PI-Grau
(Internet Protocols)

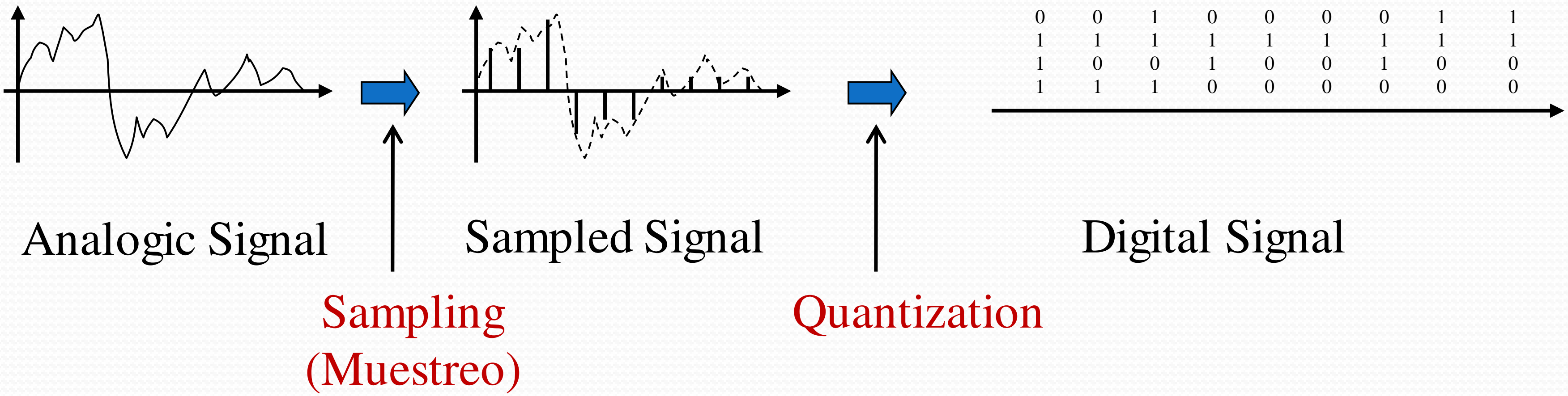
José M. Barceló Ordinas
Departamento de Arquitectura de Computadores
(UPC)

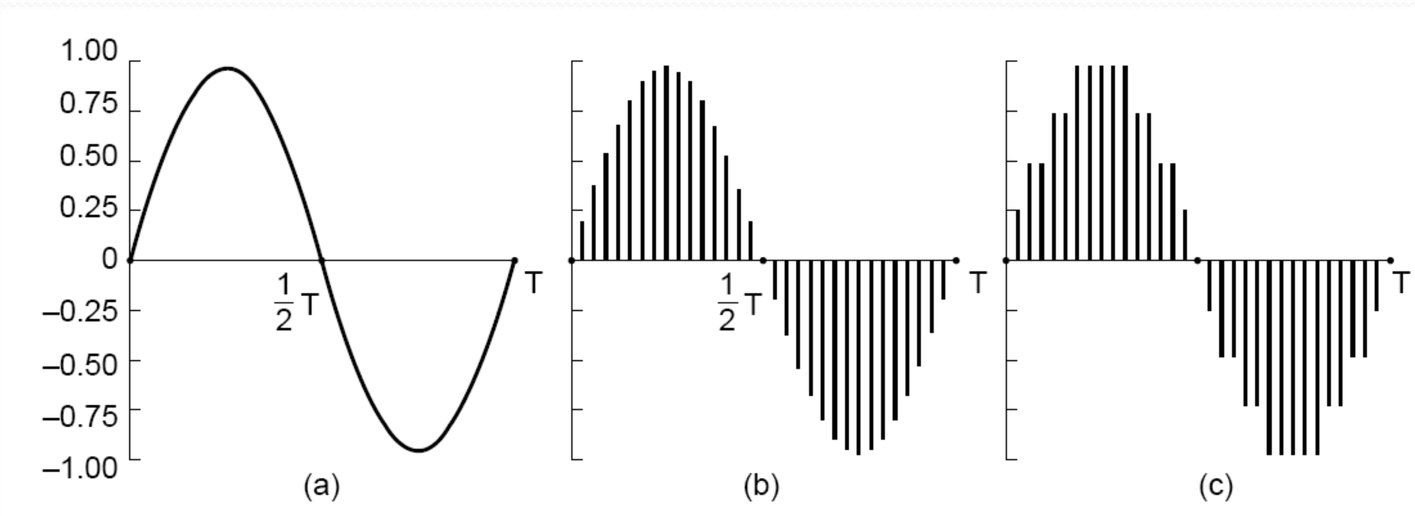
- **Topic 6: Application and Services.**
  - Objectives
    - Multimedia coding
    - Identify the main architectures and protocols involved in **multimedia applications**
    - Understand the **QoS architectures** in Internet

- **Multimedia (MM) applications and supporting protocols:**

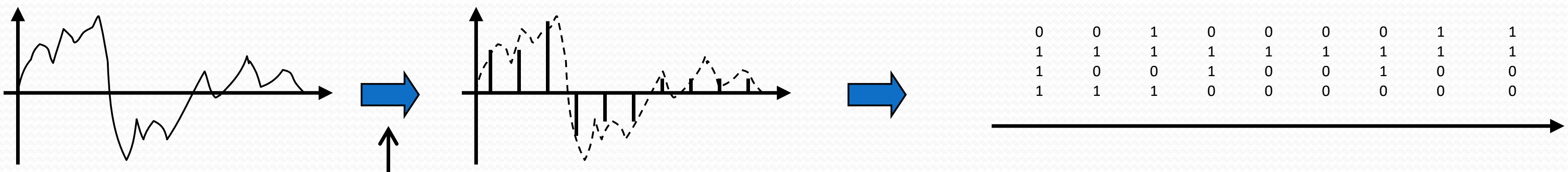
  - Includes the combination of several content forms such as text, still images, video, audio, etc
  - We will here **introduce** audio, image and video **coding techniques**
  - We will here introduce **a map with the needed protocols** in Internet to communicate multimedia content
  - We will introduce the main **Internet architectures to support QoS** (Quality of Service) content

- **Analogic/Digital (A/D) conversion:**





Analogic/Digital (A/D) conversion:



Analogic Signal — Sampled Signal — Coded Digital Signal

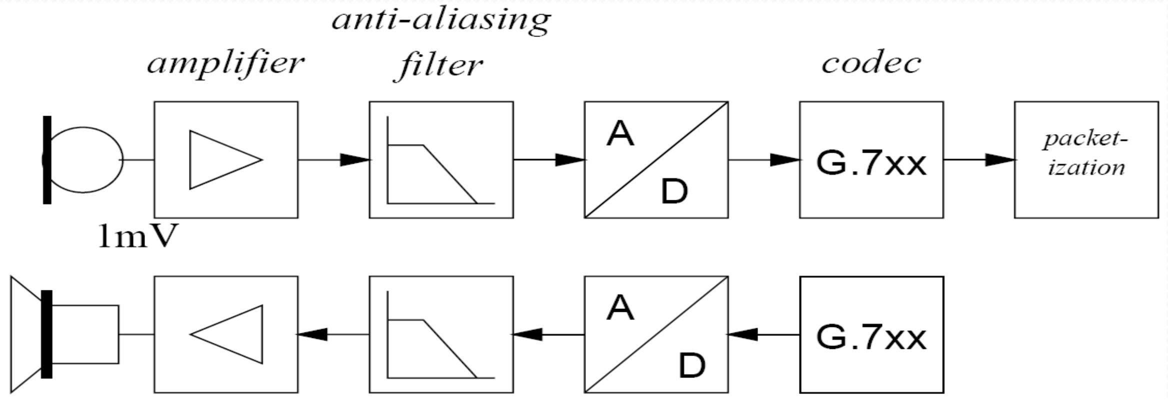
Sampling Theorem: $f_m \geq 2B_w \rightarrow T_m = 1/f_m \leq T/2 = 1/(2B_w)$

F_m : Sampling Frecuency (samples/seg) ≥ 2x Bandwidth
b_m: Bits/sample (Quantization process)

For transmitting a digital signal we need $\boxed{R = F_m \times b_m \text{ bits/seg}}$

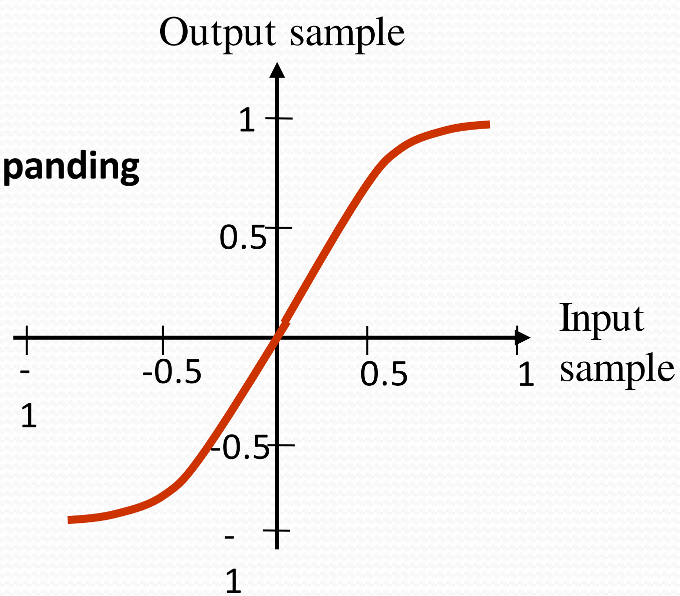
Ex: Voice with phone quality. F_m = 8 KHz, b_m = 8 → R = 64 Kbps
Ex: Compact Disk (CD): F_m = 44.1 KHz, b_m = 16 → R(estéreo) = 1.411 Mbps

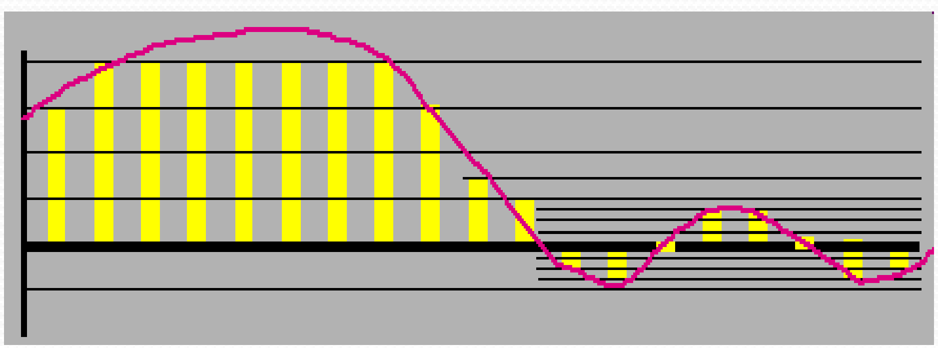
- **Audio:**
  - **G.7xx:** group of standards belonging to ITU-T (International Telecommunication Union) to compress voice
    - Sampling at 8KHz, 8-bit/sample for a maximum of 64 Kb/s data rate with a typical delay of 0.125 ms.
    - **G.711:** Logarithmic Pulse Code Modulation
    - **G.721/G.726:** AD-PCM (Adaptive Differential - Pulse Code Modulation)
    - **G.727:** same that G.726 but optimized in order to allow Packet Circuit Multiplexing Equipments (PCME) environemts



- **G.711: Phone networks.** Use of non-lineal quantization (companding=compressing+expanding): A-law (Europa), μ-Law (USA/Japan).
  - The idea is that high signal amplituds are less probable than low ones, then expand (small steps) the low amplituds (logaritmically) and compress (large steps) the high ones → uniform quantization

Companding





Equal steps

Logaritmic steps

Analogic/Digital (A/D) conversion:

- The explained coding technique is called **PCM (Pulse Code Modulation)**.

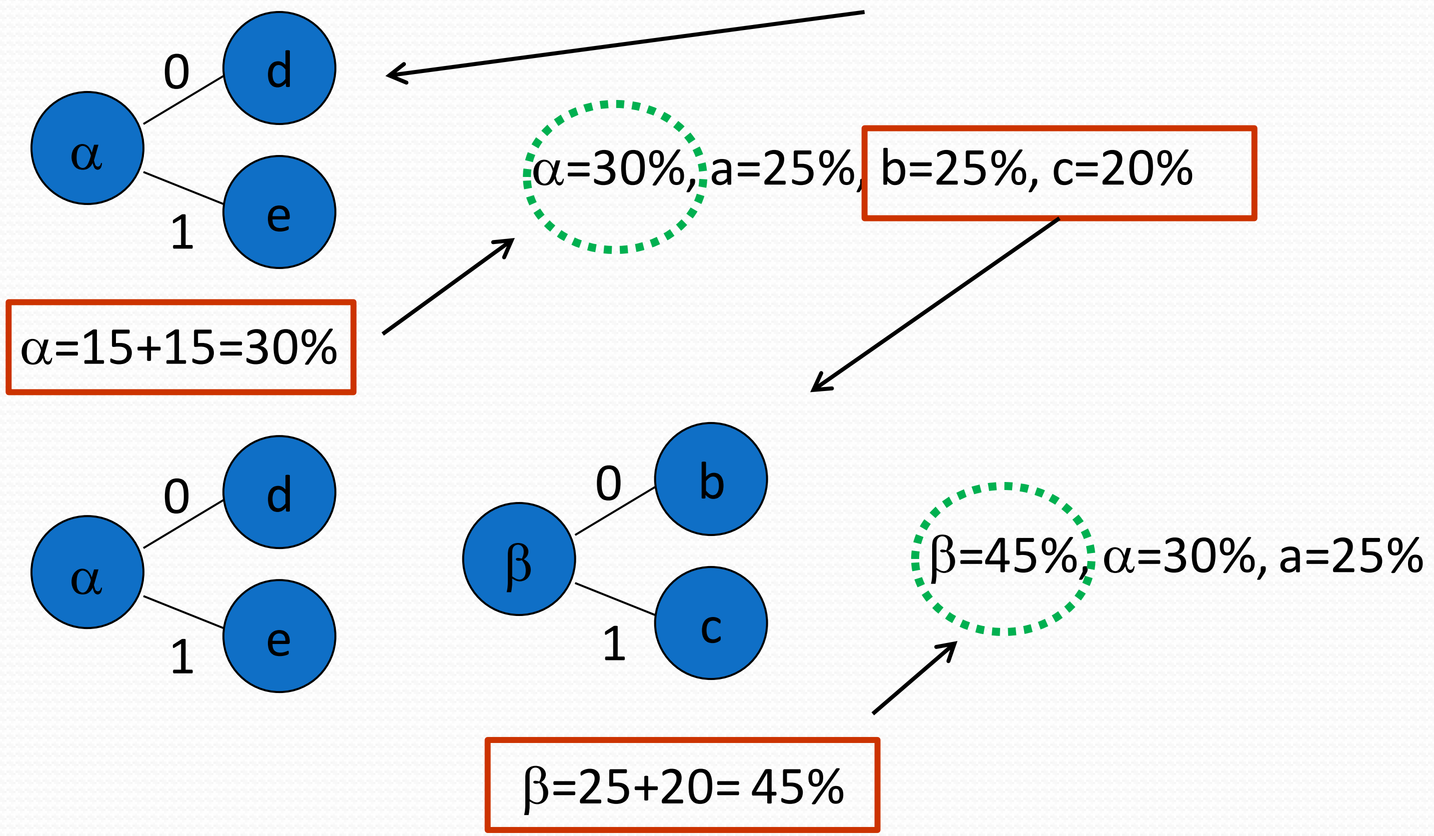
- ***Compression*** (also called ***source coding***) is used to reduce the number of bits needed for transmission or for storing information:

