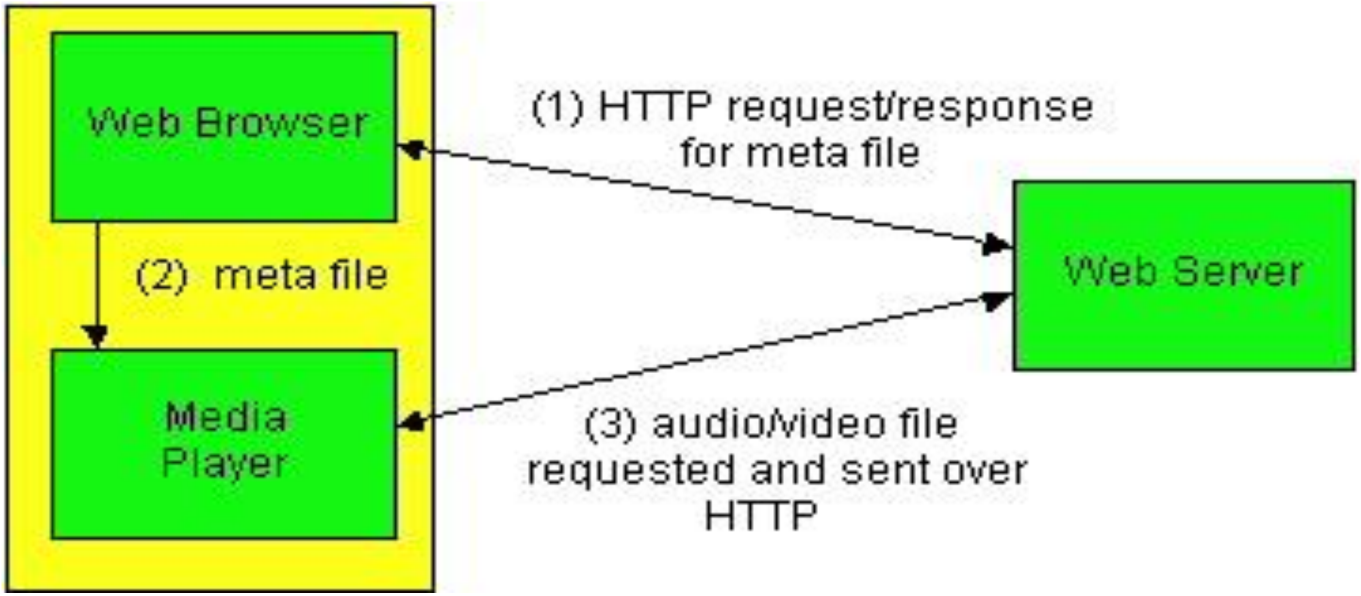
- audio or video stored in file
- files transferred as HTTP object
  - received in entirety at client
  - then passed to player

audio, video not streamed:

- no “pipelining”, long delays until playout!
- no possible solution if content is generated dynamically at time of presentation (e.g. webcam presenting weather situation at a given location)

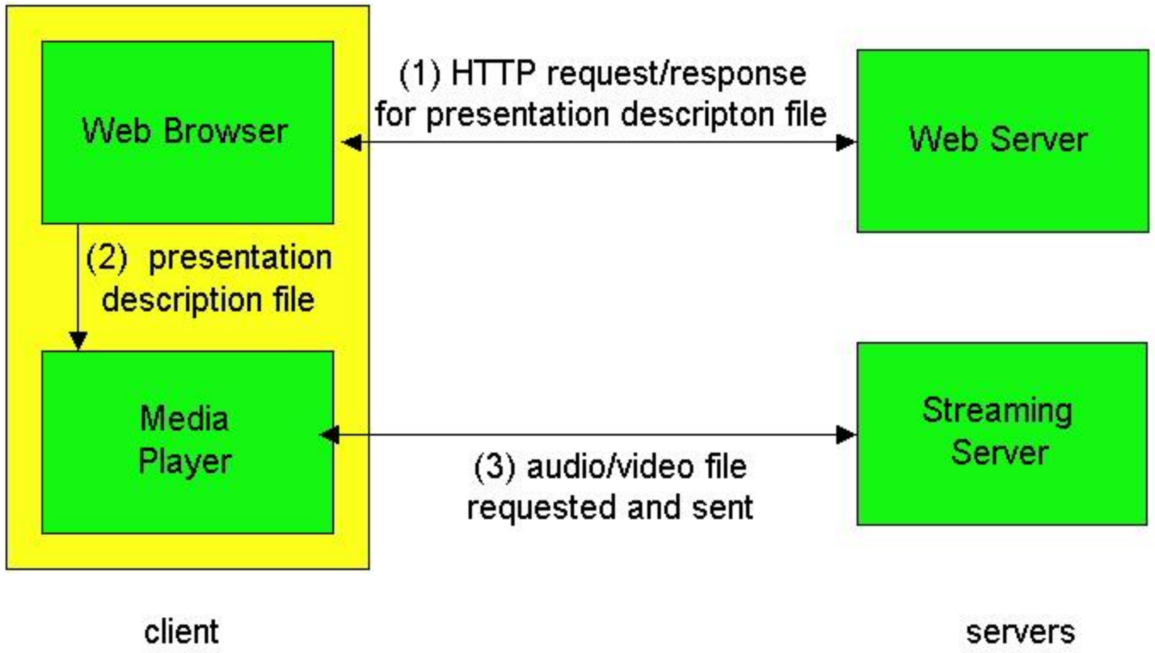
B3.2 Sending Multimedia from a Streaming Server to a Helper Application

Internet multimedia: streaming approach



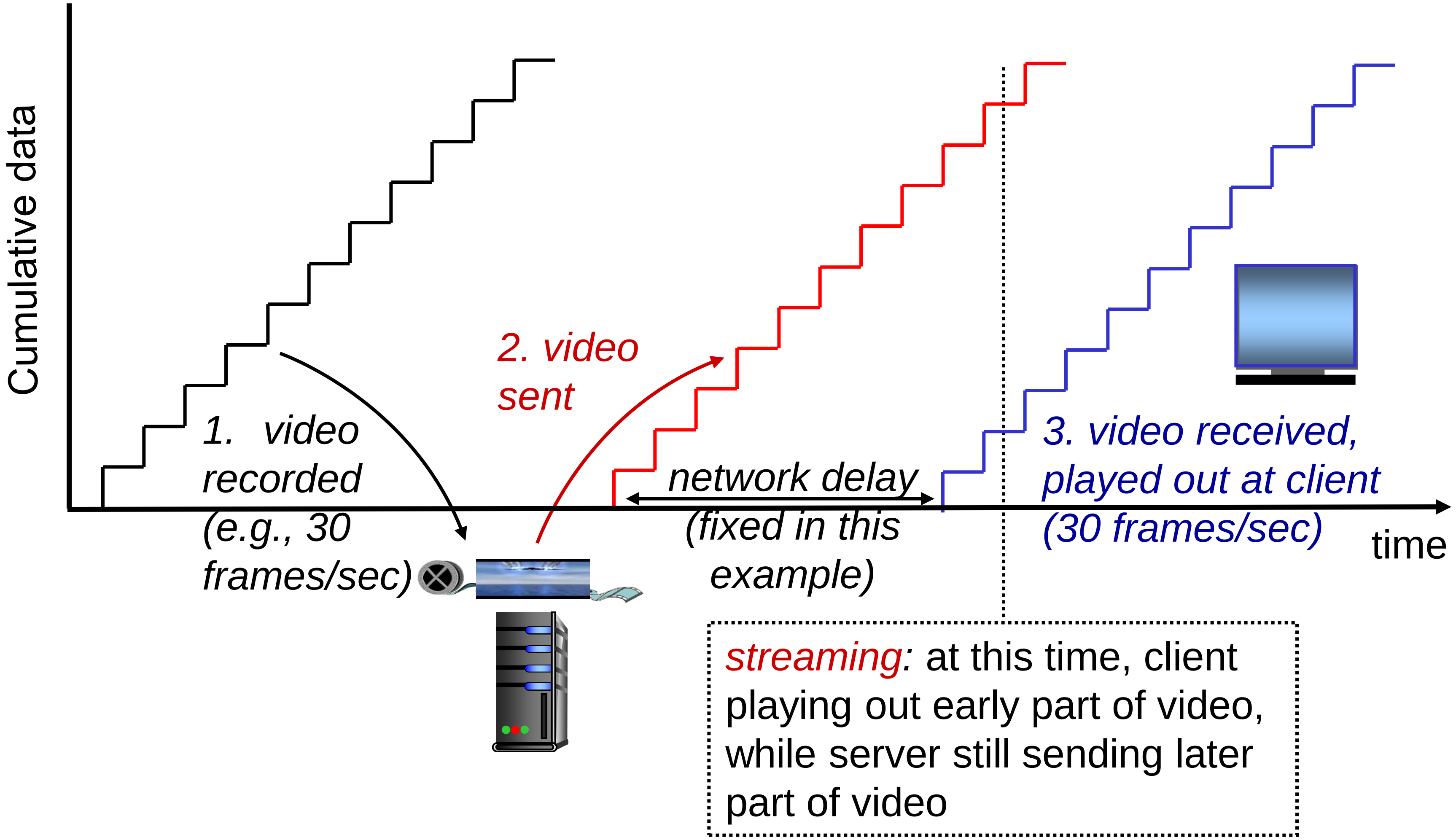
- browser GETs metafile from Web Server
- browser launches player, passing metafile
- player contacts server
- server streams audio/video to player

Streaming from a streaming server



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

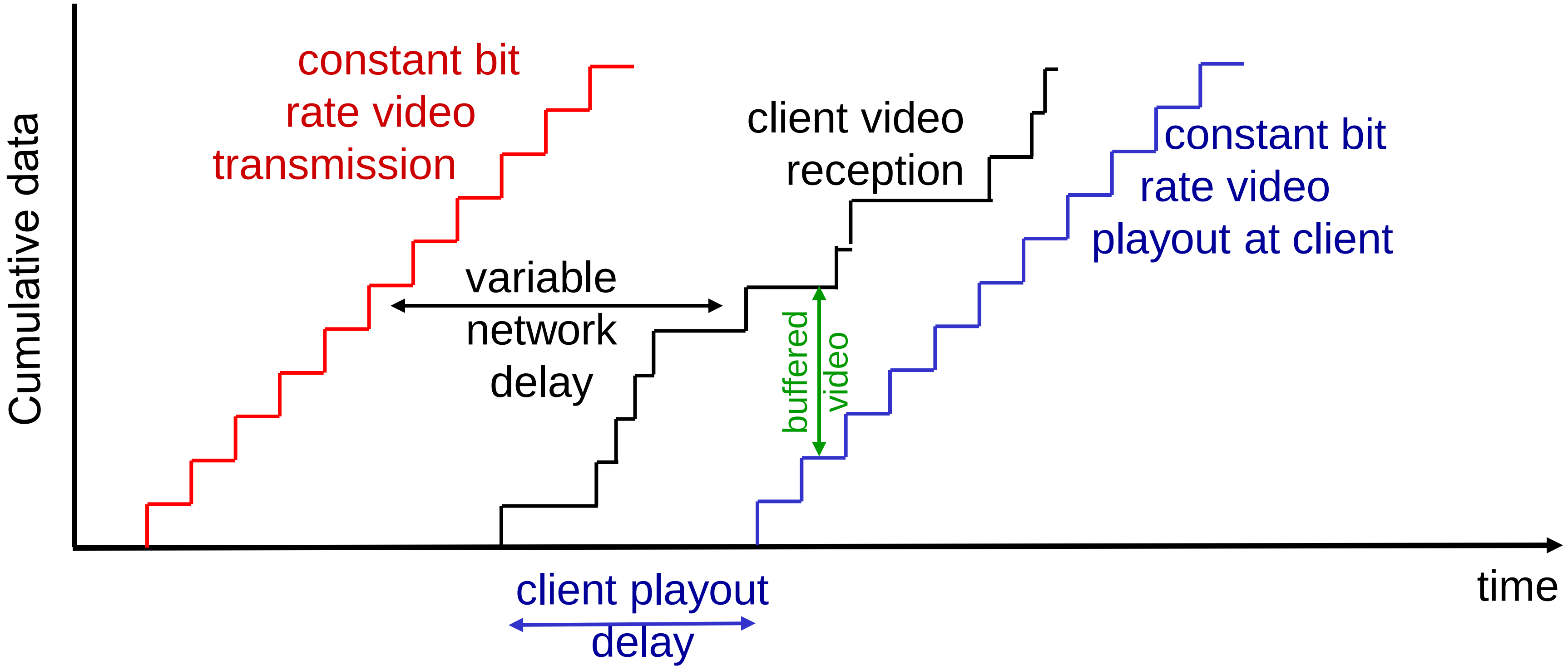
Streaming stored video:



Streaming stored video: challenges

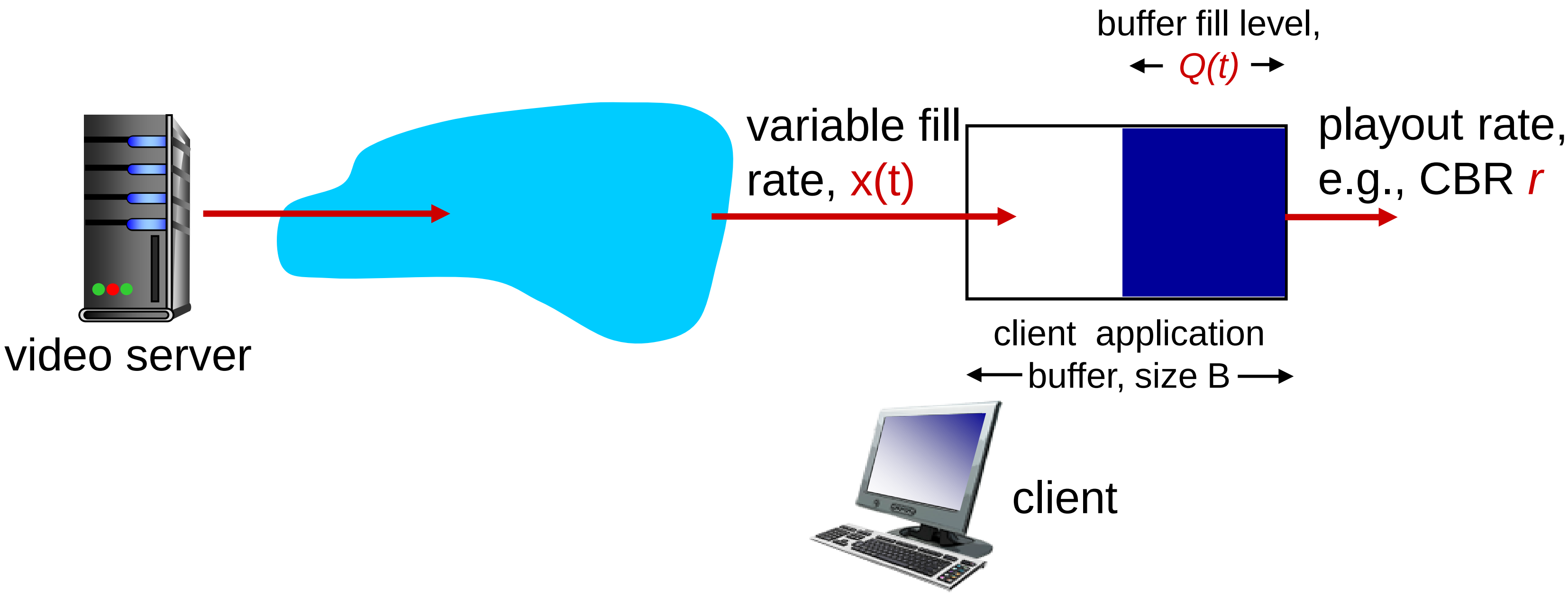
- *continuous playout constraint*: once client playout begins, playback must match original timing
  - … but *network delays are variable* (jitter), so will need *client-side buffer* to match playout requirements
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted

Streaming stored video: revisited

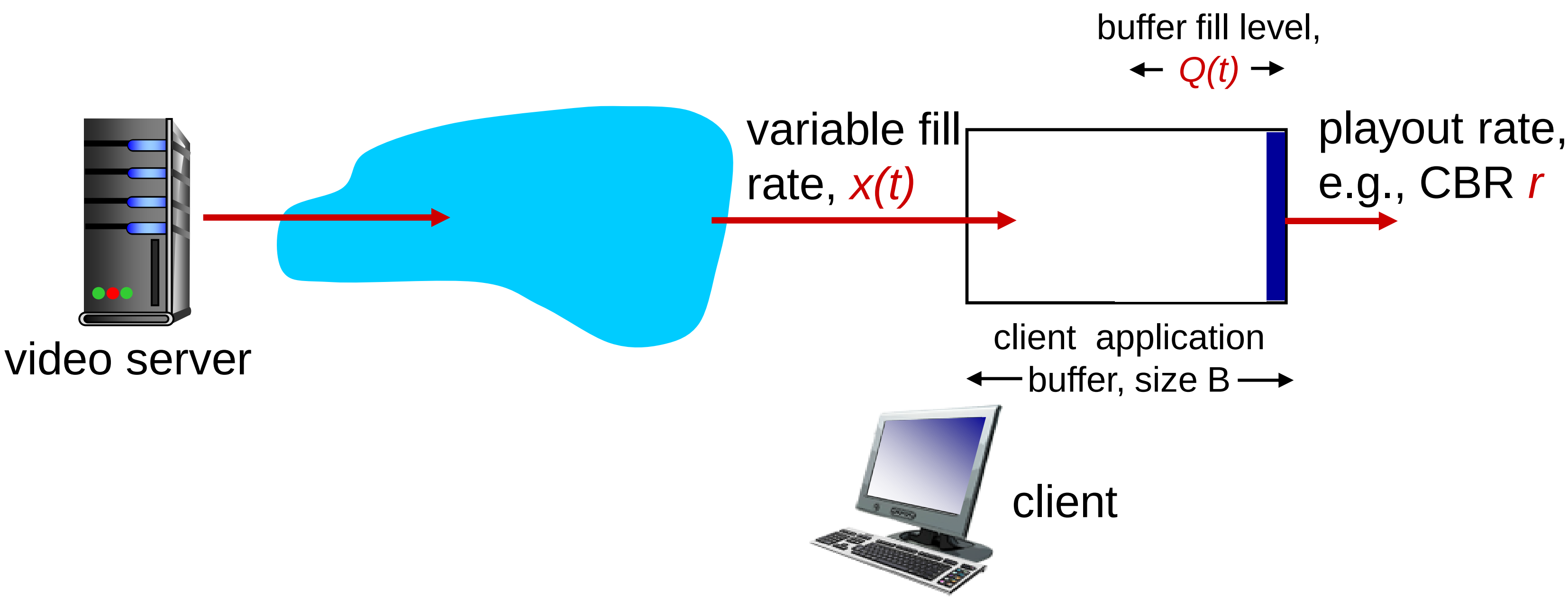


- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter

Client-side buffering, playout



Client-side buffering, playout

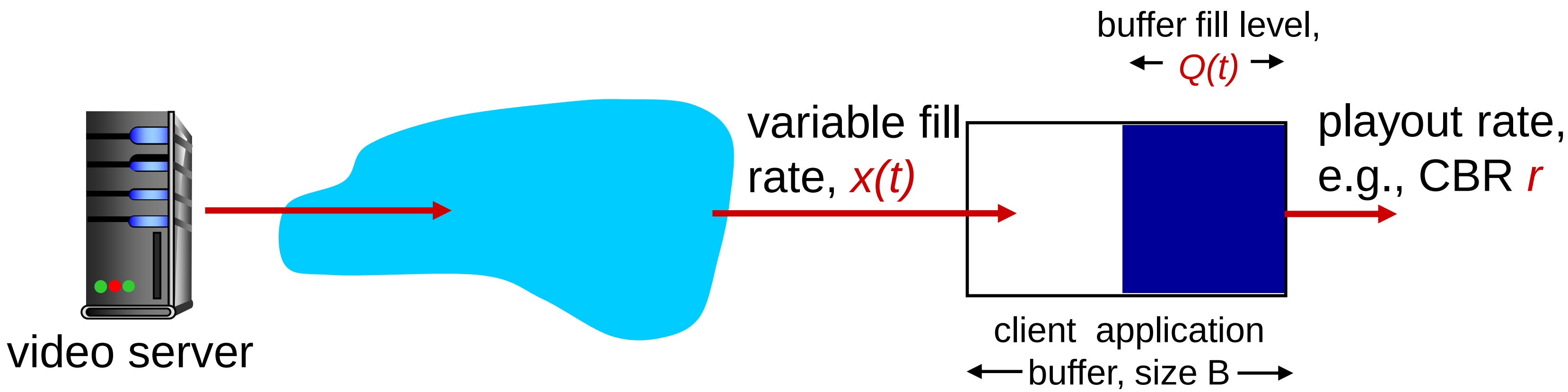


1. Initial fill of buffer until playout begins at t_p
2. playout begins at t_p
3. buffer fill level varies over time as fill rate $x(t)$ varies and playout rate r is constant

Client-side buffering, playout

→ **„Producer-Consumer-Relationship“**:

- Producer = network transmitting stream
- Consumer = decoder reconstructing stream



playout buffering: average fill rate ($\bar{x}$), playout rate (r):

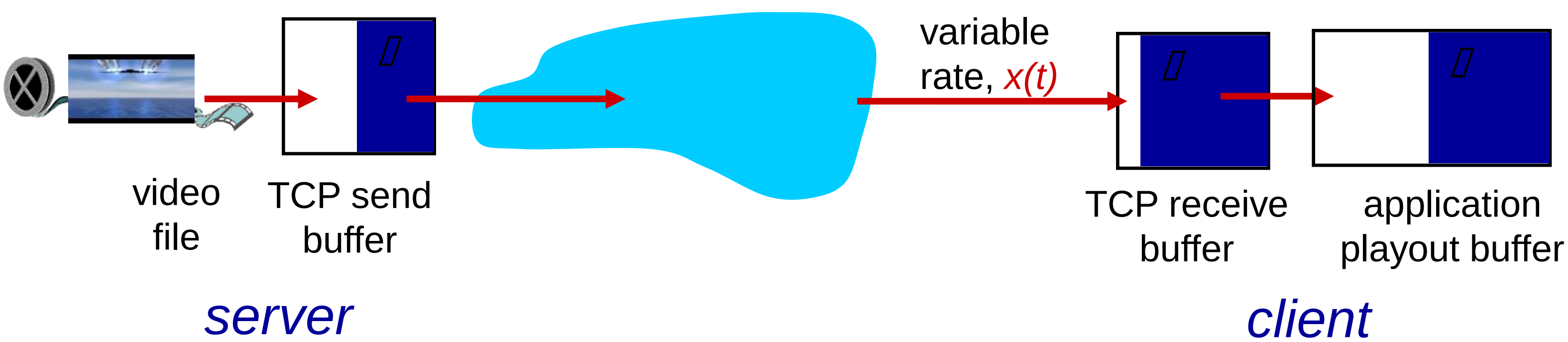
- $\bar{x} < r$: buffer eventually empties (causing freezing of video playout until buffer again fills)
- $\bar{x} > r$: buffer will not empty, provided initial playout delay is large enough to absorb variability in x(t)
  - *initial playout delay tradeoff:* buffer starvation less likely with larger delay, but larger delay until user begins watching

Streaming multimedia: UDP

- server sends at rate appropriate for client
  - often: send rate = encoding rate = constant rate
  - transmission rate can be oblivious to congestion levels
- short playout delay (2-5 seconds) to remove network jitter
- error recovery: application-level, time permitting
- RTP [RFC 2326]: multimedia payload types
- UDP may *not* go through firewalls

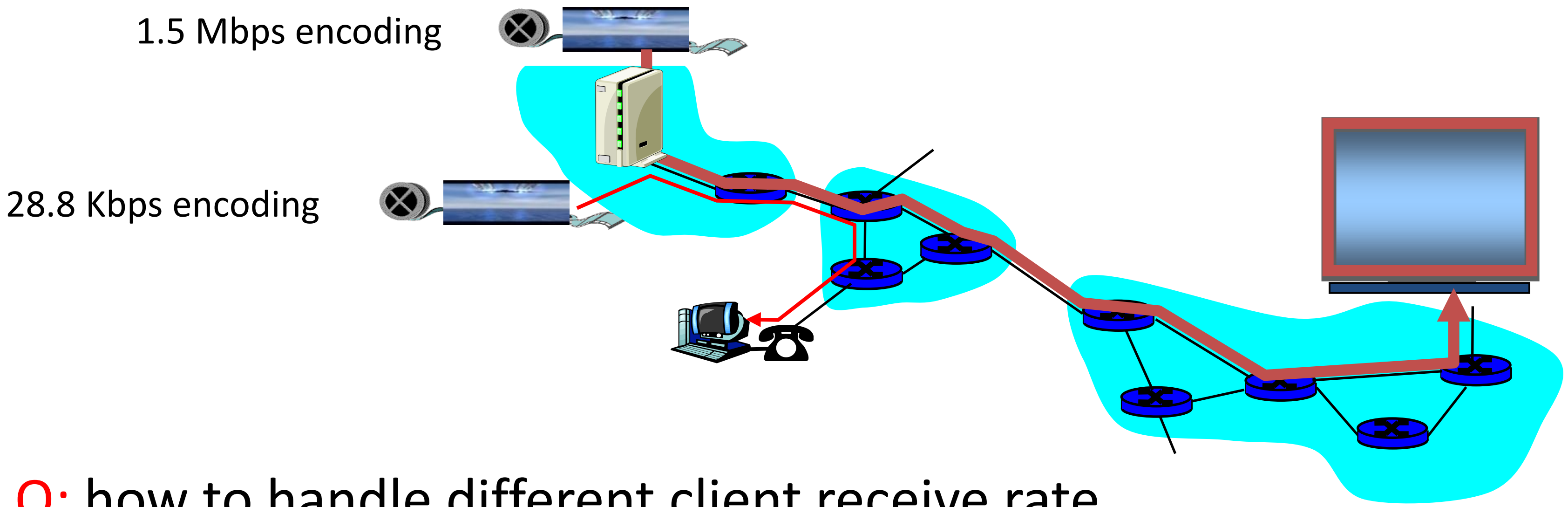
Streaming multimedia: HTTP

- multimedia file retrieved via HTTP GET
- send at maximum possible rate under TCP



- fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)
- 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

B3.3 Real-Time Streaming Protocol (RTSP)

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 RTSP 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".

