AV Transport

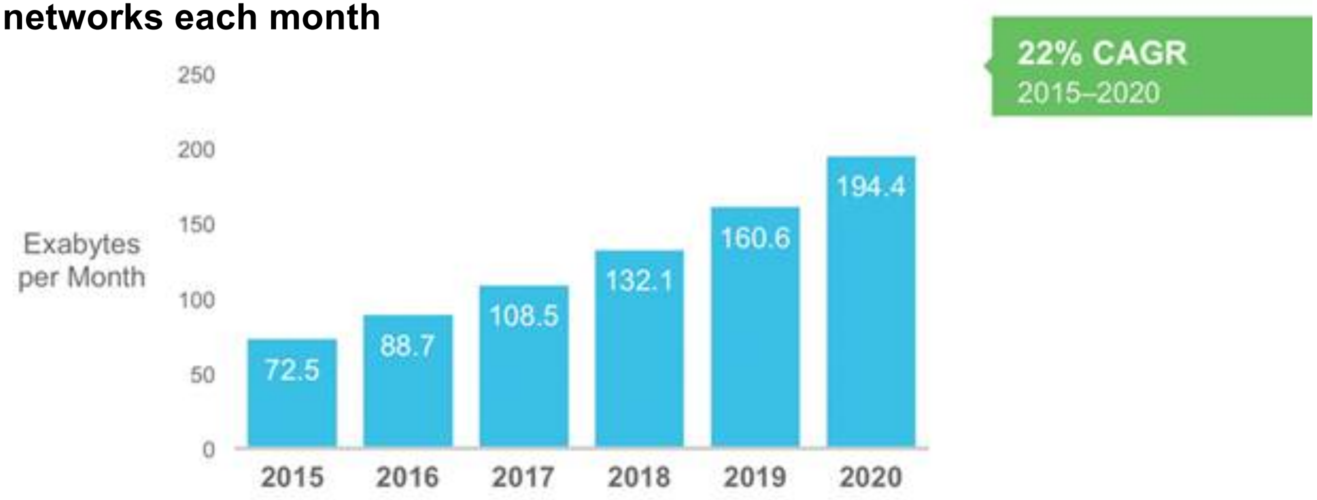
Background

jean.lefeuvre@telecom-paristech.fr
E506

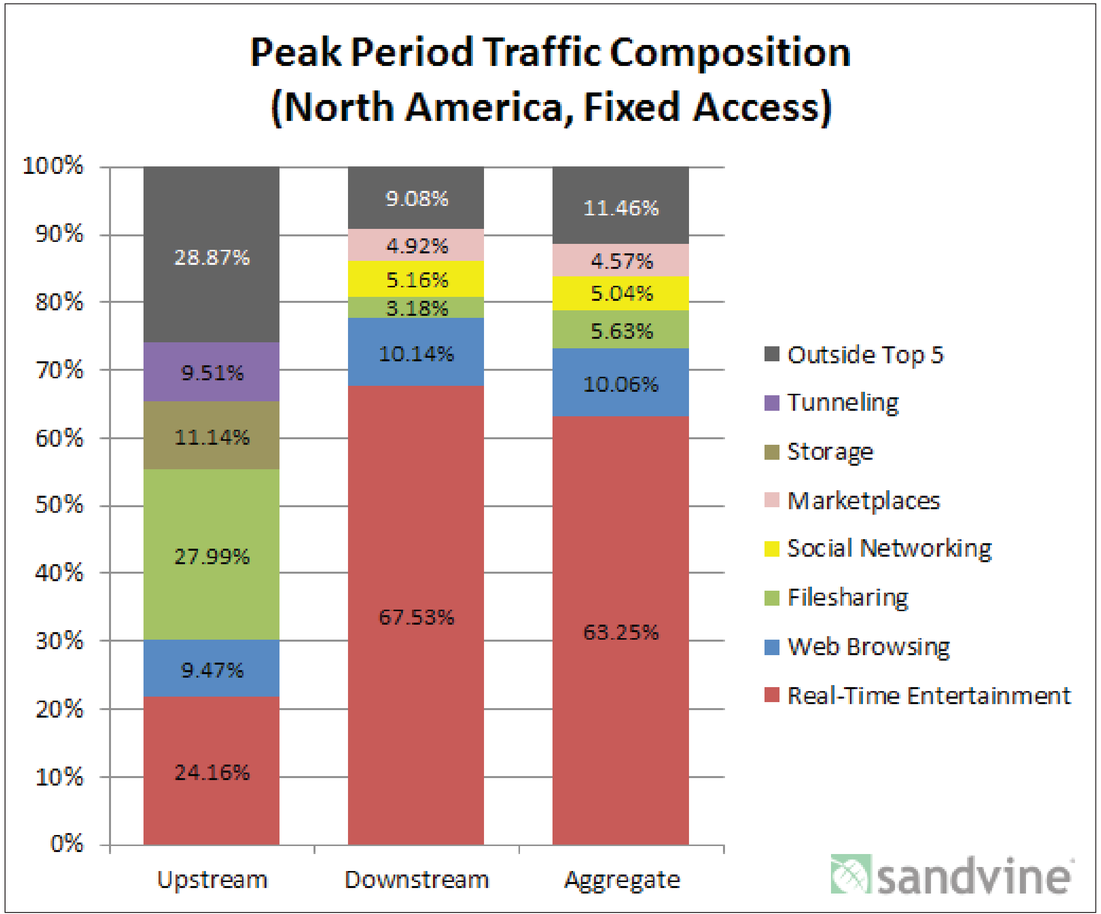
TELECOM ParisTech

Traffic Forecasts by Cisco

- **Annual IP traffic**
  - 2016: > 1 ZB (zettabyte) = 10^{21} bytes = 1 billion TB
  - 2020: 2.3 ZB
- **Mobile vs PC Traffic**
  - 2015: 53% PC, 8% Mobile
  - 2020: 29% PC, 30% Mobile
- **Wireless+Mobile:**
  - 2015: 52%
  - 2016: 66%
- **Connected devices**
  - 2015:16.3 billion (> 2x world population)
  - 2020: 26.3 billion (3x world population)
- **Hours of video in 2020**
  - **5 million years to watch the amount of video that will cross global IP networks each month**



Internet Traffic Shares: North America



Downstream	
Application	**Share**
Netflix	34.89%
YouTube	14.04%
HTTP	8.62%
Facebook	2.98%
BitTorrent	2.80%
iTunes	2.77%
MPEG - OTHER	2.66%
Amazon Video	2.58%
SSL	2.14%
Hulu	1.41%
	74.89%