  - **Lossless Compression (without losses):** the original information is exactly recovered (e.g., ZIP, GZIP, GIF, TIFF)

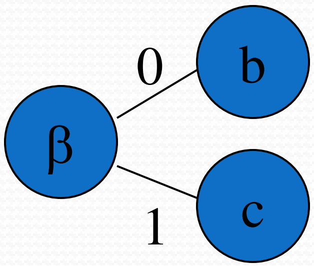
  - **Lossy Compression (with losses):** some info is lost but compression ratios are higher. Used in audio and images (e.g., JPEG, MPEG, MP3, …)

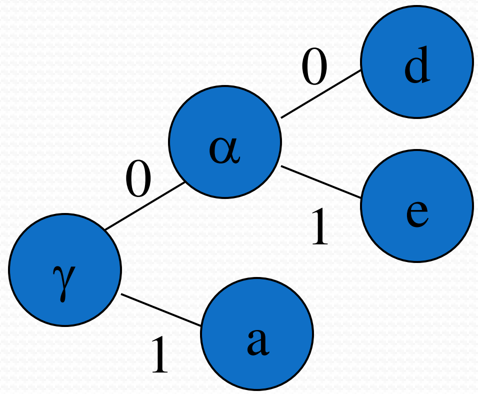
- **Huffman coding (lossless compression):**
  - Optimum technique, but relative symbol frequencies have to be known.
  - Example. Order frequencies from largest to lowest:

a=25%, b=25%, c=20%, d=15%, e=15%



- **Huffman coding (lossless compression):**

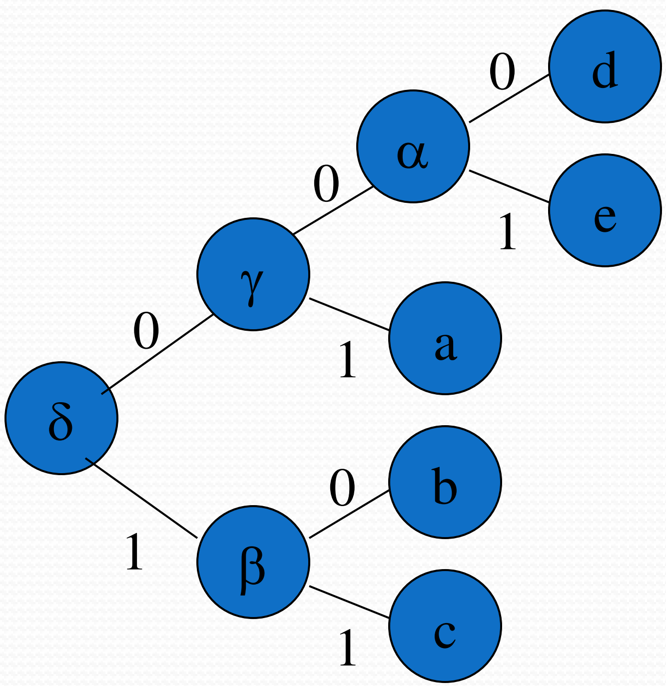




$\boxed{\beta=45\%, \alpha=30\%, a=25\%}$

$\boxed{\gamma=55\%, \beta=45\%}$

$\delta=100\%$



a=01, b=10, c=11, d=000 e=001

Average code-word length (L):

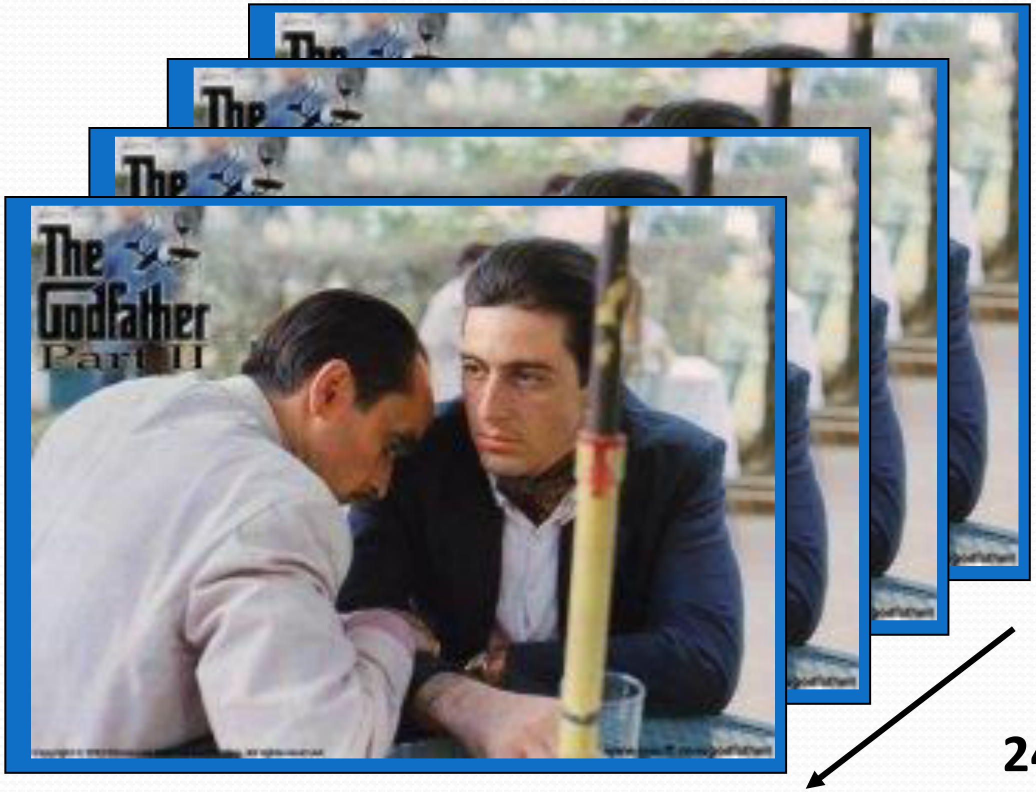
$L=\Sigma\, l_i \cdot p_i = 2\cdot 1/4+2\cdot 1/4+2\cdot 1/5+3\cdot 3/20+3\cdot 3/20=$
$= 2.3$ bit/symbol $\geq H(x)=2.2854$

Entropy (H): $H(x)= -\Sigma\, p_i \cdot \log_2 p_i = 2.2854$

- **Multimedia (MM) applications and supporting protocols:**

1. A MM file is loaded before is being played (**stored media**) (e.g.: emule).
2. If the file is big enough to be loaded before being played (disk space, latency), we can download a few seconds before playing. Thus, jitter (in the order of seconds) is compensated (**streaming**) (e.g. WindowsMedia)
3. The information have to arrive in few miliseconds since it is inter-active or because is played in real-time (**continous playout**) (e.g. VoIP, Internet radio)

- **Multimedia (MM) principles:** a video is a sequence of frames (images) at a rate of 24 to 30 frames per second.



24-30 images per second

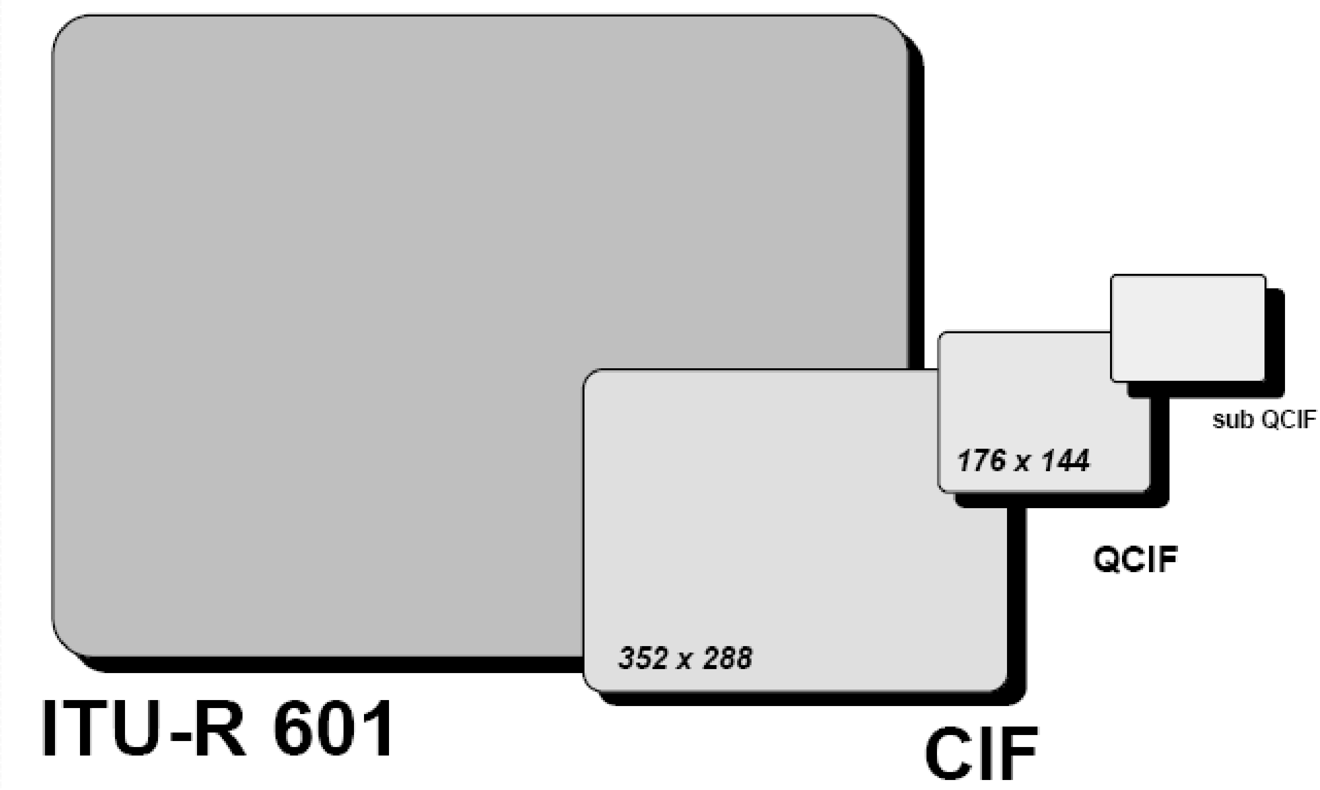
Topic 6: Applications and Services.

- **Multimedia (MM) principles:** each frame is formed by picture elements (pixels). Each pixel is represented by 3 values depending on the **color model**. For example, in the RGB (Red-Green-Blue), a matrix of NxM pixels is represented by 3 matrices, each of NxM size, with Red, Green and Blue values. HSV (hue-saturation-value), HSL (hue-saturation-lightness) or HSB (hue-saturation-Brightness) are another format with also 3 matrices.

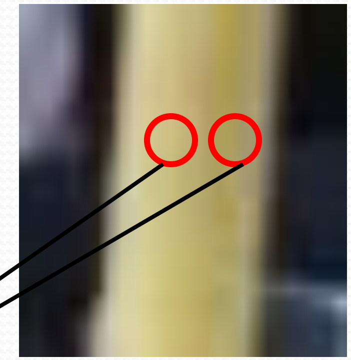
1-pixel: 10001110 10010010 00100100

Each pixel is represented by a set of bits: RGB24 implies 8-bit for Red, 8-bit for Green, and 8-bit for Blue, allowing 2^{24}=16.777.216 possible colors.

- **Multimedia (MM) applications:**
  - **Formats: Number of pixels per frame (image)**. For example a **CIF frame** has 352x288 pixels → without compression that means in 24-bit/pixel RGB = 3x352x288 = 304.128 Bytes/frame x 30frames/s = 9.123.840 Bytes/s = 72.990.720 bit/s = **73 Mb/s** → too large. **High Definition (HD)** has 1.280x720 pixels that implies: 3x1.280x720x30x8 = 663.552.000 bit/s = **663Mb/s**.



- **Multimedia (MM) principles:**



Pixel matrix

Spatial Redundancy : nearby points in the same image have similar luminance and colour. **That means that neighbor values in the same R, G or B matrix will be similar**.

Temporal Redundancy : the same points in consecutive images have similar luminance and color. **That means that they will have similar R, G or B matrices in consecutive frames**.