RTSP Example

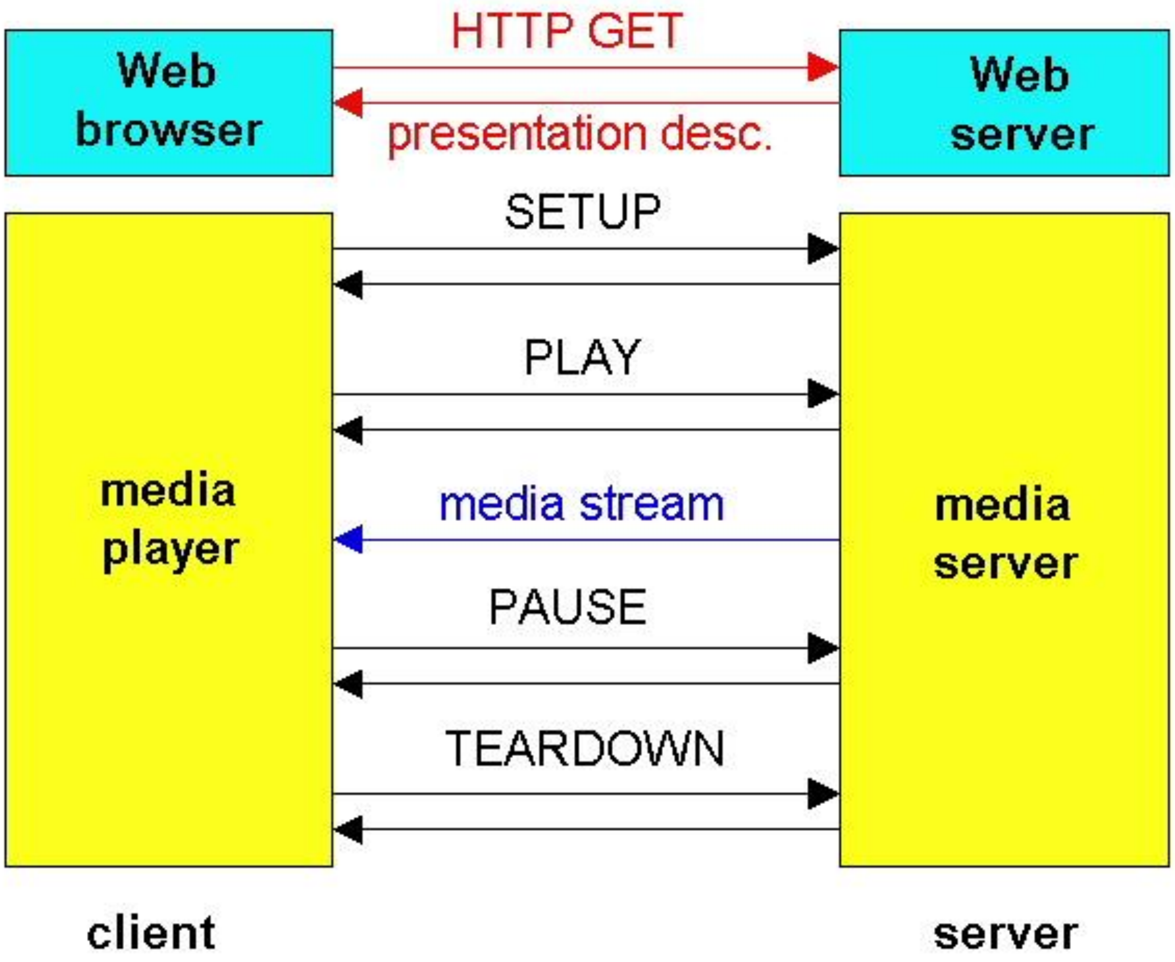
Scenario:

- 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

Part B "**Media and Real-time Communications; Service-integrated Networks**"

B4. Media Communications in Best-Effort Networks

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 (in particular, in Section B4.1 techniques discussed being useful for QoS improvement)

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.

→ For further details regarding VoIP, cf. Section B9.1

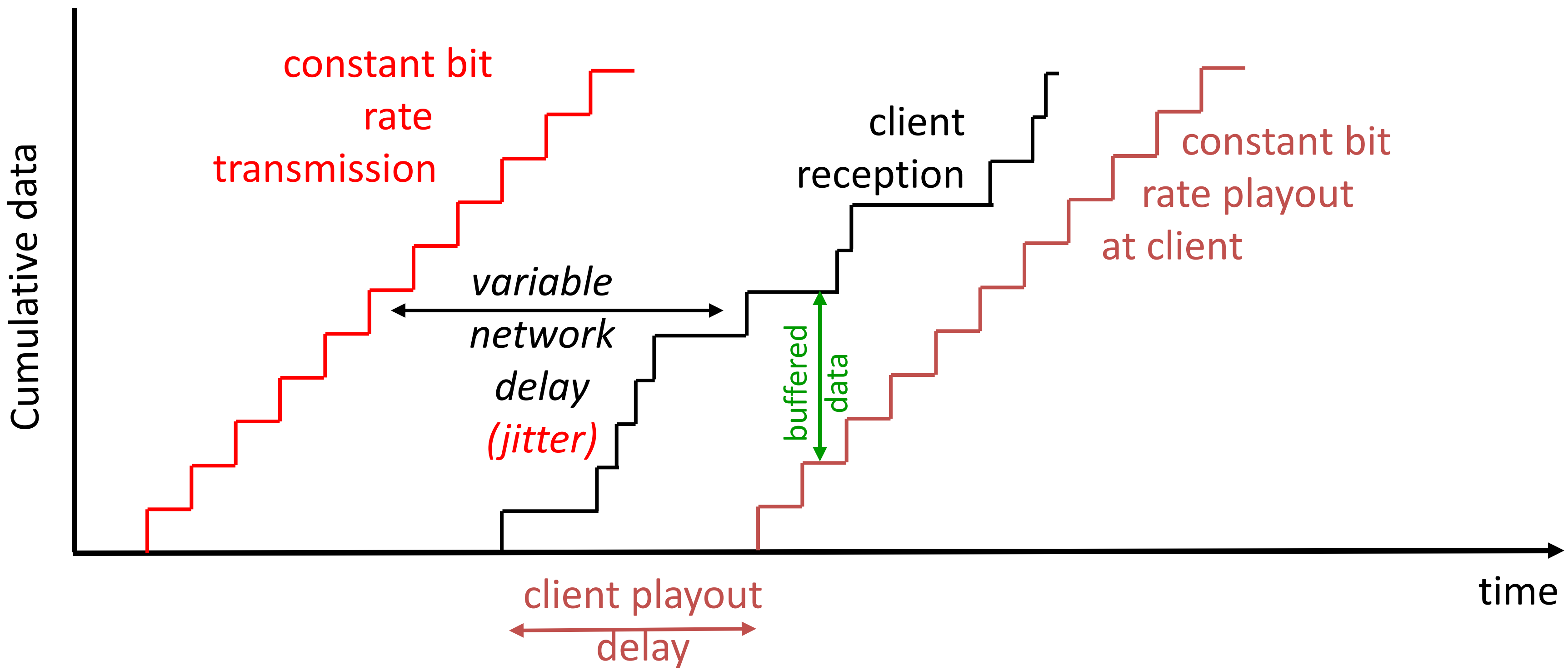
B4.1 The Limitations of a Best-Effort Service

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.

B4.2 Removing Jitter at the Receiver for Audio

Delay Jitter



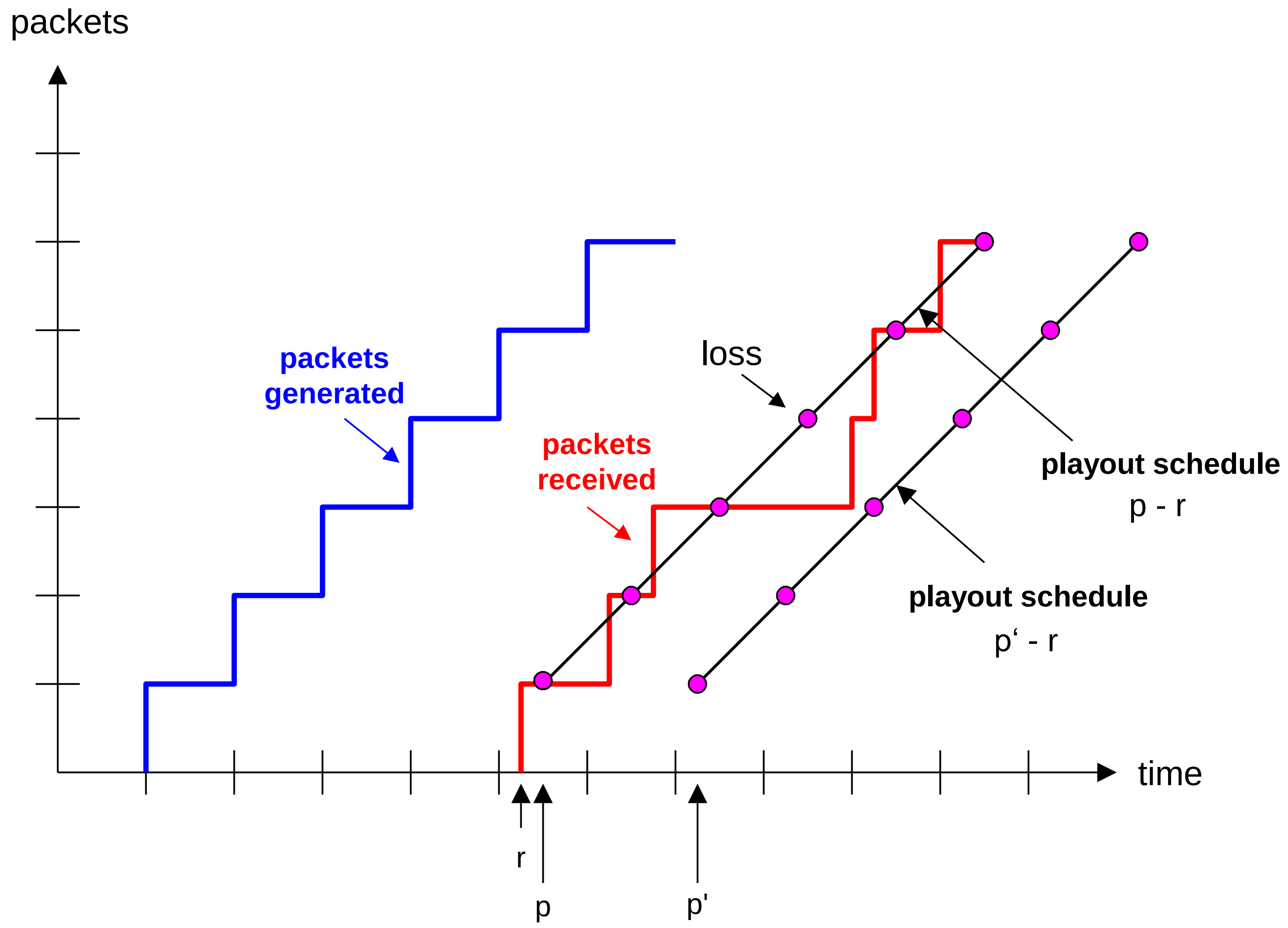
- consider end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

Voice-over-IP (VoIP): Fixed Playout Delay

- receiver attempts to playout each chunk exactly *q* msec 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.

VoIP: 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 i^{th} 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 i^{th} packet

d_i = estimate of average network delay after receiving i^{th} packet

Dynamic estimate of average delay at receiver:

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

where α is a fixed constant (e.g., α = .01); cf. also called *"geometric weighting"* (of delay measurements to estimate the current network delay).

Adaptive playout delay (2)

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

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

- 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 p_i is:

$$p_i = t_i + d_i + Kv_i$$

where K is positive constant

- remaining packets in talk spurt are played out periodically

Adaptive Playout (3)

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

A:

- if no loss, receiver looks at successive timestamps
  - difference of successive stamps > 20 msec $\rightarrow$ 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 $\rightarrow$ talk spurt begins.

B4.3 Recovery from Packet Loss (1)

Forward Error Correction (FEC): simple scheme

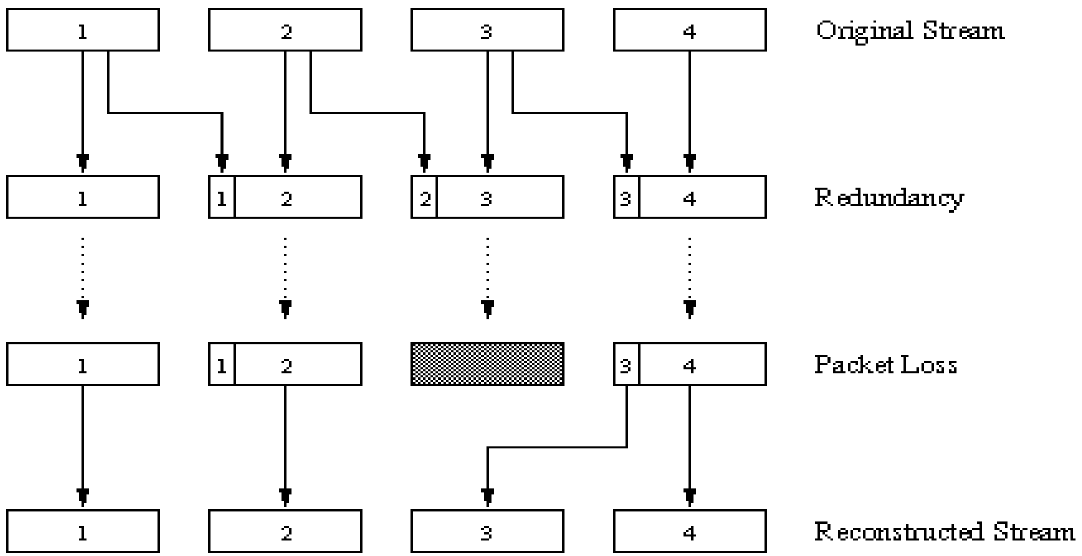
- for every group of *n* chunks create redundant chunk by exclusive OR-ing *n* original chunks
- send out *n+1* chunks, increasing bandwidth by factor *1/n*.
- can reconstruct original *n* chunks if at most one lost chunk from *n+1* chunks

- 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)

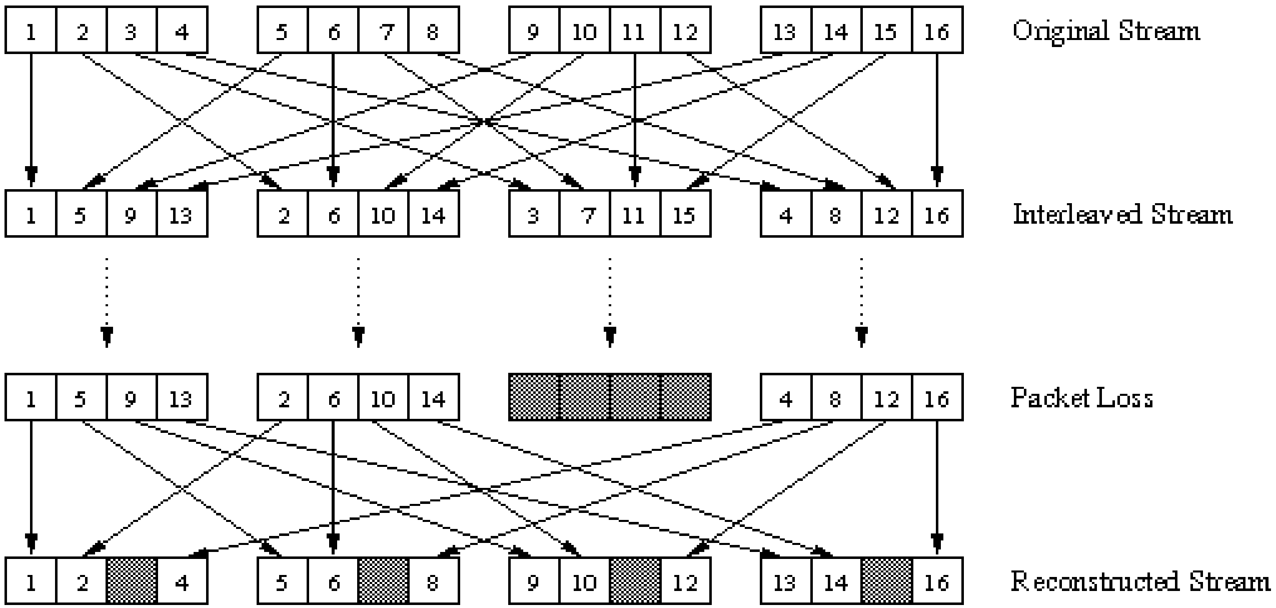
2nd FEC scheme

- "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
- generalization: can also append *(n-1)*st and *(n-2)*nd low-bit rate chunk

Recovery from packet loss (3)



Interleaving to conceal loss:

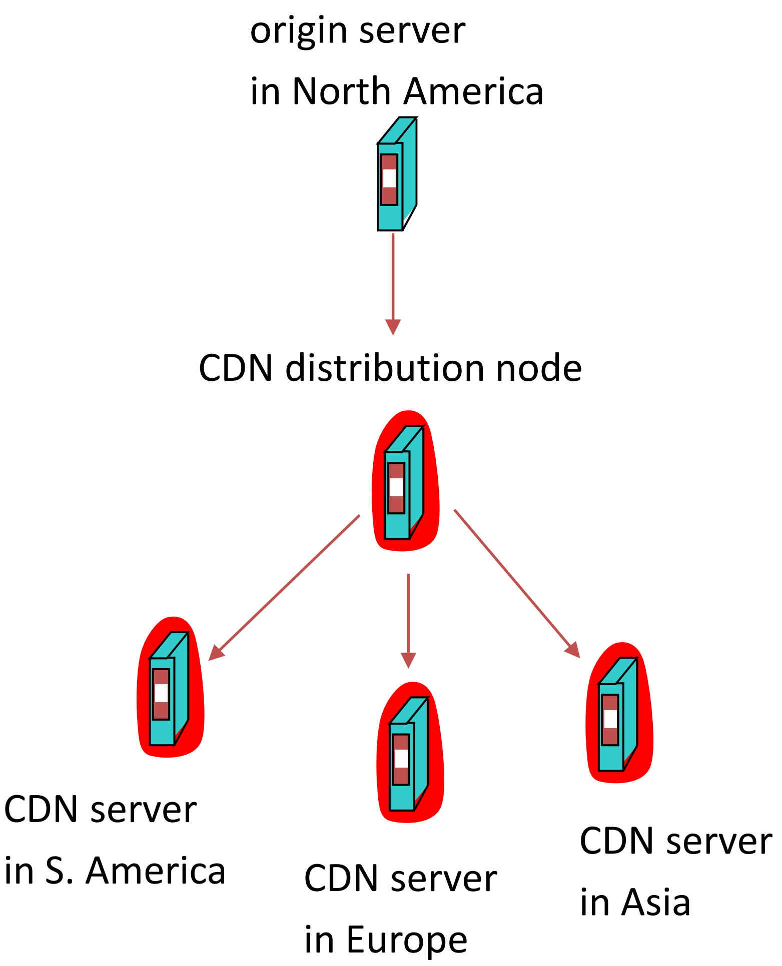
- Audio chunks divided into smaller units, e.g., four 5 msec units per 20 msec audio chunk
- packet contains small units from different chunks

- if packet lost, still have *most* of every chunk
- no redundancy overhead, but increases playout delay

B4.4 Distributing Multimedia in Today's Internet: Content Distribution Networks (CDNs)

Content replication

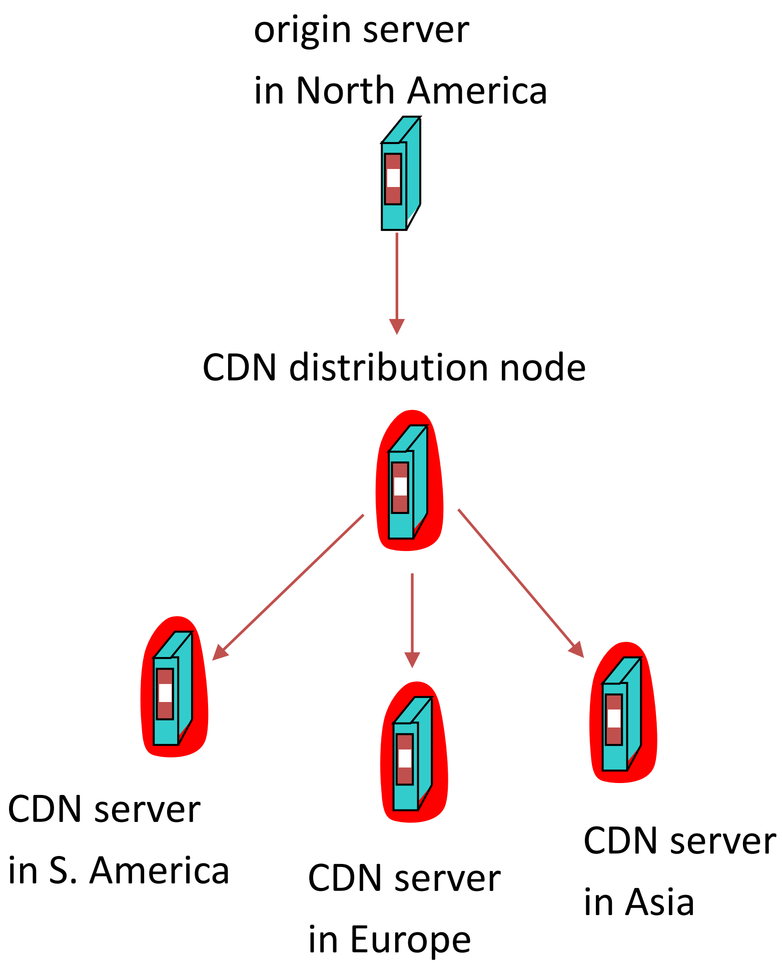
- challenging to stream large files (e.g., video) from single origin server in real time
- *solution:* replicate content at hundreds of servers throughout Internet
  - content downloaded to CDN servers ahead of time
  - *placing content "close" to user avoids impairments (loss, delay) of sending content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