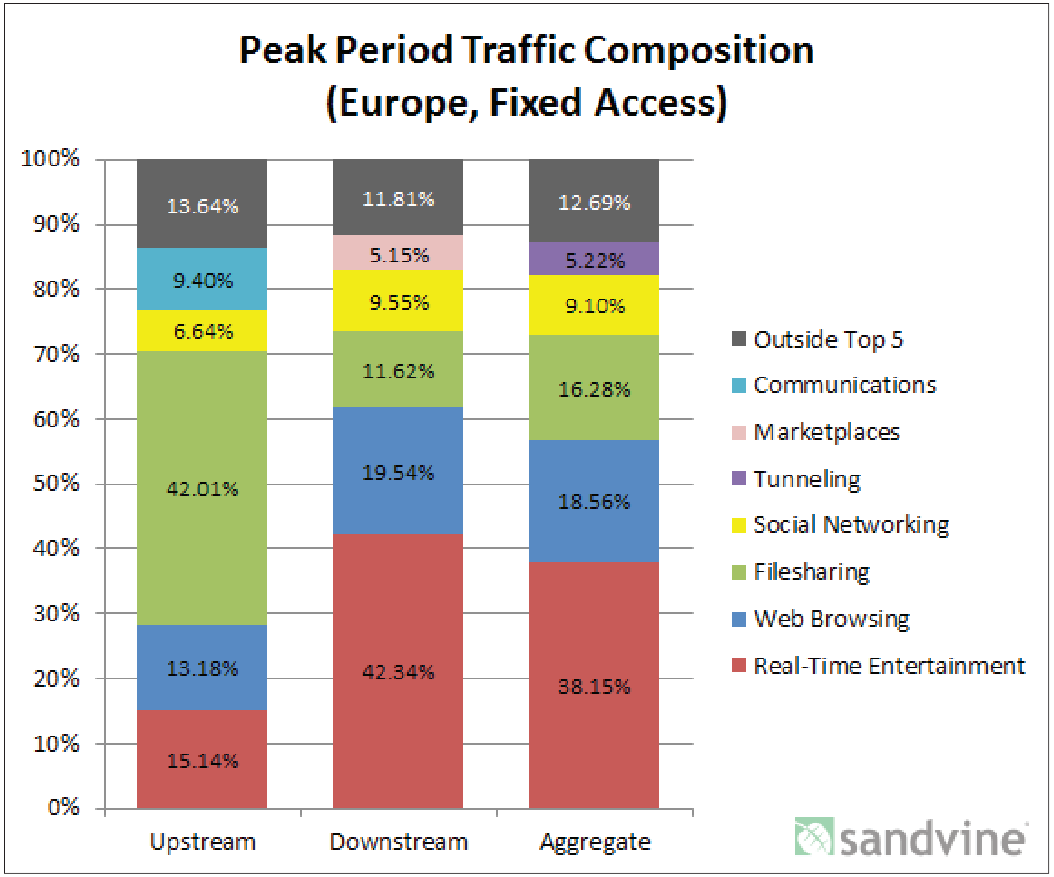
Sandvine 2015

Monthly Consumption - North America, Fixed Access		
	Median	Mean
Upstream	1.8 GB	8.5 GB
Downstream	20.4 GB	48.9 GB
Aggregate	22.5 GB	57.4 GB

AV traffic 2012: 35%
2015: 70%

TELECOM ParisTech

Internet Traffic Shares: Europe

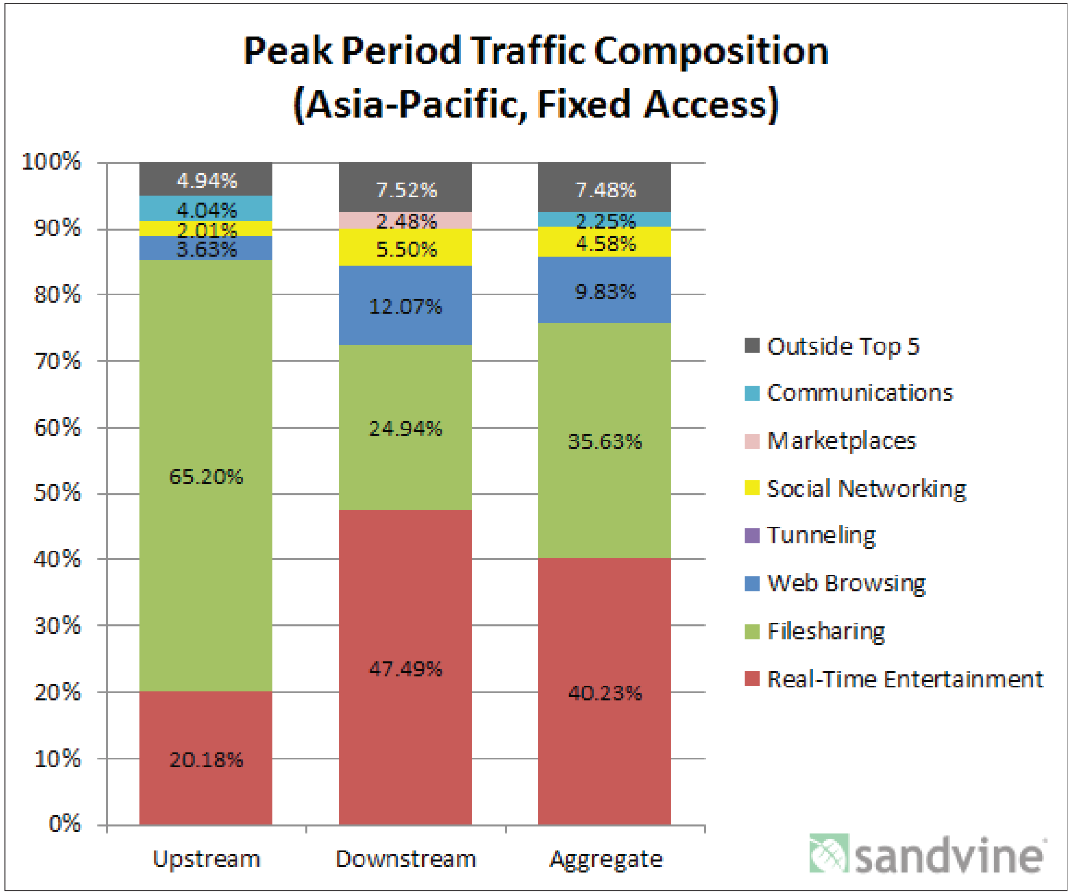


Downstream	
Application	**Share**
YouTube	22.38%
HTTP	17.27%
BitTorrent	10.39%
Facebook	7.84%
SSL	4.56%
MPEG - OTHER	3.57%
Netflix	3.44%
RTMP	2.31%
Flash Video	1.90%
PC: Valve’s Steam Service	1.73%
	75.38%