Coding and compression protocols:

Still frames: JPEG, GIF

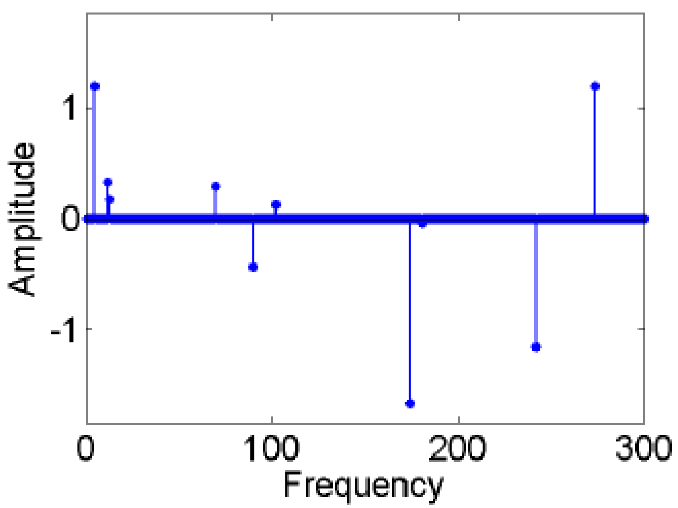
Video:

MPEG-1, MPEG-2, MPEGxx, …

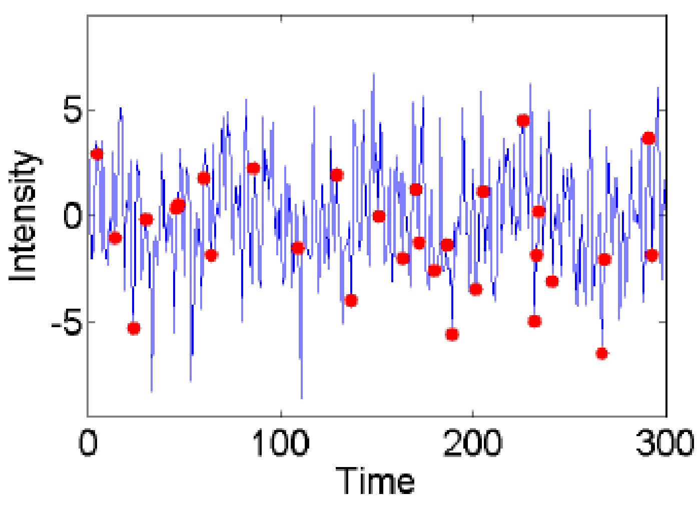
H.261, H.263, …

Spatial compression principles (lossy compression):

- Assume a sampled time signal with interval T: e.g. produces samples (coefficients) that later are coded with Huffman coding. **It is lossy because** in the quantization process we loss some information, but all samples have similar frequency → low compression ratio,
- Assume now that we transform this signal in another domain in such a way, that a high number of coefficients that represent the signal is cero and few are larger than zero, and apply Huffman coding. → high compression ratio,



(a)

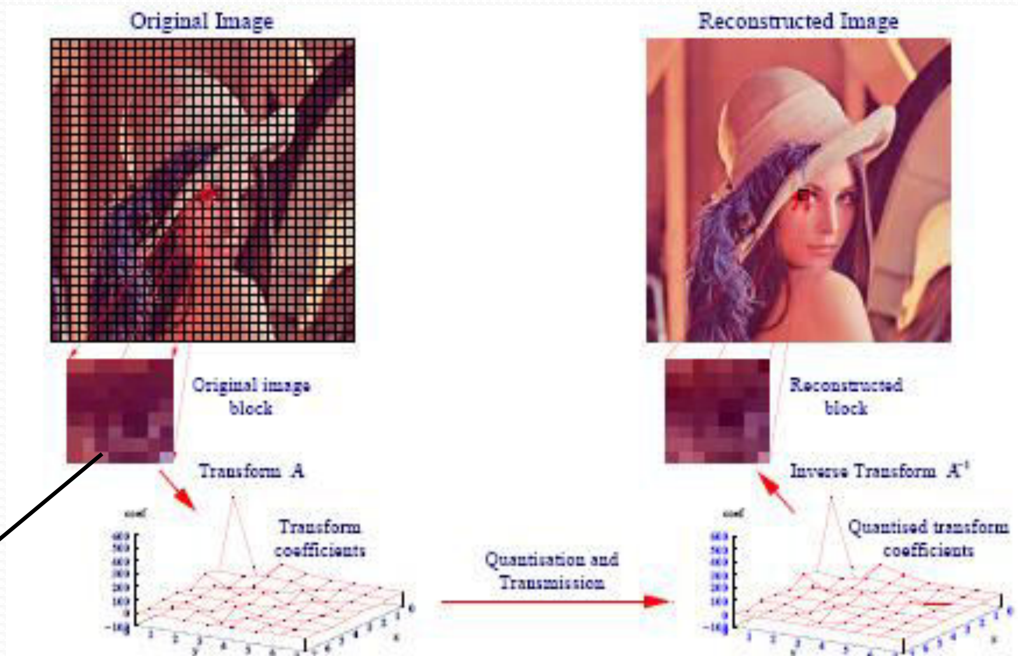


(b)

Spatial compression principles (lossy compression):

- **Transform step (no loss of information):**
  - The source X is divided into blocks of size N, $\{x_n\}$. Each block $\{x_n\}$ is mapped into a transform sequence $\{e_n\}$ using a reversible mapping.
  - Most of the energy of the transformed block (coefficients much larger than zero) are contained in few elements of the transformed values.
- **Quantization step (loss of some information):**
  - Coefficients are quantized, thus, some information will be lost,
  - The transformed sequence is quantized based on the following strategy:
    - The desired average bit rate, the statistics of the transformed elements, and the effect of distortion on the reconstructed sequence.
- **Entropy coding step:**
  - The quantized data are entropy-coded using Huffman or other techniques.

- **Spatial compression principles (lossy compression):**



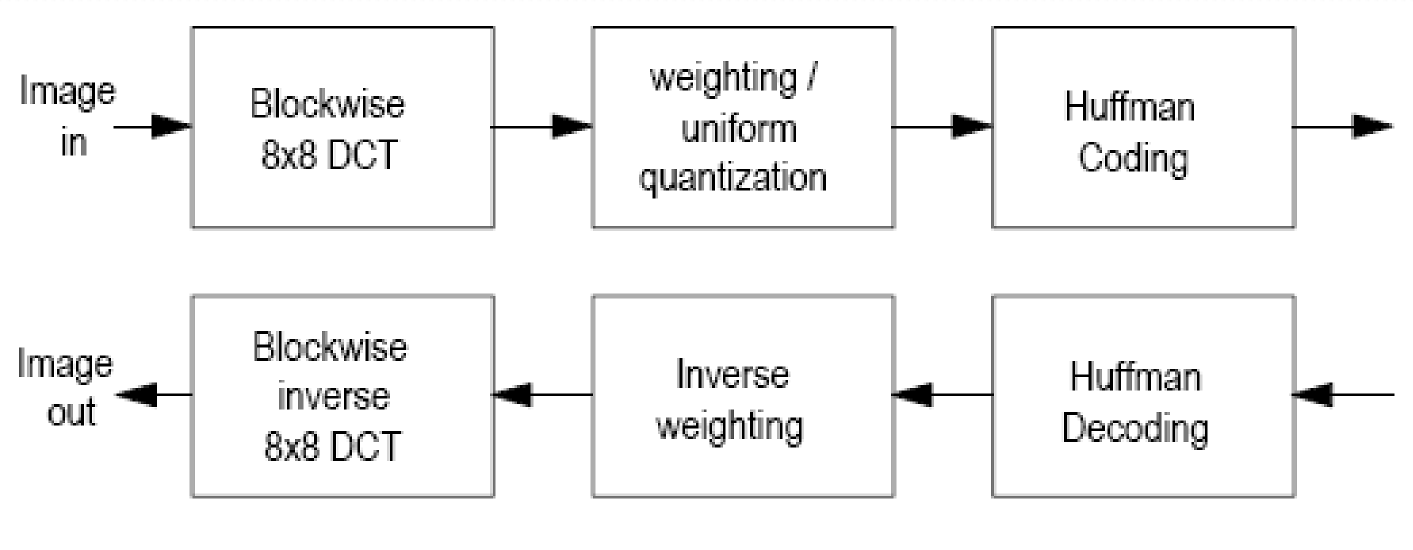
(1) **Block Transform Coding $\{x_n\}$**: Blocks of 8x8 pixels

(2) DCT (**Discrete Cosine Transform**), $\{e_n\}$ coefficients are calculated

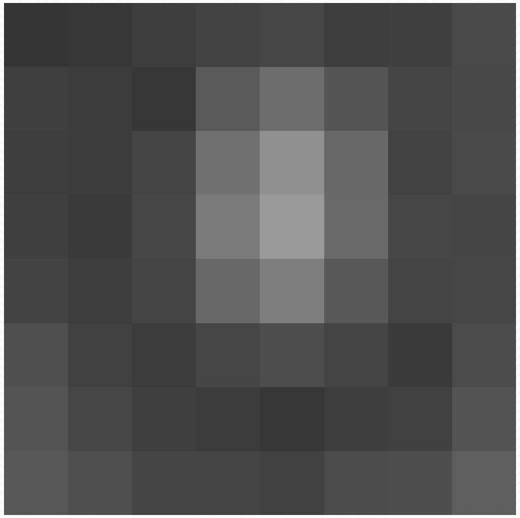
(3) **Weighting** and **quantization process**

(4) Coefficients are coded with **Huffman coding**

- **Spatial compression principles (lossy compression):**



- **Example of coding in JPEG (wikipedia):**



8x8 sub-image

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}.$$

8x8 matrix

If the range is on [0,N] = [0,255], then center the coefficients on the middle point 128. We name each coefficient as g_{xy}.

$$\begin{array}{c} \overset{x}{\longrightarrow} \\ g = \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} \Bigg\downarrow y. \end{array}$$

Example of coding in JPEG (wikipedia):

DCT-II (type II): u,v (0≤u,v<8) are the new coordinates, $g_{x,y}$ are the old coordinates

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u = 0 \\ 1, & \text{otherwise} \end{cases}$$

Matrix after DCT

$$G = \overset{\displaystyle \overset{u}{\longrightarrow}}{\begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}} \Bigg\downarrow v.$$

Example of coding in JPEG (wikipedia):

Normalize (weight) with respect a quantization matrix Q:

$$B_{j,k} = \mathrm{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right) \text{ for } j = 0,1,2,\ldots,7; k = 0,1,2,\ldots,7$$

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

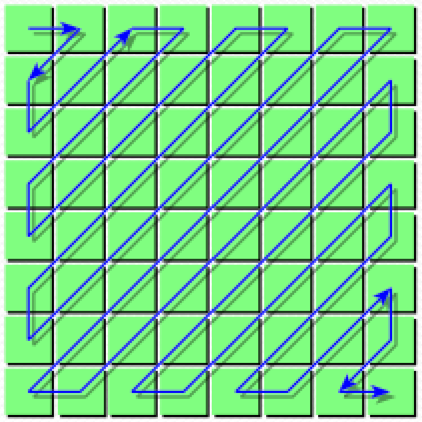
$\longrightarrow$ **Matrix after quantization**

- **Example of coding in JPEG (wikipedia):**

Store using Zig-Zag and apply Huffman compression using a dictionary :

Values taken in zigzag

-26
-3 0
-3 -2 -6
2 -4 1 -3
1 1 5 1 2
-1 1 -1 2 0 0
0 0 0 -1 -1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0
0 0 0 0
0 0 0
0 0
0



- **Multimedia (MM) principles:**
  - Motion Estimation/Compensation (ME/MC)

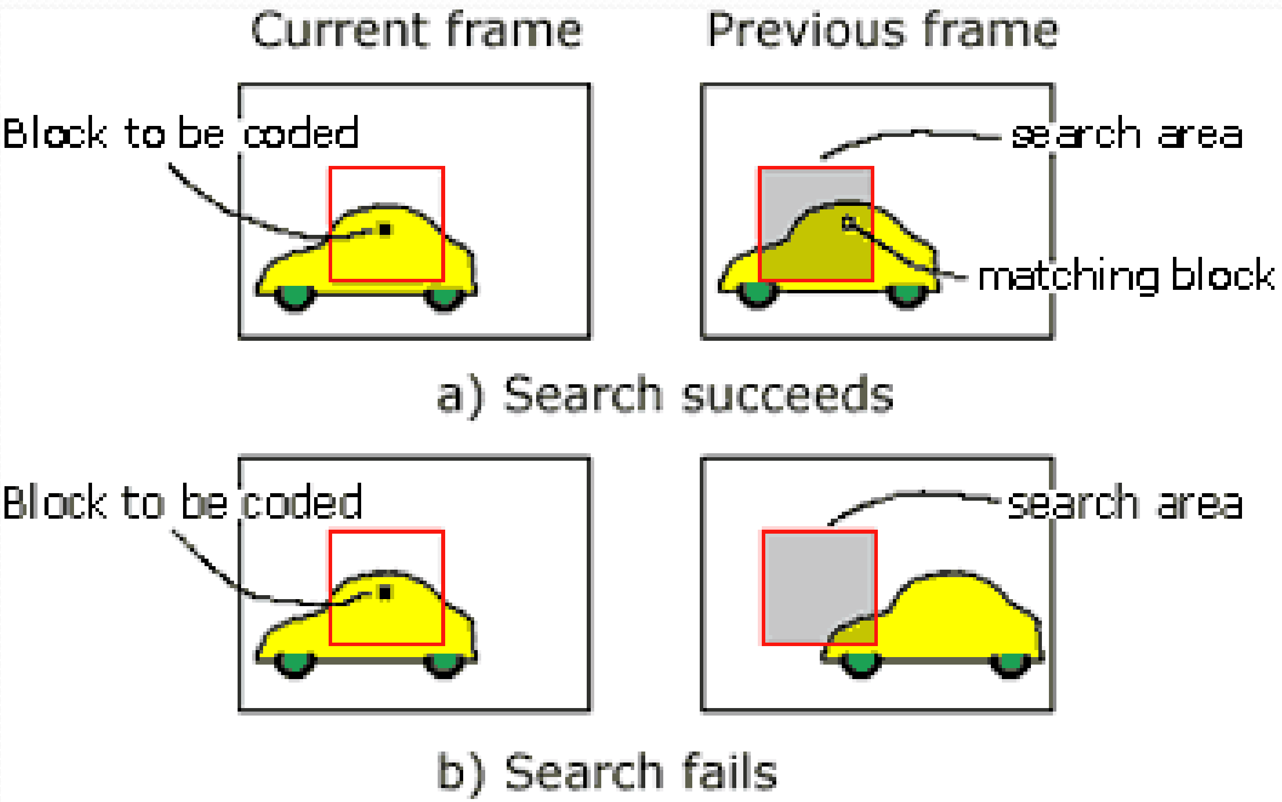


Figure: Principle of motion estimation (ME)

- **Multimedia (MM) principles:**

  - Typically first apply Motion Compensation (MC), and on the error frame apply DCT