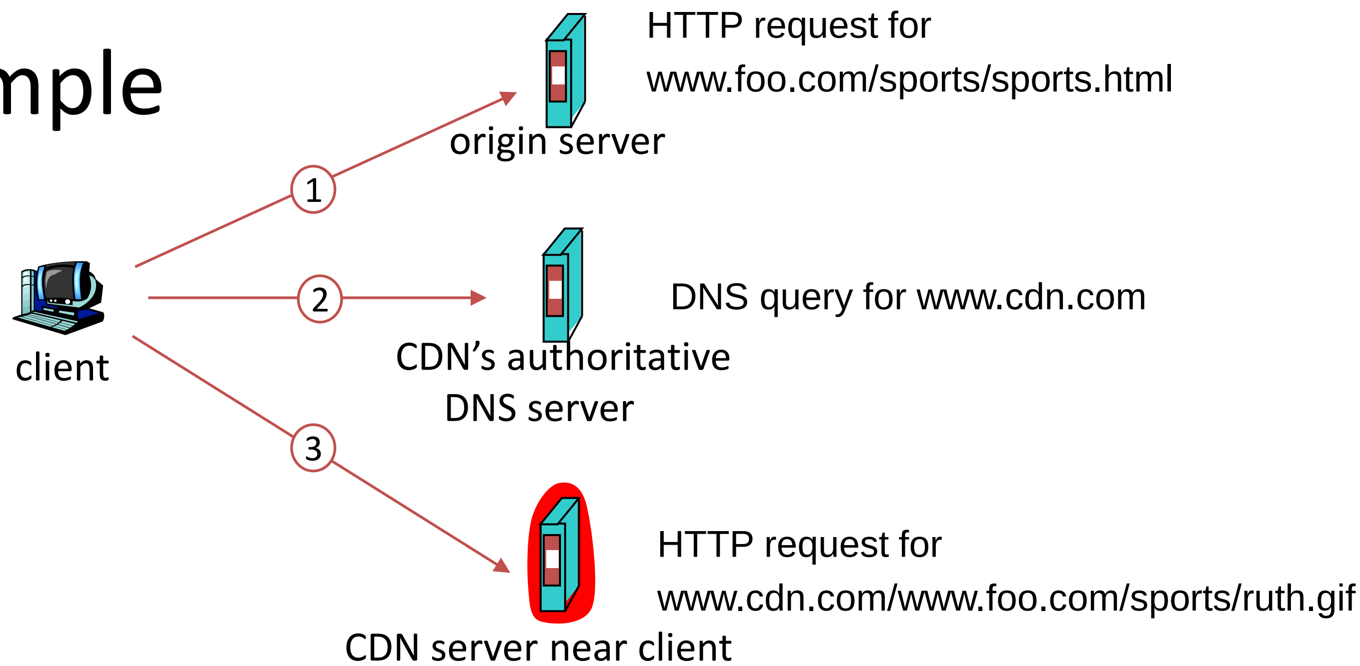
Content distribution networks (CDNs)

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



<u>origin server (www.foo.com)</u>

- distributes HTML
- replaces:

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

  with

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

<u>CDN company (cdn.com)</u>

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

More about CDNs

<u>routing requests</u>

- 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

B4.5 Dimensioning Best-Effort Networks to Provide QoS

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
  - choose 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

Part B "**Media and Real-time Communications; Service-integrated Networks**"

B5. Protocols for Real-Time Interactive Applications

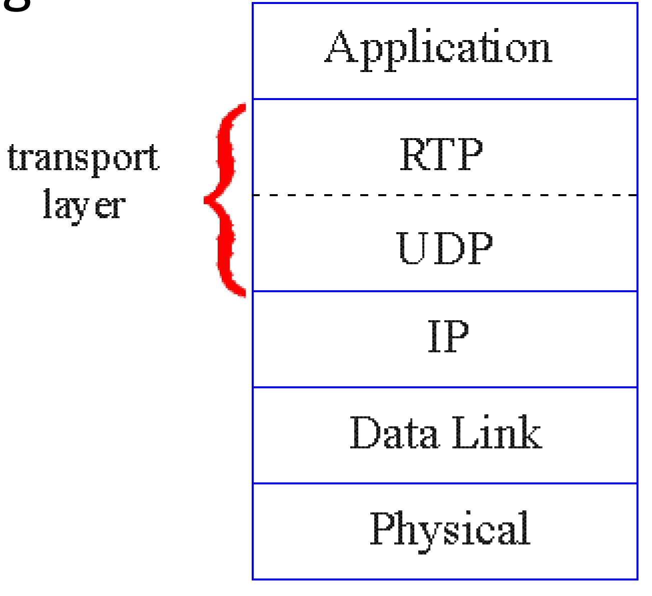
B5.1 Real-Time Protocol (RTP)

- RTP specifies packet structure for packets carrying audio, video data
- RFC 3550, cf. : *http://www.faqs.org/rfcs/rfc3550.html*
- 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

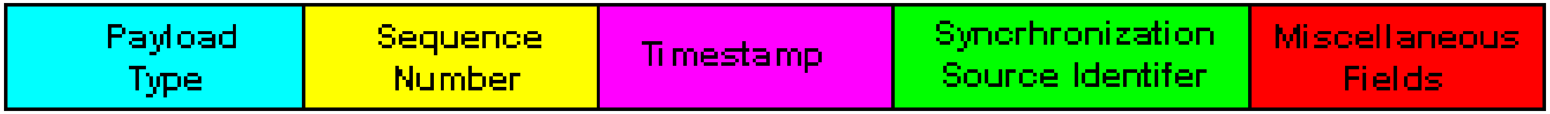
- *Example :* consider sending 64 kbps PCM-encoded voice over RTP
- application collects encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
- audio chunk + RTP header form RTP packet, which is encapsulated in UDP segment

- 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 provide best-effort service, making no special effort to ensure that RTP packets arrive at destination in timely matter.

RTP Header (1)



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 μlaw (mu-law), 64 kbps
- Payload type 3: GSM, 13 kbps
- Payload type 7: LPC (linear predictive coding), 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)

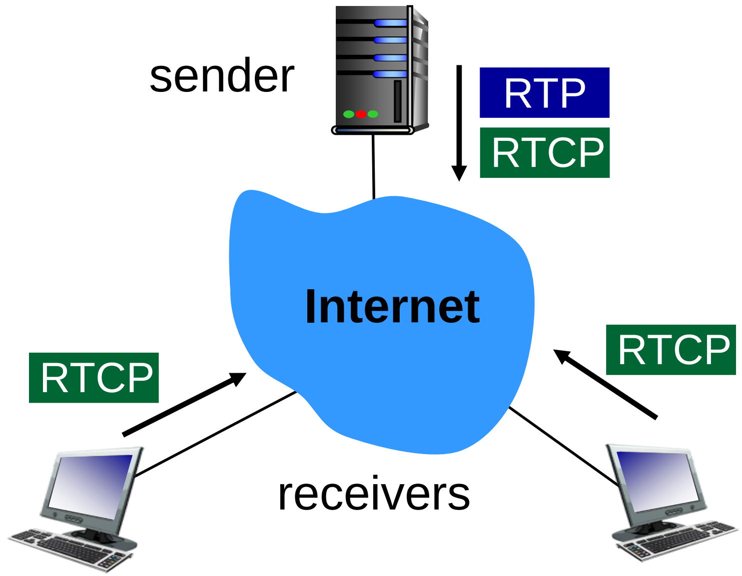
- *Timestamp field (32 bits long):* sampling instant of first byte in this RTP data packet
  - for audio, timestamp clock typically increments by one for each sampling period (for example, each 125 μsec for 8 kHz sampling clock)
  - if application generates chunks of 160 encoded samples, then timestamp 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 the RTP stream. Each stream in RTP session should have distinct SSRC.

B5.2 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: multiple multicast senders



- 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
- e.g., videoconferencing app. : 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 use association to synchronize playout of audio, video

RTCP Bandwidth Scaling

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

Example : 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.

B5.3 Session Initiation Protocol (SIP) [RFC 3261]

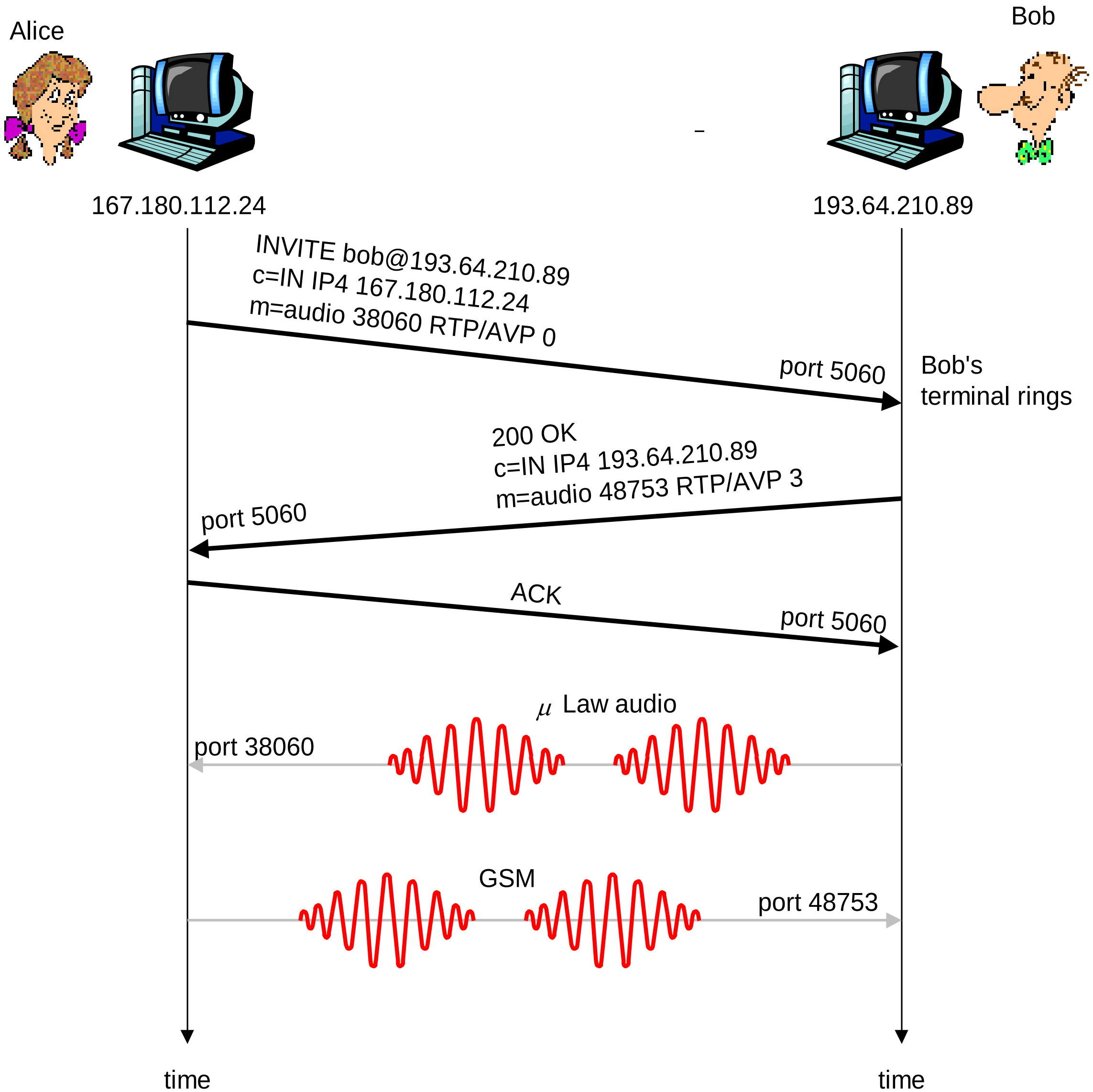
SIP long-term vision:

- 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 *(if callee so desires),* 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 µlaw, cf. RTP/AVP 0).
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM, cf. RTP/AVP 3).
- 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 μlaw 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 5060.
- Alice specifies in Via: 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

- One function of SIP server: *registrar*
- when Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server

(similar function needed by Instant Messaging)

Register Message:

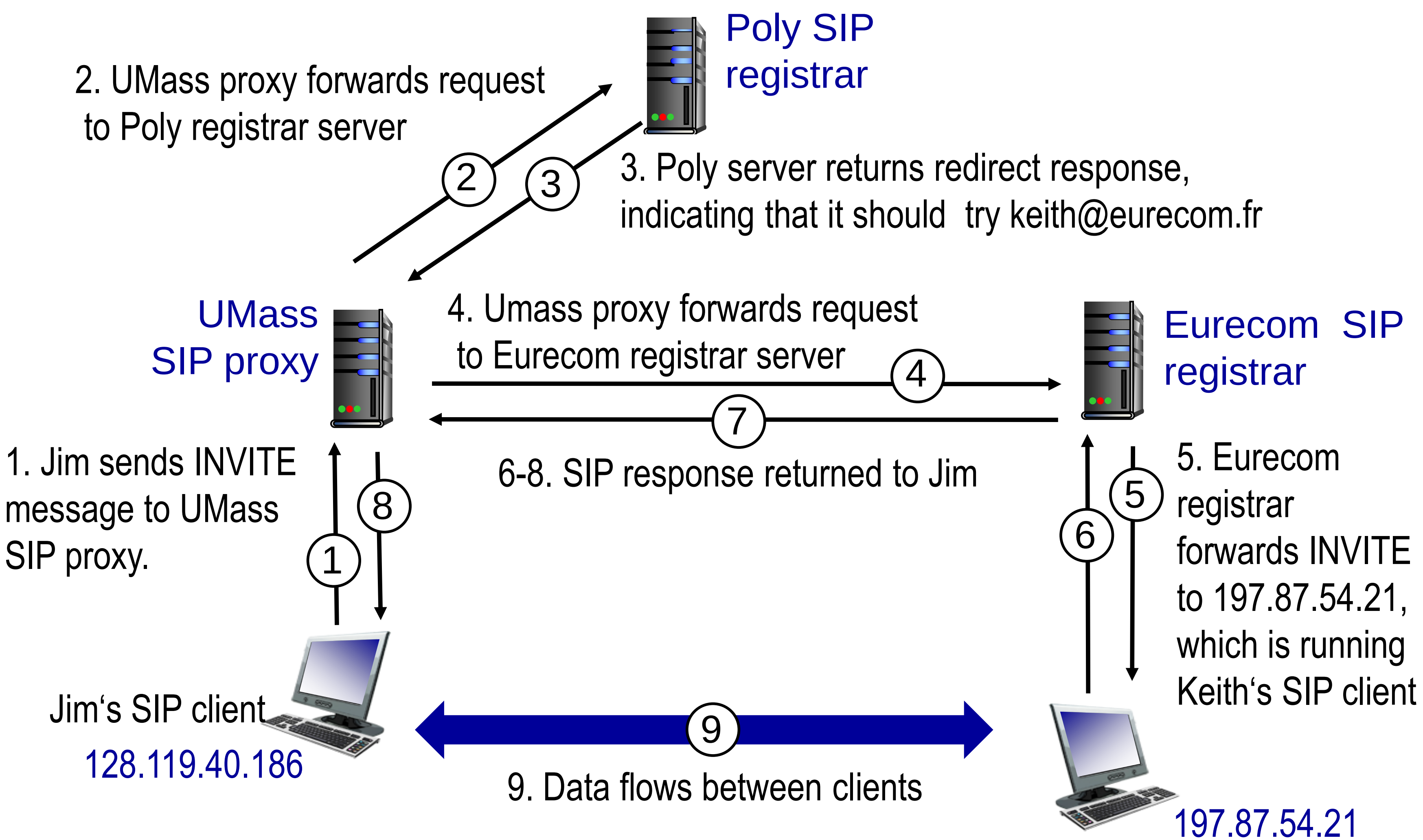
```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

SIP Proxy

- another function of SIP server: *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 (Bob) sends response back through the same set of SIP proxies
- proxy returns Bob's SIP response message to Alice
  - contains Bob's IP address
- SIP proxy analogous to local DNS server plus TCP setup

SIP example:

Caller jim@umass.edu *calls* keith@poly.edu



B5.4 H.323 for Multimedia Conferencing

Comparison of SIP 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”.

Part B "Media and Real-time Communications; Service-integrated Networks"

B6. QoS Provisioning Based on Traffic Classes and on Prioritization

B6.1 Motivating Scenarios

Network support for multimedia

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing.	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

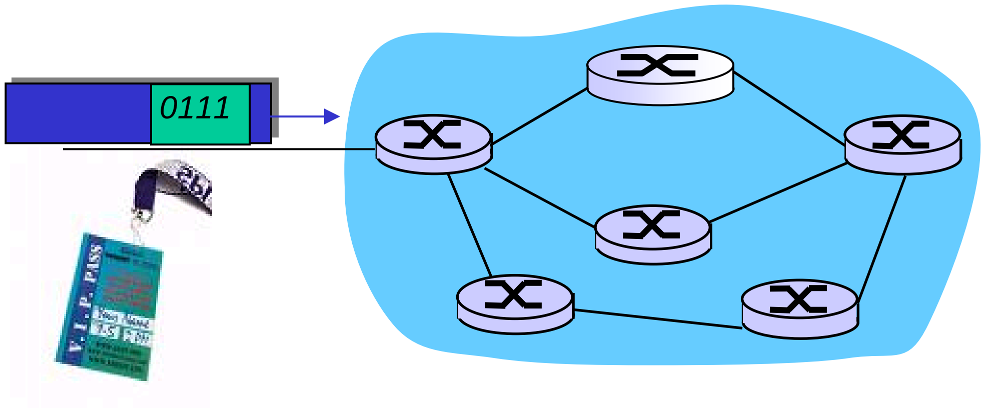
Dimensioning best effort networks

- *approach:* deploy enough link capacity so that congestion doesn't occur, multimedia traffic flows without too much delay or loss
  - low complexity of network mechanisms (use current "best effort" network)
  - high bandwidth costs
- challenges:
  - *network dimensioning:* how much bandwidth is "enough?"
  - *estimating network traffic demand:* needed to determine how much bandwidth is "enough" (for that much traffic)

Providing multiple classes of service

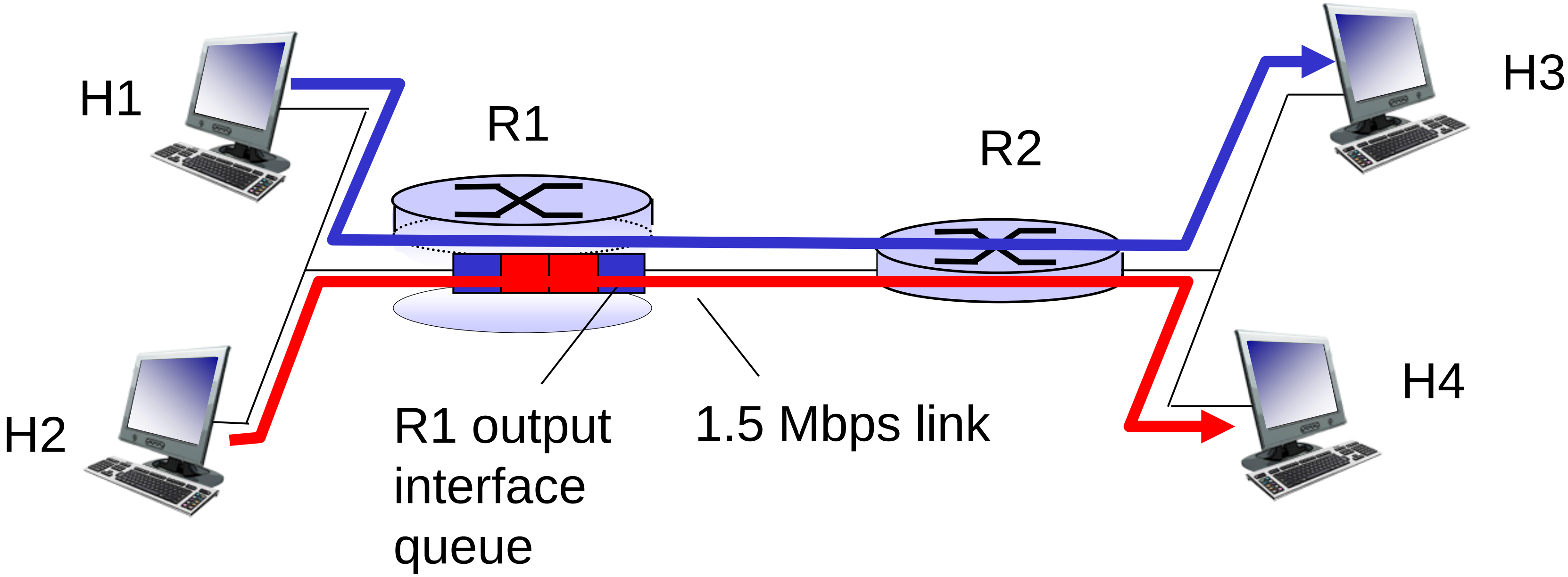
- thus far: making the best of best effort service
  - one-size fits all service model
- alternative: multiple classes of service
  - partition traffic into classes
  - network treats different classes of traffic differently (analogy: VIP service versus regular service)

- granularity: differential service among multiple classes, *not among individual connections*
- history: ToS bits (ToS: 'Type of Service' field)



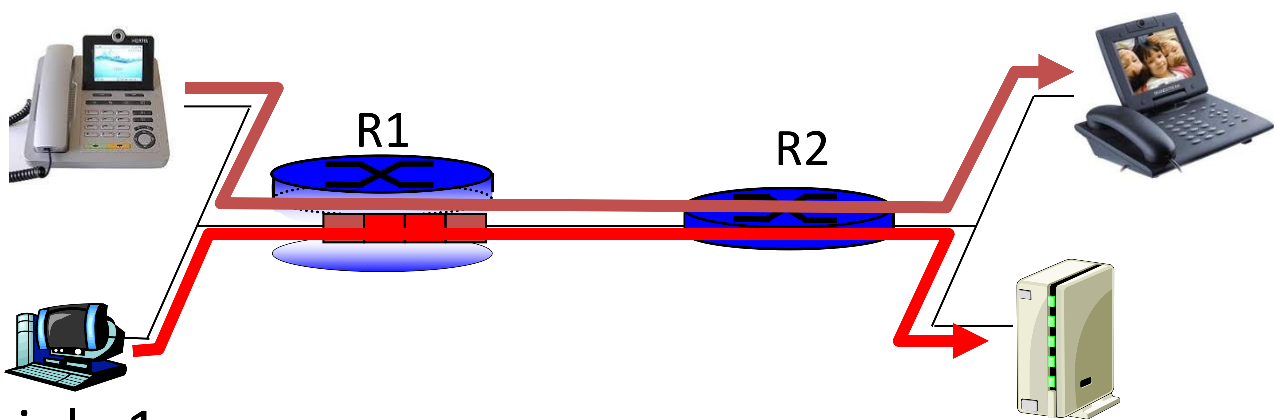
Here: VIP = Very Important Packet

Multiple classes of service: scenario



Scenario 1: mixed FTP and audio/video

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

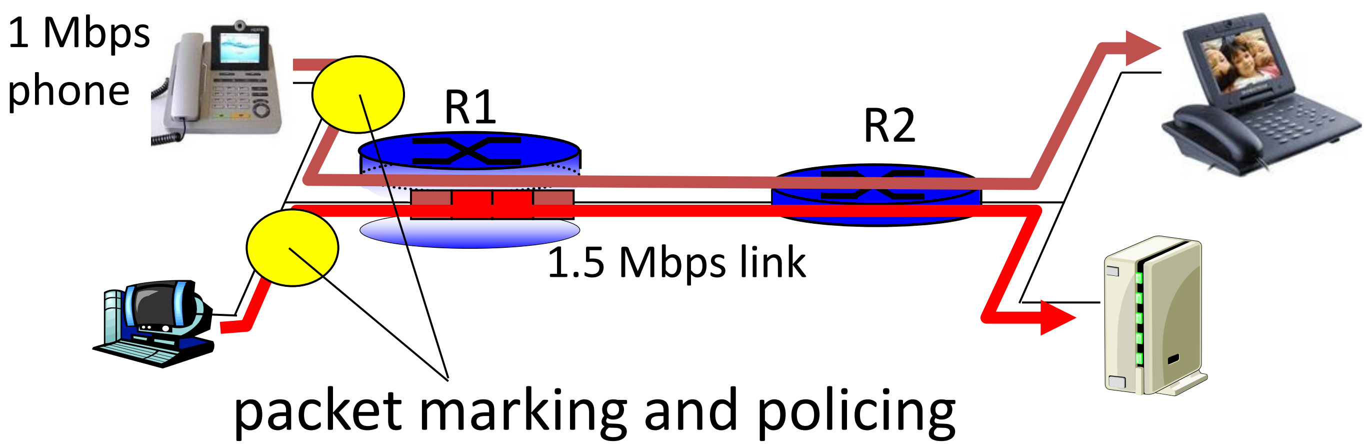


Principle 1

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/video 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), cf. Chapter B8

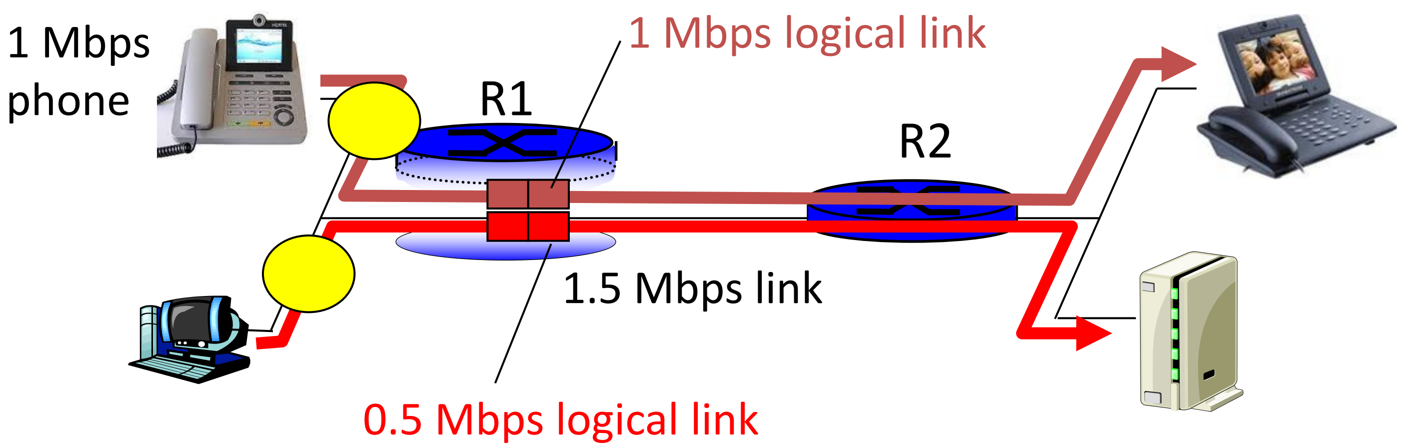


Principle 2

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 flow doesn't use its allocation

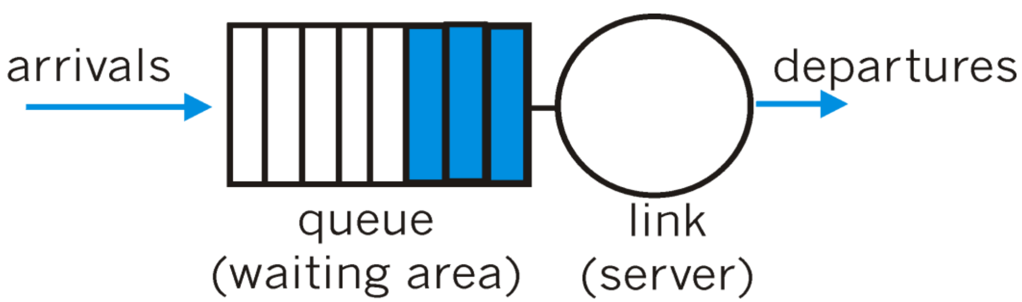


Principle 3

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

B6.2 Scheduling and Policing Mechanisms

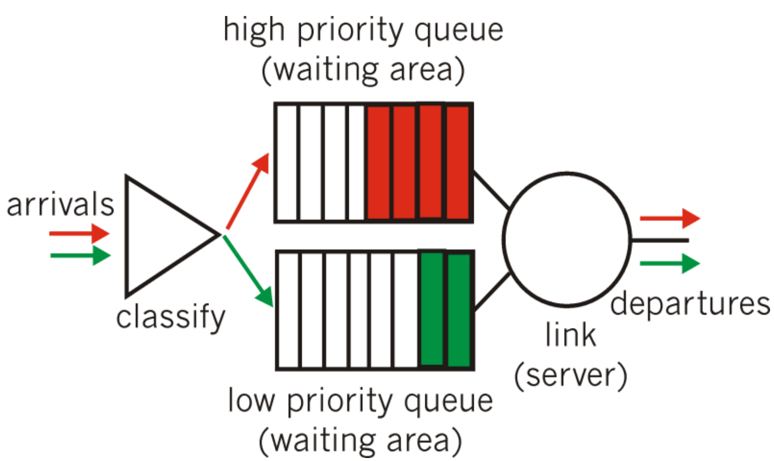
- scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - real-world example?
  - discard policy: if packet arrives to full queue: who to discard?
    - Tail drop: drop arriving packet
    - priority: drop/remove on priority basis
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

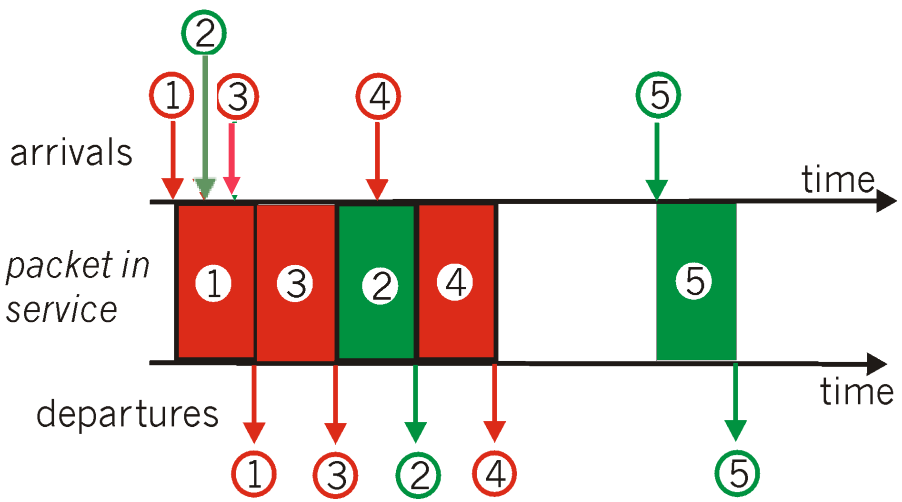


Scheduling Policies: more

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?

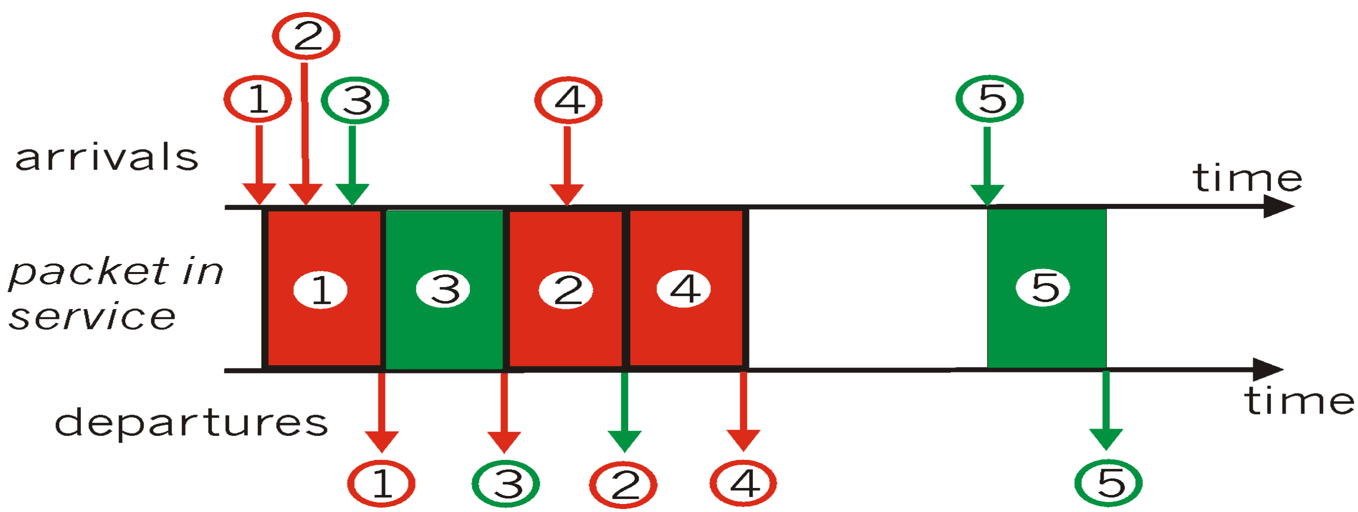




Scheduling Policies: still more

round robin scheduling:

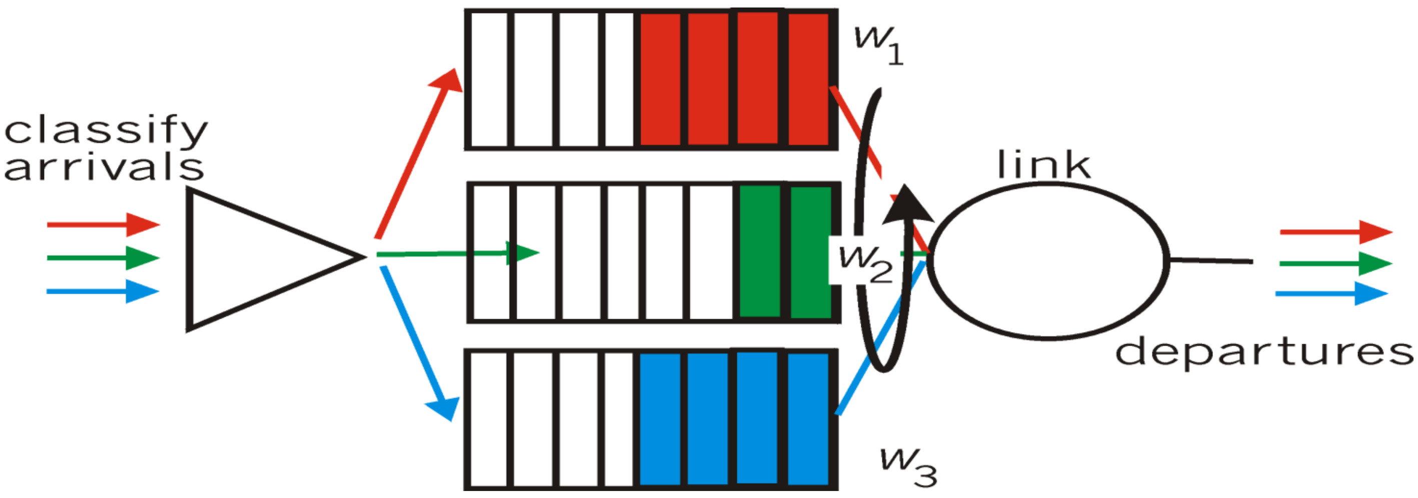
- multiple classes
- cyclically scan class queues, serving one from each class (if available)
- real world example?



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

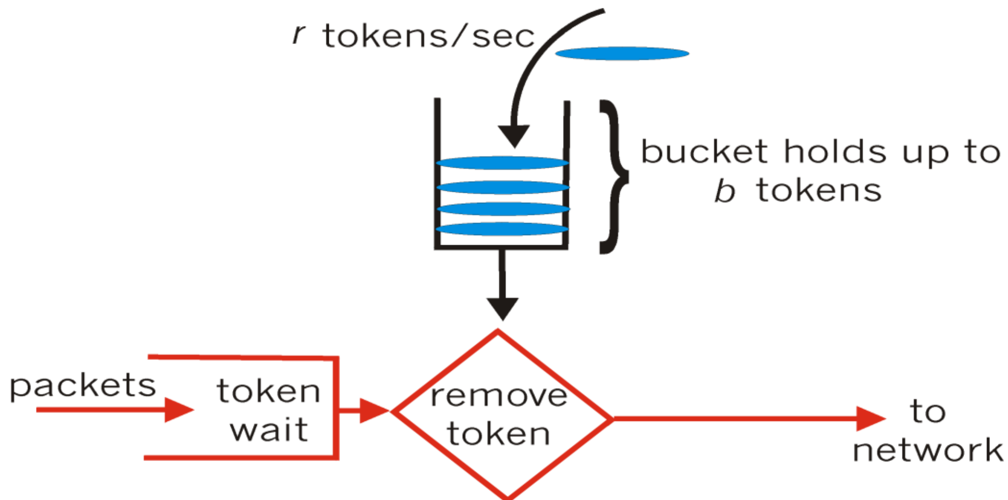
Goal: limit traffic to not exceed declared parameters.

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 pkts per sec. (pps) 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.

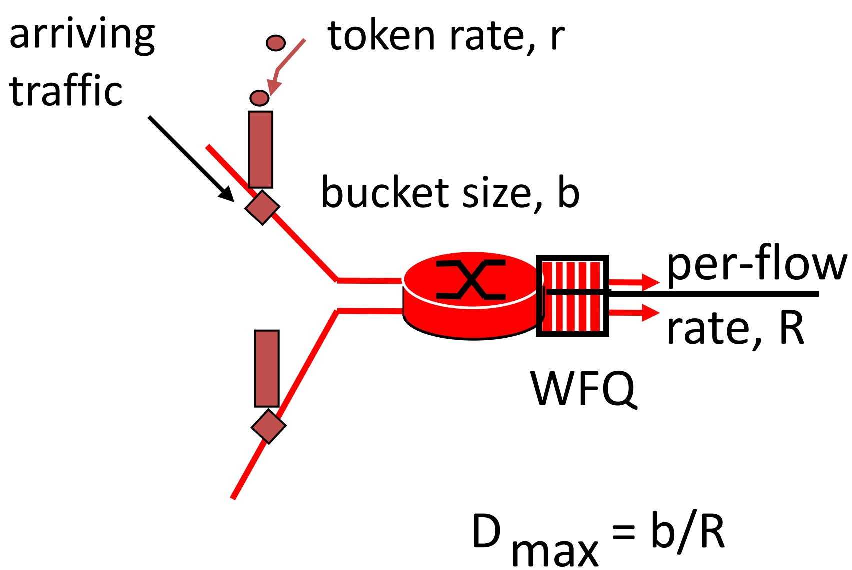


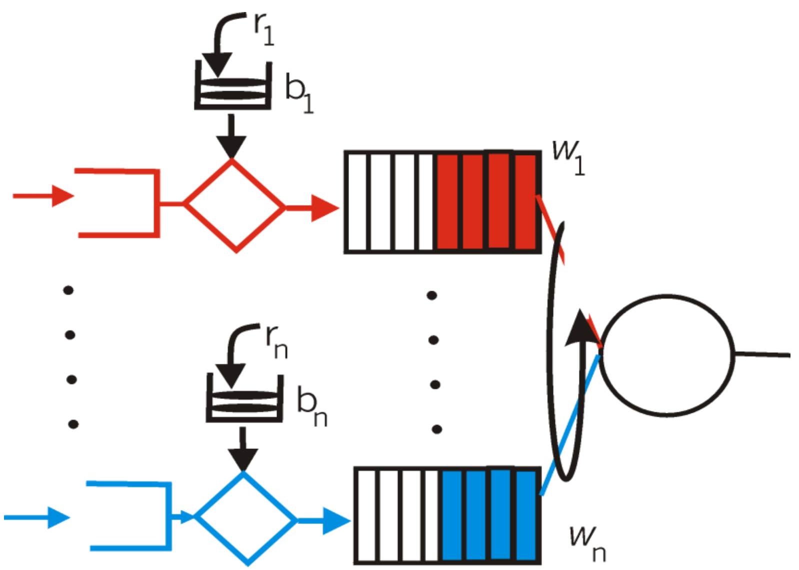
- bucket can hold *b* tokens
- tokens generated at rate *r [token/sec]* unless bucket full
- *over interval of length t: number of packets* or *number of Bytes L(t) admitted is bounded by* $L(t) \leq (r \cdot t + b)$,

cf. LBAP model (= Linear Bounded Arrival Process).

Policing Mechanisms (more)

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





Notation : D_{max} = max. delay of, e.g., packet
(where delay = time of departure – time of arrival)

B6.3 Differentiated Services (DiffServ)

- 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 service classes, provide functional components to build service classes

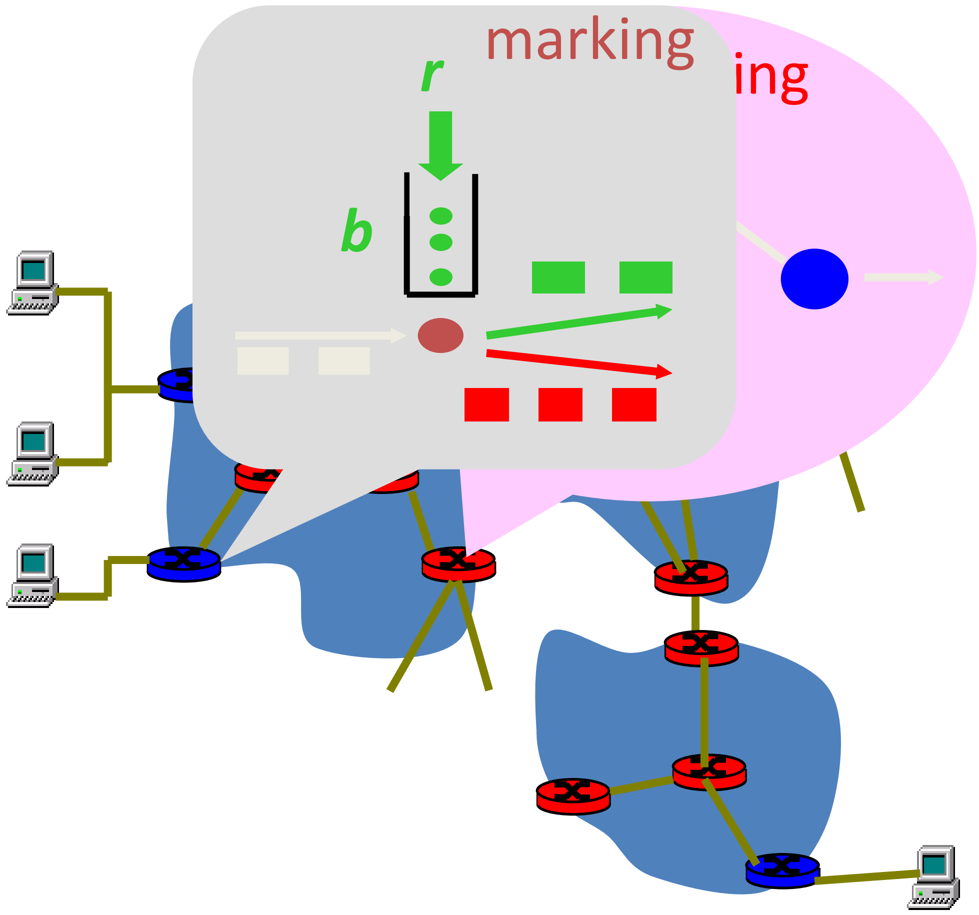
DiffServ Architecture

<u>Edge router:</u>

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

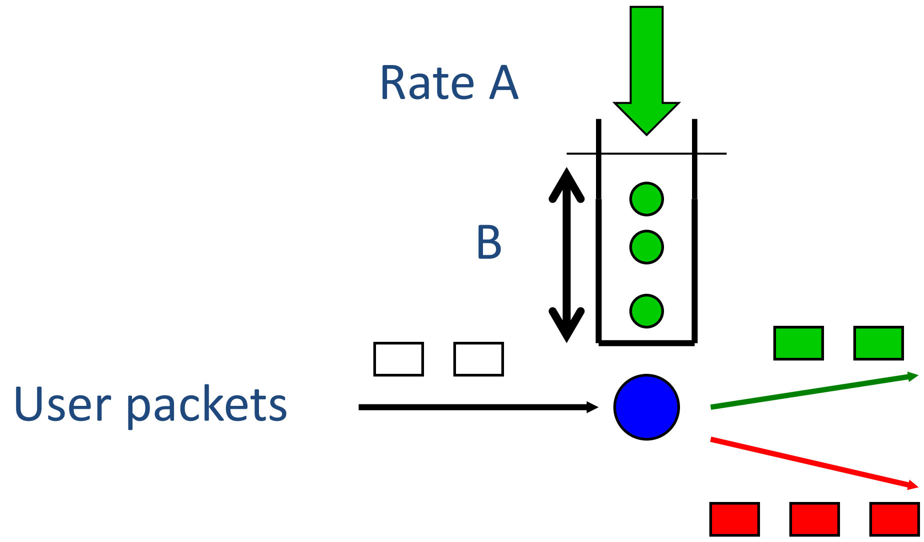
<u>Core router:</u>

- 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

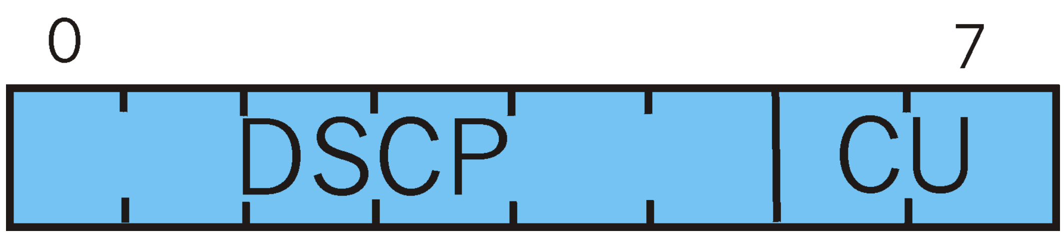


Possible usage of marking:

- 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

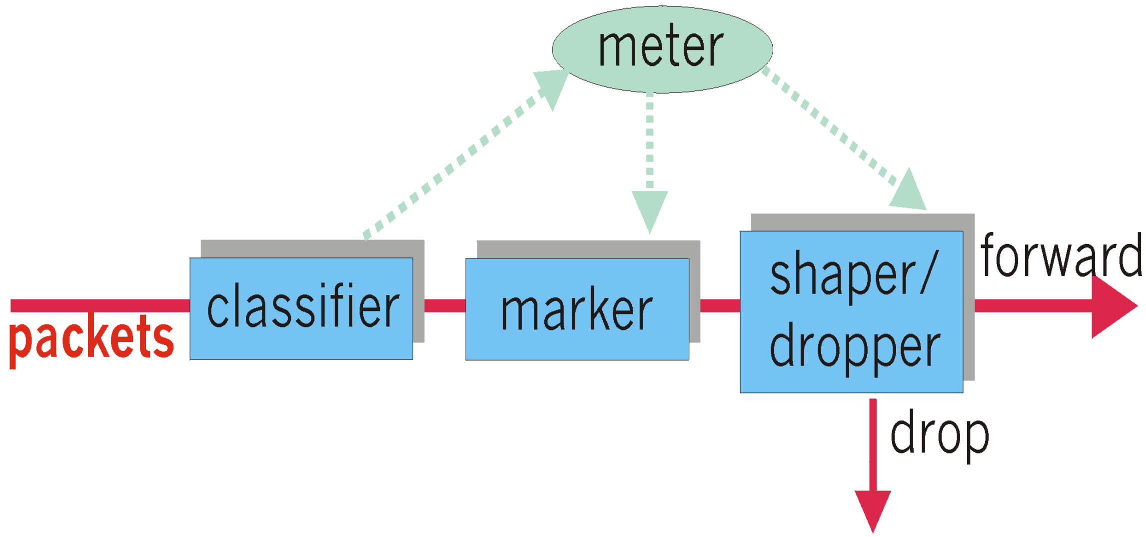
- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB (per-hop behaviour) that the packet will receive
- 2 bits are currently unused (CU)



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 results 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 (cf. table, below)
  - each guaranteed minimum amount of bandwidth
  - each with three drop preference partitions

  Remark: Service according to class prio (possibly no strict usage of prioritization, e.g. Weighted Fair Queueing / WFQ); dropping of pkts within a class according to drop preference

  → Assured Forwarding (AF) Behavior Group :

	Class 1 (lowest prio)	Class 2	Class 3	Class 4 (highest prio)
Low Drop	AF11 (DSCP 10)	AF21 (DSCP 18)	AF31 (DSCP 26)	AF41 (DSCP 34)
Medium Drop	AF12 (DSCP 12)	AF22 (DSCP 20)	AF32 (DSCP 28)	AF42 (DSCP 36)
High Drop	AF13 (DSCP 14)	AF23 (DSCP 22)	AF33 (DSCP 30)	AF43 (DSCP 38)

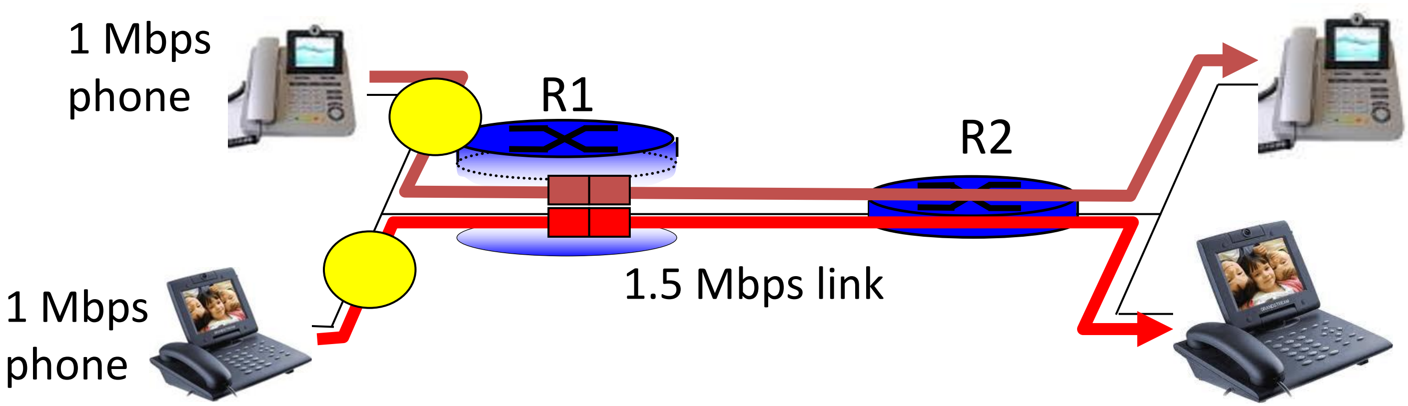
Part B “Media and Real-time Communications; Service-integrated Networks”

B7. QoS Provisioning Based on Reservation

B7.1 A Motivating Example

Principles for QoS Guarantees (more)

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



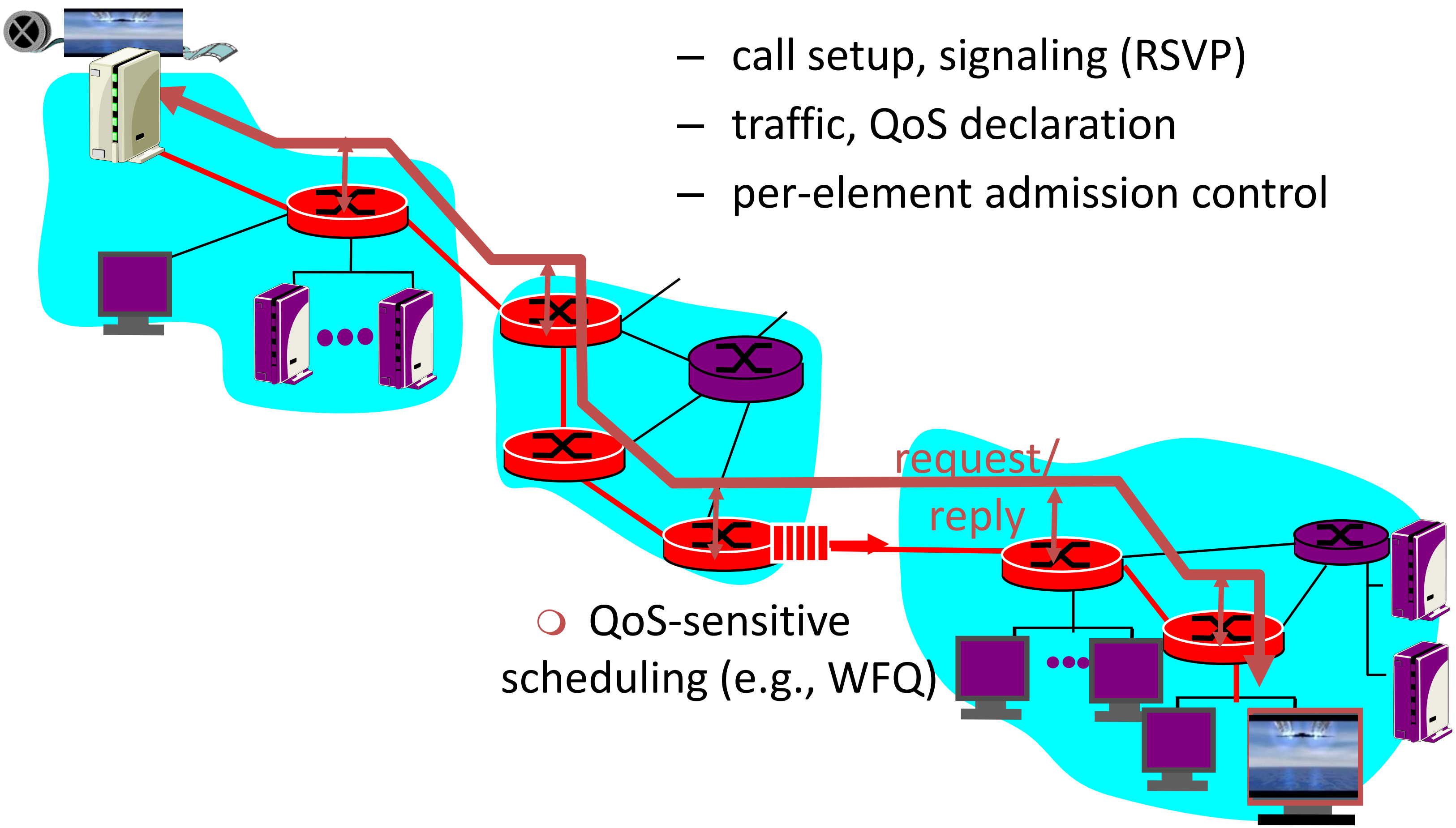
Principle 4

