AV Transport

Transport Principles

Jean Le Feuvre
jean.lefeuvre@telecom-paristech.fr

Streaming

- **Delivery of media streams**
  - AU data + meta-data (timing, RAP, …)
  - Codec configuration
  - Stream configuration (timescales, dependencies ...)

- **Generalization**
  - Media playback technology not requiring local storage of the complete media

Common Principles

- **Ensure Quality of Service**
  - Aware of data loss
  - Aware of networks delays
- **Real-time delivery**
  - « in a given time »
    - Constant delay (dD/dT=0) in theory
      - codec models: constant delay, decoding time=0
    - But not in all networks (especially IP)
  - Ensure synchronization
    - Audio and Video streams
    - Media and meta-data (configuration, annotations, subtitles, …) streams

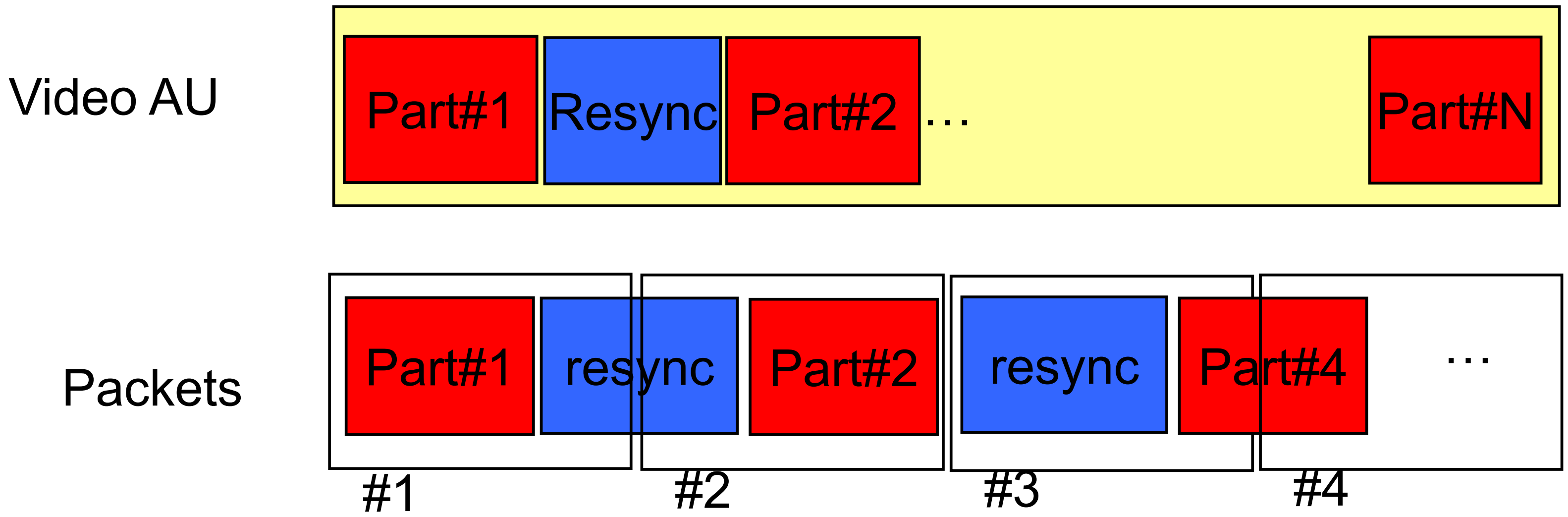
Networks (refresher)

- **Packet-based**
- **Max size of packet allowed**
  - MTU (Maximum Transmission Unit) or Max PDU (Protocol Data Unit)
  - May be configured for some networks
  - Different values
    - Ethernet: 1500 bytes by default
    - WiFi: 2346 bytes
    - IPv4: min 68 B , max 64KB
    - IPv6: min 1280 B , max 64KB (super max 4GB)

Streams and Networks

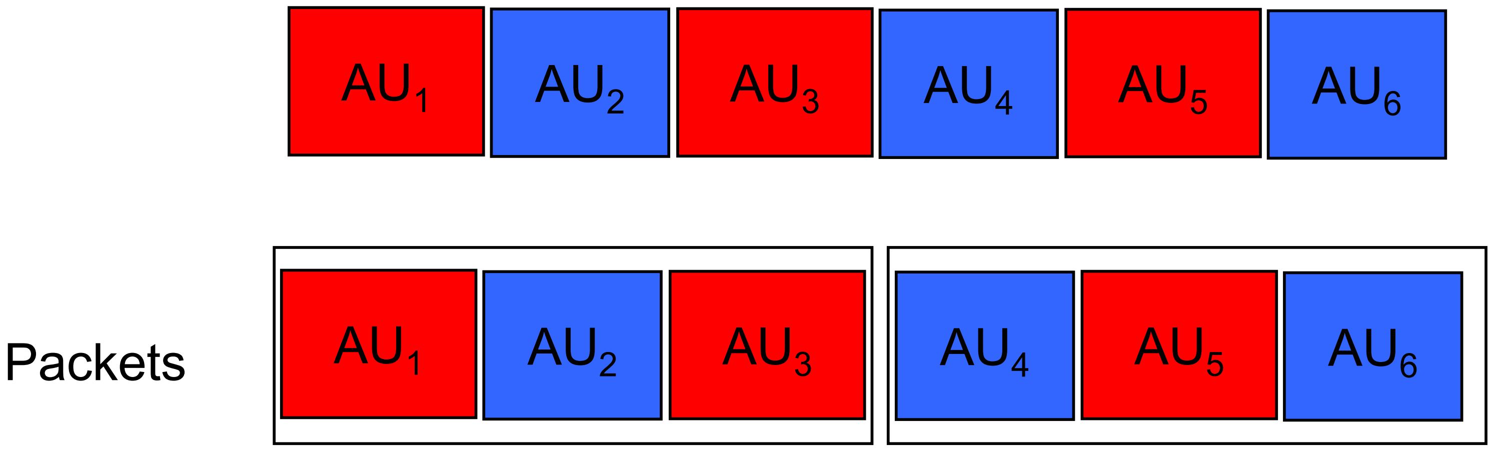
- **Large Video AUs**
  - HEVC UHD @ 15 mbps: Intra: 400->800 kB
  - AVC UHD @ 30 mbps: Intra: 800->1600 kB
  - AVC HD @ 8 mbps: Intra: 200->400 kB
- **Small audio AUs**
  - AAC 44100 Hz stereo @ 156 kbps: 300->400 B
  - AAC 22050 Hz 5.1 @ 130kbps: 500->800 B
- **Need for**
  - *Fragmentation:* cut AU into smaller packets
  - *Aggregation:* gather N AUs into a single packet

AU Fragmentation



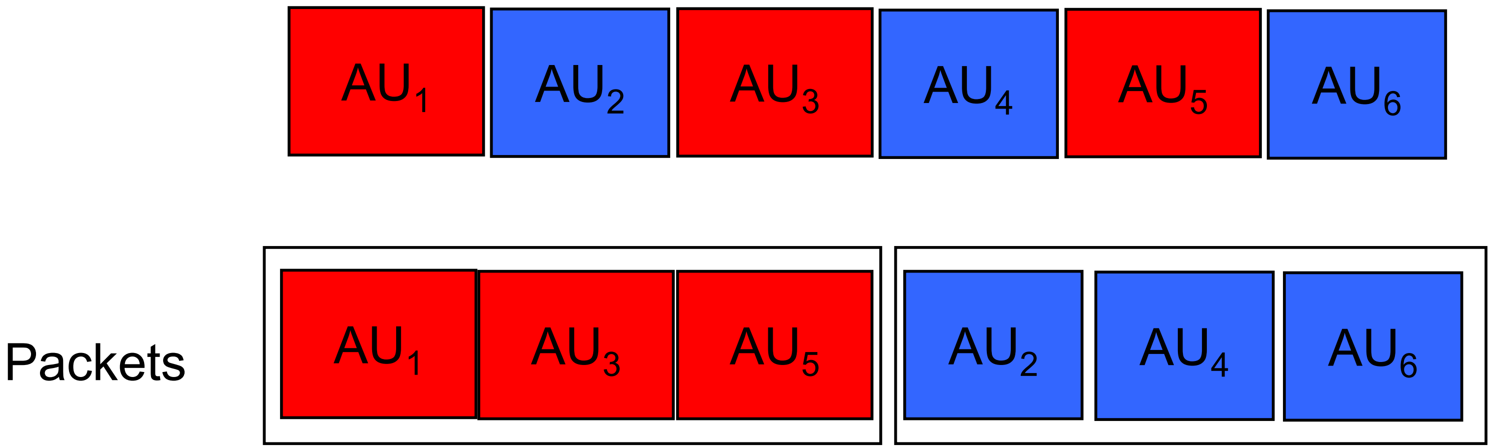
- **Take advantages media bitstream syntax**
  - start codes, resync markers …
- **Media unaware fragmentation**
  - #1 lost => #2 unusable: more losses
- **Media aware fragmentation**
  - #2 lost => #3 usable

Simple AU Aggregation



- **Packing N AUs in 1 packet**
  - Saves N-1 packet sending over the network
  - Introduces (N-1) AUs delay in the delivery chain
- **Data loss**
  - If 1 packet is lost, N AUs are lost
  - Less robust than single AU per packet

Interleaved AU Aggregation

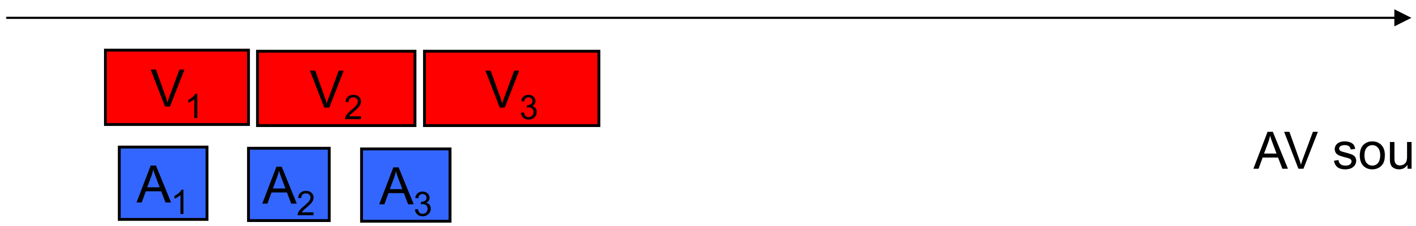


- **Spread potential data loss over time**
  - Error concealment possible with received AUs
  - Used for some audio codecs

Multiplexing

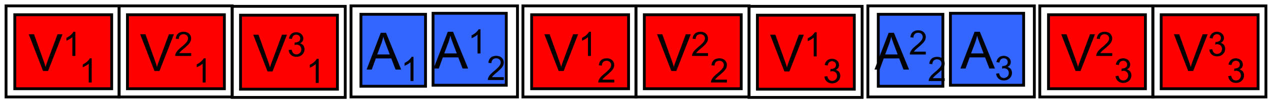
- **Many streams to deliver to a single destination**
  - Audio, Video, Subtitles, …
- **Network level multiplexing**
  - Some protocols (UDP, TCP) allow for port addressing
  - Send one stream per port
    - Simple but uses many ports
    - Problematic on some devices (OS restrictions)
    - Problematic if each port has to be explicitly opened
      - NAT traversal
- **Transport Protocol multiplexing**
  - All packets sent to the same logical network channel
  - Stream identifier assigned to each packet