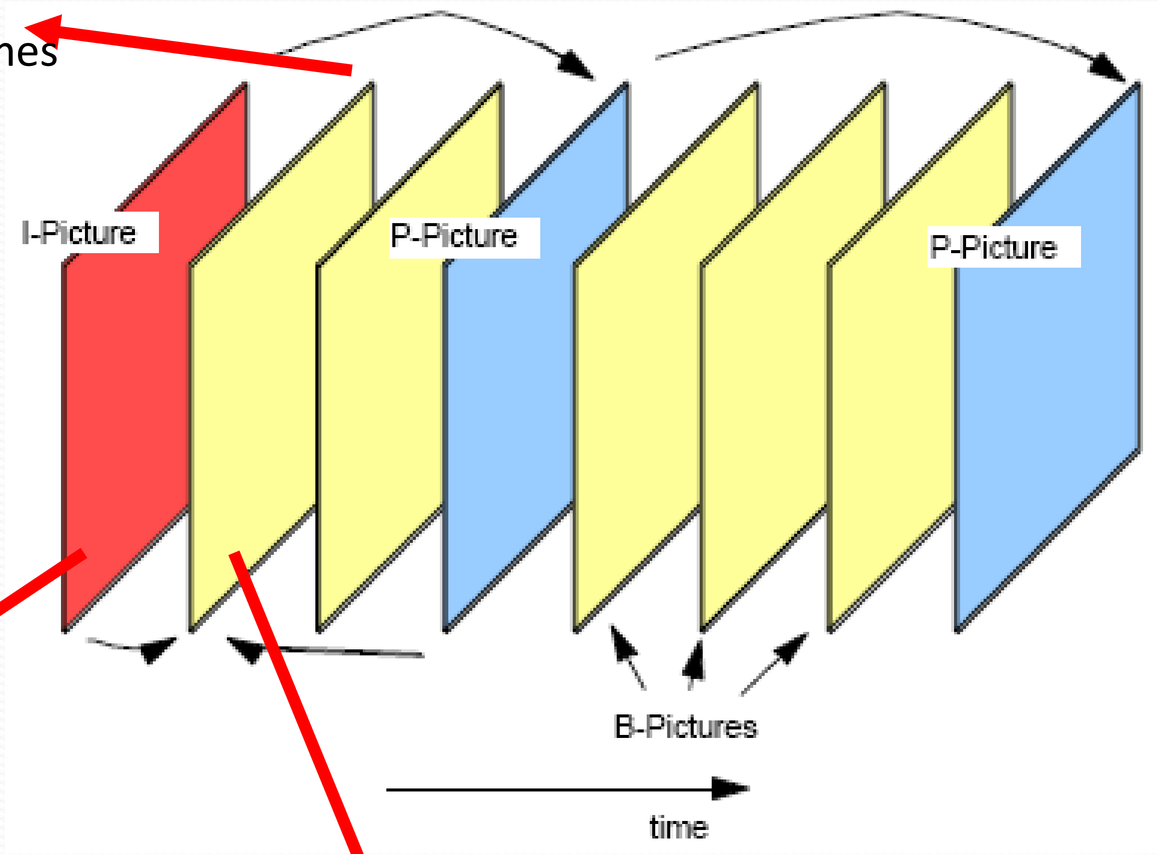
⇒ INTRA coded Frame (**I-Frame**): apply DCT to a block of NxN pixels (N=8)

⇒ Predictive Coded Frame (**P-Frame**) and Bidirectionally Predective Coded Picture (**B-Frame)**: apply MC and later DCT over the error frame. Code movement vectors and quantized coefficients of the DCT.

- **Multimedia (MM) applications:**
  - **GOP (Group of Pictures):**
    - Sequence of frames that specifies the order in which Intra frames and Inter frames are arranged
    - **I-frames** (Intra Coded Picture): reference frames, uses DCT compression
    - **P-frames** (Predective Coded Picture): contains motion-compensation from the previous I-frame or P-frame
    - **B-frames** (Bidirectionally Predective Coded Picture): contains motion compensation from the preceding and following I or P-frames. That implies buffering and reordering of frames.
    - **GOP (M,N):**
      - M distance between consecutive P frames
      - N total number of the sequence (<u>distance between consecutive I-frames</u>)
    - GOP(3,12) → IBBPBBPBBPBB IBBPBBPBBPBB IBBPBBPBBPBB ….

- **Multimedia (MM) applications:**

Prediction with motion compensation from I-frames



intra-frame coding

Prediction with motion compensation from consecutive I-frames and P-frame

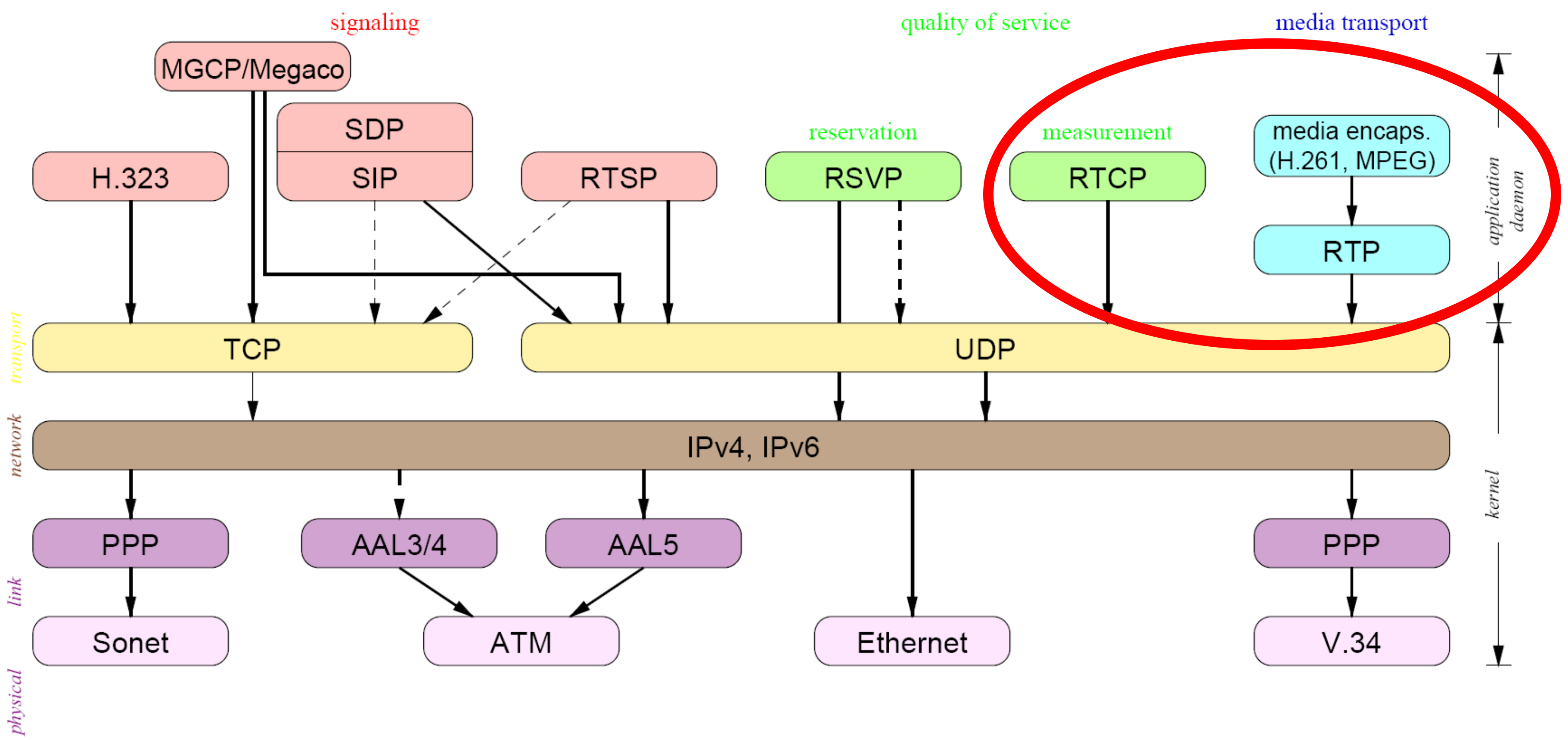
- **Multimedia (MM) applications:**
  - Still Images coding
    - **JPEG** good compression of images (but for example, would poorly compress a fax).
      - Standard for still images black/white and color
      - Lossless coding (predictive) or lossy coding (DCT).
      - Exploits properties taken from eye human perception (example: color changes are less perceptible than bright)
      - Represents colors with 24 bits
      - Variable compression ratio (24:1, acceptable quality loss)
    - **GIF** proprietary method for lossless compression (LZW).
      - Represents colors with 8 bits (**PNG**: non-propietary, 24 bits)

- **Multimedia (MM) applications:**
  - ISO MPEG
    - MPEG-1
      - 1.5 Mbps
      - CIF, no interleaving
      - 24-30 frames/seg
      - Video-storing (CD-ROM)
    - MPEG-2
      - 4-8 Mbps forTV (PAL in Europa)
      - 20 Mbps for HDTV
  - MPEG-4
    - Absorbs many features of MPEG-1 and MPEG-2
  - MPEG-7, MPEG-21, MPEG-A,B,C,D,E,V,M,U,...

- **ITU-T standards video codecs (old standards)**
  - **H.261**
    - ≤ 128 Kbps (ISDN)
    - Formats CIF or QCIF.
    - 7.5-30 frames/s
    - Videoconference systems
  - **H.263**
    - ≤ 30 Kbps (RTC)
    - CIF, QCIF, Sub-QCIF
    - ≤ 10 fps
    - Internet video streaming

- ITU-T standards video codecs (new standards)
  - H.264/AVC (Advanced Video Coders)
    - can be packaged into .mp4, .mov, .F4v, .3GP (mobile phones), and .ts containers,
    - maximum video resolution of 1280x720 pixels at 30 frames per second at a maximum data rate of 14 Mbps.
  - H.265/HEVC (High Efficiency Video Coding)
    - Successor of H264,
    - Uses many features that comes from H264 and add new ones, e.g., more intra prediction directions (from 9 to 35), etc.,
    - HEVC uses three frame types, I-, B- and P-frames within a group of pictures, incorporating elements of both inter-frame and intraframe compression.
- Other codecs
  - AV1, VP9, H266/VVC, etc.

- **Multimedia (MM) supporting protocols:**



Multimedia (MM) supporting protocols:

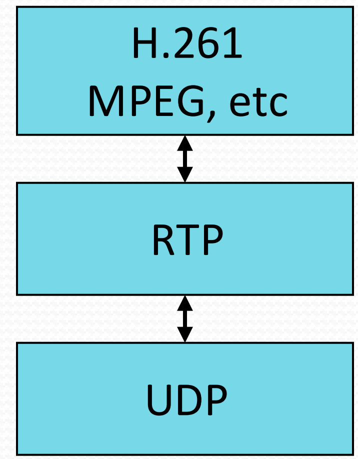
- ¿How can the process know:
  - time at which has to replay the MM information
  - lost packets
  - change in packet order
  - type of coding, … ?

- **RTP (Real-time Transport Protocol):** deals with the transfer of real-time data. Information provided by this protocol include **timestamps** (for synchronization), **sequence numbers** (for packet loss and reordering detection) and the **payload format** which indicates the encoded format of the data.

- **RTCP (RTP Control Protocol):** used to **specify quality of service** (QoS) feedback and synchronization between the media streams.

RTP (Real-time Transport Protocol):