Monthly Consumption - Europe, Fixed Access		
	Median	**Mean**
Upstream	1.5 GB	5.1 GB
Downstream	8.7 GB	23.1 GB
Aggregate	10.1 GB	28.2 GB

Sandvine
2015

Internet Traffic Shares: Asia

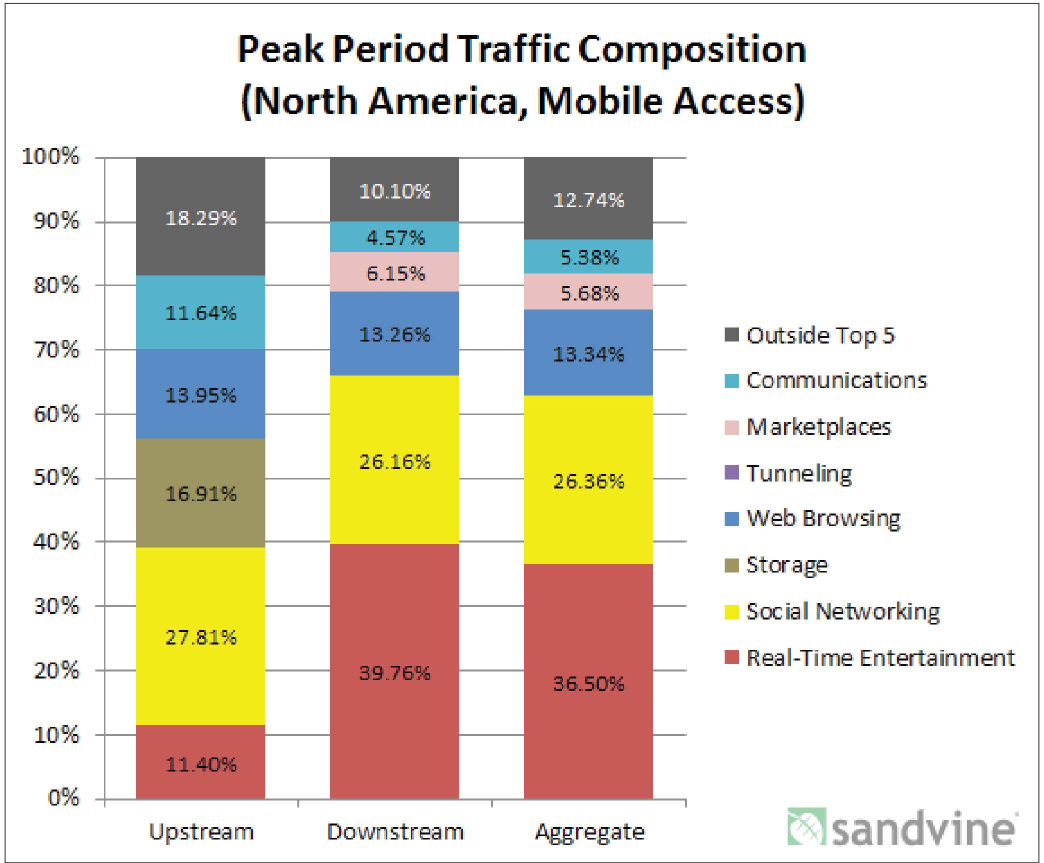


Downstream	
Application	**Share**
YouTube	23.70%
BitTorrent	22.78%
HTTP	10.94%
RTSP	7.43%
Facebook	3.22%
MPEG - OTHER	2.93%
QVoD	1.83%
Flash Video	1.82%
SSL	1.75%
RTMP	1.74%
	78.14%

Monthly Consumption - Asia-Pacific, Fixed Access		
	Median	**Mean**
Upstream	2.9 GB	13.4 GB
Downstream	17.9 GB	31.7 GB
Aggregate	20.8 GB	45.1 GB

Sandvine 2015

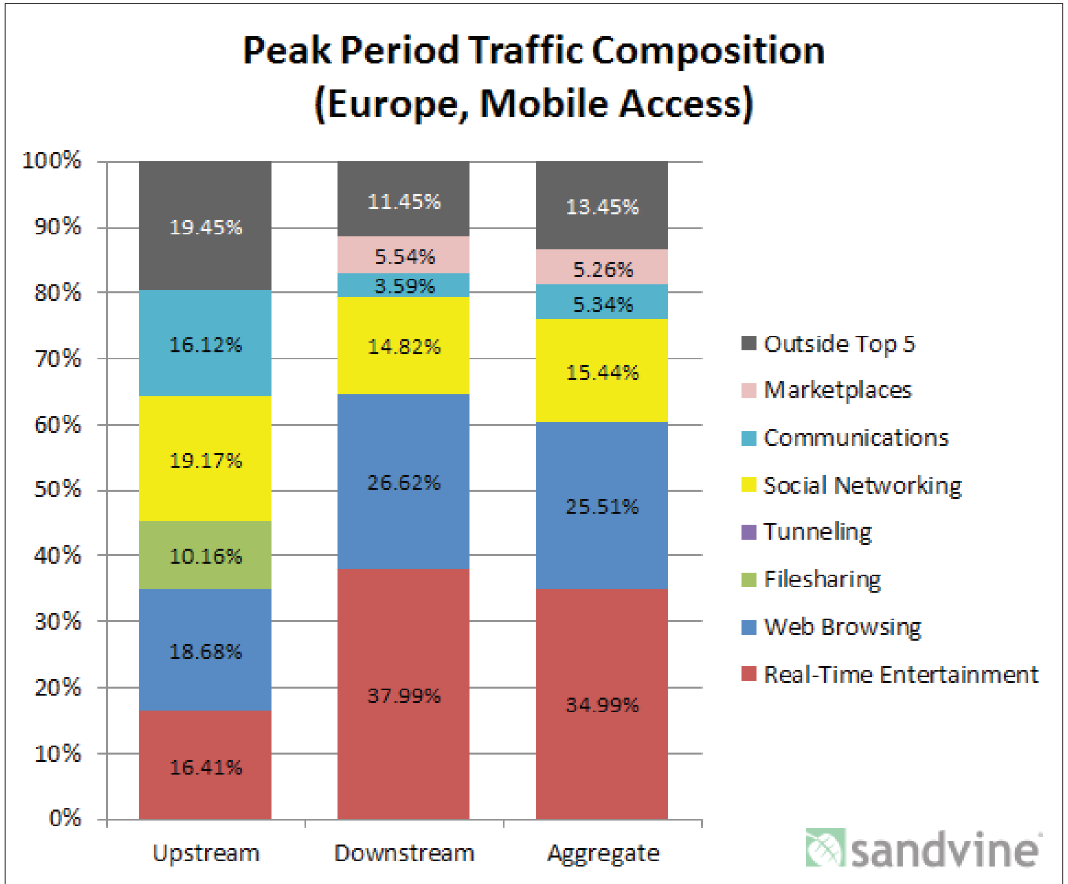
Mobile Internet Traffic Shares: North America