Multiplexing example

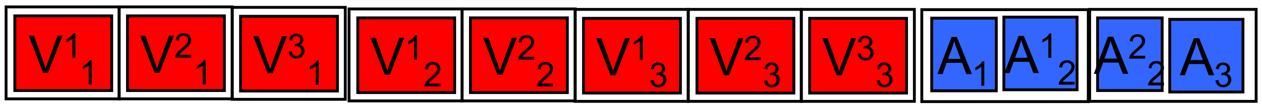


AV source

- **Time-based multiplexing**
  - Based on presentation time



- **Burst multiplexing**
  - Send by burst of N sec



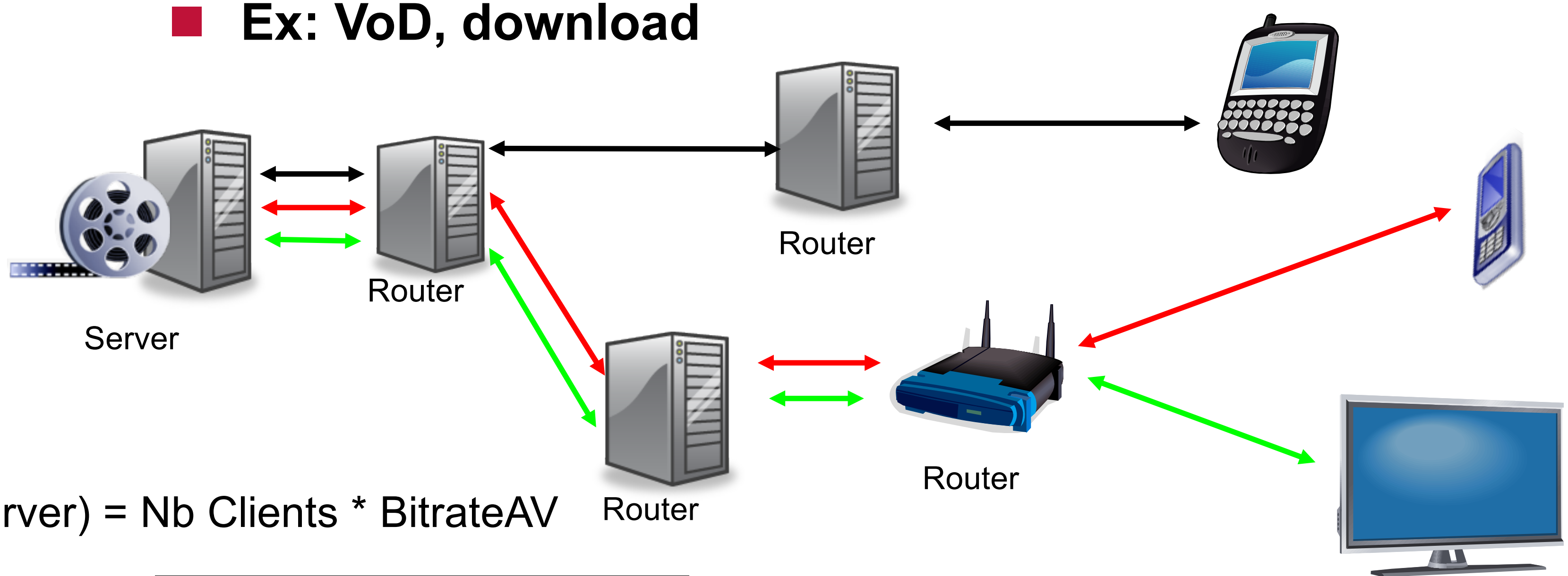
  - Higher buffering requirements
  - Higher latencies

Stream Delivery

- **Various services**
  - Point to point delivery (Video on Demand)
  - Live broad (one-to-many) delivery (TV, Live IP)
  - Broad delivery offline or slight delay (P2P, HTTP)
- **Various devices**
  - Visual capabilities
    - Codecs, spatial and temporal resolution, bits per pixels
  - Audio capabilities
    - Codecs, bits/sample, sample rate, nb channels
- **Various Networks**
  - Fixed or variable delay
  - Heterogeneous networks

Centralized on demand delivery

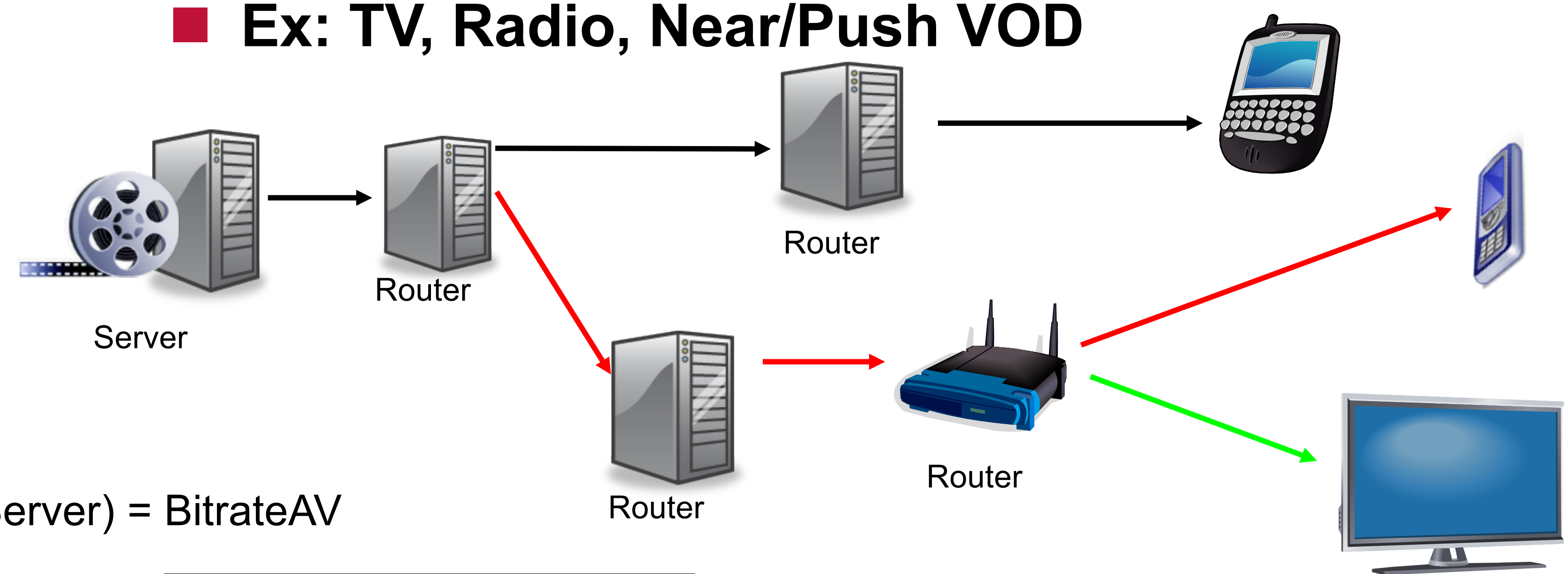
- **Principles**
  - Content retrieved per user
  - Resources allocated per session along the path
    - Important server request capacity (nb servers, load balancing)
    - Server output bandwidth high
  - Connected mode
    - Back channel information (QoS metrics)
    - Adaptation to user/terminal
- **Ex: VoD, download**



BW(Server) = Nb Clients * BitrateAV

One-to-many centralized delivery

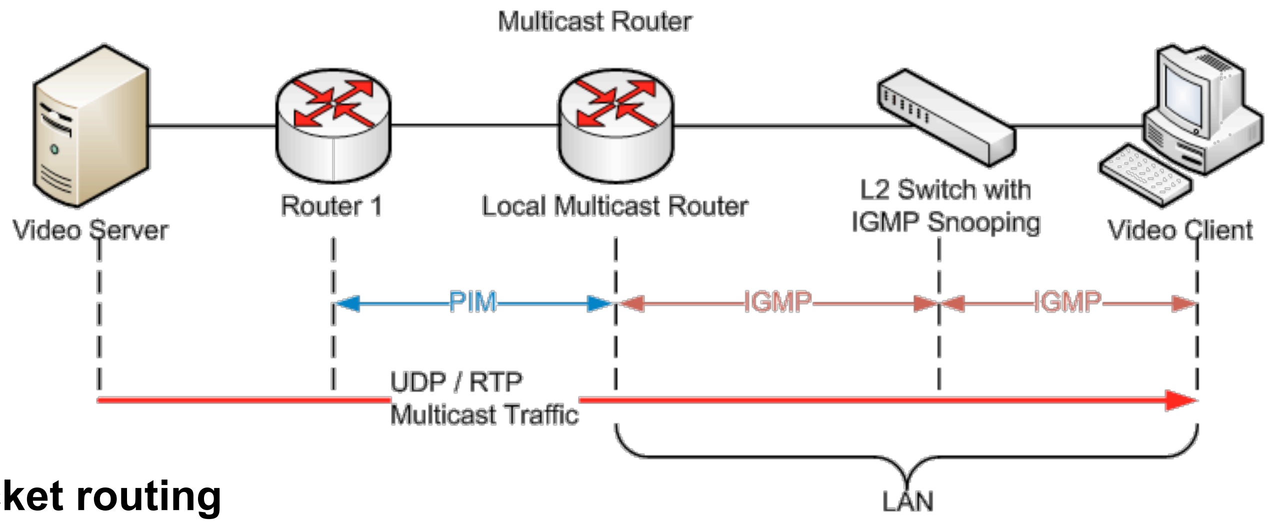
- **Principles**
  - Same content delivered to al users
  - Continuous usage of network (ex: satellite)
    - Specialized Infrastructure
    - Optimized (servers, bandwidth)
  - One-to-many architectures
    - « Broadcast » or « Multicast »
- **Ex: TV, Radio, Near/Push VOD**



BW(Server) = BitrateAV

Multicast (refresher)