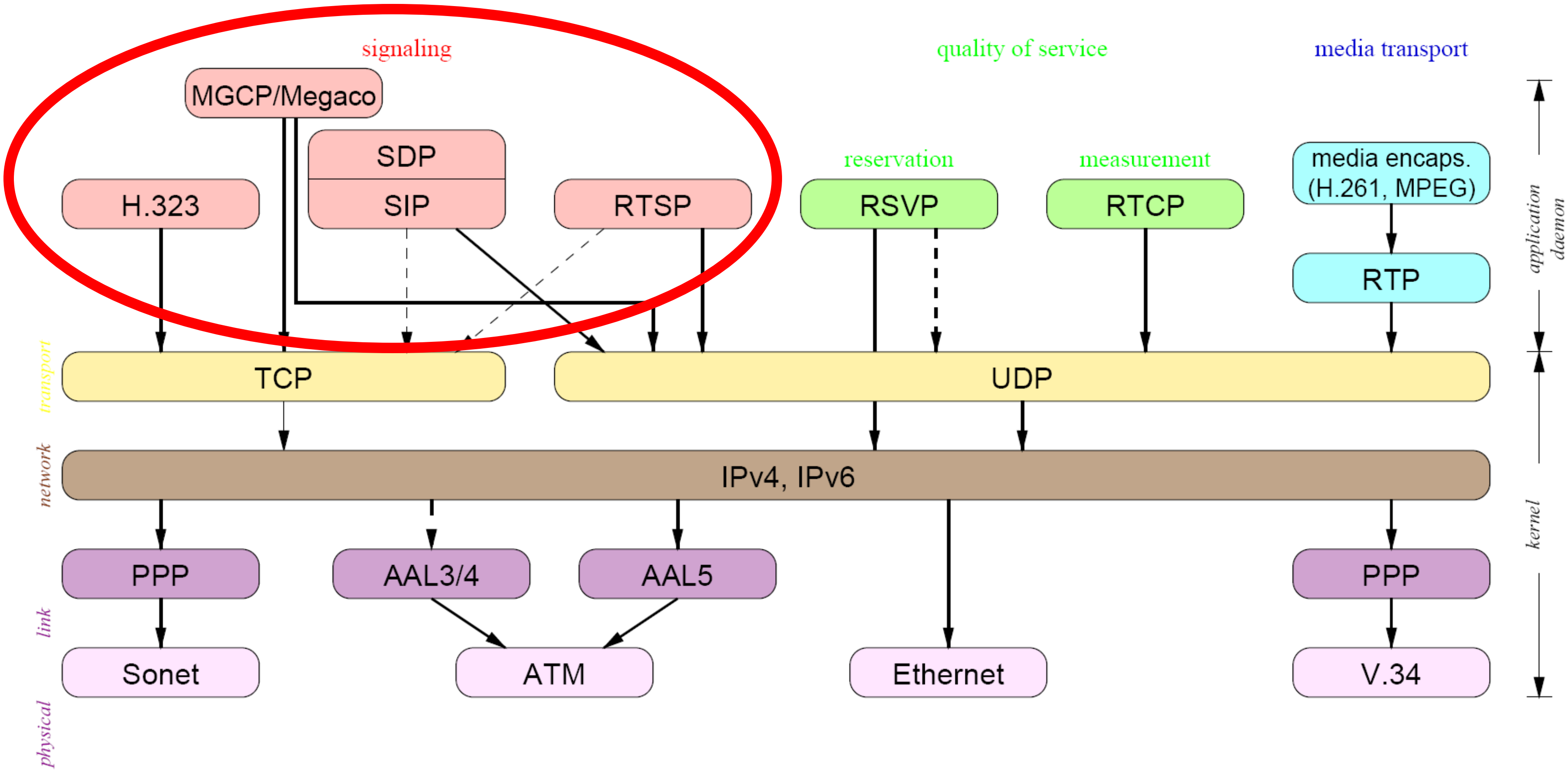
- RTP allows real-time information encapsulation.
- Adds information that facilitates the task of recovering the correct signal at the receiver:
  - Payload Type (PT):
    - 3: GSM, 14: MPEG audio, 31: H.261,
    - 32: MPEG1 video, 33: MPEG2 video, etc…
  - Timestamps: Timing
  - Sequence numbers: Losses
  - SSRC: Stream (random number)




```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

- **RTCP (RTP Control Protocol):**
  - Protocol jointly used with RTP.
  - Allows sending from Transmitter to Receiver statistics that are useful at the time of coding/decoding the transmission/reception of information on real-time
    - SSRC of the RTP flow from which we send the info
    - # of packets sent
    - # of packet lost
    - jitter

- **Multimedia (MM) supporting protocols:**



- **Streaming (signaling) protocols for MM:**
  - **Streaming protocols:** each time you watch a live stream or video on demand, streaming protocols are used to deliver data over the internet. These can sit in the application, presentation, and session layers
  - **before sending MM packets** (i.e., encapsulating the MM data using RTP-UDP-IP) **the client must communicate with the server and learn**:
    - Where is the MM data ? What kind of coding (MPEG1, 2, 4, H261, …) technique is using ? What kind of Session (e.g. RTP) and transport (e.g. UDP) protocols is going to use ?
    - Similar format than HTTP
    - It is important that they produce low latency
    - Use of UDP or TCP as transport protocol ?

- **Signaling protocols for MM:**
  - **Examples of streaming protocols (use specific streaming servers)**
    - RTSP (Real Time Streaming Protocol)
    - SRT (Secure Reliable Transport)
    - Adobe RTMP (Real-Time Messaging Protocol)
  - **HTTP-based adaptive protocols (use regular Web servers to optimize video delivery)**
    - Apple HLS (HTTP Live Streaming)
    - MPEG-DASH (MPEG Dynamic Adaptive Streaming over HTTP)
    - Microsoft Smooth Streaming
    - Adobe HDS (Real-Time Messaging Protocol)
  - **In case of VoiceIP. Signaling protocols such as:**
    - SIP (Session Initiated Protocol) for telephony (VoIP) over Internet (also allows video streaming.

<u>**Here we will only make a fast look on RTSP and SIP to grasp the idea**</u>

RTSP (Real Time Streaming Protocol)

- *RTSP* is a control protocol for the delivery of multimedia content across *IP* networks.
- It is based typically on *TCP* for reliable delivery and has a very similar operation and syntax to *HTTP*.
- *RTSP* is used by the client application to communicate to the server information such as
  - the media file being requested,
  - the type of application the client is using,
  - the mechanism of delivery of the file (unicast or multicast, *UDP* or *TCP*), and
  - other important control information commands such as DESCRIBE, SETUP, and PLAY.
- The actual multimedia content is not typically delivered over the *RTSP* connection(s), although it can be interleaved if required. *RTSP* is analogous to the remote control of the streaming protocols

RTSP example

C→ S

- OPTIONS rtsp://video.fooco.com:554 RTSP/1.0 Cseq: 1

the client will establish a TCP connection to port 554 on the server and issue an OPTIONS command showing the protocol version used for the session

S → C

- RTSP/1.0 200 OK Cseq: 1

the server acknowledges the client request

RTSP example

C → S

- DESCRIBE rtsp://video.fooco.com:554/streams/example.rm RTSP/1.0 Cseq:2

the client issues a DESCRIBE command that indicates to the server the URL of the media file being requested.