Downstream	
Application	**Share**
YouTube	19.75%
Facebook	19.05%
HTTP	11.44%
MPEG - OTHER	6.32%
Netflix	4.51%
Instagram	4.49%
SSL	4.03%
iTunes	3.20%
Google Cloud	3.07%
Pandora Radio	2.72%
	78.57%

Monthly Consumption - North America, Mobile Access		
	Median	**Mean**
Upstream	19.7 MB	75.4 MB
Downstream	99.1 MB	506.5 MB
Aggregate	118.4 MB	521.9 MB

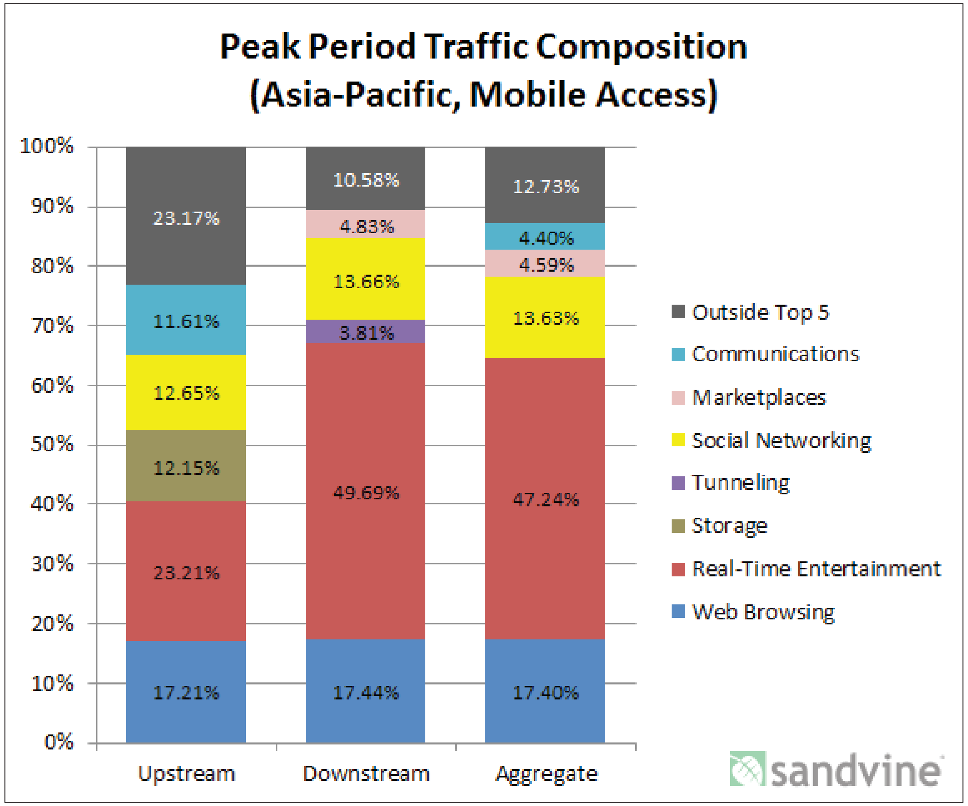
Mobile Internet Traffic Shares: Europe



Downstream	
Application	**Share**
HTTP	24.60%
YouTube	20.89%
Facebook	12.16%
MPEG - OTHER	3.77%
SSL	3.06%
Flash Video	3.03%
BitTorrent	3.01%
Google Cloud	1.90%
Google Market	1.67%
iTunes	1.61%
	75.70%

Monthly Consumption - Europe, Mobile Access		
	Median	**Mean**
Upstream	15.1 MB	69.5 MB
Downstream	108.8 MB	380.3 MB
Aggregate	122.1 MB	449.5 MB

Mobile Internet Traffic Shares: Asia



Downstream	
Application	**Share**
YouTube	17.46%
HTTP	15.59%
Facebook	9.48%
MPEG - OTHER	7.12%
SSL	5.37%
Google Market	3.56%
Dailymotion	2.59%
Instagram	1.82%
iTunes	1.54%
Google Cloud	1.47%
	66.00%

Monthly Consumption - Asia-Pacific, Mobile Access		
	Median	**Mean**
Upstream	261.7 MB	143.1 MB
Downstream	298.1 MB	1.0 GB
Aggregate	339.2 MB	1.1 GB

AV Traffic Continuous Growth

- **New Formats with more data to send**
  - 4K, 8K Video
  - High Frame Rate
  - High bit depth, HDR
  - 3D audio (22+ channels)

- **New Services**
  - Video Conferencing
  - Game streaming platforms
  - Interactive Videos
  - Virtual Reality Streaming
- **More devices with heterogeneous capabilities**
  - Different compression formats supported
  - Different screen sizes
  - Different audio loudspeakers layouts

=> Multiple encoding/delivery of the media

Media Delivery Chain

Capture → Compression → Packetization → Network → Reception Decompression Display

Service Types

- **Direct or « live »**
  - Radio, TV satellite or terrestrial
  - Internet live
- **Audio and Visio-conference**
  - Fixed and Mobile IP Phone
  - Skype / FaceTime / WebRTC
- **Video on demand**
  - User Content (ex: YouTube)
  - VoD channels (Netflix, …)
  - Catch-up & Replay
- **Storage / Archiving**
  - DVD, Blu-Ray

- **Specificities**
  - All users get the same data
  - Real-time but constant latency
  - Lossy

  - Data unique per call (and user)
  - Real-time and very low latency required
  - Lossy

  - Large amount of data
  - Not real-time, large latency possible
  - Lossless or lossy

  - No requirements on delivery speed
  - Lossless

AV Transport

Systems Definitions

Media Objects

- **Represents a media**
  - Audio
  - Video
  - Subtitles
  - Vector Graphics 2D or 3D (Flash, SVG, VRML, BIFS)
  - Various Meta-data
    - Sensors: GPS, temperature, pressure
    - Annotations: regions of interest, text, sometimes with 2D or 3D positioning in the video stream
    - Programmatic instructions or data: Java, JSON, …
- **Media and Time**
  - Static: HTML pages, image, user interface sound
  - Dynamic: media that changes over time
- **Media Stream**
  - Succession (in time) of media data
  - Most of the time, media object ⇔ media stream
    - Exception with layered coding (video, 3D audio): 1 media object = N media streams