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

B7.2 Resource Reservation, Call Admission, Call Setup

QoS guarantee scenario

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



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

B7.3 Guaranteed QoS in the Internet: IntServ and RSVP

IETF Integrated Services

- architecture for providing QoS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info (à 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 **R**equirement
  - R-spec: defines the QoS being requested
- characterize **T**raffic 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

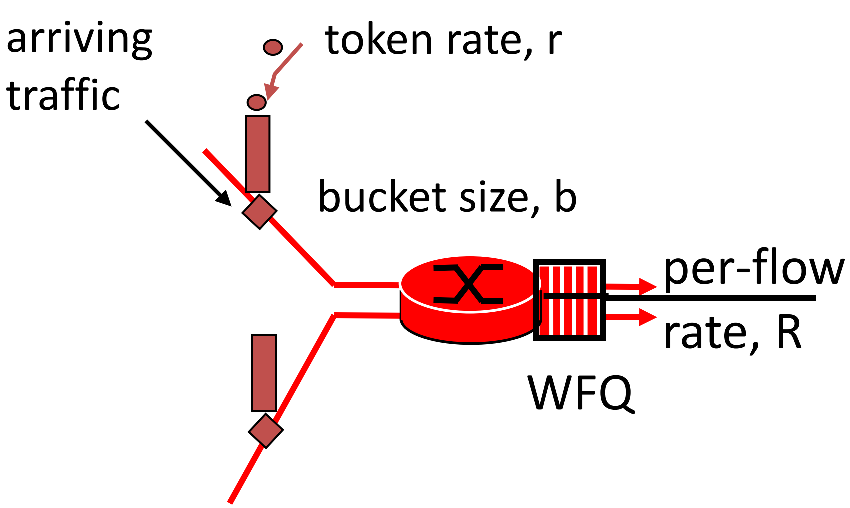
IntServ QoS: Service models [RFC 2211, RFC 2212]

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."



$D_{max} = b/R$

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

Part B1-B7: 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 $\rightarrow$ *DiffServ*
  - QoS guarantees, admission control $\rightarrow$ *IntServ* & *RSVP*

Part B “**Media and Real-time Communications; Service-integrated Networks**”

B1. Multimedia Applications and Resulting Traffic Classes
B2. Quality of Service (QoS): Measures and Assessment Methods
B3. Streaming Stored Audio and Video
B4. Media Communications in Best-Effort Networks
B5. Protocols for Real-Time Interactive Applications
B6. QoS Provisioning Based on Traffic Classes and on Prioritization
B7. QoS Provisioning Based on Reservation

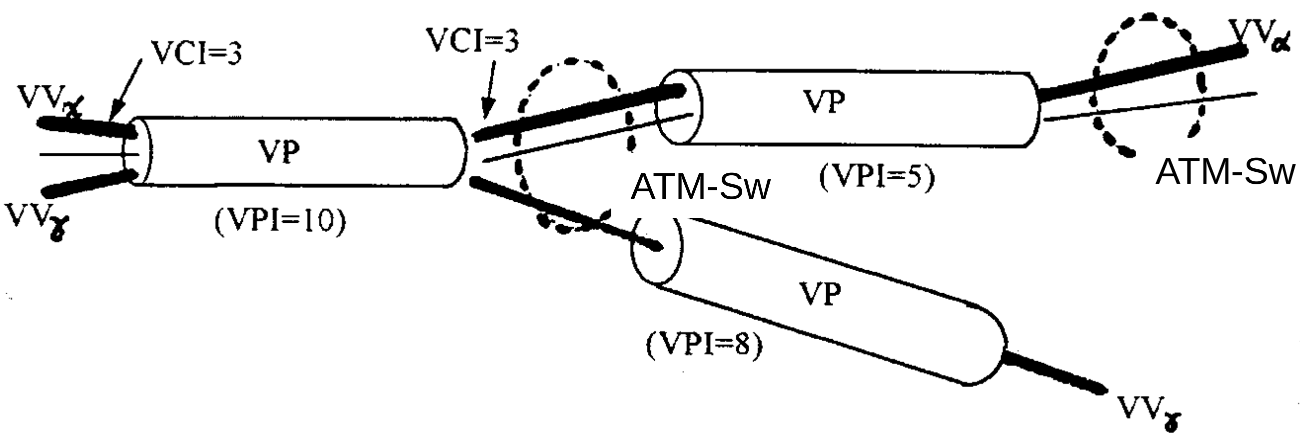
B8. (Service-integrated) Networks with Inherent QoS Guarantees
- B8.1 ATM Protocols and Networks
  - B8.1a Virtual Channels and Virtual Paths
  - B8.1b The ATM Layer
  - B8.1c The ATM Adaptation Layer and Traffic Contracts
- B8.2 QoS Extensions of IEEE 802.11: The 802.11e Standard

B9. Voice and TV Transmissions via the Internet
B10. Case Study: Adaptive QoS Management for Video Communications via Lossy Networks

B8. (Service-integrated) Networks with Inherent QoS Guarantees

B8.1 ATM Protocols and Networks

- ***ATM** (Asynchronous Transfer Mode) technology* : solution standardized by CCITT / ITU to establish B-ISDN; *please note:* standardization supported by ATM forum (non-commercial, > 100 members world-wide)
- *Switching technique*: ***cell switching***
  → variant of packet-switching with fixed-size packets (i.e. the *cells*)
- Exchange of ATM cells via *virtual connections* (VCs), *unlayered VCs*
  → identification of a VC by means of a tuple (VCI, VPI);
  VCI = **v**irtual **c**hannel **i**dentifier; VPI = **v**irtual **p**ath **i**dentifier;
  ATM-Sw = ATM switch



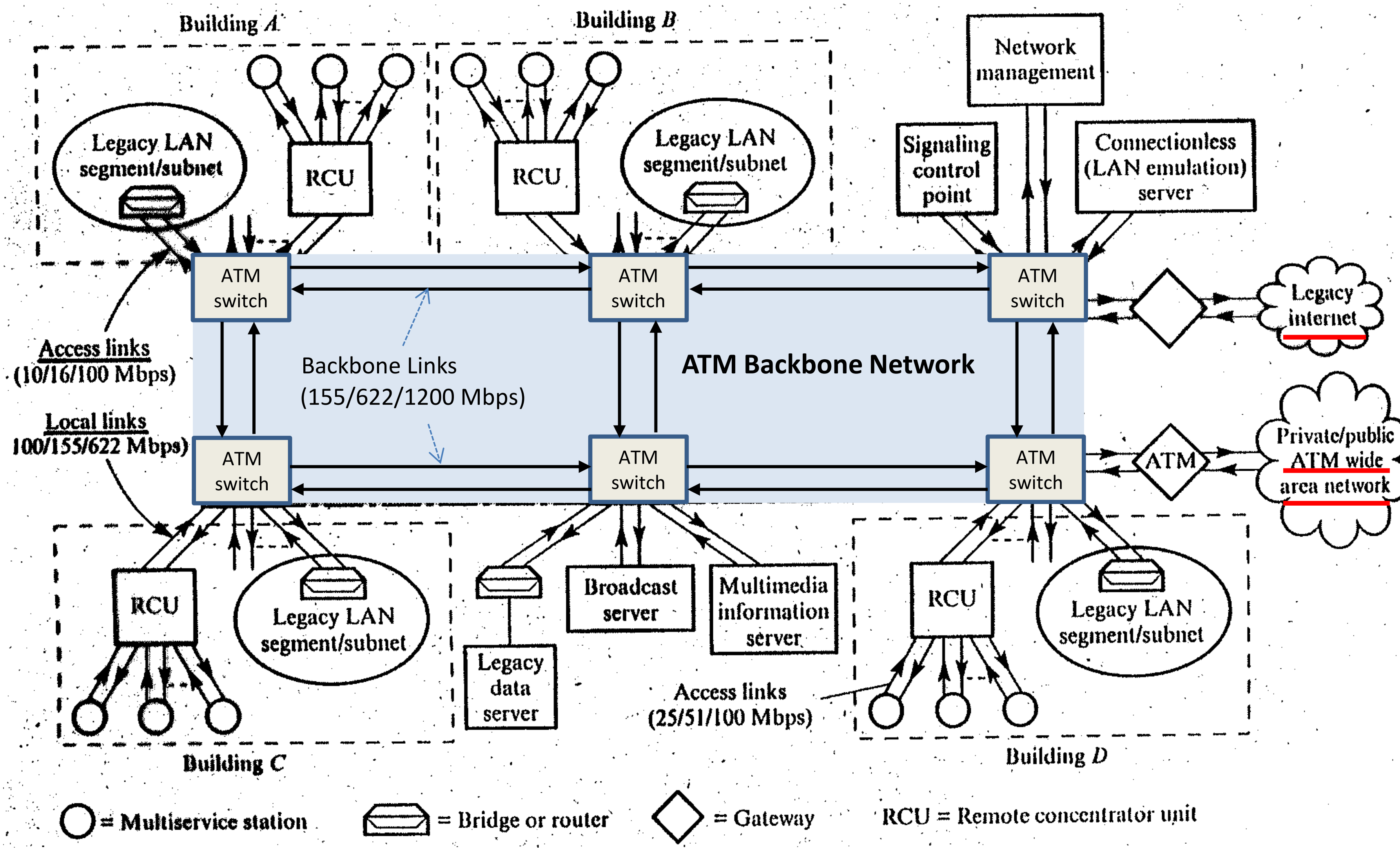
ATM Protocols and Networks, General Characteristics (continued) :

- During connection establishment:
  Specification of QoS requirements by the network user
  $\rightarrow$ network provides QoS guarantees
  (*deterministic* or *stochastic guarantees*)

- ***Examples of QoS guarantees*** by ATM networks:
  -- provisioning of a constant throughput d_0 (constant bit rate, cf. circuit-switching)
  -- cell loss rate $\leq \varepsilon_0 \rightarrow$ to be verified in what intervals ?
  -- time for connection-setup $\leq \tau_0 \rightarrow$ evidently, only with probability $p < 1$,
  etc.
  $\Rightarrow$ QoS guarantees require :
  - reservation of resources (e.g. data transmission & switching capacities)
  - control of user behavior (e.g. rate based flow control for information to be transmitted)

ATM-LAN: Example Configuration (→ non-compulsory/ *'Kür'* !)

(Original expectations, … and then, Ethernet and WLAN came !!!)



ATM Switch Architectures

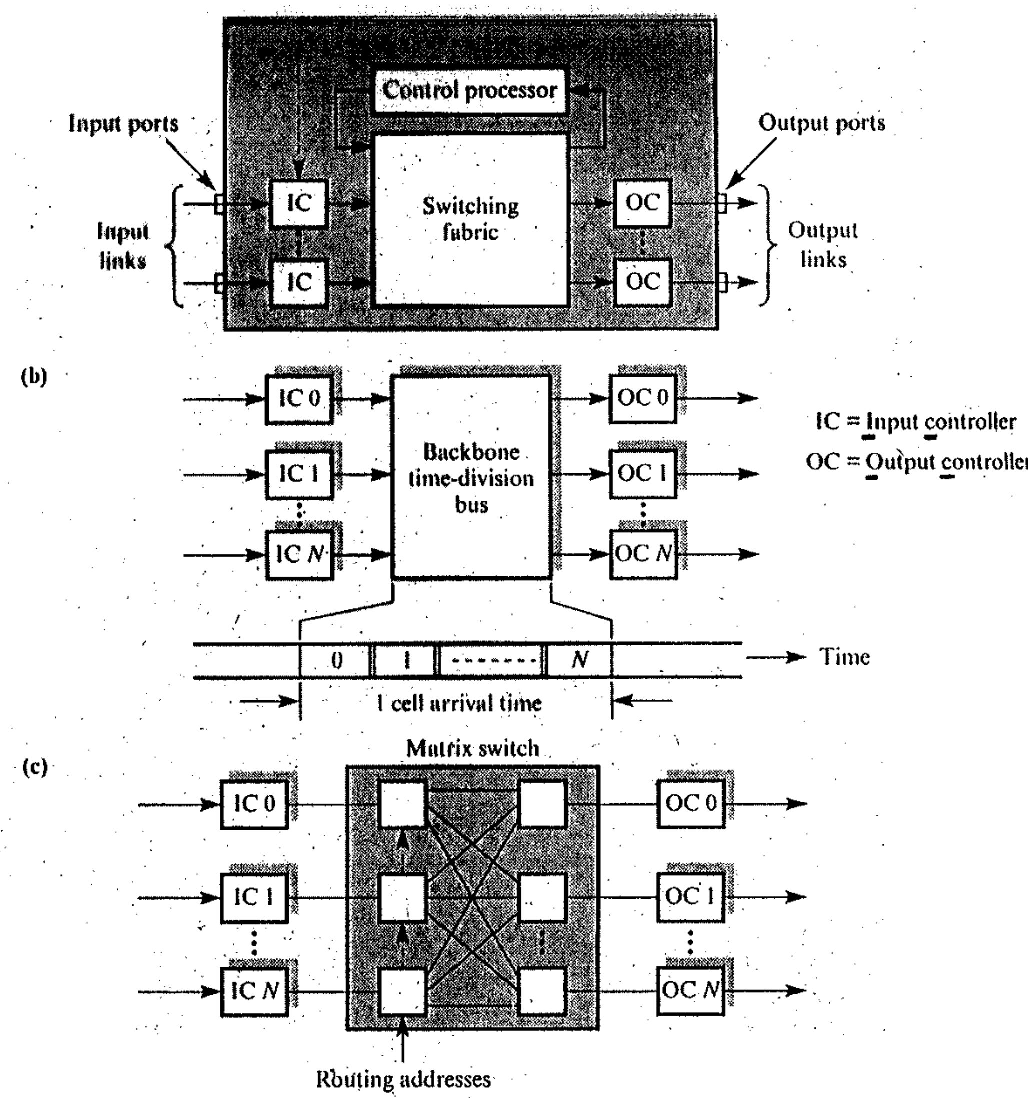
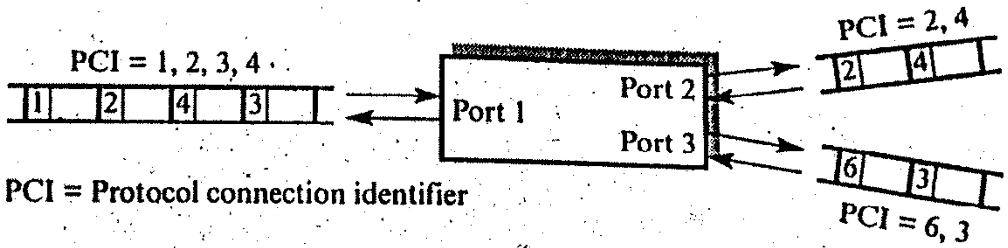


Figure 10.8
ATM switch architectures:
(a) general structure;
(b) time-division bus schematic; (c) fully connected matrix switch.

(from [Halsall 96])

B8.1a Virtual Channels and Virtual Paths

(a)



PCI = Protocol connection identifier

Link/Port 1 RT

In		Out	
Port	PCI	Port	PCI
1	1	2	2
1	2	2	4
1	3	3	3
1	4	3	6
•	•	•	•

Link/Port 2 RT

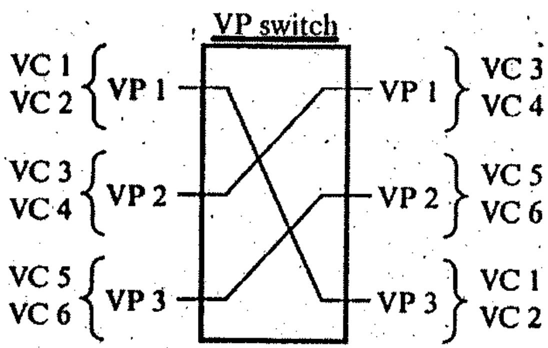
In		Out	
Port	PCI	Port	PCI
•	•	•	•
2	2	1	1
•	•	•	•
2	4	1	2
•	•	•	•

Link/Port 3 RT

In		Out	
Port	PCI	Port	PCI
•	•	•	•
3	3	1	3
•	•	•	•
3	6	1	4
•	•	•	•

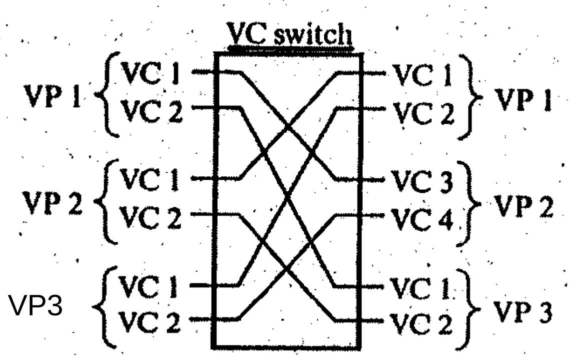
RT = Routing table

(b)



VP = Virtual path

(c)



VC = Virtual channel

Cell switching principles:

(a) Routing schematic

(b) VP routing

(c) VC routing

(from: [Halsall 96])

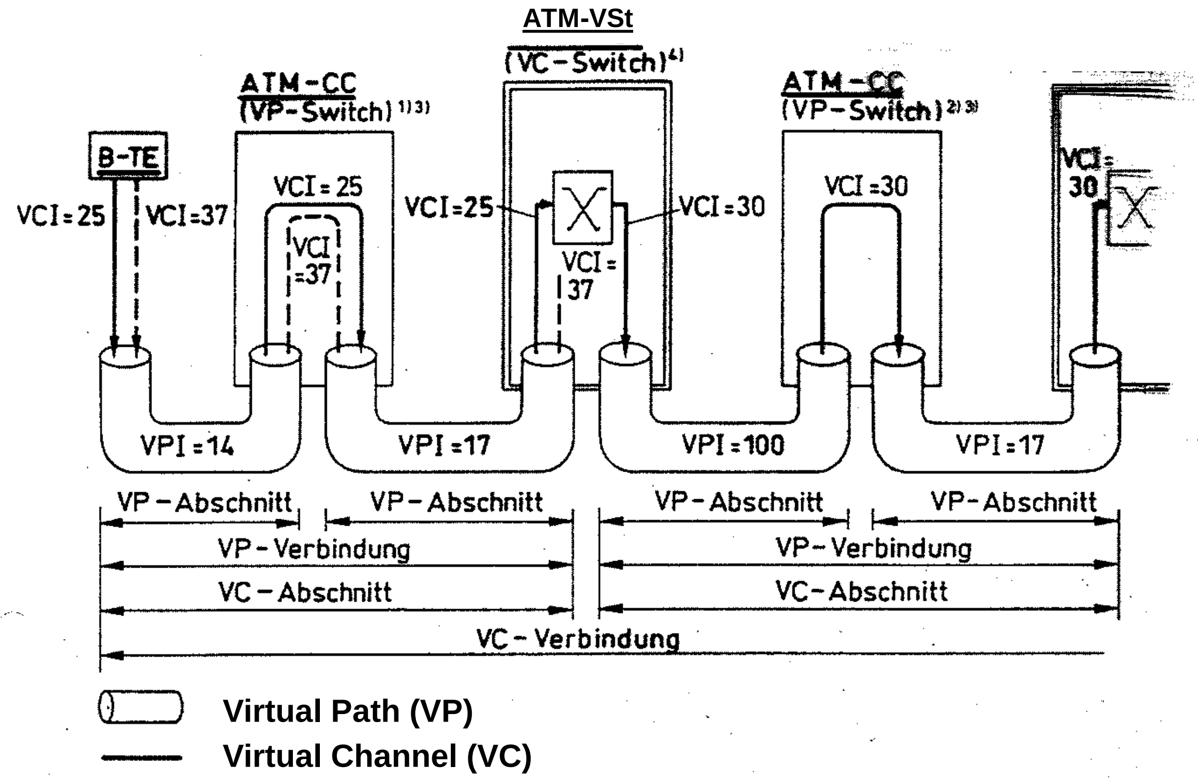


Bild 6.9. ATM-Crossconnect und ATM-Vermittlungsstelle

VCI	virtuelle Kanalkennung (virtual channel identifier)
VPI	virtuelle Pfadkennung (virtual path identifier)
VC	virtueller Kanal (virtual channel)
VP	virtueller Pfad (virtual path)
CC	Crossconnect
VSt	Vermittlungsstelle
B-TE	Breitband-Endeinrichtung
1)	Bestandteil eines Teilnehmeranschlußnetzes
2)	Virtuelles Verbindungsleitungnetz als flexible Netzinfrastruktur
3)	Steuerung per TMN (TMN: Telecommunication Management Network)
4)	Steuerung per Signalisierung

(from [Bocker 97])

Virtual Channels and Virtual Paths in ATM Networks

B8.1b The ATM Layer

Cell Structure 1 :

Zellkopf (Cell Header)

Im Bild unten ist auch der detaillierte Aufbau des Zellkopfes an der Benutzer-Netz-Schnittstelle (UNI) dargestellt. Der Zellkopf besteht aus folgenden Feldern:

- Generic Flow Control (GFC)
- Virtual Path Identifier (VPI)
- Virtual Channel Identifier (VCI)
- Payload Type (PT)
- Cell Loss Priority (CLP) → 1 ≡ niedrige Prio.; 0 ≡ hohe Prio.
- Header Error Control (HEC) → CRC; $P(X) = X^8 + X^2 + X$

<table>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>Bit
Oktett</td></tr>
<tr><td colspan="4">GFC</td><td colspan="4">VPI</td><td>1</td></tr>
<tr><td colspan="4">VPI</td><td colspan="4">VCI</td><td>2</td></tr>
<tr><td colspan="8">VCI</td><td>3</td></tr>
<tr><td colspan="4">VCI</td><td colspan="3">PT</td><td>CLP</td><td>4</td></tr>
<tr><td colspan="8">HEC</td><td>5</td></tr>
<tr><td colspan="8">Informationsfeld</td><td>6
⋮
53</td></tr>
</table>

Header (5 Byte)