S → C

- RTSP/1.0 200 OK Cseq: 2 Content-Type: application/sdp Content-Length: 210 <SDP Data...>

The server responds with another 200 OK acknowledgment and includes a full media description of the content, which is presented in either Session Description Protocol (SDP) or Multimedia and Hypermedia Experts Group (MHEG) format.

RTSP example

C→S

- SETUP rtsp://video.fooco.com:554/streams/example.rm RTSP/1.0 Cseq: 3 Transport:

  rtp/udp;unicast;client_port=5067-5068

the client issues a SETUP command that identifies to the server the transport mechanisms, in order of preference, the client wants to use. We won't list all of the available transport options here (the RFC obviously contains an exhaustive list), but we'll see the client request RTP over UDP on ports 5067 and 5068 for the data transport.

S → C

- RTSP/1.0 200 OK Cseq: 3 Session: 12345678 Transport: rtp/udp;client_port=5067-5068;server_port=6023-6024

The server responds with confirmation of the RTP over UDP transport mechanism and the client-side ports and includes the unique Session ID and server port information.

- **RTSP example**

C→S

- PLAY rtsp://video.fooco.com:554/streams/example.rm RTSP/1.0 Cseq: 4 Session: 12345678

the client is now ready to commence the receipt of the data stream and issues a PLAY command. This simply contains the URL and Session ID previously provided by the server.

S → C

- RTSP/1.0 200 OK Cseq: 4

The server acknowledges this PLAY command, and the RTP stream from the server to client will begin.

<DATA>

- **RTSP example**

C->S

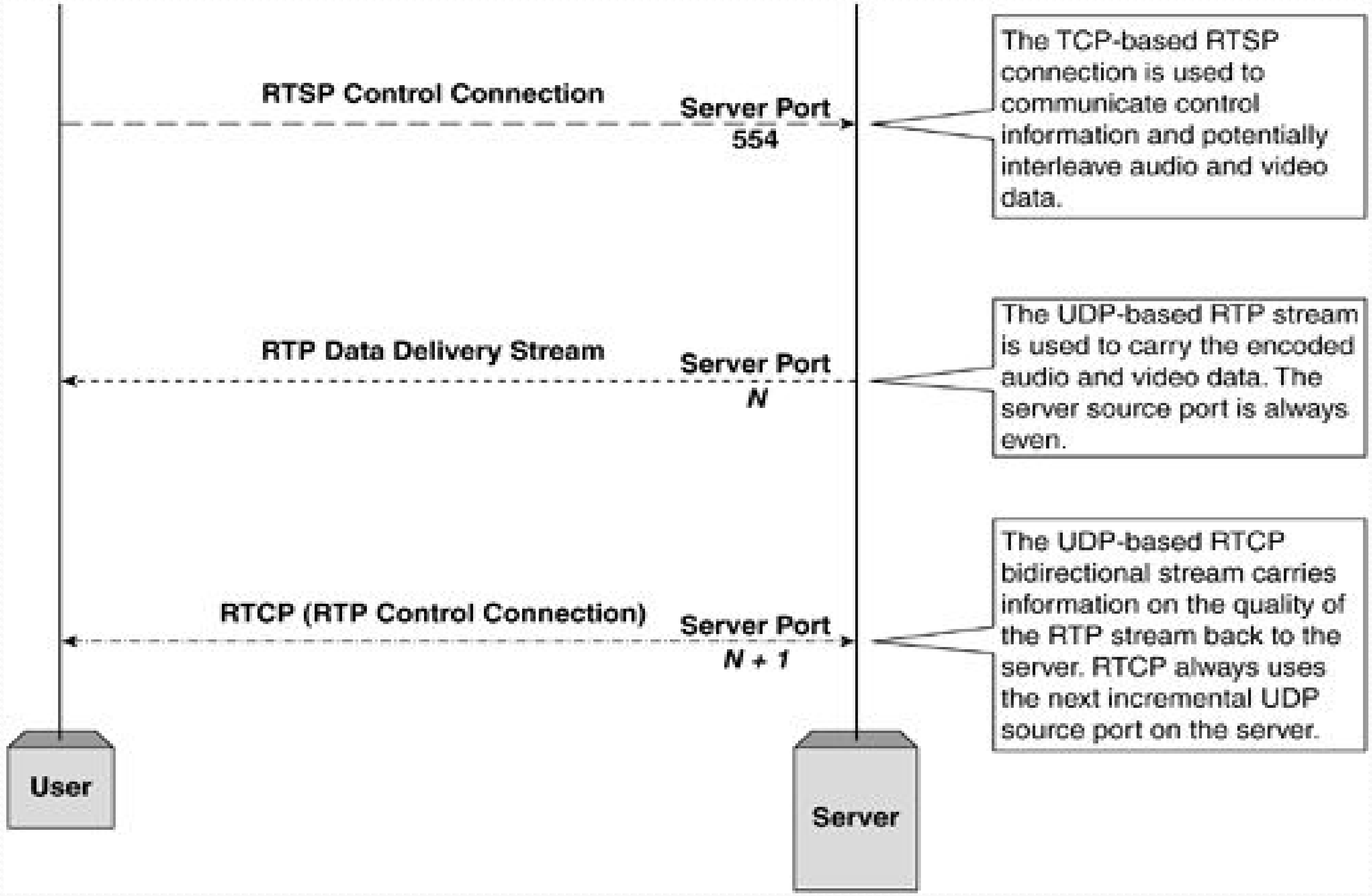
- TEARDOWN rtsp://video.fooco.com:554/streams/example.rm RTSP/1.0 Cseq: 5 Session: 12345678

S->C

- RTSP/1.0 200 OK Cseq: 5

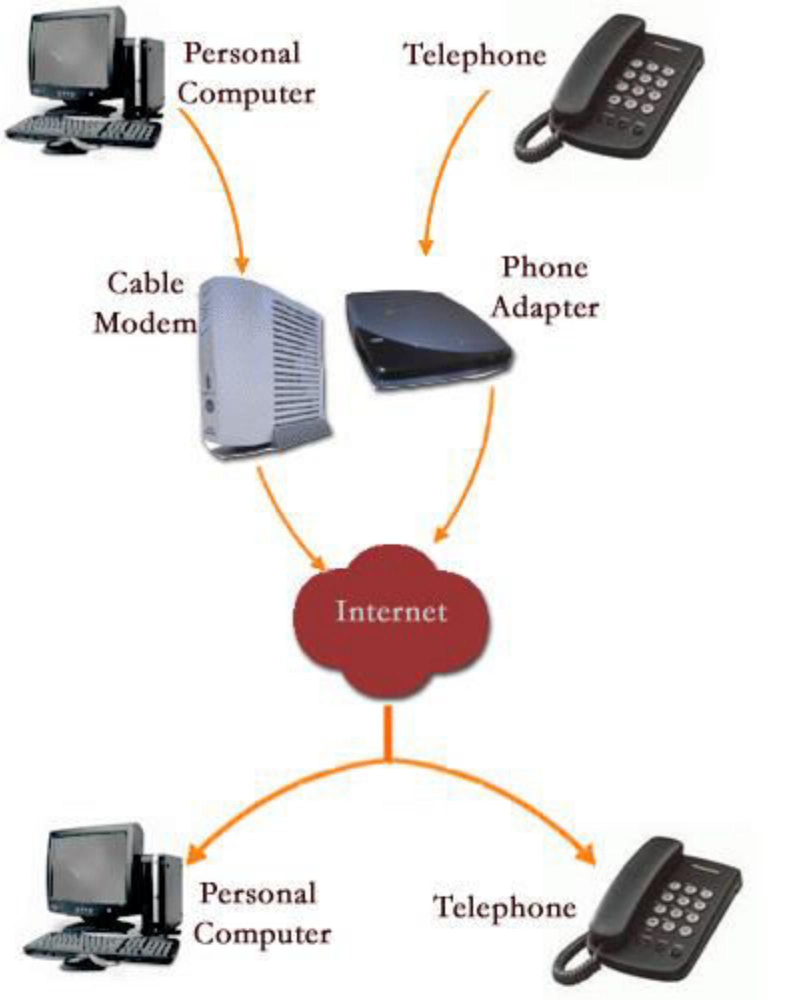
Once the client decides that the stream can be stopped, a TEARDOWN command is issued over the RTSP connection referenced only by the Session ID

- RTSP (Real Time Streaming Protocol)



- **HTTP-based adaptive protocols**
  - the video resides on any HTTP server and the technology remains **stateless**,
  - all HTTP-based **streams are broken into chunks**, either separate files or segments within a larger file,
  - rather than retrieving a single large file with a single request, HTTP-based technologies retrieve consecutive short chunks on as needed basis,
  - the technologies (Smooth Streaming, HLS, and HDS) also enable the efficient switching between streams, thus **they stream adaptively**,
  - HTTP-based technologies are **firewall friendly** and can leverage HTTP caching mechanisms,
  - Because no streaming server is required, they are **less expensive** to implement and can **scale more cheaply and effectively** to serve available users

- ## VoIP (Voice iP)



VoIP is a technology that allows you to make telephone calls using a broadband Internet connection instead of a regular (or analog) phone line.

Some services using VoIP may only allow you to call other people using the same service, but others may allow you to call anyone who has a telephone number - including local, long distance, mobile, and international numbers.

Also, while some services only work over your computer or a special VoIP phone, other services allow you to use a traditional phone through an adaptor

- **VoIP (Voice iP)**
  - **H.323 (ITU-T), SIP (IETF)**
    - Both describe terminal equipments and other entities that support multimedia communications over packet networks.
    - Possibility of joint operation with telephone networks.
  - **MGCP, Megaco/H.248**