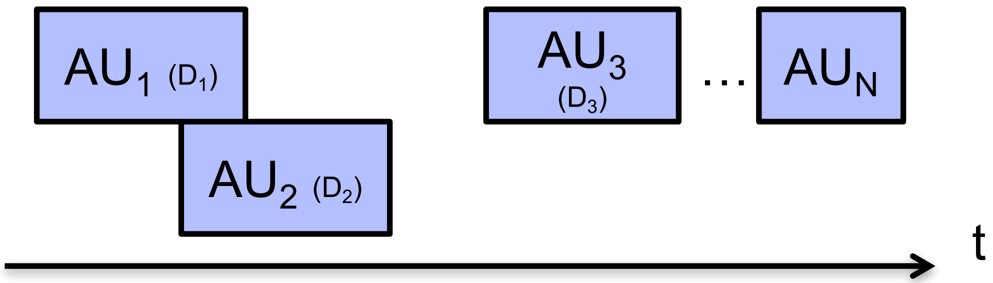
Media Streams

- **Succession of Access Unit**
- **Access Unit**
  - ⇔ Each « state » or update of the media
  - « smallest data chunk to which a unique time can be assigned »
- **Examples of AU**
  - Progressive Video: 1 frame
    - Interlaced Video: 1 field or 1 frame
  - Audio: N (>=1) samples
  - Subtitle:
    - Bitmap: One image (bitmap subtitle)
    - Text: N (>=1) lines of text
- **Rate of access units (or Frame Rate)**
  - Constant Frame Rate (CFR): same time interval between AUs
  - Variable Frame Rate (VFR): different time intervals
  - Terminology mostly used for video and animations

Timing of Access Units

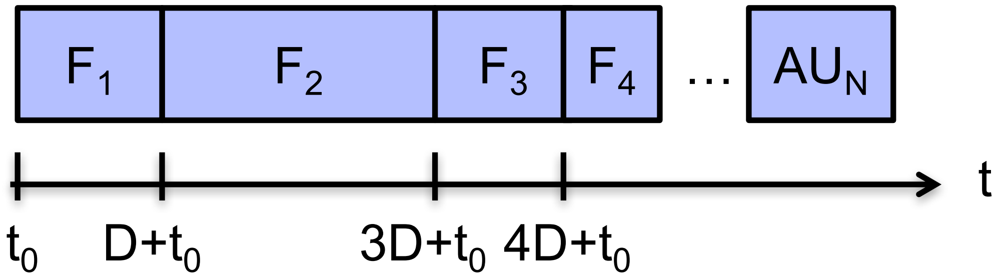
- **Implicit Duration**
  - Video, Audio, text, meta-data
  - No “holes” in the timeline, an AU lasts until the next AU begins
  - Typically used for CFR/VFR media

- **Explicit Duration**
  - Text, meta-data
  - Can express empty moments in the media timeline
  - Can be hard to express as implicit duration (editing of media data often required)
  - Typically used for VFR media

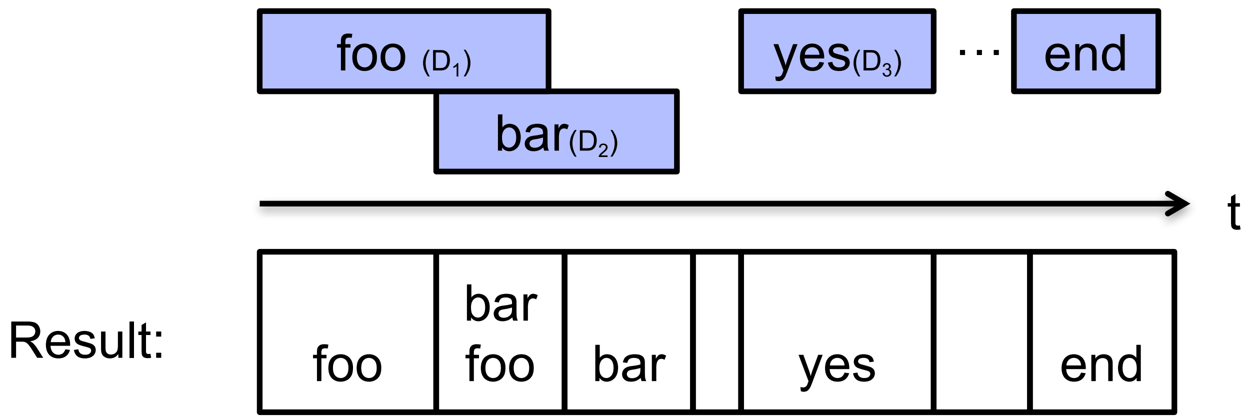


Examples

- **Variable video frame rate : Implicit Duration**



- **Subtitles: explicit duration**



Uncompressed streams (refresher)

Capture

- **Video Characteristics**
  - Resolution (width x height)
    - Ex: 640x480,720x576,1280x720,1920x1080
  - Sampling Format
    - Ex: RGB, YUV420, YUV422, YUV444
  - Color Bit Depth
    - Ex: 8, 10, 12 bit per color channel
  - Color Gamut
    - Ex: SDR (BT.720), HDR (BT.2020), …
  - Frame rate
    - Ex: 23.976, 24, 25, 29.97, 30, 50, 60, 100, 120
    - Interlaced (i) or progressive (p)
      - Ex: 720p, 1080i
- **Quick numbers**
  - Video frame 1080p YUV420: 3.1 Mbytes
  - Video 720p50 YUV420: 553 Mb/s

Capture: Interlaced Video

Capture

Top field	Bottom field	Interlaced Frame		Progressive Frame
line 1		line 1		line 1
	line 2	line 2		line 2
line 3		line 3		line 3
	line 4	line 4		line 4
line 5		line 5		line 5
	line 6	line 6	VS	line 6
line 7		line 7		line 7
	line 8	line 8		line 8
line 9		line 9		line 9
	line 10	line 10		line 10
line 11		line 11		line 11
	line 12	line 12		line 12
D = 1/2f	D = 1/2f	D = 1/f		D = 1/f

Uncompressed streams (refresher)

Capture

- **Audio Characteristics**
  - Sample Rate in Hz (i.e. capture frequency)
    - Ex: 44100, 48k, 24k, 22050, 96k
  - Sampling Format
    - Ex: 8 bit int, 16 bit int, 24 bit int, 32 bit float
  - Channels and their positioning
    - Ex: stereo, 5.1, 7.2, 22.2
  - Ambisonic representations
- **Quick numbers**
  - audio stereo 16bpc 44.1 KHz: 1.4 Mbps

Uncompressed streams (refresher)