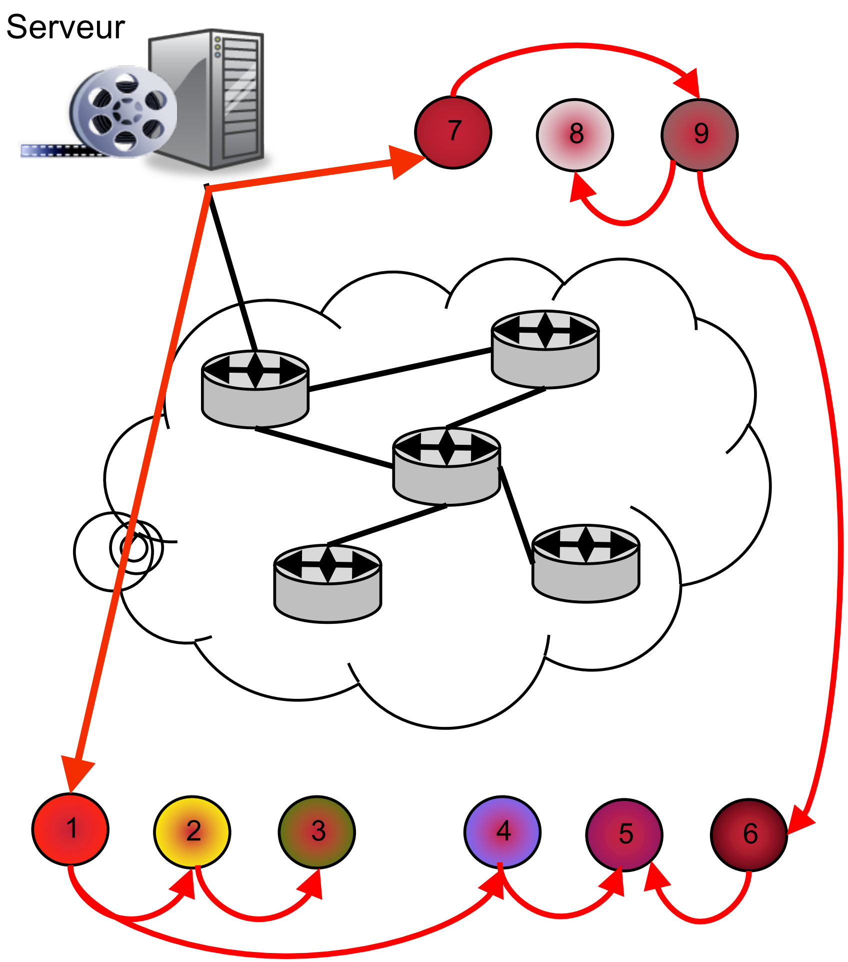
- **Deployment**
  - First IPv4 routers: NO MULTICAST
  - IPv6: all routers shall support it
  - In practice, some providers may block multicast traffic
- **Multicast Groups**
  - Explicit joint requests by hosts through IGMP protocol
  - Sender is not required to belong to the group



- **Packet routing**
  - Each packet has a “Time To Live” counter
  - If TTL = 0, packet is discarded
  - Otherwise, packet forwarded with TTL -= 1

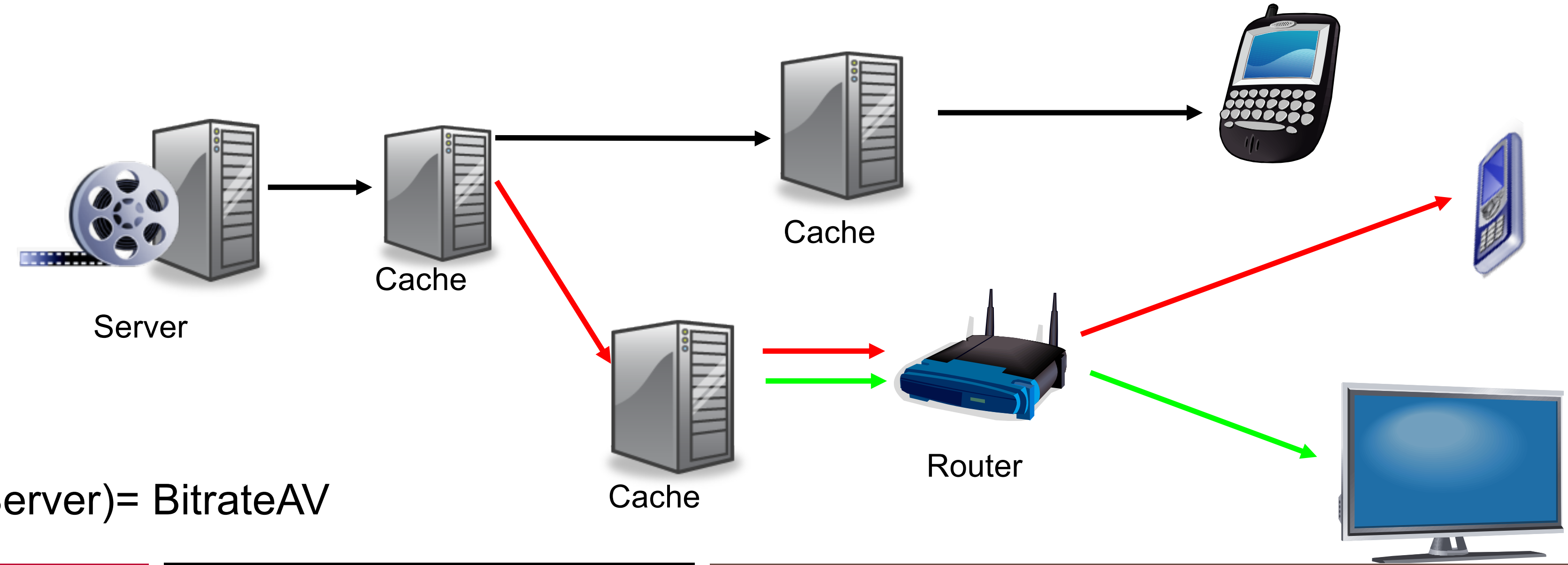
Distributed delivery

- **Principles**
  - Each client is a server
  - Usage of both download and upload links even at network edges
  - Lower server load
- **Examples:**
  - P2P Networks
  - Some video-conferencing systems



Content delivery networks

- **Principles**
  - Re-use existing web architecture
  - Content is cached in the network
    - Either after client request
    - Or pushed by the origin server
- **Ex: HTTP Streaming**