“Payload“ (48 Byte)

Structure of a Cell (from [Bocker 97])

Nota bene: **48 Byte** as a **compromise between**
64 Byte (as desired by USA)
and 32 Byte (as desired by Europe)
→ That‘s standardization ! ☺

Cell Structure 2 :

Values of the OAM field "**Payload Type**" (PT) **(→ non-compulsory / *'Kür'* !)**

000	Zelle des Benutzers der ATM-Schicht keine Überlast Indikator zum Informationsaustausch zwischen den Benutzern der ATM-Schicht = 0
001	Zelle des Benutzers der ATM-Schicht keine Überlast Indikator zum Informationsaustausch zwischen den Benutzern der ATM-Schicht = 1
010	Zelle des Benutzers der ATM-Schicht Überlast Indikator zum Informationsaustausch zwischen den Benutzern der ATM-Schicht = 0
011	Zelle des Benutzers der ATM-Schicht Überlast Indikator zum Informationsaustausch zwischen den Benutzern der ATM-Schicht = 1
100	abschnittsweiser OAM-Fluß F5
101	End-zu-End-OAM-Fluß F5
110	Managementzelle für Systemresourcen
111	Reserviert für zukünftige Funktionen

(from [Bocker 97])

Cell Structure 3 :

Standardized Values of the Cell Header (→ non-compulsory / *'Kür'* !)

	GFC	VPI	VCI	PT	CLP
Leerzelle	0000	00000000	00000000 00000000	000	1
OAM-Zelle der physikalischen Schicht	0000	00000000	00000000 00000000	100	1
zukünftige Nutzung der physikalischen Schicht	PPPP	00000000	00000000 00000000	PPP	1
Metasignalisierung	GGGG	XXXXXXXX	00000000 00000001	0A0	C
allgemeiner Rundsendekanal	GGGG	XXXXXXXX	00000000 00000010	0AA	C
Punkt-zu-Punkt-Signalisierungskanal	GGGG	XXXXXXXX	00000000 00000101	0AA	C
abschnittsweiser OAM-Fluß F4	GGGG	XXXXXXXX	00000000 00000011	0A0	A
End-zu-End-OAM-Fluß F4	GGGG	XXXXXXXX	00000000 00000100	0A0	A
abschnittsweiser OAM-Fluß F5	GGGG	XXXXXXXX	ZZZZZZZZ ZZZZZZZZ	100	A
End-zu-End-OAM-Fluß F5	GGGG	XXXXXXXX	ZZZZZZZZ ZZZZZZZZ	101	A
Managementzelle für Systemresourcen	GGGG	XXXXXXXX	ZZZZZZZZ ZZZZZZZZ	110	A
unassigned cell	GGGG	00000000	00000000 00000000	BBB	0

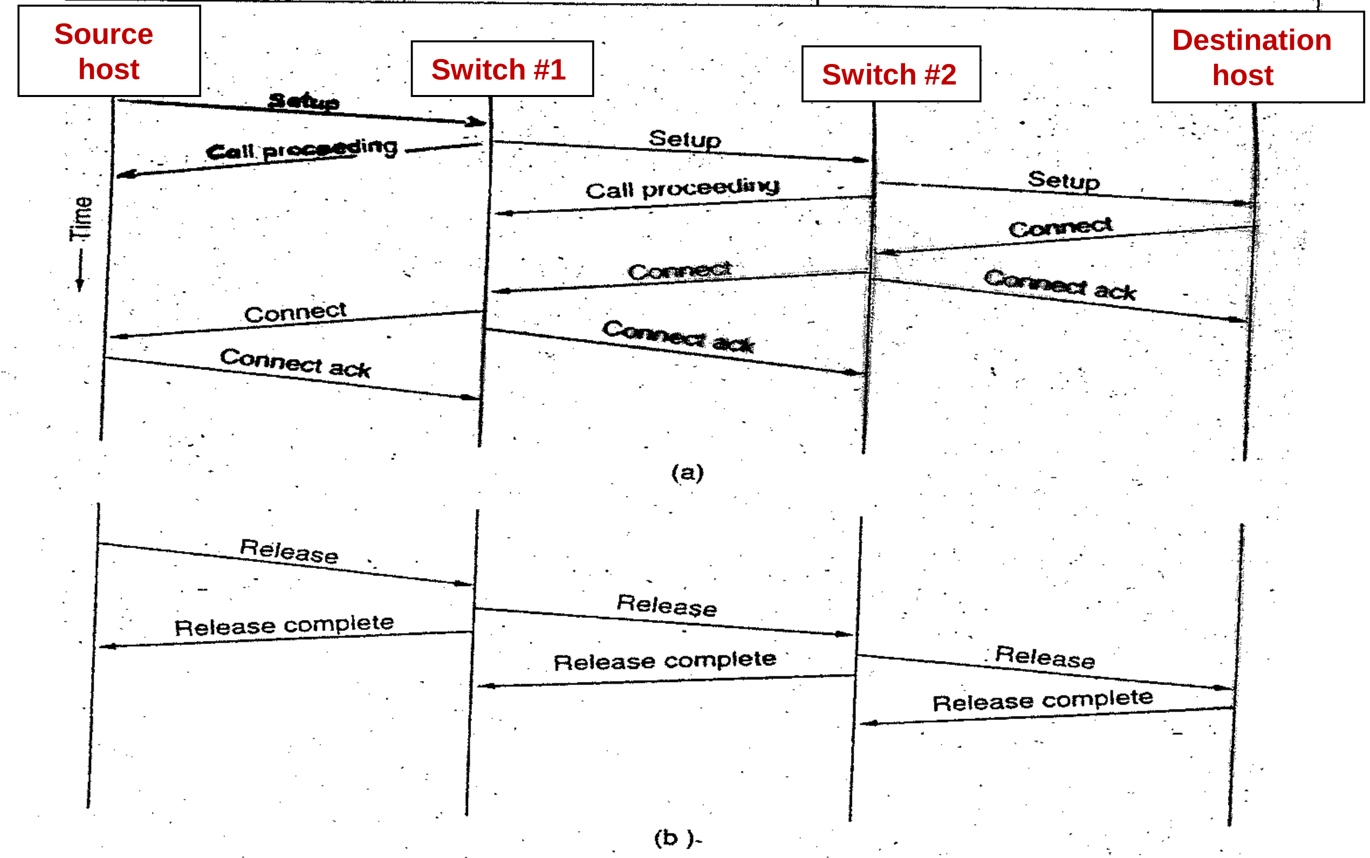
A	0 oder 1; entsprechend der Funktion der ATM-Schicht
B	beliebiger Wert
C	wird von der Quelle 0 gesetzt, kann im Netz geändert werden
G	0 oder 1 entsprechend der GFC-Funktion
P	steht der physikalischen Schicht zur Verfügung
X ... X	beliebiger VPI-Wert
Z ... Z	beliebiger VCI-Wert, außer 0

OAM: operations, administration, and management

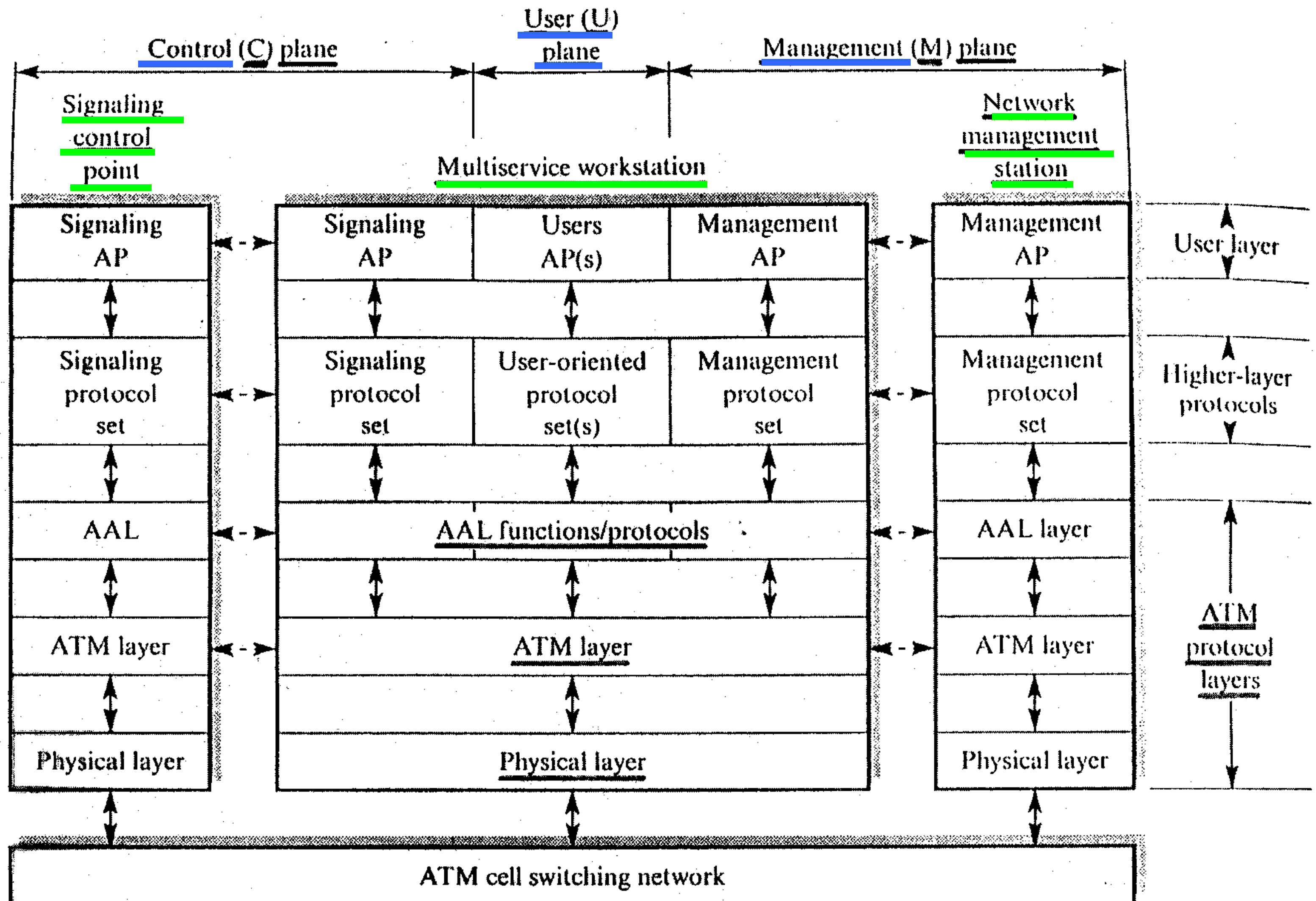
(from [Bocker 97])

Connection Establishment / Release in ATM Networks

Message	Meaning when sent by host	Meaning when sent by network
SETUP	Please establish a circuit	Incoming call
CALL PROCEEDING	I saw the incoming call	Your call request will be attempted
CONNECT	I accept the incoming call	Your call request was accepted
CONNECT ACK	Thanks for accepting	Thanks for making the call
RELEASE	Please terminate the call	The other side has had enough
RELEASE COMPLETE	Ack for RELEASE	Ack for RELEASE



ATM – Protocol Hierarchy



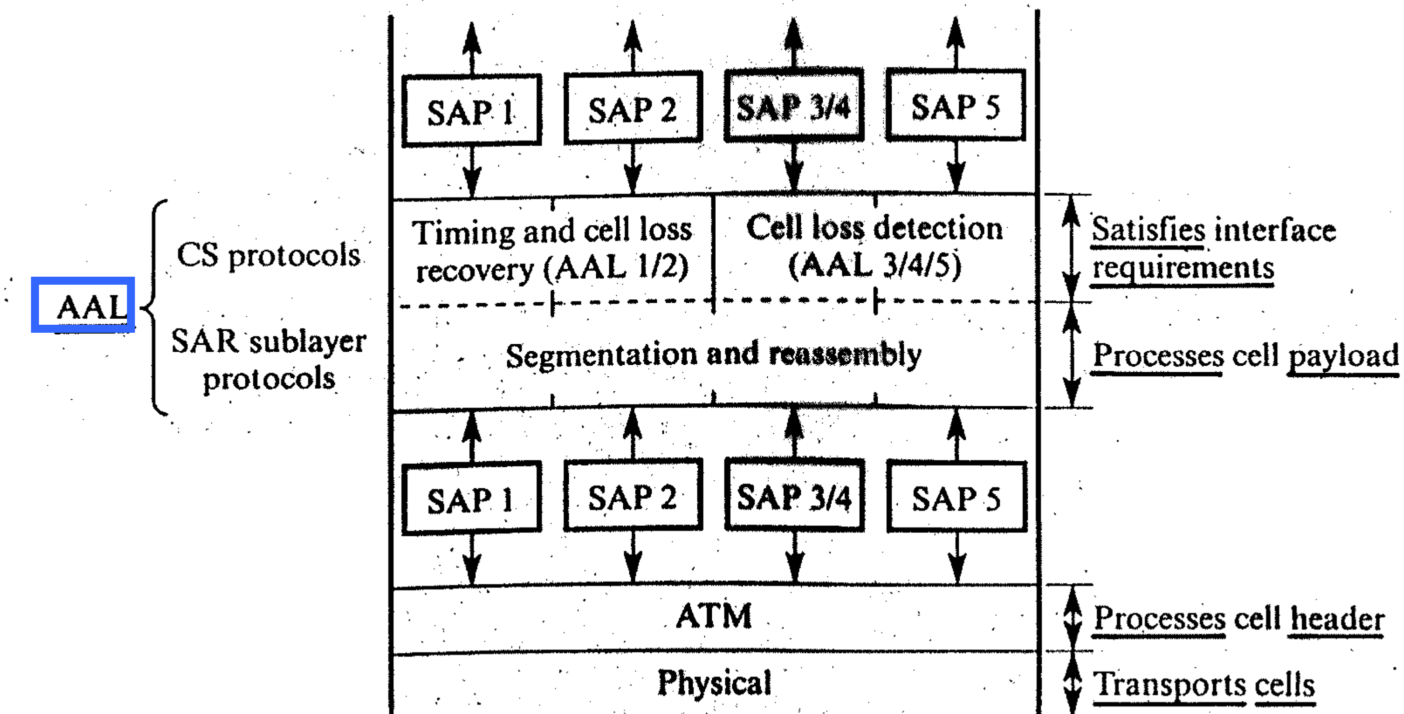
AP = Application process
AAL = ATM adaptation layer
ATM = Asynchronous transfer mode

B8.1c The ATM Adaptation Layer and Traffic Contracts

(a) AAL-Characteristics:

	Service type			
	AAL 1	AAL 2	AAL 3/4	AAL 5
Timing relationship	Yes		No	
Bit rate	Constant CBR	Variable (VBR: variable bit rate)		
Mode	Connection-oriented		Connectionless	

(b)



CS = Convergence sublayer SAR = Segmentation and reassembly

ATM Service Categories and QoS Parameters

Class	Description	Example
CBR	Constant bit rate	T1 circuit
RT-VBR	Variable bit rate: real time	Real-time videoconferencing
NRT-VBR	Variable bit rate: non-real time	Multimedia email
ABR	Available bit rate	Browsing the Web
UBR	Unspecified bit rate	Background file transfer

Fig. 5-69. The ATM service categories.

Service characteristic	CBR	RT-VBR	NRT-VBR	ABR	UBR
Bandwidth guarantee	Yes	Yes	Yes	Optional	No
Suitable for real-time traffic	Yes	Yes	No	No	No
Suitable for bursty traffic	No	No	Yes	Yes	Yes
Feedback about congestion	No	No	No	Yes	No

Fig. 5-70. Characteristics of the ATM service categories.

Parameter	Acronym	Meaning
Peak cell rate	PCR	Maximum rate at which cells will be sent
Sustained cell rate	SCR	The long-term average cell rate
Minimum cell rate	MCR	The minimum acceptable cell rate
Cell delay variation tolerance	CDVT	The maximum acceptable cell jitter
Cell loss ratio	CLR	Fraction of cells lost or delivered too late
Cell transfer delay	CTD	How long delivery takes (mean and maximum)
Cell delay variation	CDV	The variance in cell delivery times
Cell error rate	CER	Fraction of cells delivered without error
Severely-errored cell block ratio	SECBR	Fraction of blocks garbled
Cell misinsertion rate	CMR	Fraction of cells delivered to wrong destination

Fig. 5-71. Some of the quality of service parameters.

B8.2 QoS Extensions of IEEE 802.11: The 802.11e Standard

This topic has been covered in the seminar part of MdNE during the current summer semester

(cf. seminar talks presented)

Part B "**Media and Real-time Communications; Service-integrated Networks**"

B9. Voice and TV Transmissions via the Internet

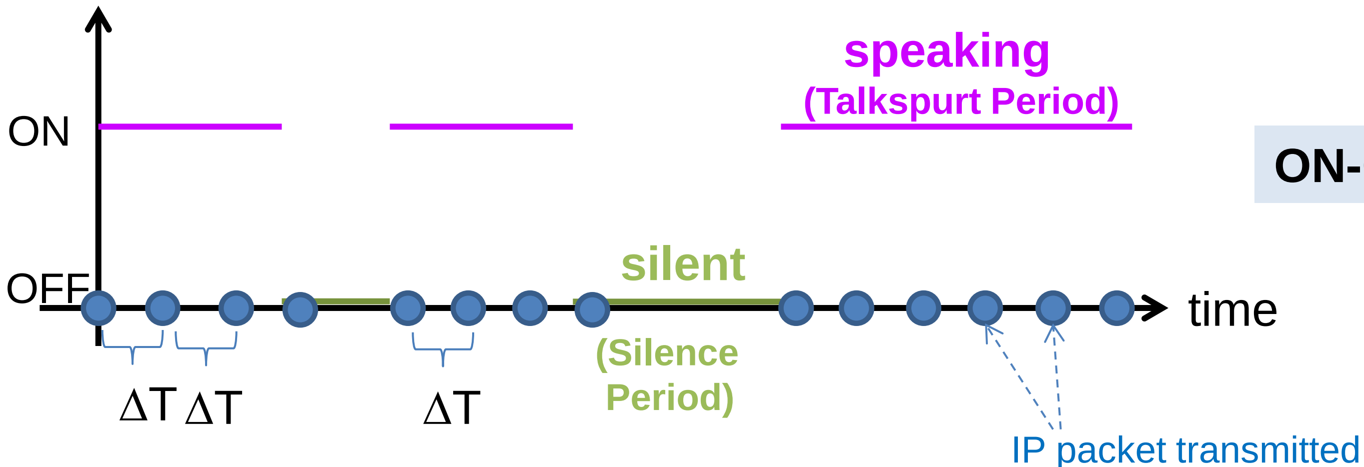
B9.1 Voice over IP (VoIP)

- For encoding use *PCM* (pulse-code modulation), cf. Section B1.4
- For session management (e.g. session establishment & tear-down) and call handling use *SIP* (Session Initiation Protocol); “session“ here: voice connection, cf. Section B5.3
- For transmission of voice data the following protocol stack is typically used, *RTP* (Real-Time Protocol), cf. Section B5.1:

RTP
UDP
IP

- PCM-encoded voice data handed over to RTP for transmission to correspondent (dialog partner at phone); decoding at receiver; cf. also remarks w.r.t. *Internet Phone* (beginning of B4)

Talkspurt-Silence Model for Voice



ON-OFF-Model

Data rate in case of silence compression:

$\Delta T = n \bullet \Delta t$, where $\Delta t = 125\ \mu s$ (i.e. time between 2 subsequent PCM samples)
n = const. (number of PCM samples per IP packet), e.g. n=160, i.e. $\Delta T = 20$ ms

Therefore, length L(n) of an IP packet (in Byte) with its n PCM samples:
L(n) = n [Byte] + HL [Byte] , where HL = header-length (in Byte)

Thus, during a talkspurt period, we get a CBR data rate of
d(n) = 8 [kByte/s] + 8000/n • HL [Byte/s] = (64 + 64/n • HL) [kbit/s]

Note: *large n* → losses more serious & less real-time quality of VoIP service
small n → more overhead (resulting from header transmission)

Voice-over-IP (VoIP)

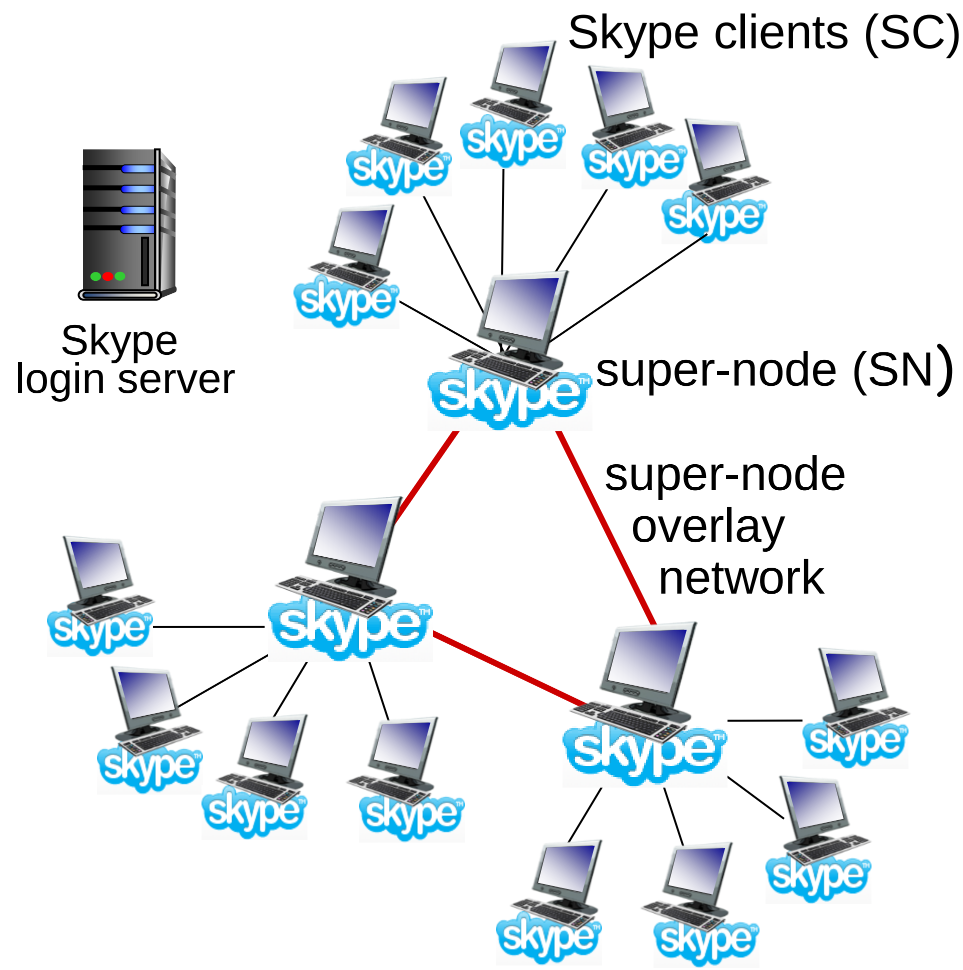
- *VoIP end-end-delay requirement*: needed to maintain "conversational" aspect
  - higher delays noticeable, impair interactivity
  - < 150 msec: good
  - > 400 msec bad
  - includes application-level (packetization, playout), network delays
- *session initialization:* how does callee advertise IP address, port number, encoding algorithms?
- *value-added services:* call forwarding, screening, recording
- *emergency services:* 911

VoIP: packet loss, 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, loss concealment, packet loss rates between 1% and 10% can be tolerated

Voice-over-IP: Skype

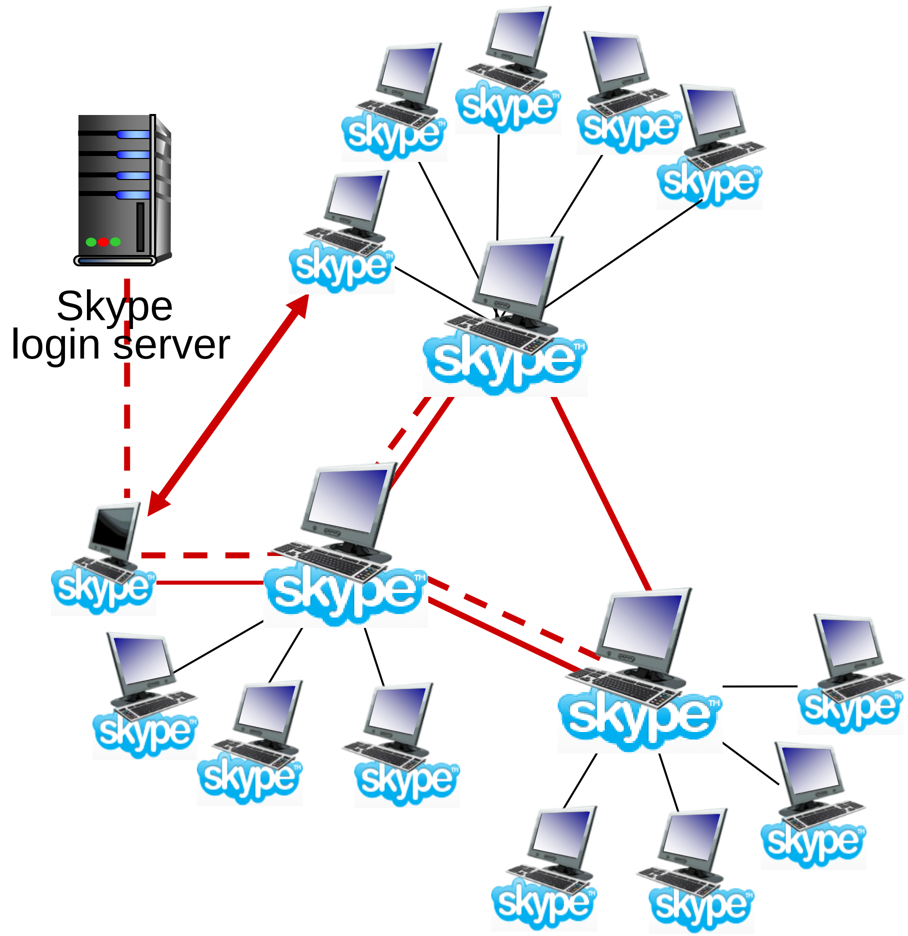
- proprietary application-layer protocol (inferred via reverse engineering)
  - encrypted msgs
- P2P components:
  - clients: Skype peers connect directly to each other for VoIP call
  - super-nodes (SN): Skype peers with special functions
  - overlay network: among SNs to locate SCs
  - login server



P2P voice-over-IP: Skype

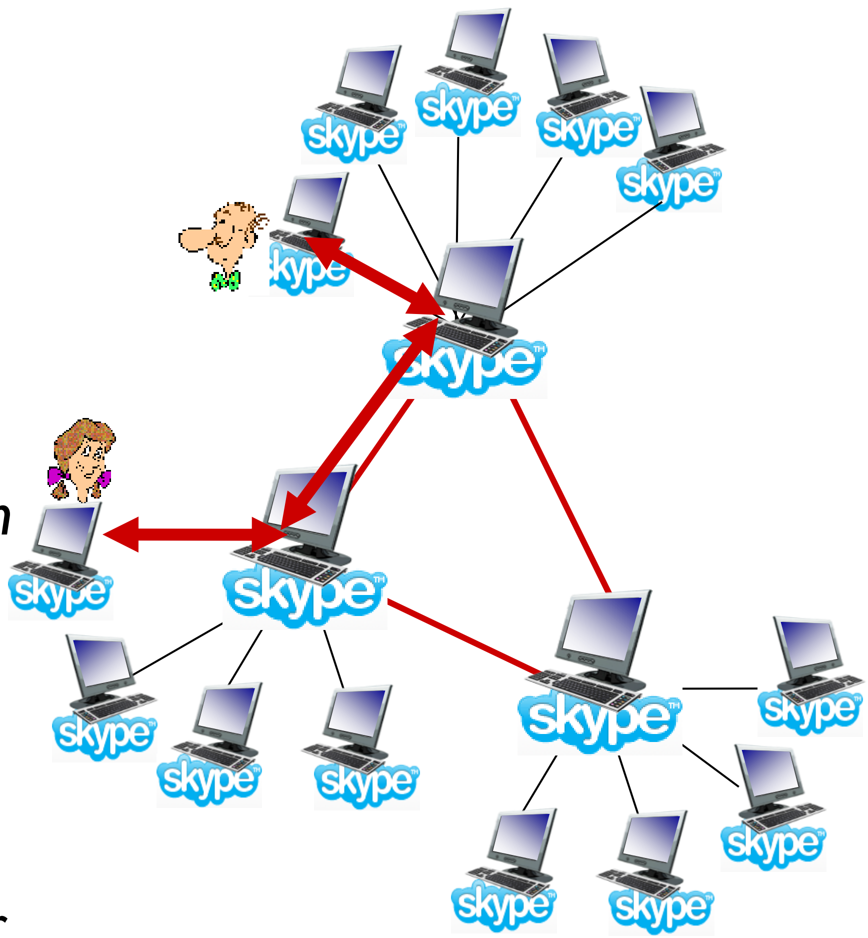
Skype client operation:

1. joins Skype network by contacting SN (IP address cached) using TCP
2. logs-in (user name, password) to centralized Skype login server
3. obtains IP address for callee from SN, SN overlay
   - or client buddy list
4. initiate call directly to callee



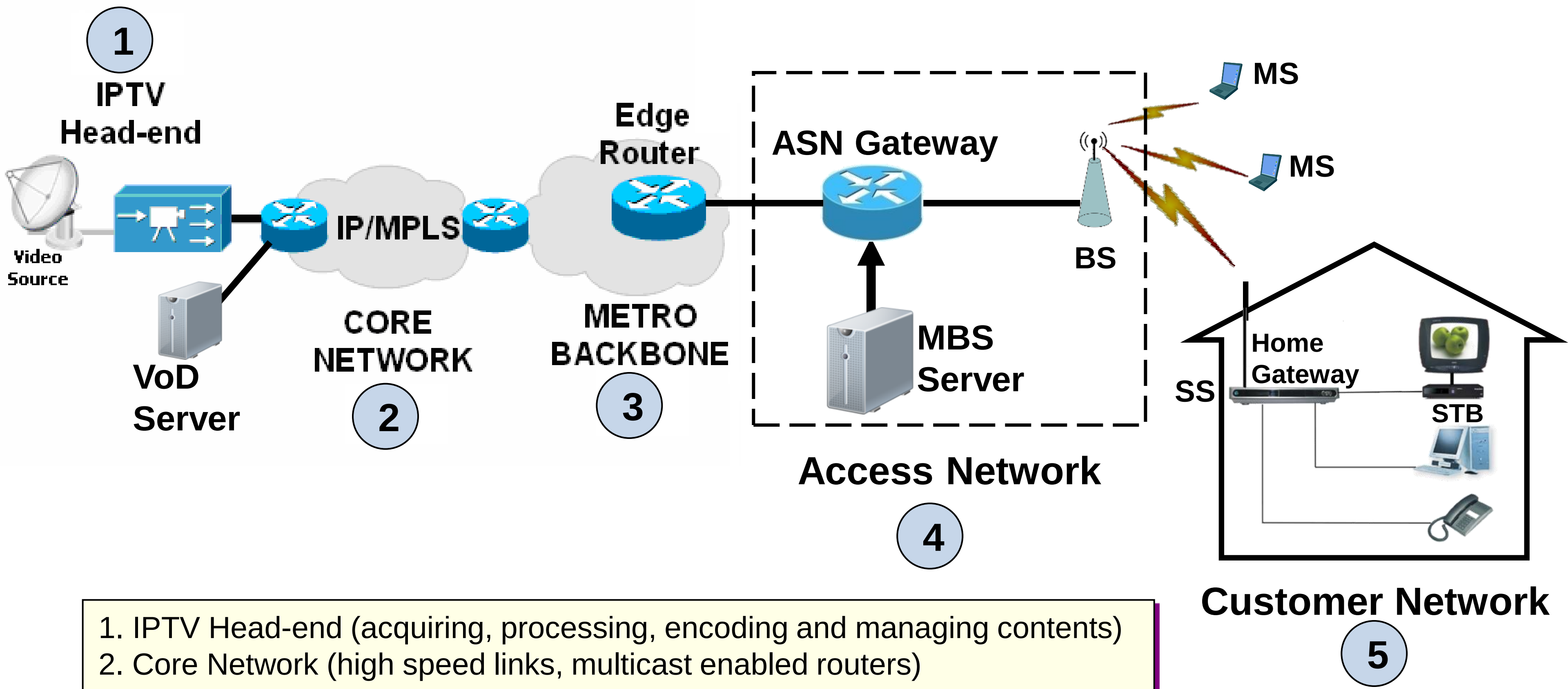
Skype: peers as relays

- *problem:* both Alice, Bob are behind "NATs"
  - NAT prevents outside peer from initiating connection to insider peer
  - inside peer *can* initiate connection to outside
- *relay solution: Alice, Bob maintain open connection to their SNs (super-nodes)*
  - Alice signals her SN to connect to Bob
  - Alice's SN connects to Bob's SN
  - Bob's SN connects to Bob over open connection Bob initially initiated to his SN



B9.2 IPTV: Systems and Protocols

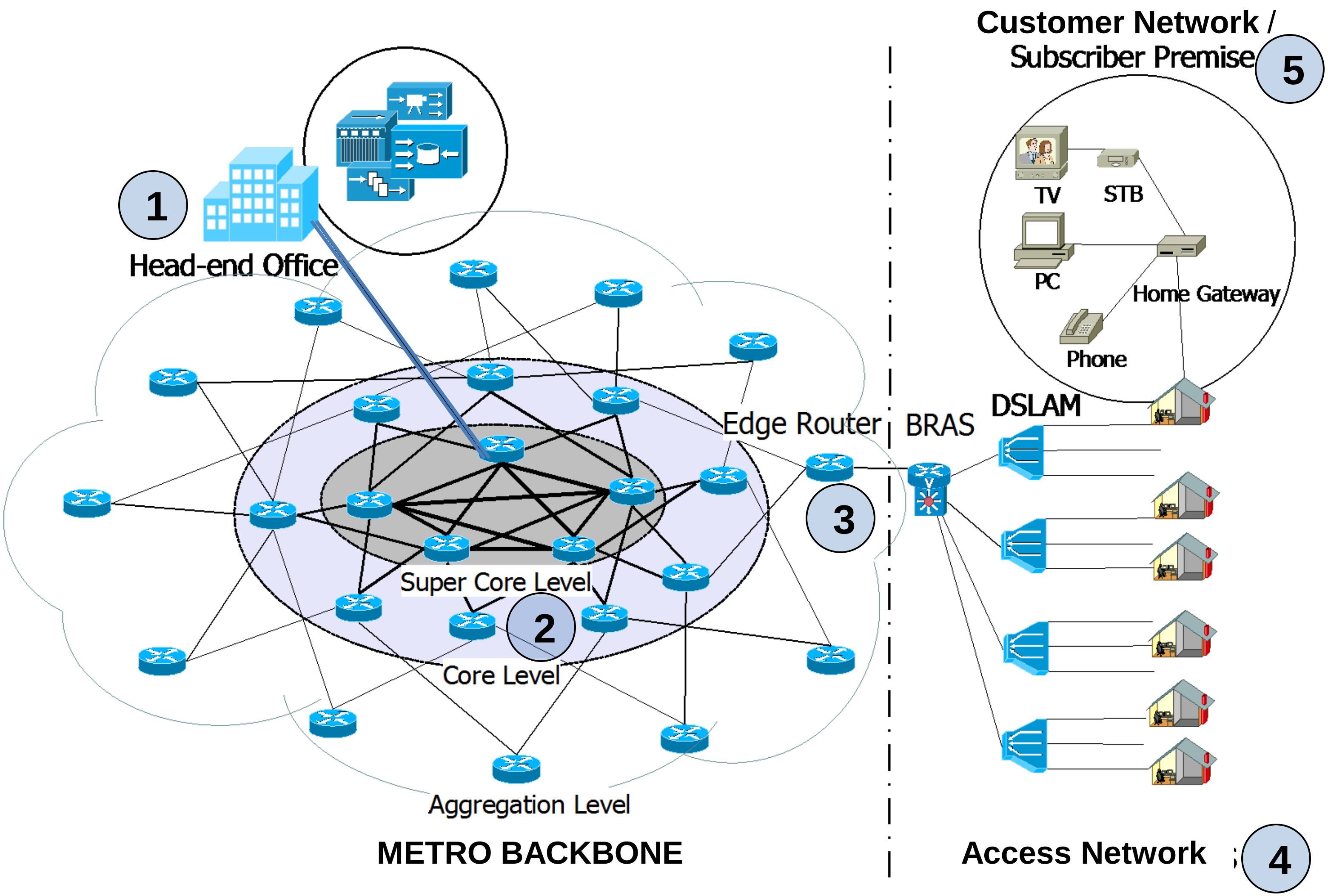
IPTV System Structure :
WiMAX based Access Network



1. IPTV Head-end (acquiring, processing, encoding and managing contents)
2. Core Network (high speed links, multicast enabled routers)
3. Distribution Network / Metro Backbone (local content addition)
4. Access Network (provides “last mile access”)
5. Customer Network (provides TV, IP phone and Internet services)

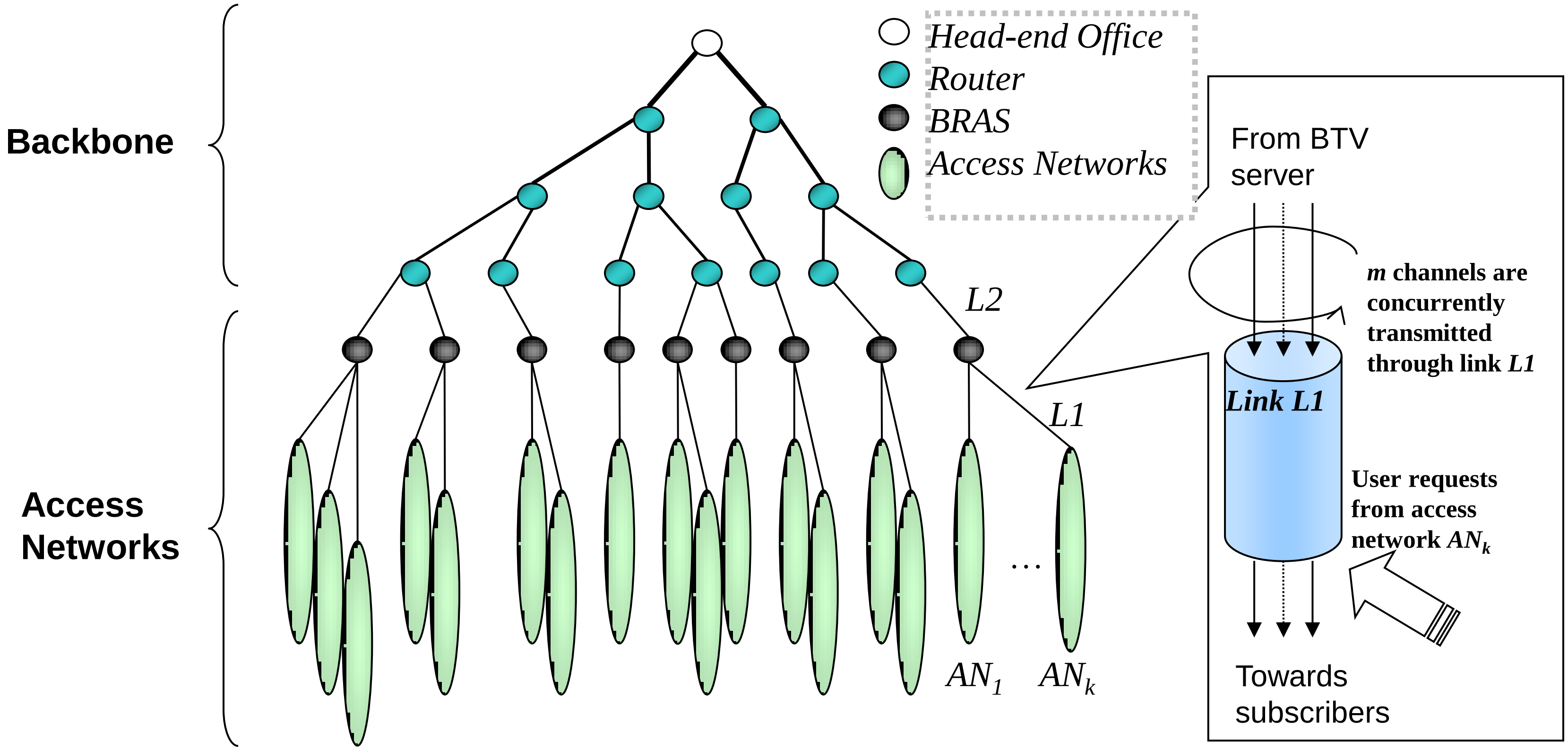
Abbreviations: ASN: Access Service Network; MBS: Multicast & Broadcast Service (with WiMAX); MPLS: Multi-Protocol Label Switching; SS: Subscriber Station; STB: Set-Top-Box

IPTV System Structure :
DSL based Access Network



Abbreviations: BRAS: Broadband Remote Access Server; DSLAM: DSL Access Multiplexer

Multicast Tree Topology of a Broadband TV (BTV) Distribution Network Architecture



Potential Bottleneck (PB)-link:
Link on which blocking of user requests may occur

Requirements to IPTV Services and Measures for their Quality

- Requirements of IPTV users:

  Get same quality as in conventional TV broadcast systems, e.g.

  - *R1:* get <u>all</u> channels delivered upon request
  - *R2:* get (at least) comparable audio/video quality
  - *R3:* quick switching between different channels demanded by a single user

- Quality measures for

  - *R1:* **TV Channel Blocking Probability** (CBP), i.e. *probability that a desired channel cannot be provided*
  - *R2:* **Quality of Experience** (QoE) Measures such as Mean Opinion Score (MOS)
  - *R3:* **Channel Switching Delay**

Solutions for the Provisioning of IPTV Services

- ***Fixed reservation of link bandwidth*** for the IPTV service in the network of the ISP (Internet Service Provider) who provides the Triple (or Quadrupel) Play Service, i.e. data, voice, TV (and mobile) communication services → QoS/QoE can be assured

- Deliver TV channels to the customers/users using ***IP multicast*** for channels being currently required by > 1 user (if only 1 user watches a given unpopular channel then unicast might be the better choice); the multicast trees resulting for the different TV channels may have to be changed dynamically → ***LEAVE & JOIN MULTICAST GROUP operations*** issued by the TV viewers

- Mechanisms are applied to ***shorten the switching delay*** occuring when changing the TV channels

Future Research Challenges in IPTV

Trends to be expected:

- Strong changes in the way future IPTV services will be offered $\rightarrow$ e.g. more often non-real-time (TV on Demand), feed-back channel for viewers, …
  $\Rightarrow$ user behavior will strongly change, too !
- Speed and throughput of future (IP based) networks will continue to strongly increase, BUT: networks also much more heavily loaded (e.g., by video traffic)
- “Anytime & anywhere access” will be demanded for IPTV services

Resulting research challenges, e.g.:

- ✓ Which TV channels should be multicasted and which ones unicasted ?
- ✓ Analyses of bottlenecks and new mechanisms for their avoidance (in particular, within access networks)
- ✓ New user behavior models required
- ✓ IPTV to be provided for highly different end-systems
- ✓ Security problems (e.g. sniffing of pay-TV channels), privacy of end-users ?
- ✓ How to tread heavy zappers? ($\rightarrow$ distr. denial of service attacks)

Part B "**Media and Real-time Communications; Service-integrated Networks**"

- B1. Multimedia Applications and Resulting Traffic Classes
- B2. Quality of Service (QoS): Measures and Assessment Methods
- B3. Streaming Stored Audio and Video
- B4. Media Communications in Best-Effort Networks
- B5. Protocols for Real-Time Interactive Applications
- B6. QoS Provisioning Based on Traffic Classes and on Prioritization
- B7. QoS Provisioning Based on Reservation