Capture

- **Text / Subtitle Characteristics**
  - Format: bitmap or text
  - For bitmap formats
    - Compression method, color palette
  - For text formats
    - Character encoding (UTF8, ASCII)
    - Representation method: SRT, WebVTT, TTXT (iTunes)
- **Quick numbers**
  - DVB subtitle: 50 to 100 kbps
  - SRT: << 1 kbps

Media Delivery Chain

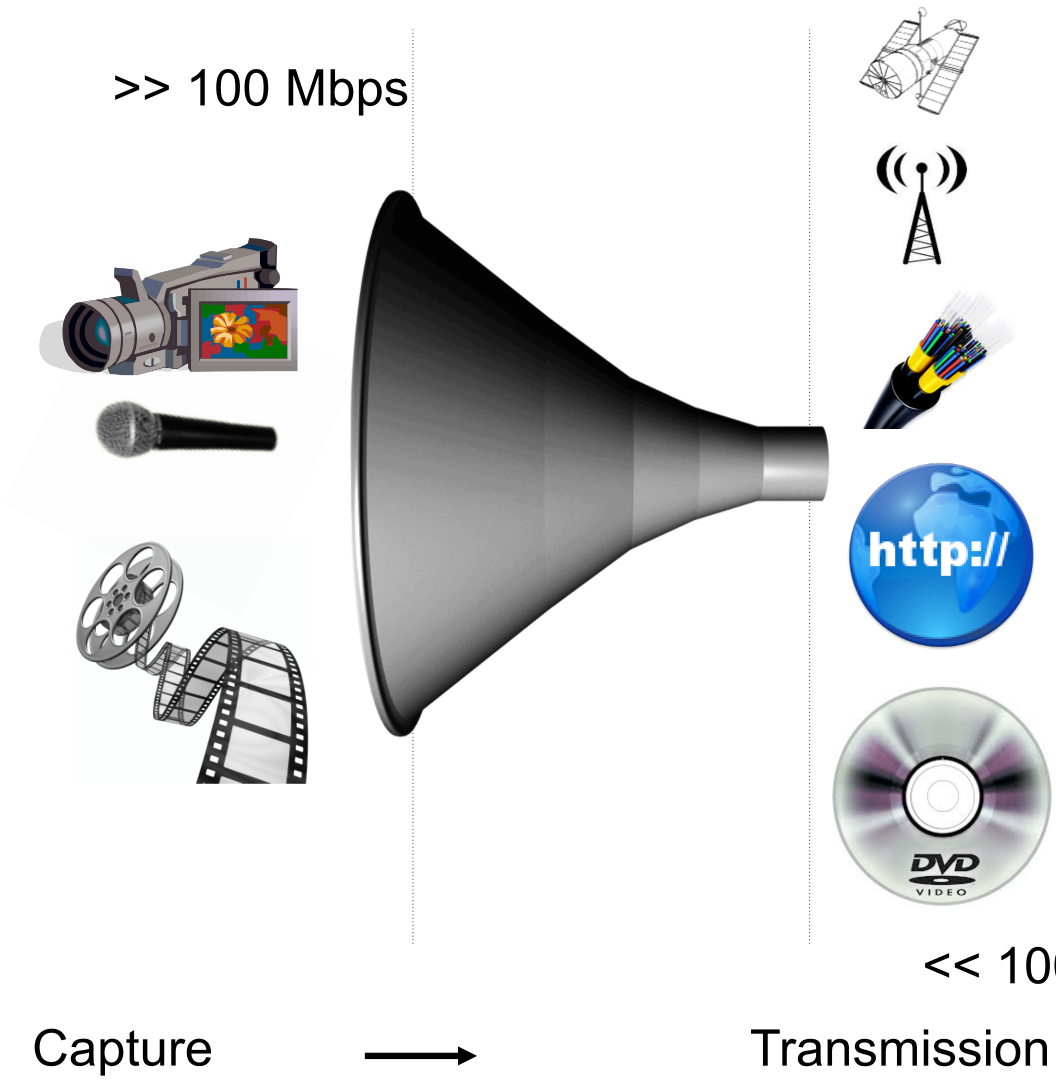
Capture → Compression → Packetization → Network → Reception Decompression Display

Network Characteristics

Network

- **Satellite and Terrestrial Broadcast**
  - Losses without retransmissions
    - FEC « Forward Error Correction »
  - Fixed bandwidth available
  - Fixed latency
- **IP Network**
  - Managed:
    - Few losses, low latency and network jitter
    - Bandwidth guaranteed
  - General Internet:
    - losses, important jitter and latencies
    - Variable bandwidth
- **Physical support**
  - No losses
  - Fixed bandwidth

How to choose the right compression?



- **Application type**
  - Movie Theatre (~= lossless compression)
  - TV Broadcast
  - VoD (variable bitrate)
- **Network Type**
  - Digital TV : MPEG-2 or MPEG-4 compression
  - IP/HTTP
    - More codecs and transport possible
    - Depends on target devices

Compressed Streams

Compression

- **Different needs:**
  - Reduce the bitrate: audio, video, animations
  - Change modality: text->images
- **Codec**
  - Entity in charge of **co**ding and or **dec**oding the signal between a compressed and an uncompressed representation
- **Coding modes**
  - Constant Bit Rate (CBR): output rate of the encoder is constant
  - Variable Bit Rate (VBR): output rate of the encoder can vary
  - Capped VBR: VBR used but maximum bitrate is set

Random Access in decoding process

Compression

- **Decoding a stream from any moment in time**
  - Some AUs have coding dependencies to past AUs
  - Need to identify «random access point » AUs
    - with no coding dependencies
    - After which subsequent AUs in display order can properly be decoded
  - This may imply
    - rewinding in a stream to get previous RAP
    - Waiting for the next RAP
- **Audio, Text, Images, Meta-data**
  - Usually every AU is a RAP
- **Video, compressed graphics**
  - Not every AU is a RAP

Codec Configuration

Compression

- **Most modern codecs**
  - Video: since MPEG-4 part 2 (1998)
  - Audio: Since 11C (1997)
  - Other: depend on format
- **Fixed data chunk**
  - Stream properties
    - Resolution, bit depth
    - Sample rate, channel configuration
  - Codec configuration
    - Profile: subset of all tools in standard
    - Level: various limits on the subset (max size, rate …)
    - Other
  - Decoder cannot be configured without this !
- **Located**
  - In the stream: usually with each RAP AU
  - Outside the stream: depends on the transport protocol

TELECOM ParisTech

Audio Coding Specificities (refresher)

Compression

- **Basic principles**
  - Remove perceptually irrelevant signals
  - Remove redundancies
  - This implies working on much more than one sample!
- **Audio Coding Window**
  - Audio codec encodes N audio samples at once
  - This defines the access unit for the compressed stream i.e., 1 Audio AU ⇔ N Audio samples
  - Examples
    - MP3: 576 or 1152 samples per AU
    - AAC: 1024 samples per AU
- **Random Access**
  - (most codecs) All AUs are RAP
  - There can be dependencies between two consecutive AUs, but usually (e.g. most codecs) the second AU can still be decoded, although not perfectly