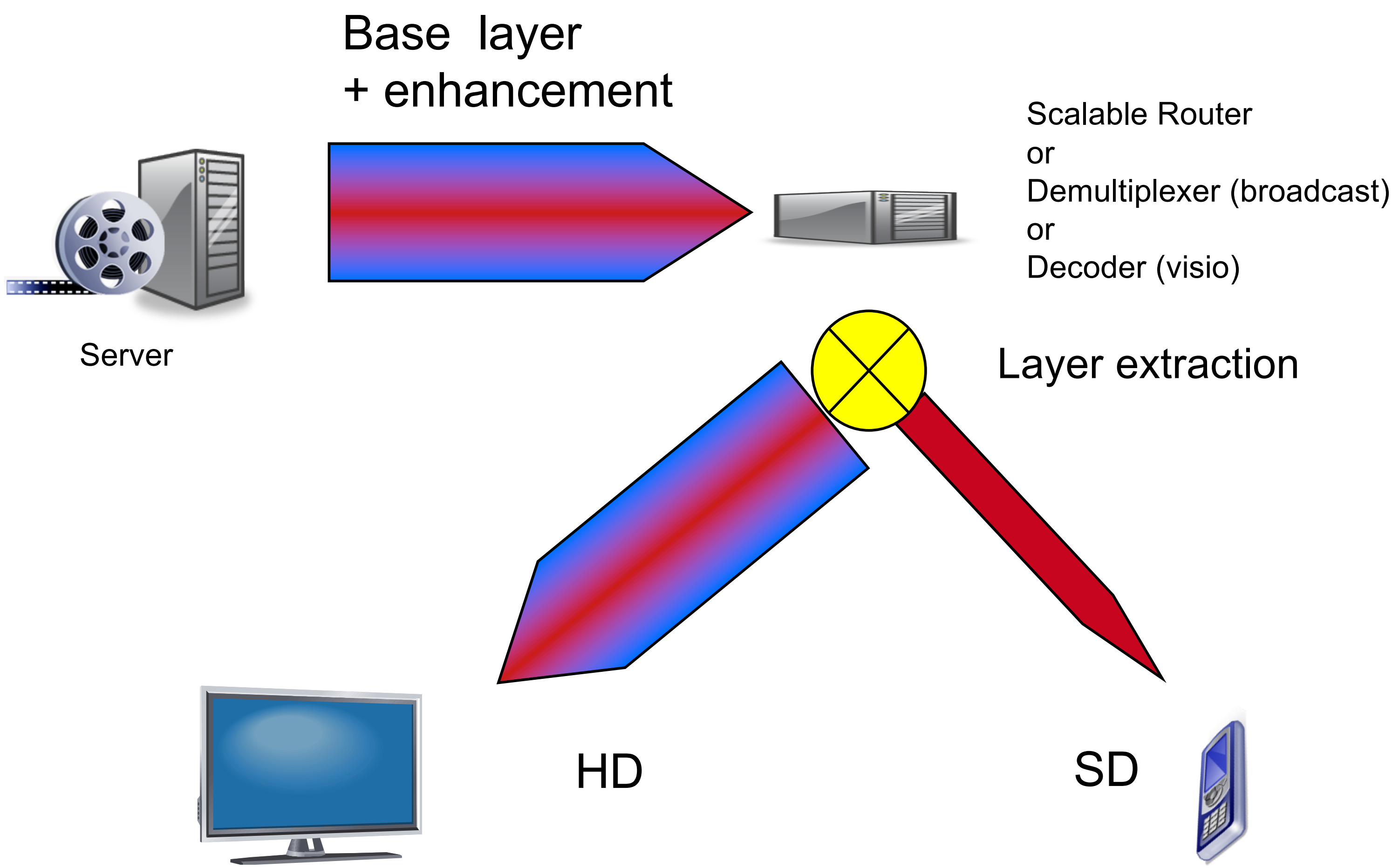


BW(Server)= BitrateAV

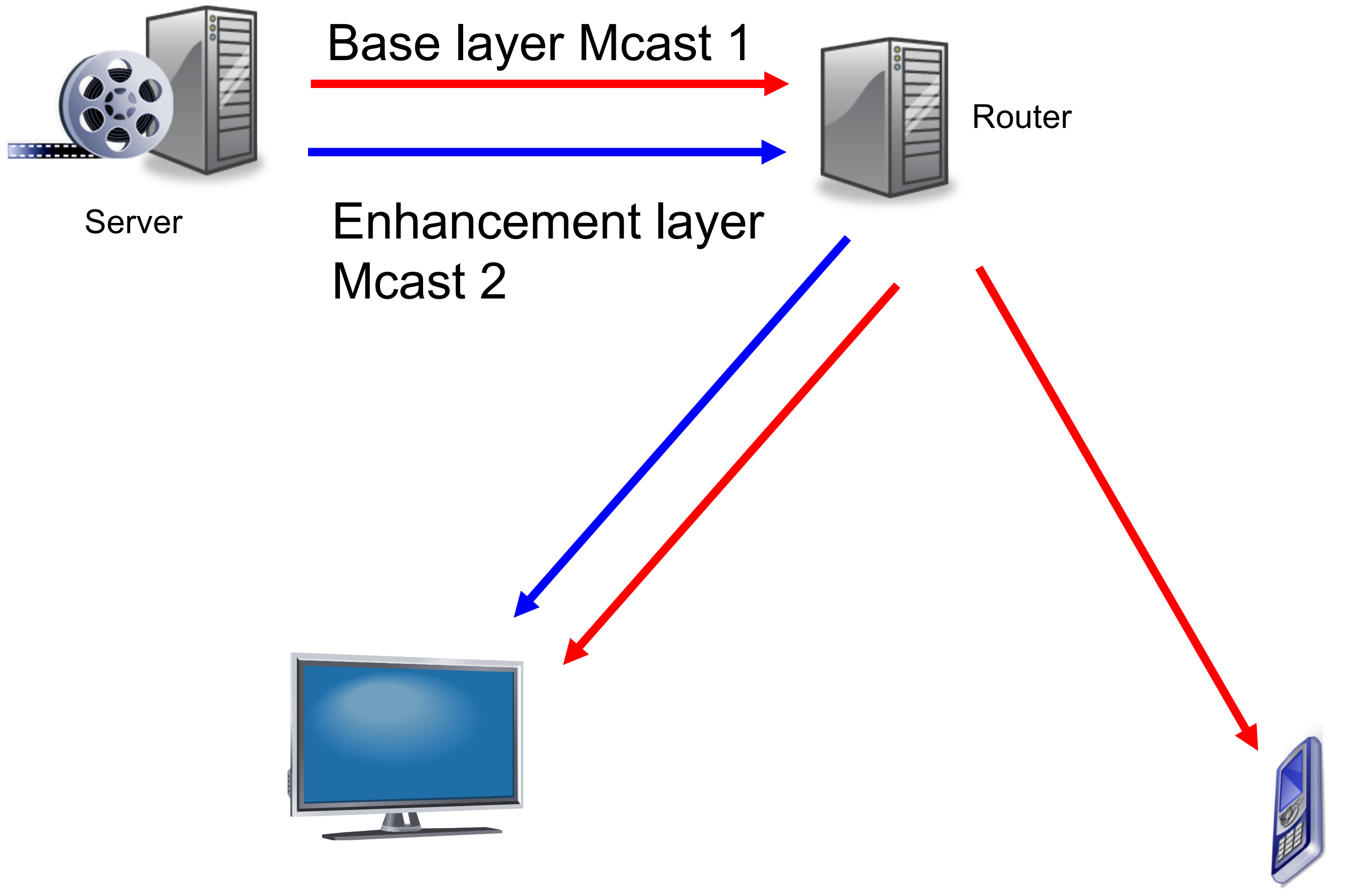
Scalable Delivery

- **Variable delivery conditions**
  - Network (Congestions, heterogeneous networks)
  - User/Terminal (CPU, memory, display settings)
- **Solutions**
  - Alternate versions for each media (« Stream Switching »)
    - More storage required
    - Not convenient for multicast distribution
    - Compression friendly
  - Hierarchical Coding
    - Base layer common to all delivery scenarios
    - Enhancement layers for all other cases
    - Well suited for multicast distribution
    - Less efficient in terms of compression
      - Typical overhead is 20%

Scalable Delivery



Scalable delivery over multicast



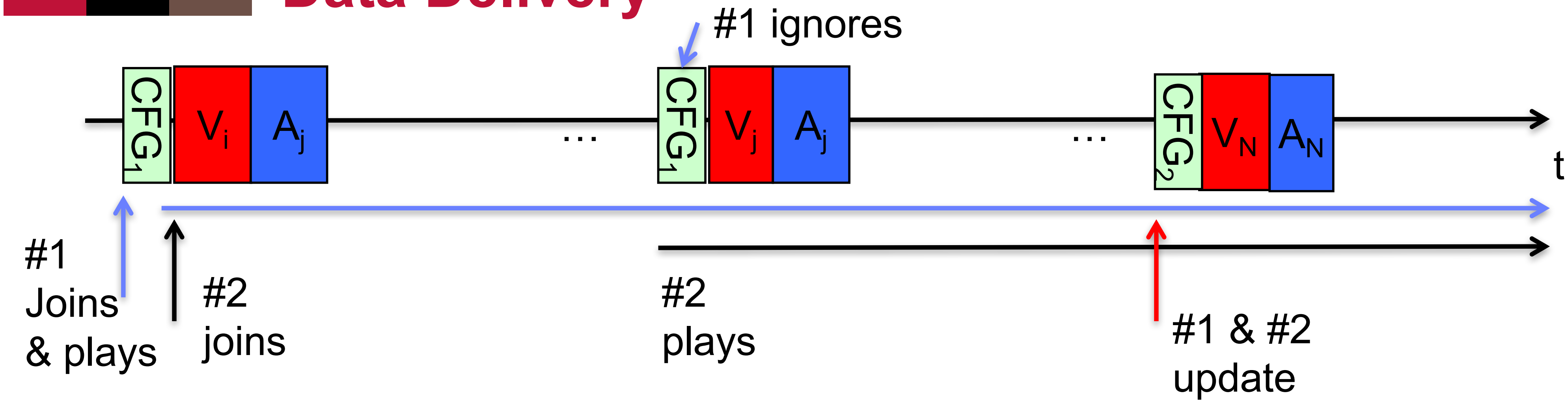
Uses both multicast groups

Uses only multicast group 1

Broadcast Delivery

- **Unidirectional link**
  - No feedback on quality
    - Lost packets cannot be retransmitted
    - Usage of FEC tools
  - No dialog client-server:
    - All required info shall be in the content
      - Service Configuration
      - Stream Configuration (type, language, …)
      - Codec configuration
    - **! All users receive the same data !**

Data Delivery



- **In broadcast/multicast**
  - Data sent with other AV streams (configuration, logo, …)
  - Some shall be repeated
    - Configuration
    - Low repeat rate may increase join/tune-in time
  - May be updated
    - Configuration changed
    - Usually done through versioning
- **Terminology**
  - “Carousel”

Processing a stream

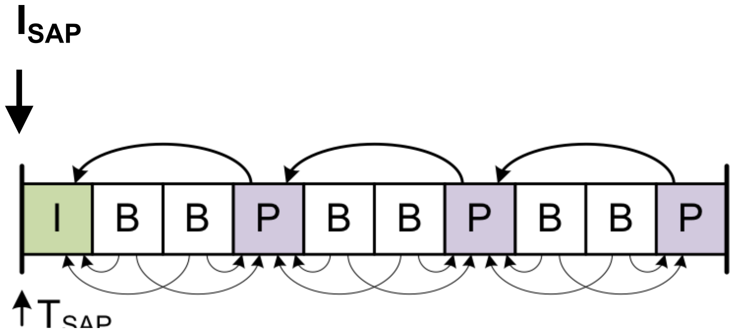
- **Media stream**
  - Wait for packets
  - Gather packets of one AU, possibly re-ordering them
  - AU completed, check type
    - Not random access: discard, go to step#1
    - Random access: start buffering
  - Buffering done, playback ready
- **Carousel Data**
  - Wait for packets
  - Gather packets for carousel data, possibly re-ordering them
  - Carousel completed, check version
    - Same as previously processed carousel: discard, go to step#1
    - Otherwise process data
- **Caution**
  - In some scenarios, carousel data may contain media streams (movie files, ring tones)

Stream Access Point

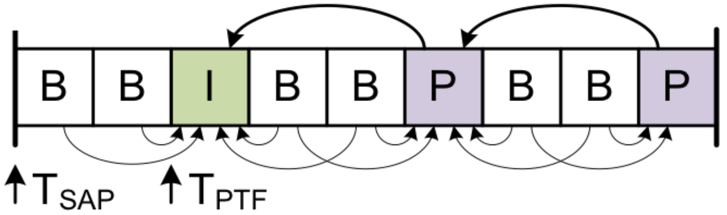
- **Generalization of RAP (Random Access Point)**
  - I / IDR frame
  - Boolean
- **Common other cases**
  - few non-decodable picture following RAP (in decode order) but to be discarded (because before the RAP in display)
  - A succession of non-RAP AUs may result in a « RAP » state
- **SAP: Stream Access Point**
  - Allows quick identification of random access types
  - Mainly
    - Regular RAP (I-frame / IDR)
    - OpenGOP
    - Gradual Decoding Refresh