- B8. (Service-integrated) Networks with Inherent QoS Guarantees
- B9. Voice and TV Transmissions via the Internet
- B10. Case Study: Adaptive QoS Management for Video Communications via Lossy Networks
  - B10.1 The Problem to be Solved
  - B10.2 Methods for Adaptive QoS Management in Video Communications and Assessment of Quality Improvement

B10. Case Study: Adaptive QoS Management for Video Communications via Lossy Networks

B10.1 The Problem to be Solved

Motivation

High demand for video communications,

e.g., picture-phone, video-conferencing ...

however: significant parts of nowadays network infrastructures not designed for this type of traffic and therefore no adequate basis for supporting audio/video transmissions,

e.g., because of real-time requirements.

Areas for Improvement of Video Communications:

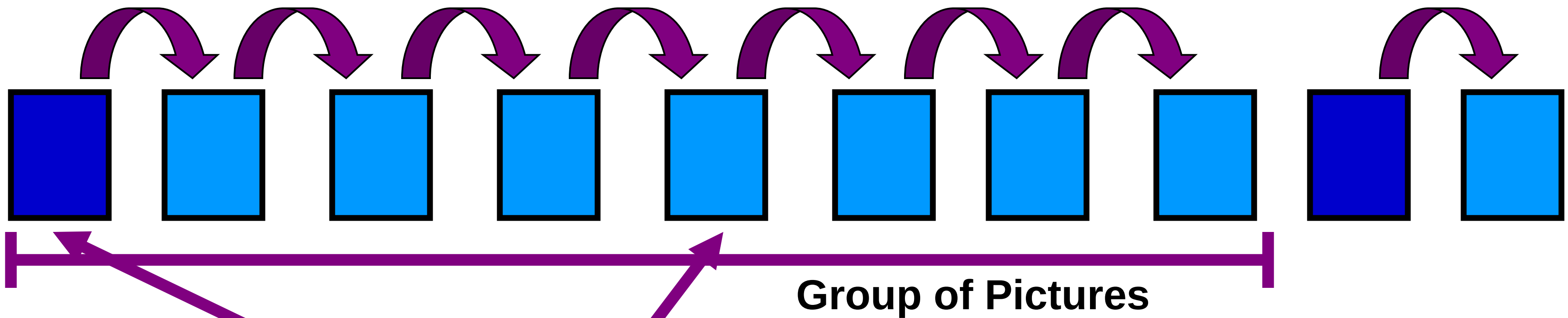
- Network-internal changes (IntServ, DiffServ, ...) $\rightarrow$ cf. Chapters B6. and B7.
- **Network-external** (application-oriented fault-tolerance) on

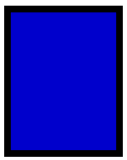
  -- *Sender* side (FEC – Forward Error Correction $\rightarrow$ static *or* dynamic; ECC – Error Correction Code, I-Frame distance)

  -- *Receiver* side

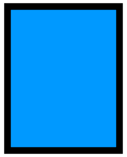
Structuring of Video Sequences (Illustration)

Sequence of Pictures





Intra-coded Pictures (**I-Pictures** for short) contain complete information needed for their decoding.



Inter-coded Pictures (**P-Pictures** for short) encoded as a difference to the I- resp. P-Picture immediately preceding.

Dependency !

B10.2 Methods for Adaptive QoS Management in Video Communications and Assessment of Quality Improvement

A. Mechanisms for Fault-tolerance of the Sender

I-Pictures do represent **Recovery/Restart Points** in case of errors/losses during real-time transmissions of video sequences.

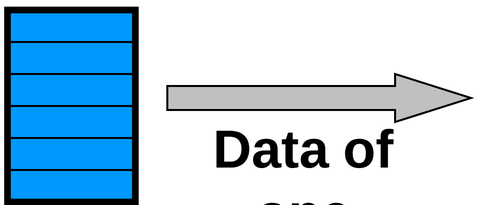
The sender does influence the fault-tolerance (redundancy) by means of **Changes in the distance between I-Pictures**.

Goal:

Keep the estimated **perceived accumulated interference** ζ **per Group of Picture** (GOP) below a given threshold $\zeta_{\mathbf{ref}}$.

Therefore, we need a ***Model*** **to calculate the user-perceived accumulated interference** per GOP.

Model for losses and their impact



Data of one picture

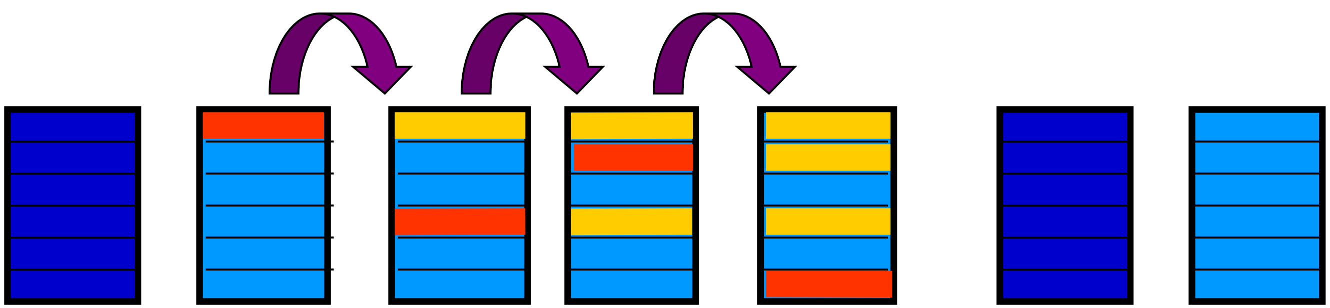
a data packets lost per group of blocks

Communication System

Measure for percentage of errors per picture (if *a* packets are directly lost within the *c* packets transmitted per picture):

$\varepsilon = a / c$

Indirect Losses



Losses per Picture: ε 2ε 3ε 4ε

Accumulation of losses per GOP: $\zeta = \varepsilon \, n(n+1)/2$

Control of Distance between I-Frames

Goal:

Keep the estimated **perceived accumulated interference** ζ **per GOP** under a given threshold $\zeta_{\mathbf{ref}}$.

Model:

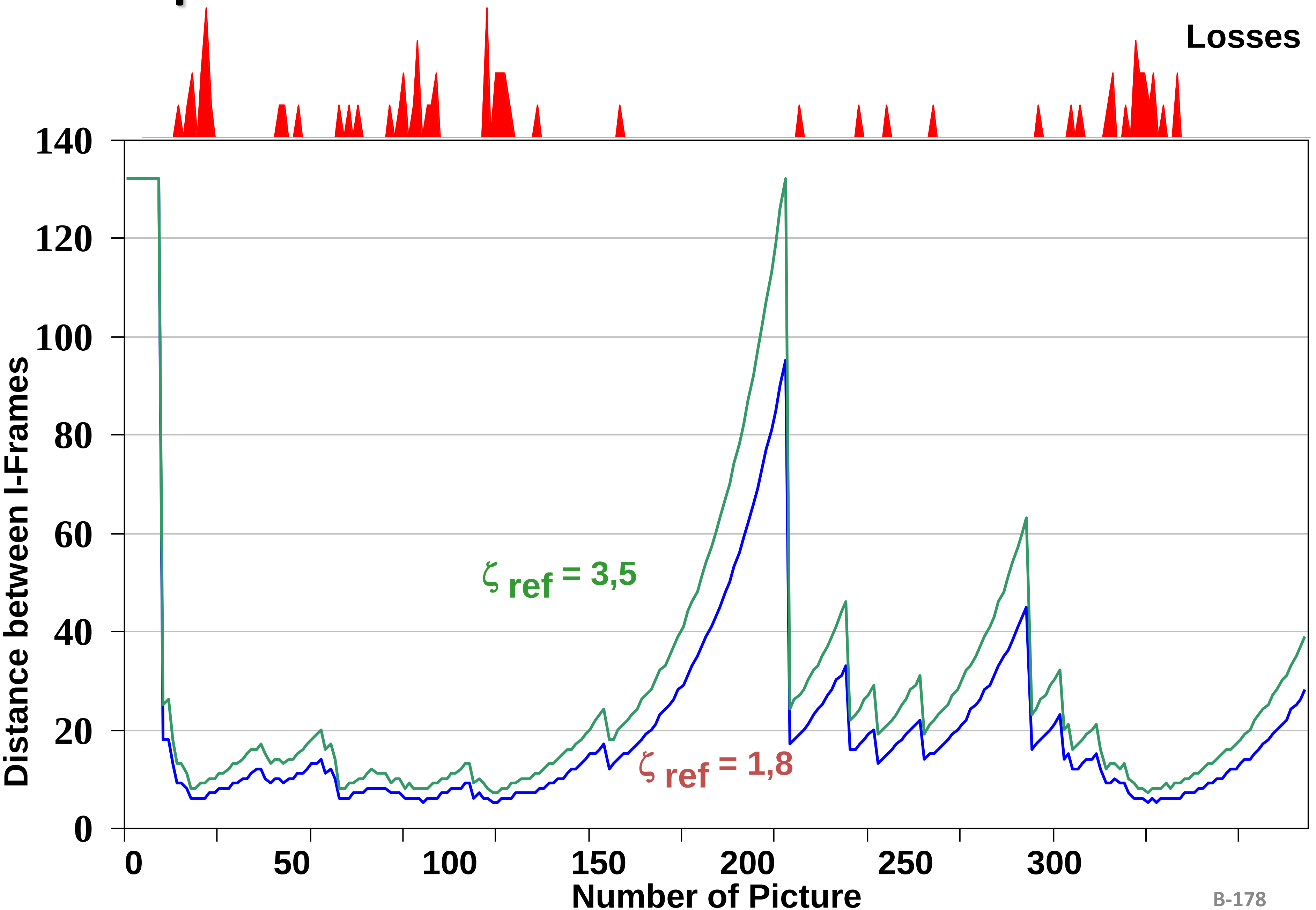
$\zeta = \varepsilon\, n(n+1)/2$, $\zeta_{\mathbf{ref}} \geq \varepsilon\, n_{\mathbf{max}} (n_{\mathbf{max}}+1)/2$

Parameters for control:

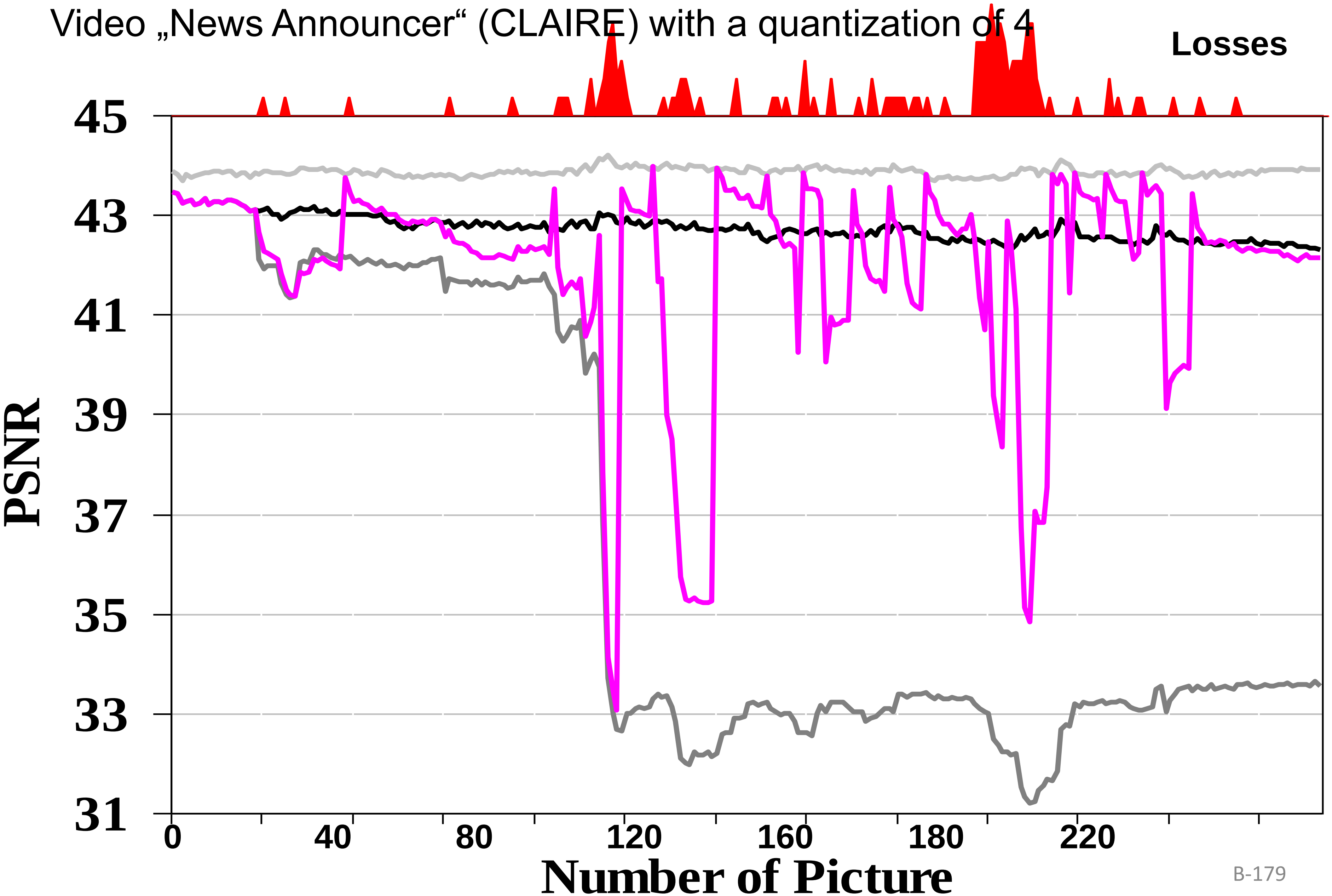
Thus, maximum distance between **I-Frames** is given by:

$\mathbf{n_{max} = min\{}\lfloor\sqrt{(1/4)+2\,\zeta_{\mathbf{ref}}\,/\varepsilon}\rfloor, 132\}.$

Example for Control of Distance between I-Frames



Comparison of Video Quality



Picture-quality: Fault-tolerance by means of mechanisms at the *Sender*

5 % packet losses,

no fault-tolerance

119 341

5 % losses, usage of our mechanism to achieve fault-tolerance

B. Mechanisms for Fault-tolerance of the Receiver

Without fault-tolerance of the receiver:

Stop of Decoder !

Fault-tolerance at the receiver:

- Insertion of monochrome blocks to replace erroneous areas
- **Replay (Repetition)** of preceding picture information
- Use of **unidirectional** motion vectors (**TCON**)
- **Median** of **three** neighbouring motion vectors (**MeBeV**)

Picture-quality: Fault-tolerance by means of mechanisms at the *Receiver*

Picture 247

Without Loss

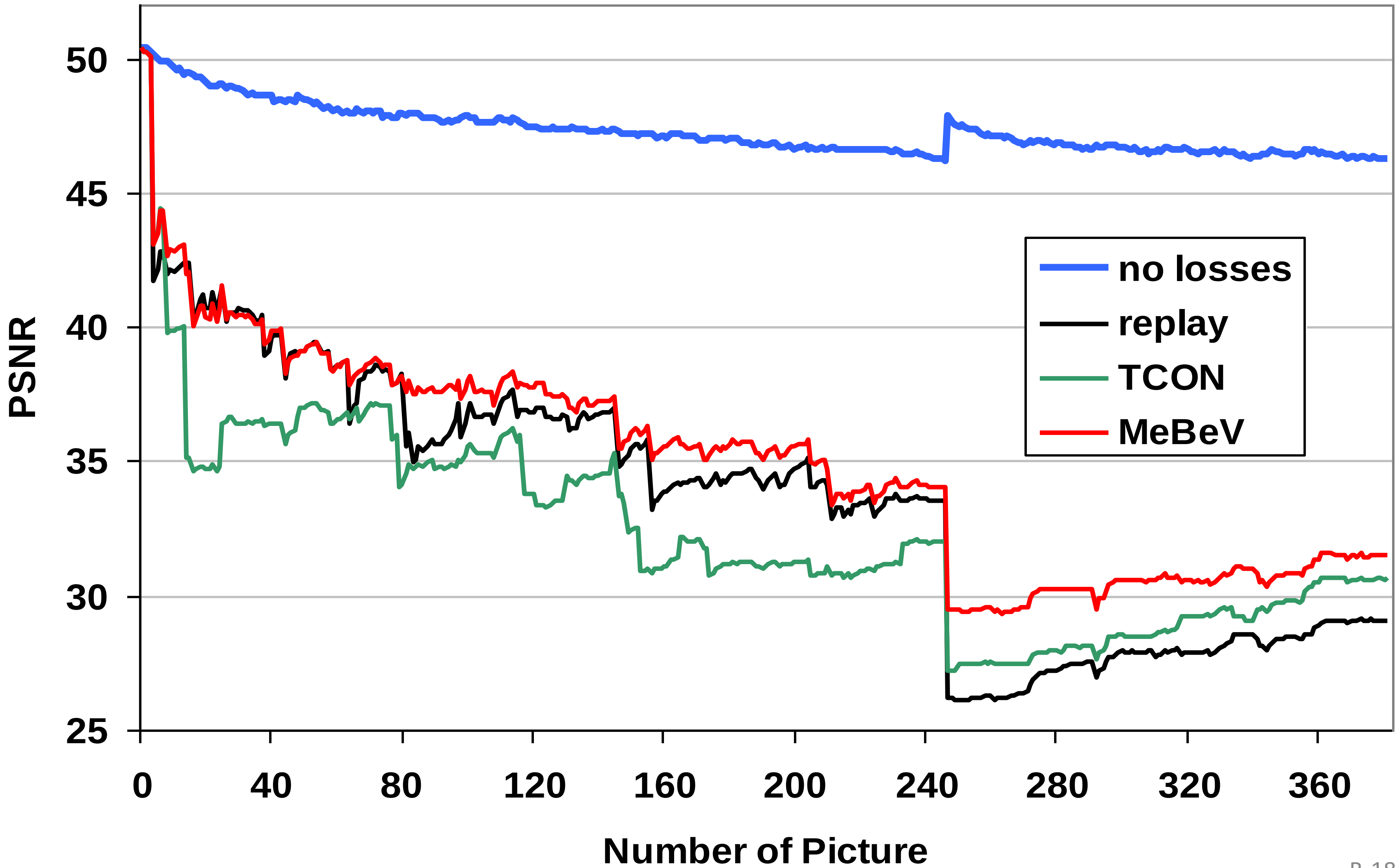
5% Losses

Replay

TCON

MeBeV

Comparison of Video Quality using different Techniques for Fault-tolerance at the Receiver



Summary

- Construction of a model to control fault-tolerance in video communications by consideration of
  - losses during transmissions (direct losses) and
  - their negative impact on the process of decoding (indirect losses)

- Development of a model-based mechanism for fault-tolerance for the sender by using this kind of loss modeling

- Realization of fault-tolerance at the receiver

- Evaluation and comparison of mechanisms for fault-tolerance by means of PSNR-based measures of picture quality

- Demonstration of the quality improvements thanks to fault-tolerance (experiments with a large variety of video sequences)

Lessons Learned

- The model-based mechanisms for fault-tolerance at the sender allows for very good amelioration of quality.

- The price to be paid: additional I-frames, i.e. more data.

- The mechanisms for fault-tolerance at the receiver are only able to mask losses up to some quite limited level.

- Thus, the resulting areas of application basically are different: connections with high resp. low percentage of losses.

- The mechanisms for fault-tolerance at receiver are able to ameliorate the sender‘s activities if mechanisms are combined.

Outlook

- Measurement of error statistics in relevant communication systems (direct losses)
- Further investigations to model error propagation (indirect losses)
- Usage of resulting model variants for control of mechanisms to achieve fault-tolerance on sending site
- Reduction of amount of data needed for additional I-Frames, e.g., by means of a refined tuning of the quantization
- Combination of sender- and receiver-based mechanisms
- Investigation of the quality of the mechanisms developed by means of using them in simulated as well as in existing communication systems