    - Protocols that allow controling devices from a Media Gateway Controller (MGC) or Call Agent (CA).

SIP (Session Initiated Protocol)

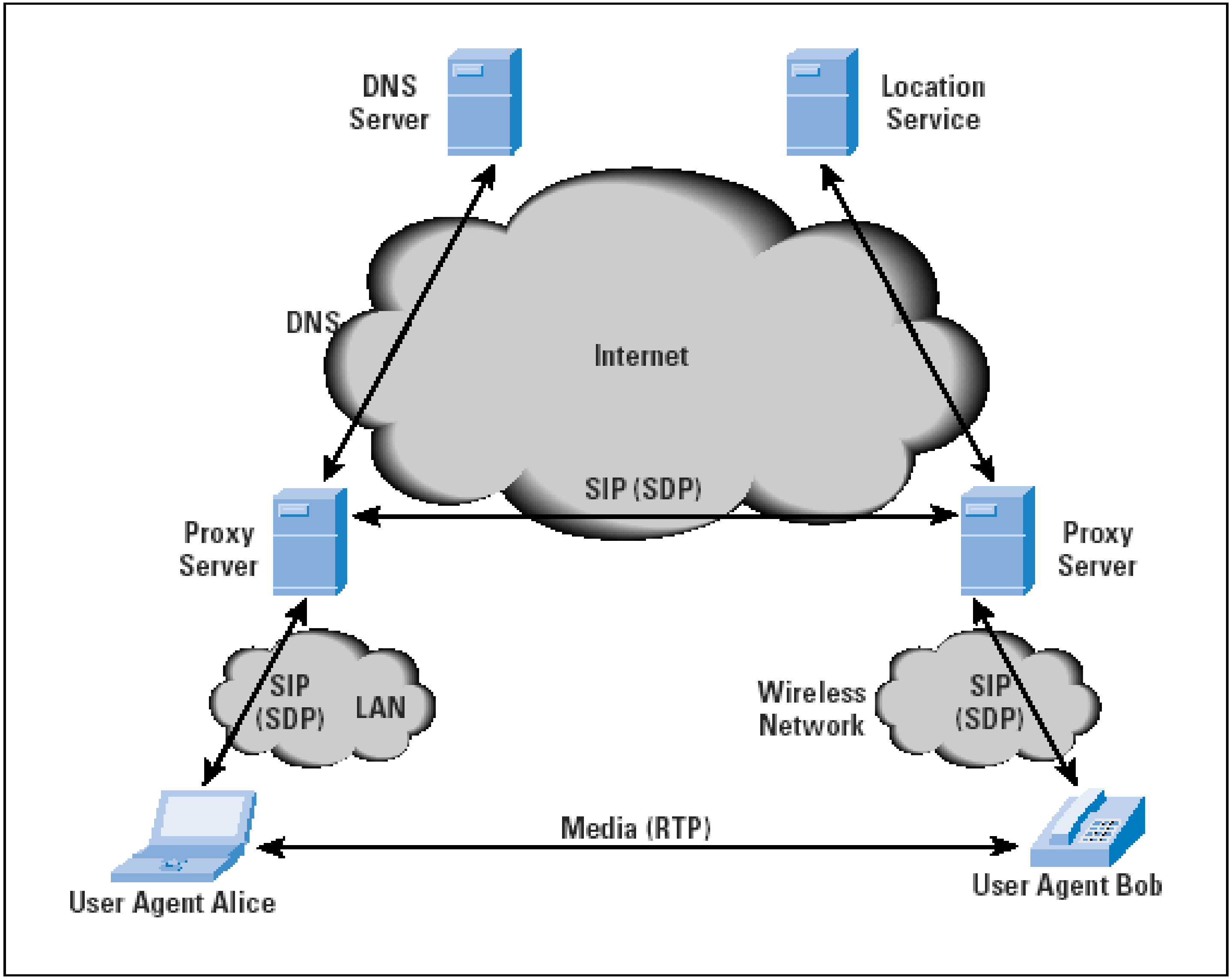
- Is an application level signaling protocol for:
  - Setting-up
  - Modifying
  - Terminating

  real-time sessions between participants over an IP data network.

- SIP can support any type of single-media or multi-media session, including teleconferencing
- **Session Description Protocol (SDP):** defines session content using a set of type similar to MIME
- The actual multimedia content is exchanged using appropiate transport protocols (eg: RTP)
- Key driving force: Internet telephony (Voice over IP, VoIP)

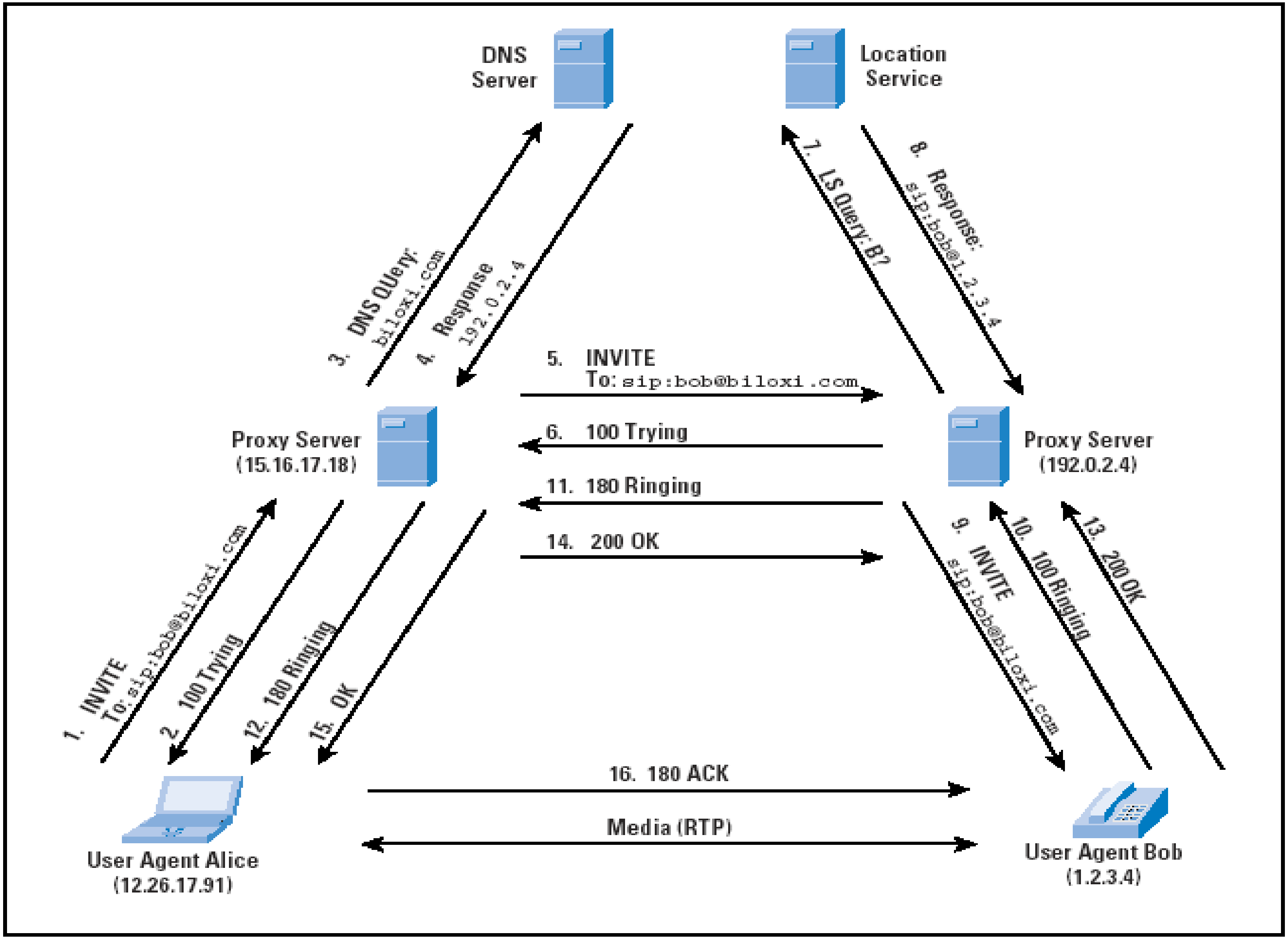
Topic 6: Applications and Services.

Figure 1: SIP Components and Protocols

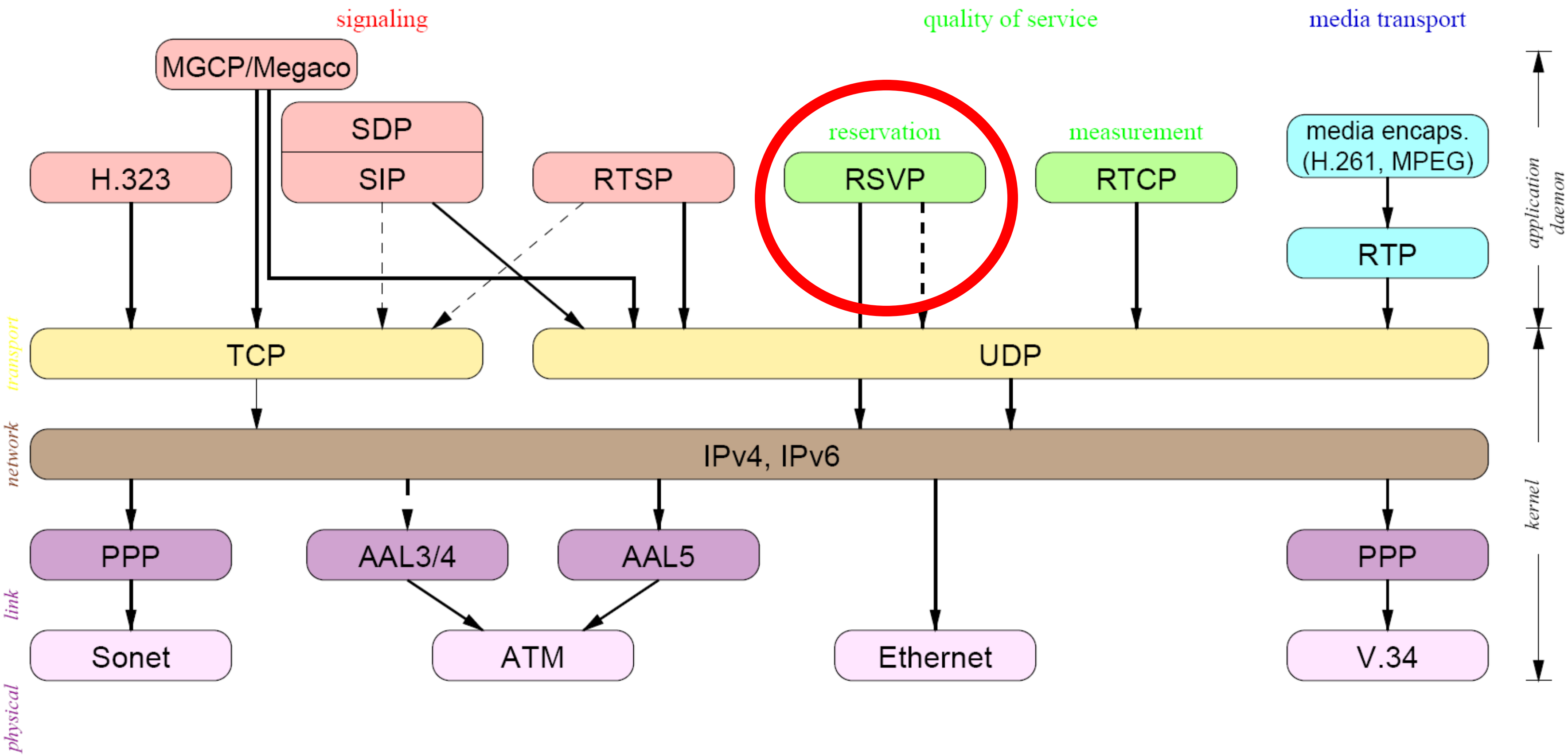


Topic 6: Applications and Services.

Figure 2: SIP Successful Call Setup



- **QoS Internet Architectures:**



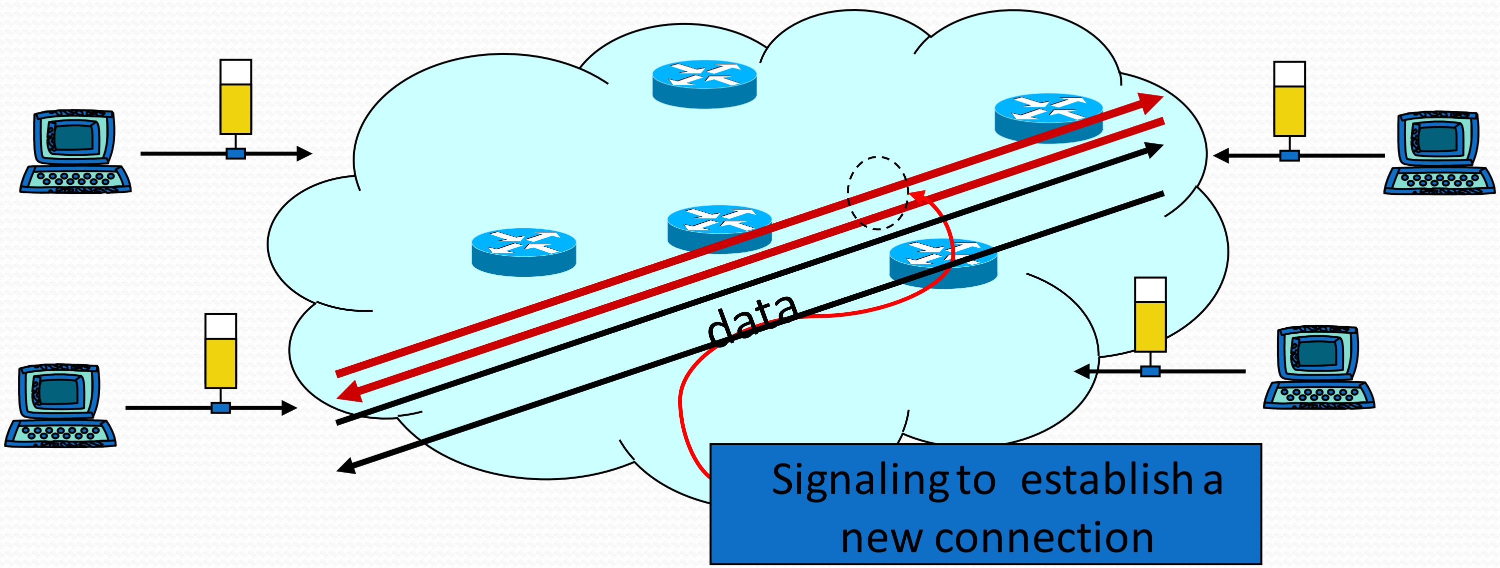
QoS Architectures:

- **Quality of Service (QoS):** is the ability to provide different priority to different applications, users, or data flow. It is measured as jitter, end-to-end packet delay, packet losses, etc

- **IntServ (Integrated Services):** is an architecture that specifies the elements (**fine-grained**) to guarantee QoS on networks.
  - **Per-flow architecture**: that means that every **flow** (5-tuple form by source/destination IP@ + source/destination ports + transport protocol) is treated specifically at each route to guarantee it a QoS.
  - Uses **RSVP** as main protocol

- **DiffServ (Differentiated Services):** architecture that specifies a simple, scalable and **coarse-grained** mechanism for classifying, managing network traffic and providing QoS.
  - **Per-class architecture:** that means that flows are grouped in classes that received the same treatment in the network

- **IntServ (Integrated Services):**

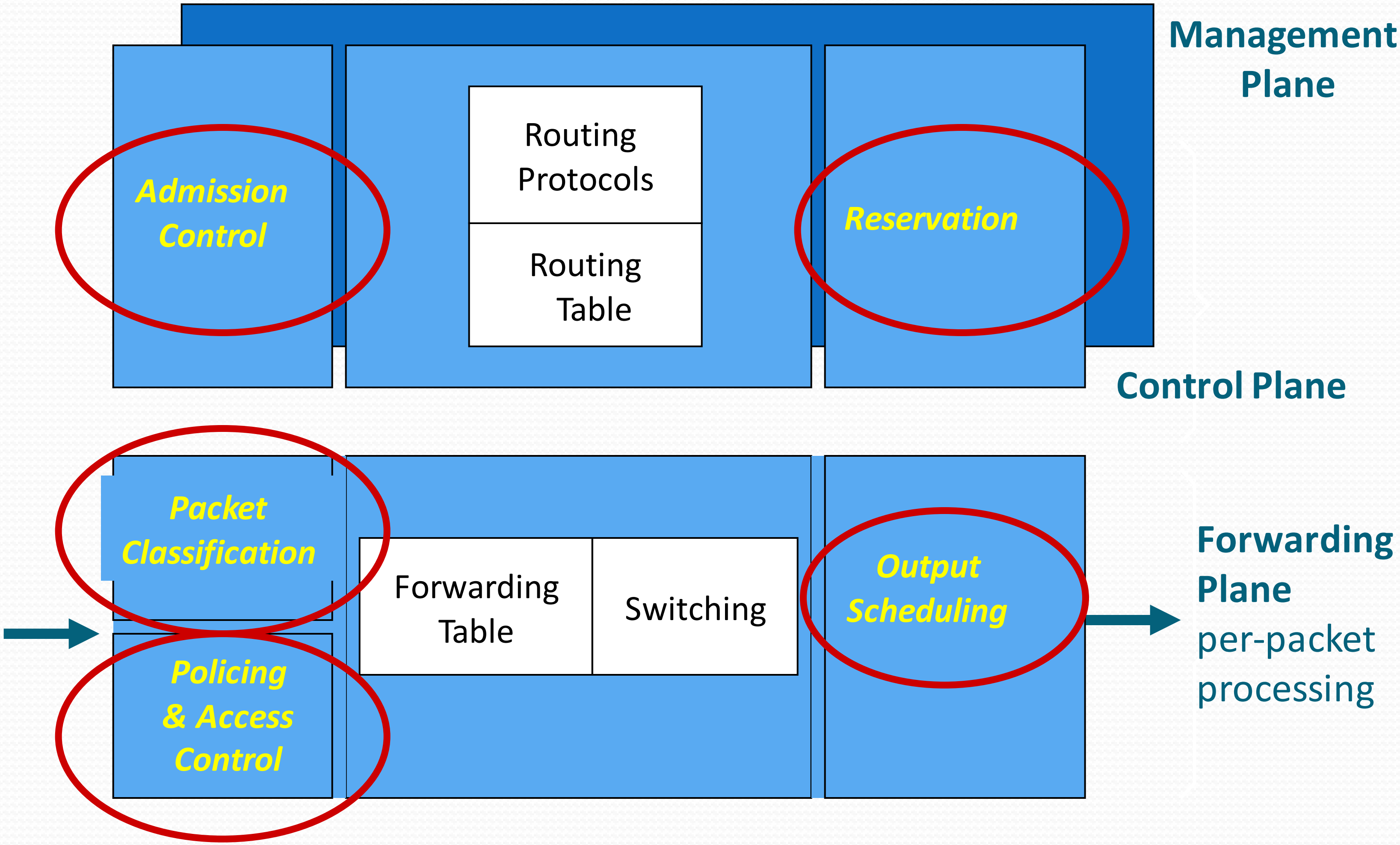


IntServ implemets the following mechanisms:

- **Signaling for Resource Reservation** (**RSVP,** Resource Reservation Protocol)
- **Admission Control** of new connections (**CAC**, Connection Admission Control)
- **Packet classification**, and **queuing mechanisms**, e.g., priority queues
- **Policy function** (**UPC**, Usage Parameter Control, ej: **TBF**: token bucket filter or Leaky Bucket)
- **Scheduling** policies in the routers (**WFQ**, Weighted Fair Queuing)

- **IntServ (Integrated Services):** An application wishing to obtain a certain QoS:
  - Uses the **RSVP signaling protocol** for reserving resources in the routers and partners (it supports multicast)
  - The routers will maintain a **reservation for a flow** (defined as a one-way stream of data with defined destination (IP@, Port, Transport protocol)
  - Routers must implement an **Admission Control** function for verifying that the resource needed are available.
  - In the new connection is accepted, routers must thus **create and maintain** a certain **state** per each flow
  - When a router receives packets, **classify** them depending on the flow they belong to, **police** them to see whether the packets satisfy QoS parameters, and then **process** them using specific schedulers such WFQ.

- **IntServ (Integrated Services):**



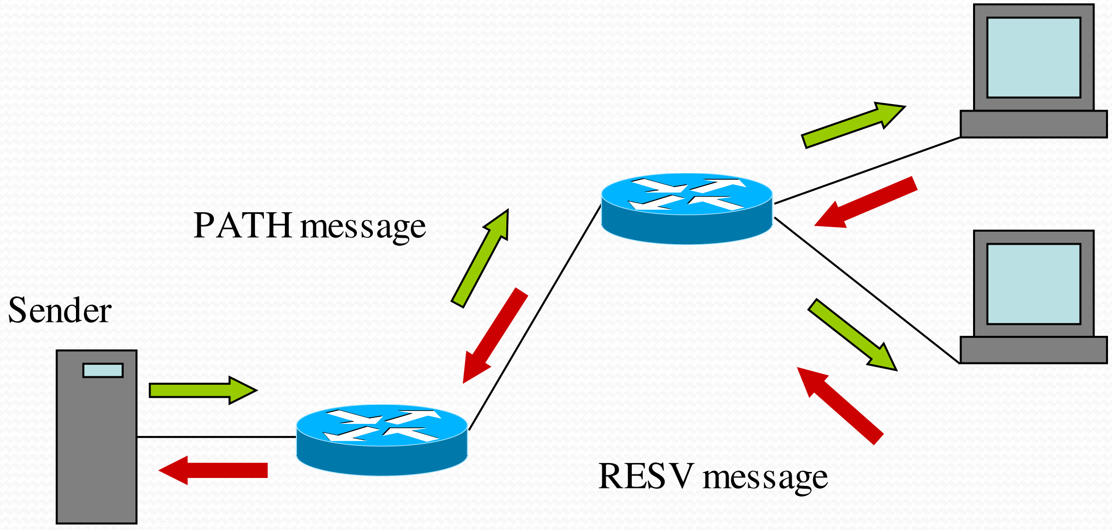
- **Resource Reservation Protocol (RSVP)**
  - It is a signaling protocol which allows the sender/receiver to reserve resources along a communication path
  - It is used by a host on behalf of an application data flow