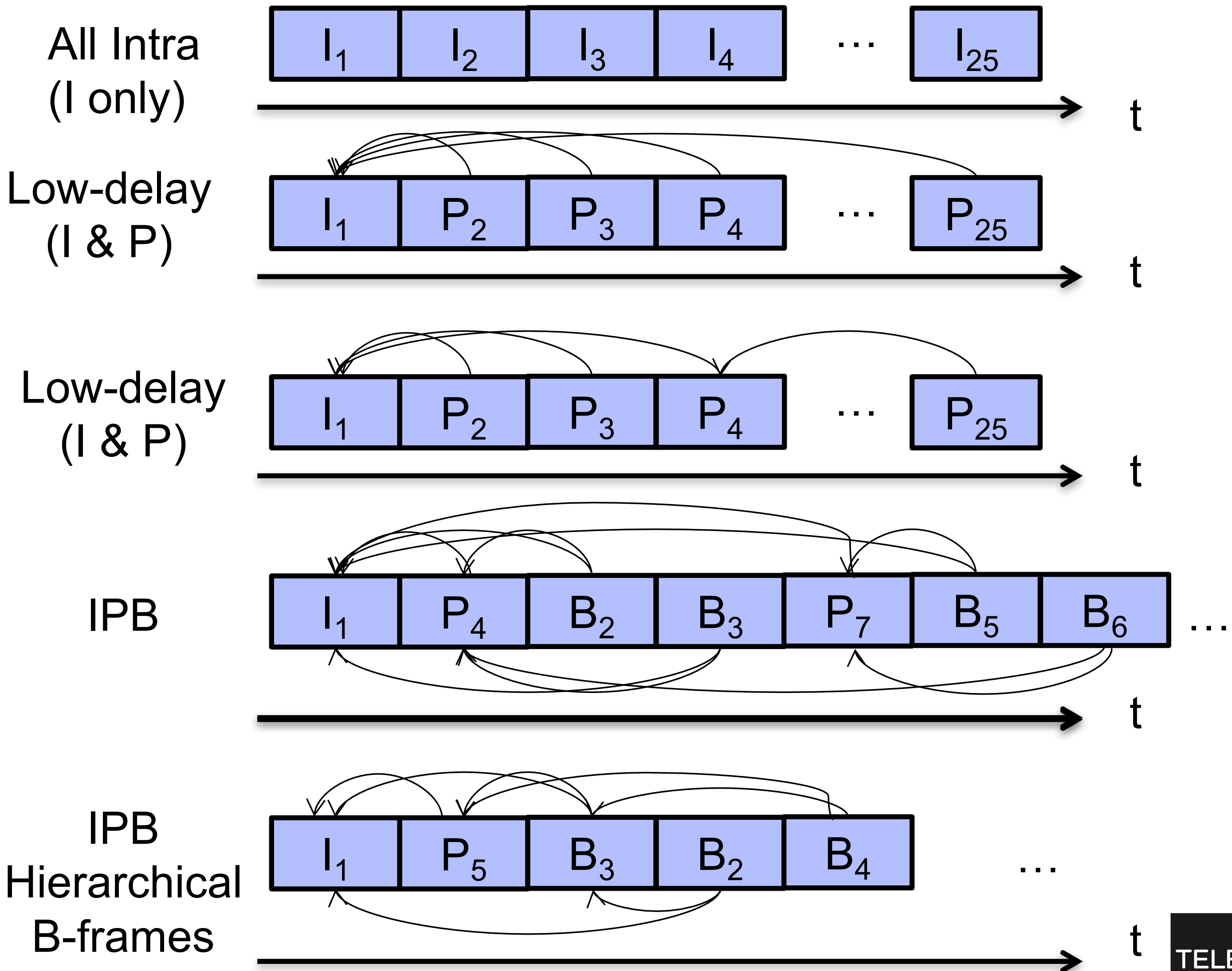
Video Coding Specificities (refresher)

Compression

- **Basic principles**
  - Remove redundancies by locating similar pixels inside a frame or between frames
  - Some frames do not depend on any other frame for the decoding (intra frames)
  - Better efficiency is achieved when using reference frames from the past and the future
    - Knowing “future frames” at the encoder implies buffering of many frames, hence higher latency
    - Frames are not always delivered in presentation order
- **Group of picture (or GOP)**
  - Sequence of picture, in decode order, starting with an I-frame, up to but excluding the next I-frame
- **B-pictures or Bidirectional Pictures**
  - Pictures using frames from the future and from the past for reconstruction
  - In both AVC and HEVC, B-pictures can be used as reference pictures for other B
  - **If B-pictures, decoding order != presentation order**

Example of GOPs

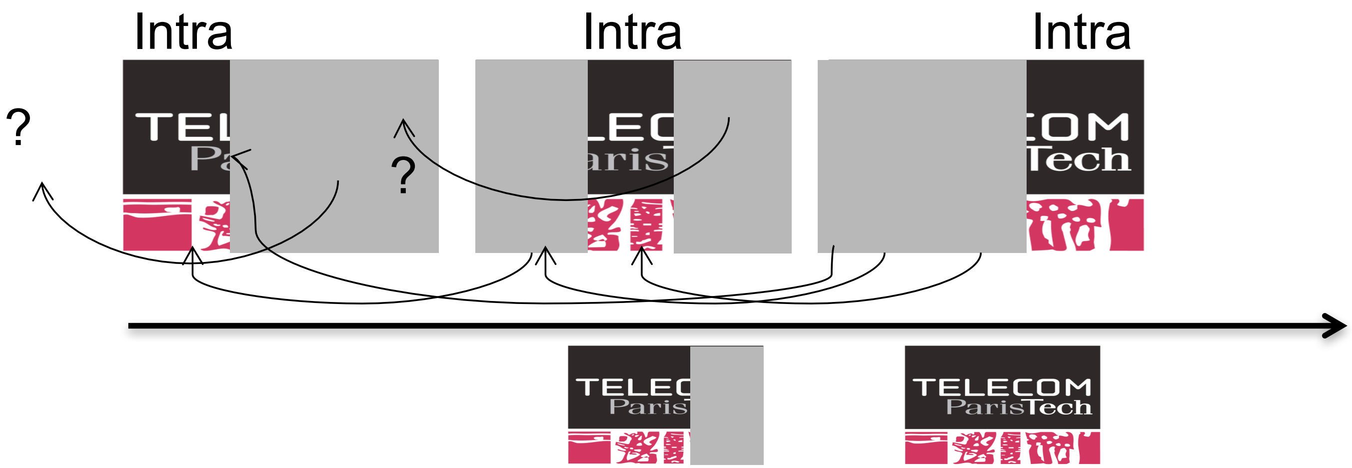
Compression



Random Access in video

Compression

- **Point at which decoding may start**
  - I-frame (before AVC), IDR (AVC, HEVC)
- **Gradual Decoding Refresh**
  - Rebuild reference image over N frames without any previous RAP
  - Smooth data rate (avoid burst of I-frame)
  - Allows gradual random access (partially correct) after the RAP