SAP Types

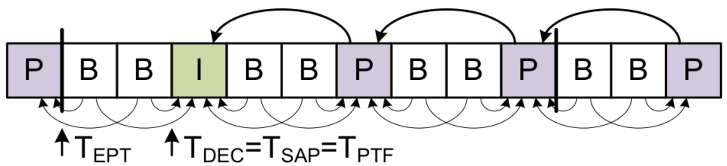
- 1: Closed GOP



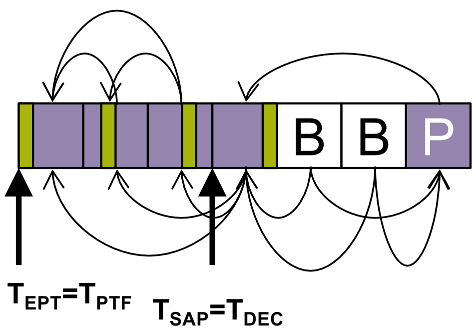
- 2: Closed GOP



- 3: Open GOP



- 4: GDR

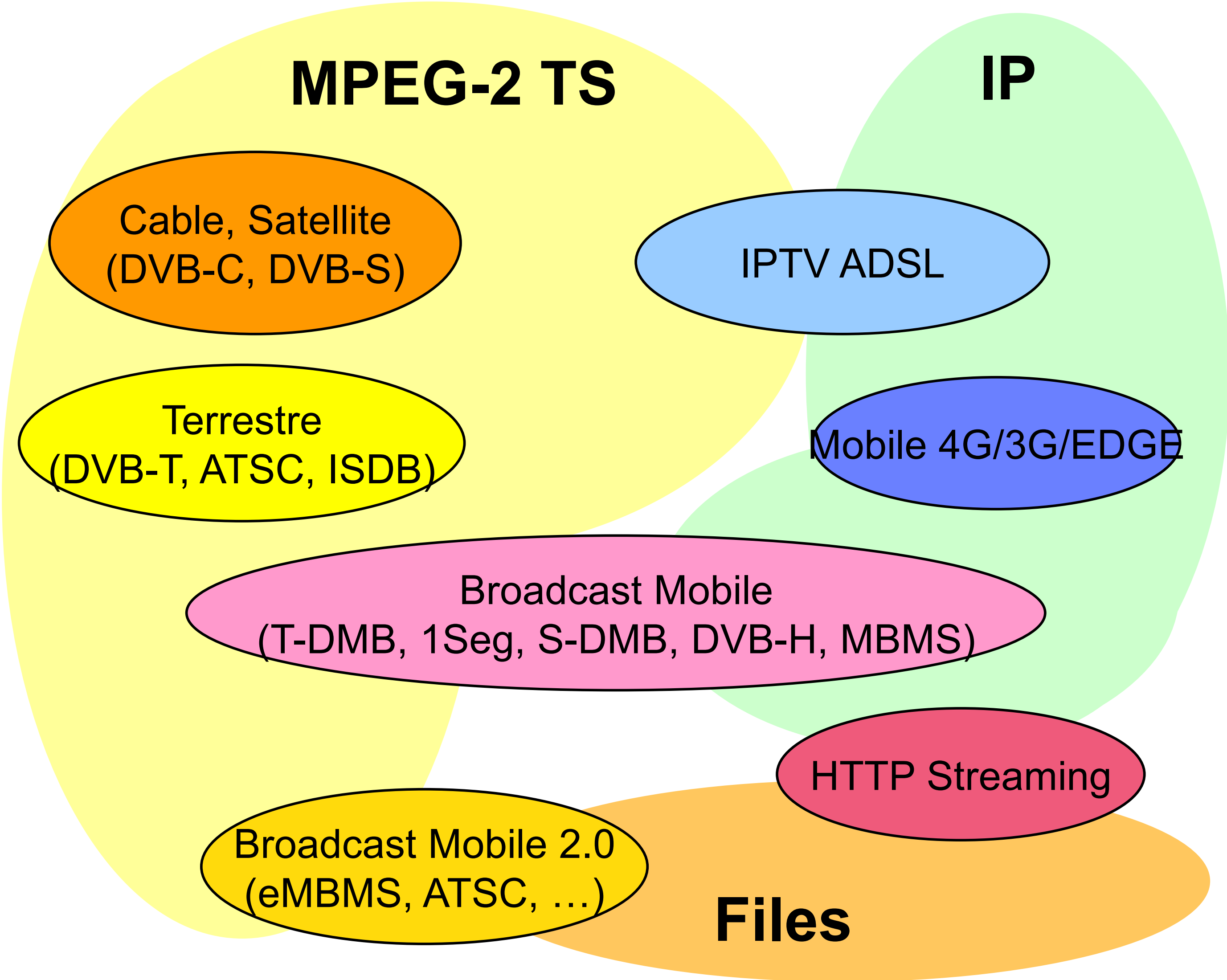


- I_{SAP}: byte in bit stream from which decoding can start with no dependency on data before I_{SAP}
- T_{SAP}: PTS from which decoding can start with no dependency on data before I_{SAP}
- T_{PTF}: smallest PTS of frames following I_{SAP}
- T_{EPT}: PTS of the frame corresponding to I_{SAP}

Common packet header fields

- **Destination info**
  - If multiplexing used
- **Timing information**
  - Presentation time: always
  - Decoding time: only for strict buffer management
- **Data information**
  - AU size or end of AU
  - Random access type (SAP)
- **Error detection if needed**
  - Packet sequence number
  - Packet checksum

AV Transport protocols



AV Transport

Real-Time IP Streaming

Requirements

- **Real-time AV Transport**
  - handle jitter
  - handle loss
- **Various media types**
  - Standard formats for media description
  - Adapt media to transport
    - Fragmentation rules
- **Video on Demand**
  - Stream Selection
  - Stream Control
- **Stream Protection**

File download

- **Pros**
  - Audio/video synchronization is guarantee
    - interleaving done in the file
  - Deployment costs
    - regular HTTP servers
  - No packet loss
- **Cons**
  - Playback may re-buffer
  - Not efficient for real-time transport
  - Random access difficult
  - Requires storage capacity
  - Scalability difficult
    - How to “thin” a file / remove layers ?

Stream Transport

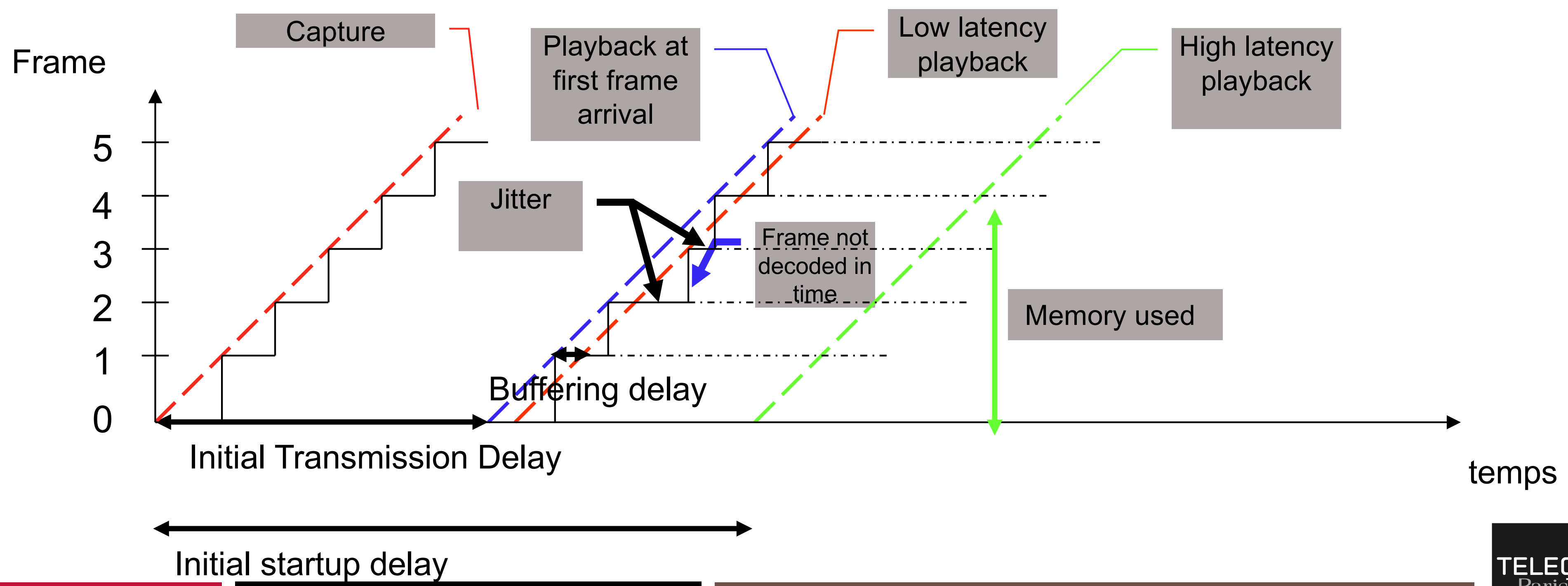
- **Audio Visual Stream**
  - Sequence of data (frames, application instructions)
  - Strong timing constraints
- **Pros**
  - Low memory usage
  - Low Latency
  - Enables Scalability
- **Cons**
  - Packet loss
  - Requires dedicated servers
    - Deployment cost

Real-time Transport

- **Lower transport transport time**
  - Prefer UDP rather than TCP
- **Handle link quality**
  - Latency and Jitter
    - Unpredictable changes, loss of order: use sequence numbers per packet
    - Variable (jitter): need to “de-jitter”
  - Packet losses
    - FEC or other redundant sending mechanism
    - Loss retransmission ?
      - Possible but heavy and not commonly deployed
      - Impacts latency
  - Tell the sender about the link quality!

De-jittering

- **IP delays origin**
  - Packet queuing and scheduling in routers
  - Route modifications
- **Jitter: delay variation**
  - Use de-jittering buffer to « never » experience buffer under run
  - Increase global latency



Reminders: IP, UDP, TCP

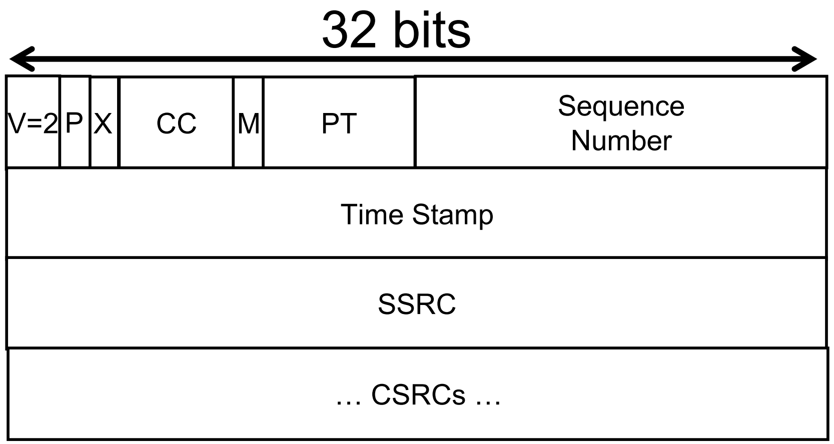
- **Internet Protocol**
  - RFC 791 (IPv4)
  - Data fragmentation
  - Source and destination addressing
- **User Datagram Protocol**
  - RFC 768
  - Data Multiplex (port)
- **Transmission Control Protocol**
  - RFC 793
  - Reliable Transmission
    - Scheduling Rules
    - Loss detection and recovery
    - Flow Control

RTP

- **Real-time Transport Protocol**
  - IETF, RFC 1889 -> RFC 3550
- **Transport de données média temps-réel**
  - Mode unicast et multicast
  - Conférence Audio/Vidéo, diffusion média
  - Indépendant des médias transportés (audio, vidéo, application, …)
    - Segmentation/assemblage des données via autres RFC

RTP – RFC3550

- **Real-time data transport**
  - Mode unicast et multicast
  - Audio/Video conference, media streaming
- **Synchronization**
  - Time stamps (Presentation time)
    - **No DTS**
  - NTP as common clock
- **Packet sending:**
  - Sequence number for re-ordering
    - Random seed !
  - Sent in decoding order



- **M (marker)**
  - For video, usually indicates end of frame
- **PT (payload type)**
  - Indicates media type in payload
  - Association through SDP

Formats Media sur RTP

- **Indicated via « PayloadType »**
  - Range [0, 95] for pre-defined media types (H263, …)
    - Except 72 et 73 (RTCP reserved). RFC 3551
  - Other media types use range [96, 127] via out-of-band configuration (SDP)
- **Packetization rules**
  - AU Fragmentation OK (video, …)
  - AU Aggregation OK (audio, …)
  - Fragments aggregation forbidden
    - No end of frame+start of next frame in 1 packet
- **Configuration via SDP**
  - Decoder Configuration
  - Payload Configuration
  - Media Profiles and Levels

- **RFC 2190: H263**
- **RFC 2250: MPEG-1, MPEG-2**
- **RFC 3119: MP3**
- **RFC 3267: AMR Audio**
- **RFC 3640: MPEG-4 ES**
- **RFC 3984: AVC/H264**