  - It provides reservations for bandwidths in multicast trees
  - It is **receiver-oriented**, i.e. the receiver initiates and maintains the resource reservation used for that flow
  - Note that RSVP **does not specify** *how* nodes will provide the reservation it requests

- # Resource Reservation Protocol (RSVP)

After receiving a PATH message, receivers can request a reservation
By sending back RESV messages. These messages:

- **PATH messages** specify the resource requirements (bandwidth, delay, losses)
- **RESV messages** set-up the state of routers along the path

After receiving the RESV, sender can start to transmit packets

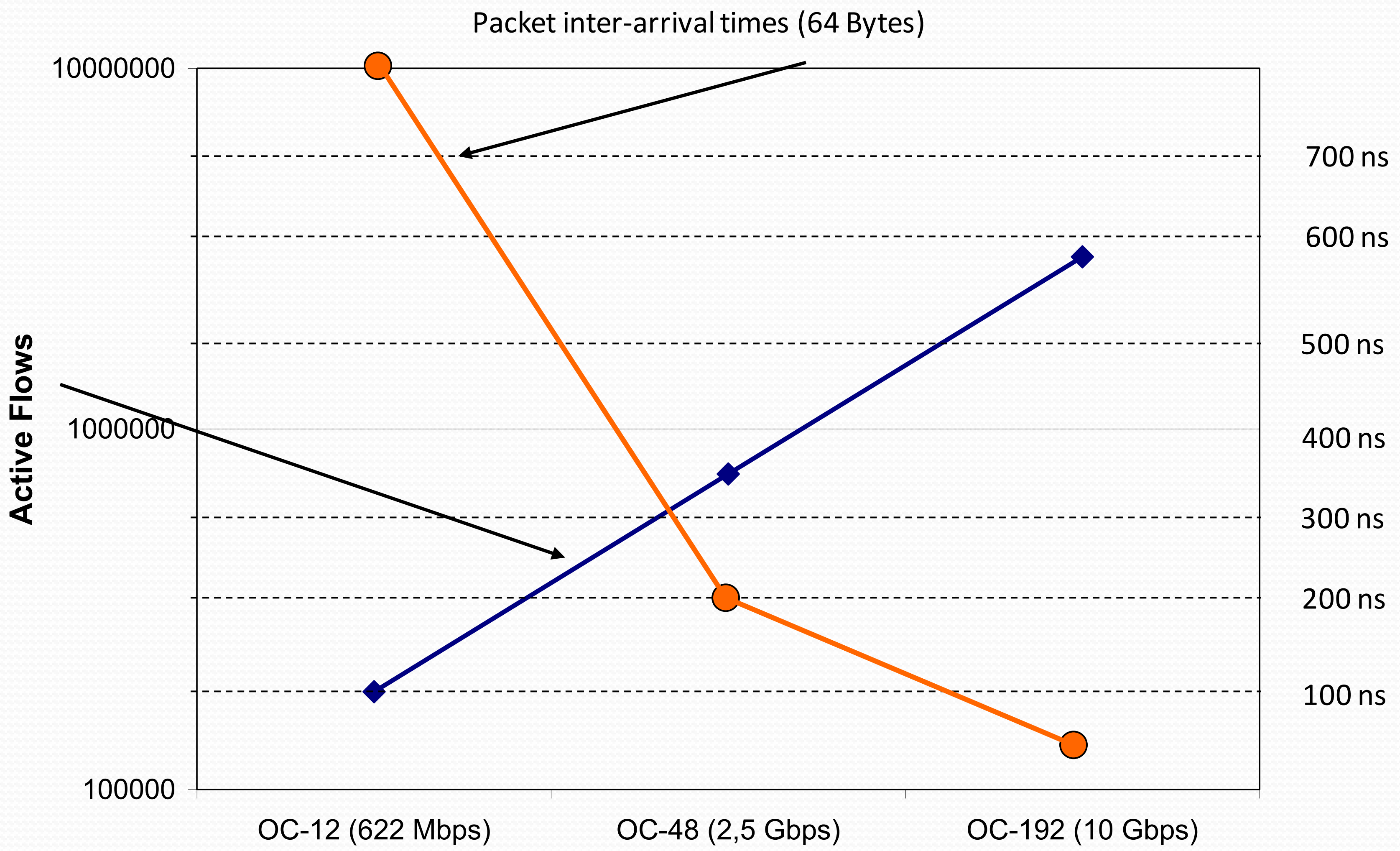


- **Main drawbacks with RSVP:**

  - In the “core network”, is where routers should process x100.000 flows for each link, IntServices presents thus problems of scalability:

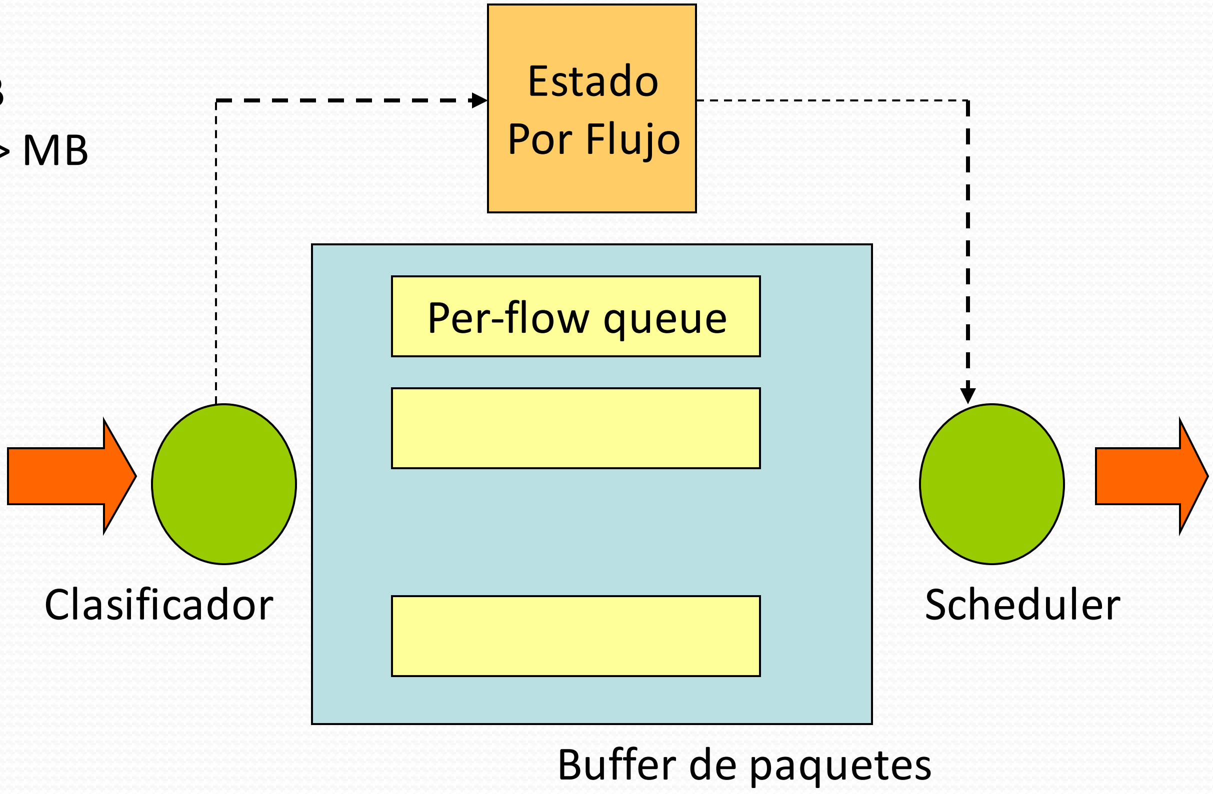
    - control plane (RSVP)
    - forwarding plane (per-flow scheduling)

- **Number of active flows:**

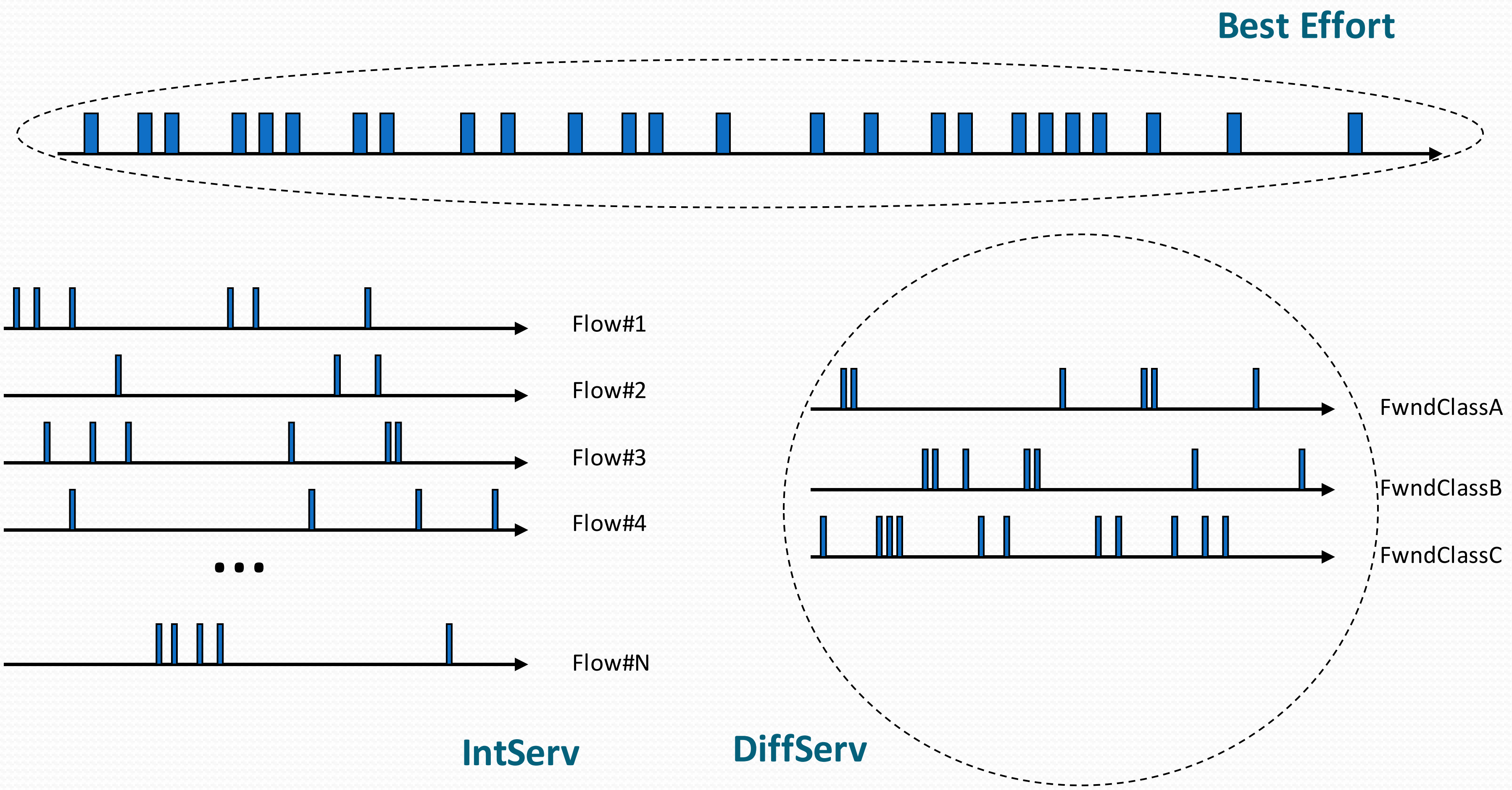


- **Scalability problems in the forwarding plane**
  - Per-flow scheduling: for each new packet read/written the router has to access to the flow variable state (in real-time)

SRAM: 5ns, KB
DRAM: 50ns, > MB

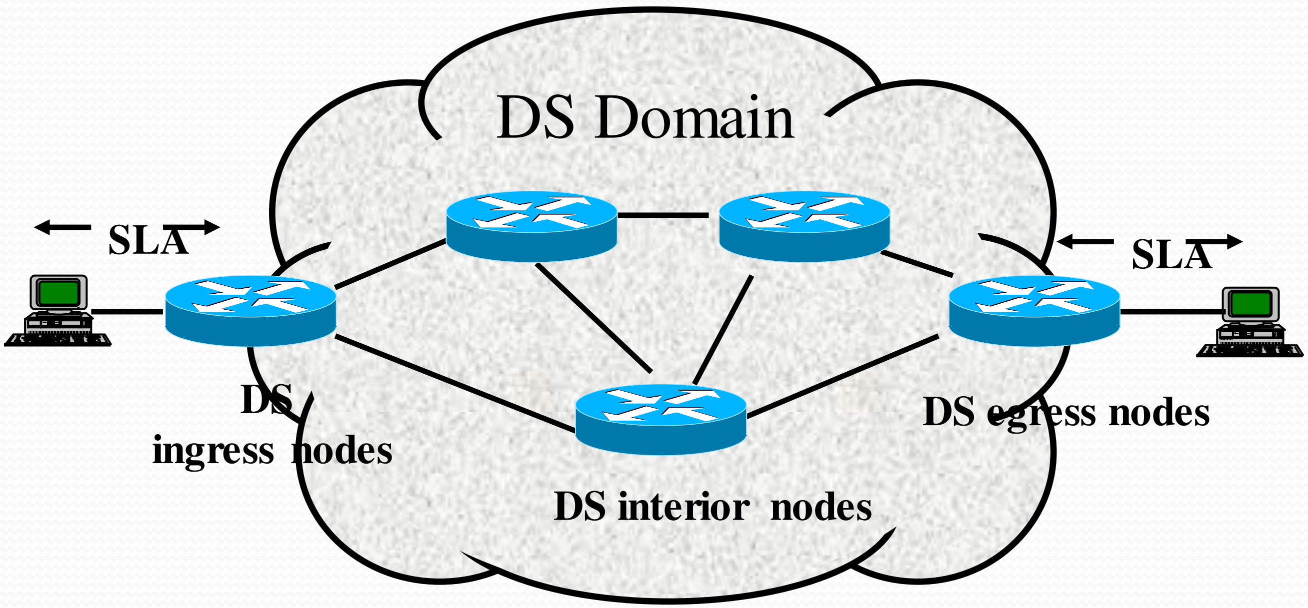


- **DiffServ (Differentiated Services)**
  - Resource reservation is performed as a function of traffic aggregation (**Forwarding classes**) instead than over flows



DiffServ (Differentiated Services):

- Classes of traffic are classified and marked in the IP header (ToS field)
- Each class of traffic receives a specific treatment in the domain (Autonomous System) defined in the SLA (Service Level Agreement)
- The treatment is given using schedulers, policing functions, etc

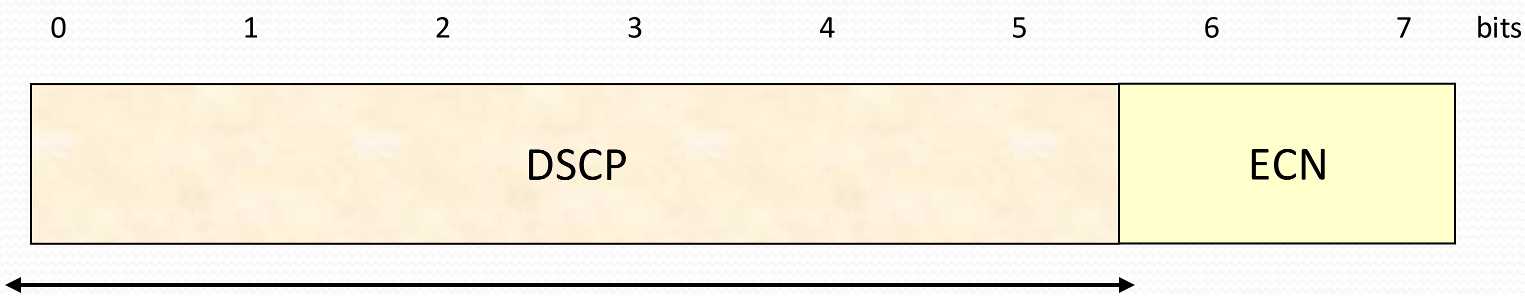


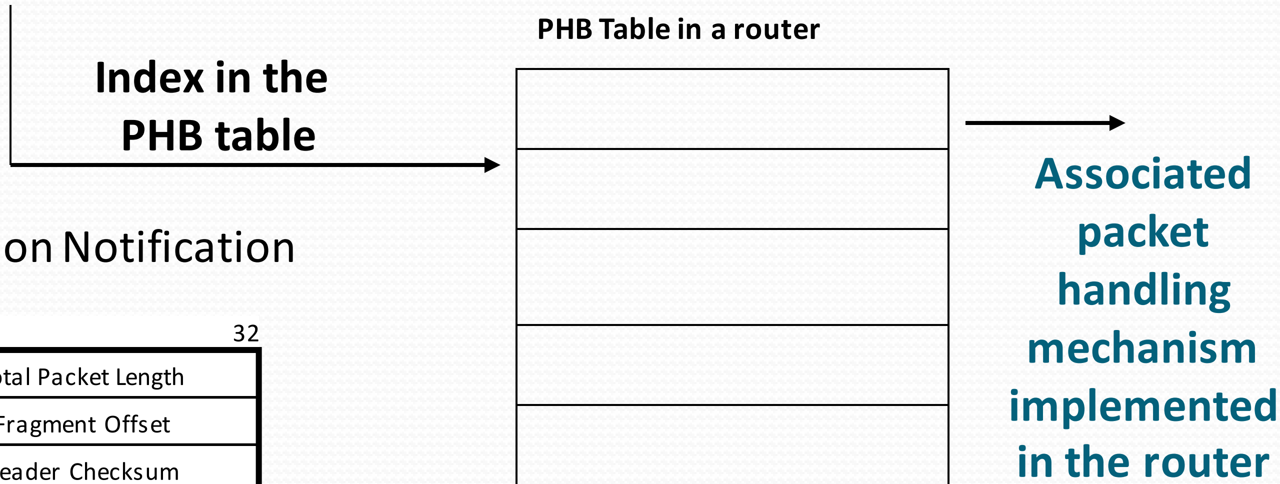
- **Core routers** give different ***treatment*** (**PHB: Per Hop Behavior**) to each packet depending of the byte **Type of Service** (ToS) in the IP header. The ToS associate the packet to a specific **Forwarding Class**

- **Access routers** has the responsibility of classifying and marking packets in one of the defined *Forwarding Classes*

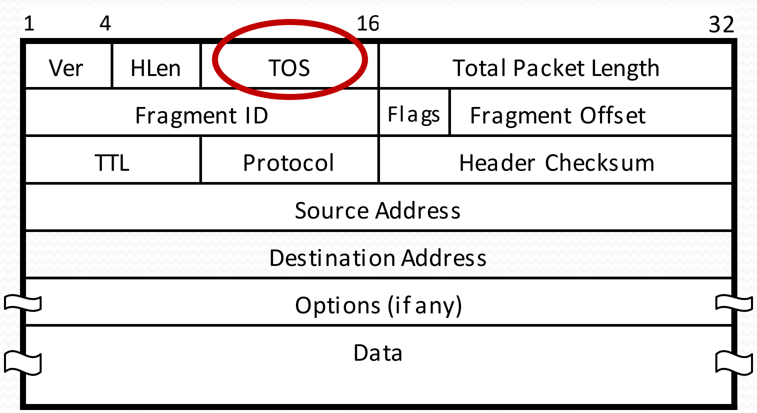
  - → The most costly functions are implemented in the edge of the network, while in the ***core network*** the routers only have to **give different treatments** to the packets depending on a reduce number of classes

- **DS (DiffServ) field:** substitutes the TOS IPv4 field





ECN: Explicit Congestion Notification



- **DiffServ allows defining bussines models, example:**
  - SLA defines a class called **"VoIP"**. The Service Provider (SP) and the client have the following agreement
    - An Ingress Committed Rate (ICR) is defined and an Egress Committed Rate (ECR). Normally, the traffic will be symmetric, so ICR=ECR. The SP guarantees a maximum delay per each direction (i.e, 15 or 30 ms), and a loss ratio (i.e., 0.1 %).
  - The SLA also defines a specific class called **"Business Latency-optimized"**, with specific parameters such as for example, a maximum delay of 30 or 80 ms
  - A third class, **"Business Throughput-optimized"**, does not define a specific guaranteed bandwidth. On the other side, It is agreed that this class will have a 80% of the bandwidth behind of the classes "VoIP" and "Buss-Lat". But there are no delay or loss ratio guarantees with this class.
  - A fourth class, **"standard"**, will receive the rest of the bandwidth (i.e., 20%) after removing "VoIP", "Buss-Lat" and "Bus-throughput"