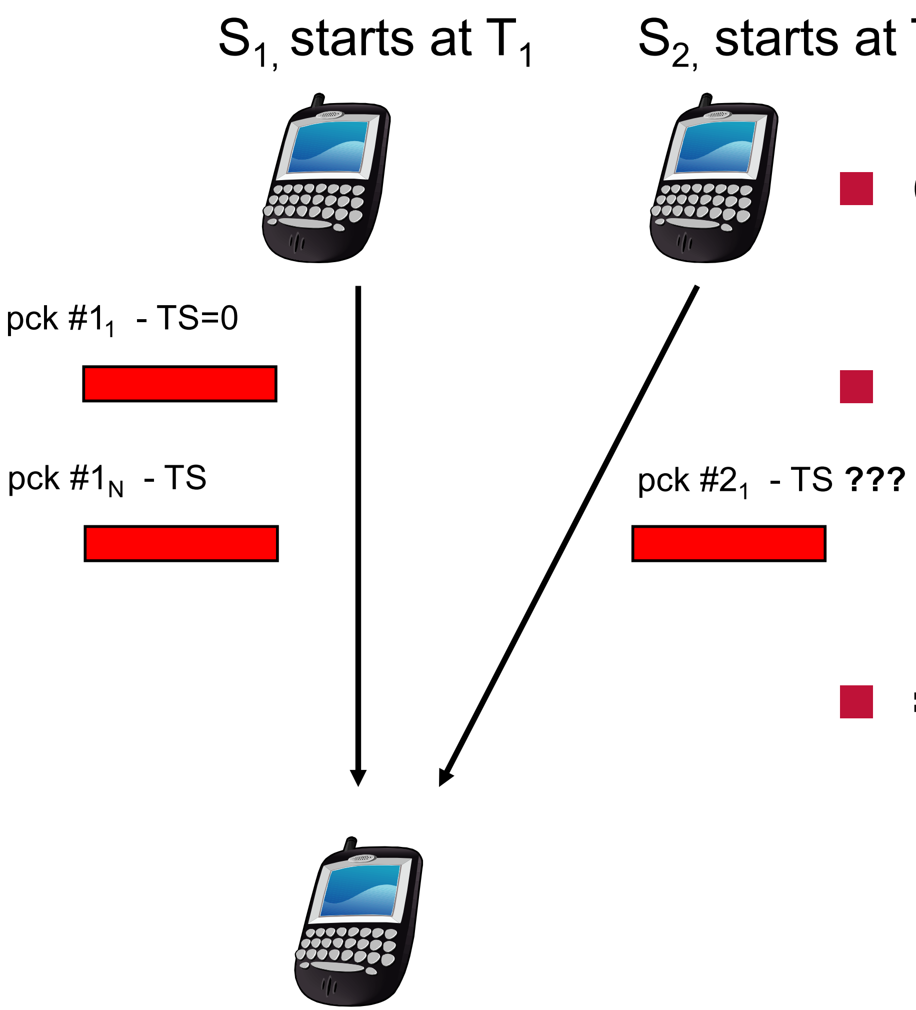
SDP (Session Description Protocol) RFC 2327

```
v=0
o=DarwinStreamingServer 3357901904 1146138454000 IN IP4 137.194.65.248
s=Movie Channel #1
u=http://media.enst.fr/channels
e=admin@media.enst.fr
c=IN IP4 137.194.65.248
b=AS:435
t=0 0
a=control:*
m=video 8000 RTP/AVP 96        //video
b=AS:313                       //bandwidth video
a=rtpmap:96 H264/90000   //format de transport
a=fmtp:96 profile-level-id=4D401E; packetization-mode=1; sprop-parameter-
    sets=J01AHppigRBIsMSgoKC+AAADAAIAAAMAZSg=,KO48gA==
m=audio 8002 RTP/AVP 97
b=AS:122
a=rtpmap:97 MPA/44100/2
```

RTCP, Real-time Control Transport Protocol

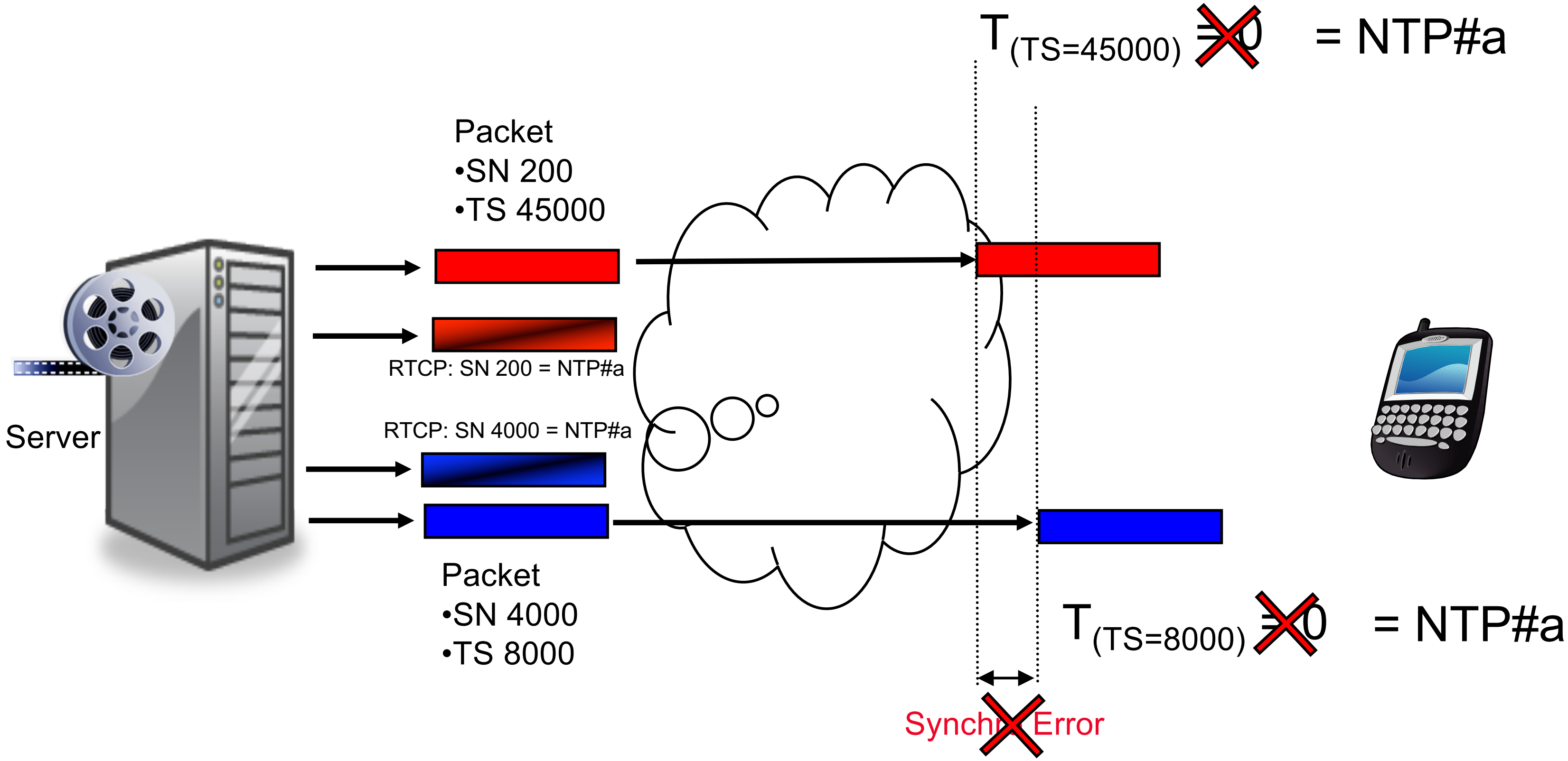
- **Real-Time Transport Control Protocol**
  - Defined with RTP for QoS measurement
- **Identification tools**
  - Textual ID of each source (CNAME)
  - Peer departure messages (BYE)
- **Synchronization tools**
- **Constraints**
  - Regular RTP : each RTP has an RTCP stream
    - Even port for RTP
    - (Port RTP)+1 for RTCP
  - RTP Visio: multiplexing on single port OK

Multiple streams synchronization
Multiple Sources



- **Choosing the timestamp**
  - TS=0: synchro shift T_2-T_1
  - TS=T_2-T_1
- **How to estimate T_2-T_1 ?**
  - Communication S_1 <-> S_2
  - Requires location of S1
    - Complexity if Nb S>2
    - S_1 can leave at any time
- **=> Synchro RTP/RTCP**
  - TS(S_i,1) = Random Offset
  - RTCP Report
    - Per source
    - Gives TS(Si, j) ⇔ NTP

Multiple streams synchronization
Single Source



RTCP

- **QoS measurements through reports**
  - Receivers (s) <-> Sender
  - Info set
    - Synchronization: associate NTP and RTP time
    - Number of expected and loss packets
    - Jitter computed based on NTP time of reports
    - Last report time (for round-trip delay estimation)
- **Bandwidth adjustment**
  - Depends on number of users
  - RTCP bandwidth < 5% of RTP
    - 25% for active sources
    - 75% for receivers

Jitter Estimation

- **Jitter/packet**
  - $NTP_R(i) = L(i) + NTP_S(i)$
    - NTP_R : NTP at reception of packet i
    - NTP_S : NTP at sending of packet i
    - L(i): latency induced by network for packet i
- **Simplified estimation**
  - $NTP_{S \text{ is unknown}}$ is unknown
  - But TS(i) is known
    - Real-time sender assumed

$$NTP_S(i+1) - NTP_S(i) = TS'(i+1) - TS'(i)$$

TS' RTP timestamp in same scale as NTP

$$NTP_R(i+1) - NTP_R(i) = L(i+1) - L(i) + TS'(i+1) - TS'(i)$$

$$D(i) = L(i+1) - L(i) = NTP_R(i) - TS'(i) - NTP_R(i-1) - TS'(i-1)$$

In each reception report: J at last received packet

$$J(i+1) = J(i) + (|D(i+1)| - J) / 16$$

Round-trip Estimation

- **Sources and Receivers exchange "Reports" messages**

```
Sender-only                                                              Reciever-only

                                  RR1 = pck(nb_lost, …,
                                  Last_SR, DLSR)
last_SR = LAST_SR_TIMESTAMP=0     <------------------------     DLSR = DELAY_SINCE_LAST_SR =
        S_recvtime(RR1)=NTP                                             NTP - R_recvtime

                                  ------------------------>
DLSR = DELAY_SINCE_LAST_SR=0      SR1 = pck(nb_sent, rtp_time,   last_SR = S_sendtime(SR1)
 + S_sendtime(SR1)=NTP             last_SR=0, DLSR=0,           + R_recvtime=NTP
                                  S_sendtime(SR1) )
```

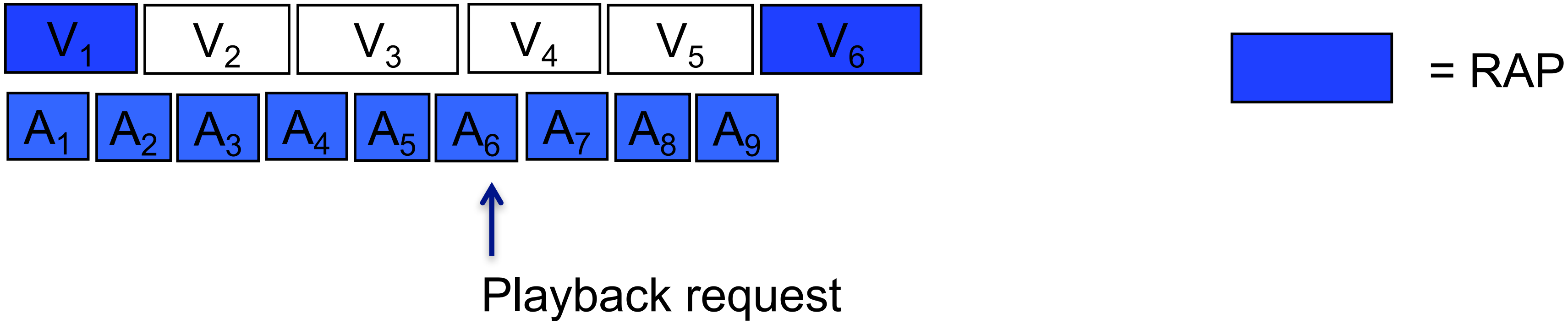
```
RTT = S_recvtime(RR1) – last_SR(SR1) – DLSR(SR1)
= S_recvtime(RR1) – S_sendtime(SR1) – processing_time(client)
```

- Symmetrical delays are assumed
- Computation only on Sender Reports

RTSP Real Time Streaming Protocol RFC 2326

- **How to control RTP stream delivery**
  - Distribution and announces of media sessions
    - Session description, Session updates
  - Stream playback: Play, Pause, Record, Teardown
    - Range playback: PLAY [0-10] [20-30]
  - Parameters SET/GET
    - Bandwidth selection MTU, etc …
    - !! Parameters are not normative !!
- **Handling synchronized streams**
  - Single command for N streams
    - Video, audio lang#1, subtitle #2
  - RTSP Session
    - all streams requested for playback
    - Enables synchronized sending of streams
    - Enables playback range adjustment to first RAP in all streams

RTSP Range adjustment



- **Streams in different sessions**
  - Video starts at V_1
  - Audio starts at A_6
  - Higher buffering of audio (waiting for V_4)
- **Streams in single session**
  - Video starts at V_1
  - RTSP Server adjusts audio starts to A_1
  - No extra buffering required
- **Server reply to PLAY [10-]**
  - Adjusted range : [8.9-]
  - RTP timestamp matching start range

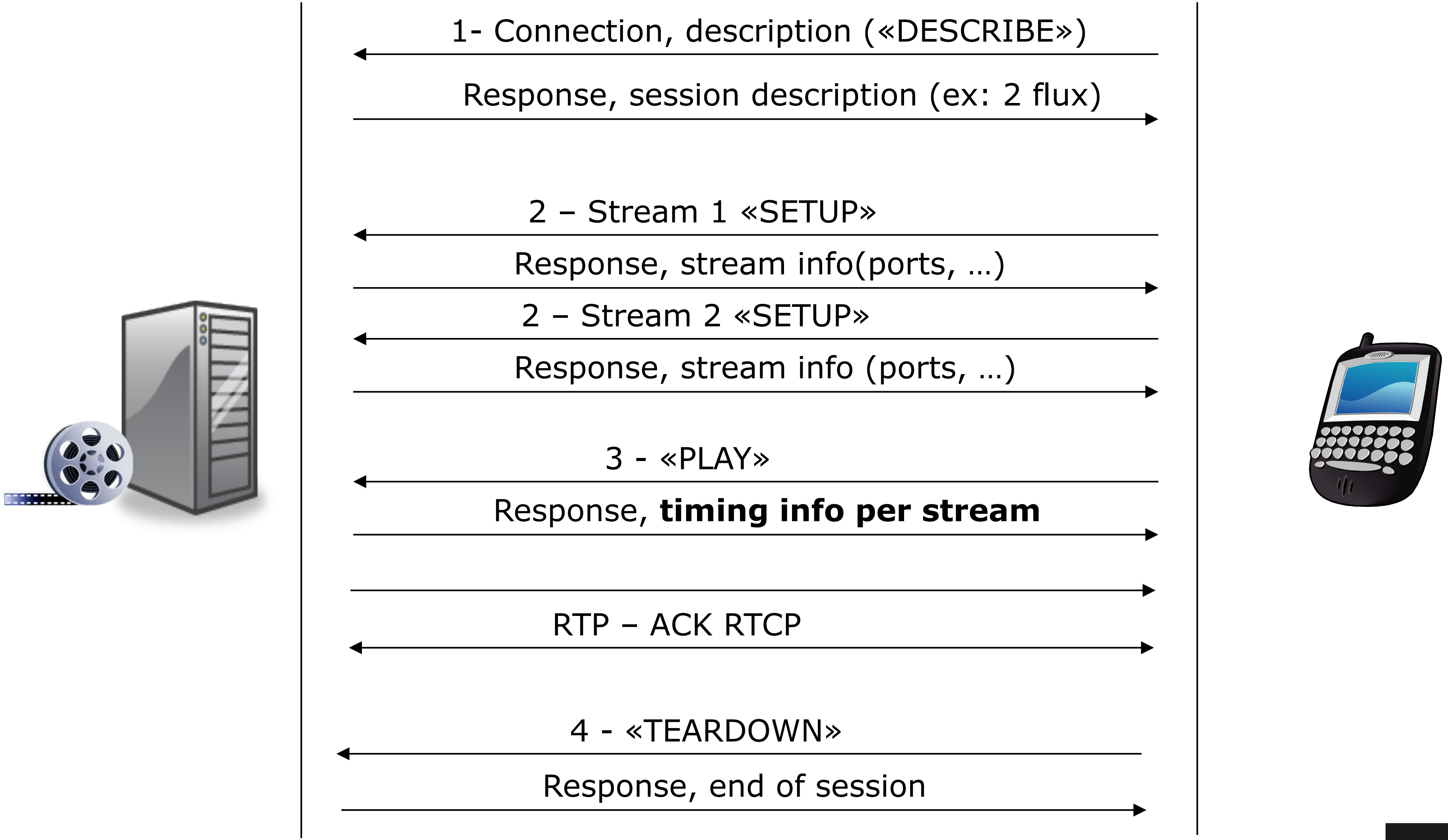
Some specificities

- **Connected (TCP) or not (UDP)**
  - Disconnection between requests possible
    - Usage of command number ID
  - Permanently connected
    - Possibility to change session info (ANNOUNCE)
- **Interleaved Mode**
  - RTP data over TCP connection
    - Packets identified with "$NXX"+raw data
      - N (1 byte) gives RTP stream ID
      - XX (2bytes) gives packet length
  - Allows RTP traffic when UDP is blocked

- **RTSP State Machine**

Method	State Req.	Description
DESCRIBE	NO	C->S: request session description
ANNOUNCE	NO	C->S: send or update session description
SETUP	YES	C->S: setup streams (transport parameters) and create session if needed
PLAY	YES	C->S: request playback of all streams in session
PAUSE	YES	C->S: pauses playback (now or differed)
RECORD	YES	C->S: requests recording of stream from client
SET/GET PARAMETER	NO	C->S, S->C: parameters handling
REDIRECT	NO	S->C: indicate new location of session
TEARDOWN	YES	C->S: stops playback, session is destroyed

Controlling the streams: RTSP (RFC 2326)



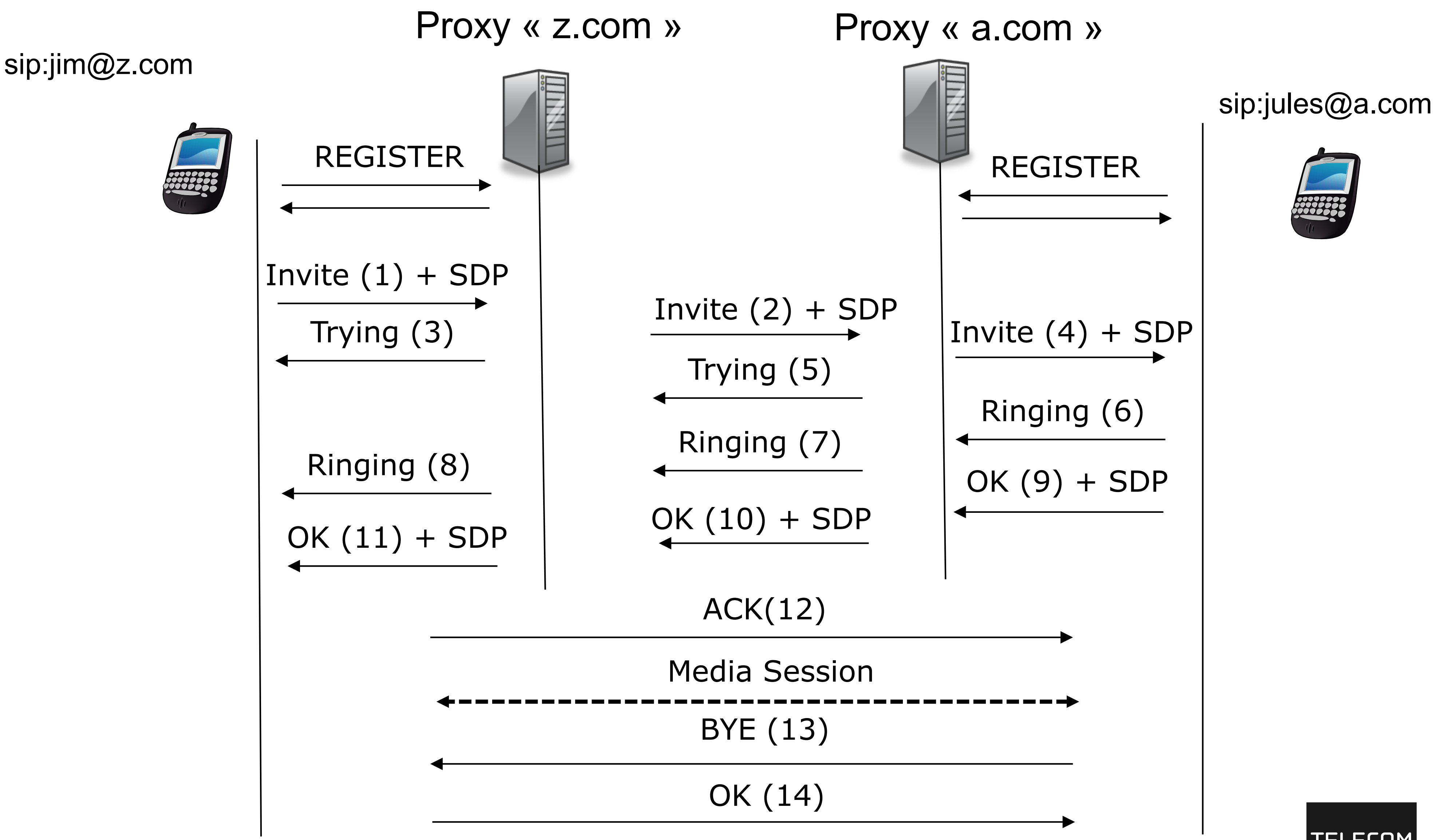
SAP

- **Session Announcement Protocol**
  - IETF, RFC 2974
- **Periodical announce of session over a network**
  - Multicast UDP port 9875
  - Bandwidth shall be <= 4 kbps
  - Packet payload: SDP
- **Compatibility**
  - IPv4 and IPv6
  - Recommended Addresses
    - 224.2.127.254 : IPv4 global scope
    - IPv6: FF0X:0:0:0:0:0:2:7FFE , X < 2^4

SIP

- **Session Initiation Protocol**
  - IETF, RFC 2543 -> RFC 3261
- **Usages**
  - Videoconferencing
  - Voice over IP
  - Multimedia streaming
- **Session management**
  - REGISTER
  - INVITE, ACK, CANCEL
  - BYE
  - OPTIONS
- **Media Stream**
  - SDP for description
  - RTP/RTCP for media
- **Presence is not handled**
  - Other protocols required

Voice over IP: SIP (RFC 3261)



AV Transport

Exercice

Exercice 1

- **MP3 over RTP**
  - sample rate 44100
  - Stereo
  - Constant bitrate 128kbps
  - No media-specific header in payload

  - Compute the RTP overhead
  - Propose a method reducing the overhead

Exercice 2

- **H263 over RTP**
  - Frame rate 25 fps
  - Bitrate 1 mbps
  - No media-specific header in payload

  - Compute the RTP overhead
    - Is it accurate ?

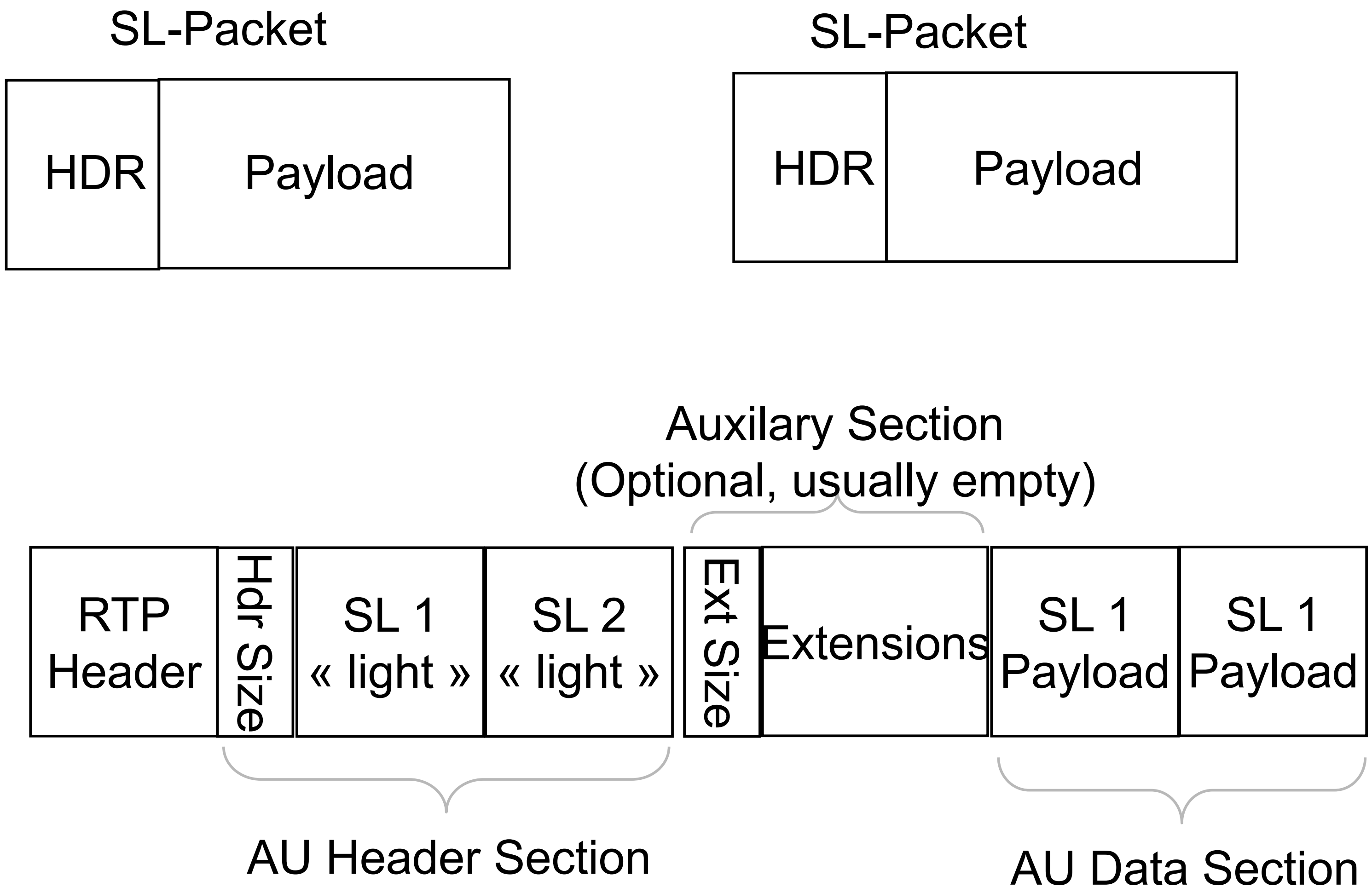
AV Transport

Payload Formats

RFC 3640

- **MPEG-4 carriage over IP**
- **Supports MPEG-4 SyncLayer (SL)**
  - Base tools supported:
    - Timestamps, RAP indication, AU size, AU duration
  - Possible extensions for other SL info
- **Supports MPEG-4 Stream**
  - In practice, only used for AAC

RFC 3640: Syntax (1/2)

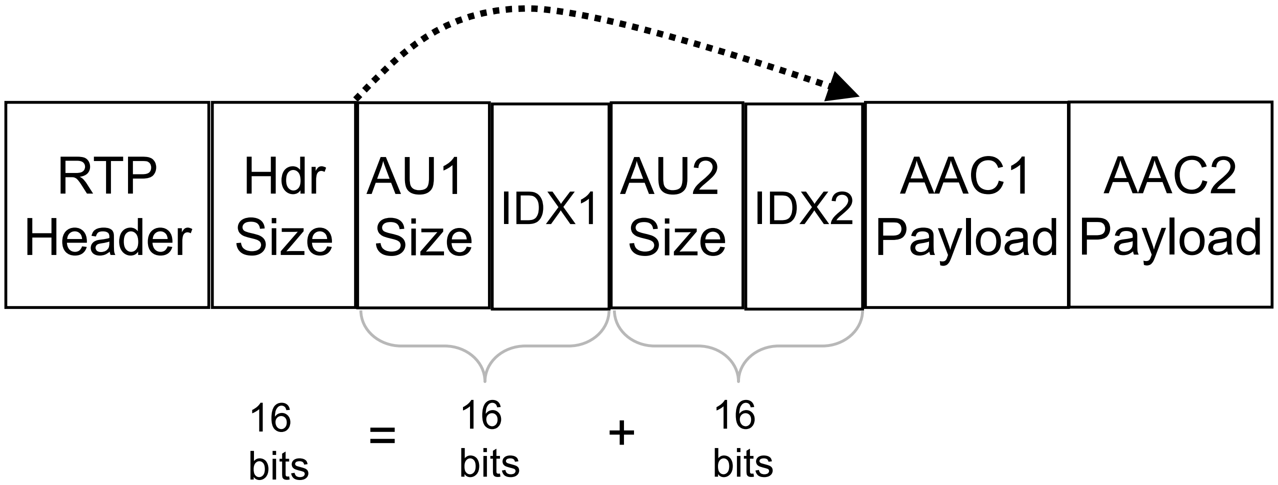


RFC 3640: Syntax (2/2)

- **SyncLayer Information:**
  - CTS, DTS: offset to RTP TS
  - AU sequence number
  - AU Size
  - Random Access Point
  - StreamState
    - ~ carousel signaling: repeated or new AU
- **M bit: end of AU**

RFC 3640 –AAC High BitRate Example

- **Fixed Configuration**
  - Max AU size coded on 13 bits
  - 3 bits to indicate decoding order (IDX)
    - Supports audio AU interleaving
    - Meilleure résistance aux pertes



RFC 3984

- **MPEG-4 AVC / H264 carriage over RTP**
- **AVC data model:**
  - Data encapsulated in Network Abstraction Layer Units
  - Codec parameters in
    - Sequence Parameter Set NAL: valid for entire video sequence
    - Picture Parameter Set NAL: valid for a coded picture

F	NRI	Type

- F: forbidden 0 bit
- NRI: NAL reference index (2 bits)
- Type: NAL unit type (5 bits)

RFC 3984 - Principes

- **New NAL units defined**
  - NAL Aggregation
    - STAP: NALU with same composition time
    - MTAP: NALU with different composition time
  - NAL fragmentation
- **Supports interleaving**
  - DON: Decode Order Number
- **Parameter Set (SPS, PPS)**
  - Transport in band
  - Transport out-of-band (SDP)

RFC 3984 – STAP NALU

- **STAP-A: Envoi en ordre de décodage**

RTP Header	STAP-A NAL Header	NALU 1 Size	NALU 1 Header	NALU 1 Payload	NALU 2 Size	NALU 2 Header	NALU 2 Payload

- STAP-B: Envoi hors ordre de décodage
  - ⇨ Pas d'entrelacement au sein du paquet RTP

RTP Header	STAP-B NAL Header	DON	NALU 1 Size	NALU 1 Header	NALU 1 Payload	NALU 2 Size	NALU 2 Header	NALU 2 Payload

RFC 3984 – MTAP NALU

- **Composition time of each NAL**
- **Decoding order of each NAL**

RTP Header	MTAPX NAL Header	Decoding Order Number Base		
NALU 1 Size	NALU 1 DON Diff	NALU 1 TS Diff	NALU 1 Header	NALU 1 Payload
NALU 2 Size	NALU 2 DON Diff	NALU 2 TS Diff	NALU 2 Header	NALU 2 Payload

- MTAP16: 16 bits for NAL TS – RTP TS
- MTAP24: 24 bits for NAL TS – RTP TS

RFC 3984 – FU NALU

- **Handling fragmentation**
  - FU-A: sending in decode order
  - FU-B: sending in any order

RTP Header	FU-A NAL Header	FU header	NALU 1 Payload Fragment

- FU NAL Header:
  - ⇨ S Bit: start of NALU
  - ⇨ E Bit: end of NALU
  - ⇨ Type: type of the fragmented NAL

S	E	R	Type

RFC 3984

- **Payload MIME type: « H264 »**
- **Profiles/Levels and decoder config in SDP**
- **Recommendations:**
  - Single NALU: mandatory
  - FU-A et STAP-A: recommended
  - Autres types: optional

Exercice 3

- **H264 RTP**
  - frame syntax:
    - NAL1: SPS, 50 bytes
    - NAL2: PPS, 6 bytes
    - NAL3: SEI, 180 bytes
    - NAL4: VCL 56000 bytes
    - NAL5: VCL 58000 bytes
  - RTP max payload size
    - 1460 =1500 (Ethernet MTU) – 20 (IP) – 8 (UDP) – 12 (RTP)

  - Describe the packets being produced
  - Propose a high level view of a ISOBMFF hint track sample for that frame