Typical bitrates

- **Terrestrial Digital TV**
  - SD program: 4 Mbps in MPEG-2
  - HD program: 8 Mbps in AVC, 4 Mbps in HEVC
  - UHD program: 16->22 Mbps in HEVC, 30 Mbps + in AVC
- **Typical Netflix rates (AVC + AAC)**
  - 512x384: 560 Kbps, 750 Kbps
  - 640x480: 1050 Kbps, 750 Kbps
  - 720x480: 1750 Kbps
  - 1280x720: 3 Mbps, 2.35 Mbps
  - 1920x1080: 5.8 Mbps, 4.3 Mbps
- **Audio**
  - 128->512 kbps for stereo MP3
  - 64->256 kbps for stereo AAC or AC3
  - 220->1200 kbps for FLAC
- **cf http://www.digitalbitrate.com**

AV Transport

Synchronization

Intra Stream Synchronization

- **Correct playback of one media**
  - Audio, video, text, …
- **Render AUs at the right time**
  - Ex: video 25 FPS -> one frame every 40 ms
- **Deal with AU losses**
  - Numbering or assigning a time
- **Frames with variable durations (subtitles)**
  - ⇒ Numbering is not enough
  - ⇒ **Assign a presentation/rendering time to each AU**

Synchronization: Core Concepts

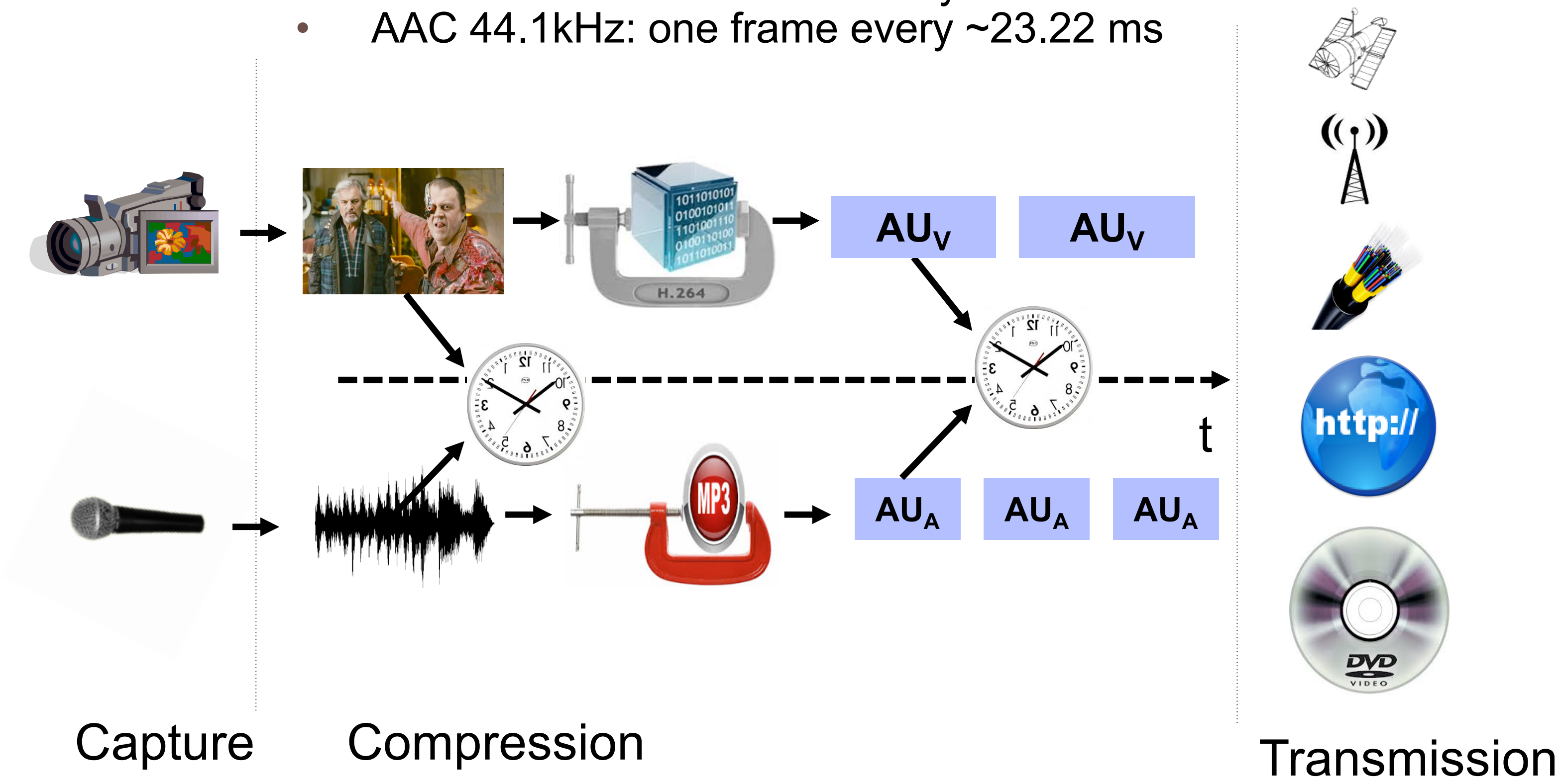
- **Time Scale**
  - Number of ticks in one second
  - Ex for 24 FPS video: 24, 24000, 90000 ticks/s
- **Time Stamps**
  - **Rational Number** (not real)
  - T_{sec} = TimeStamp / TimeScale
  - Precision errors if bad time scale
- **Clock**
  - Reference time to which timestamps are compared
  - May be explicit (RTP, MPEG-2 TS)
  - May be implicit (eg, first AU TimeStamp = 0), mostly for files
- **Presentation Time or Composition Time**
  - PTS, CTS, RTP TS
  - Time at which the frame data shall be rendered/presented

Importance of the time scale

- **Example with audio (AAC)**
  - Coding window 1024 samples
  - Capture Sample rate 44100 Hz
  - Window length 1024/44100=0,023219954648526
  - TimeScale 1000, AU TimeStamps:
    - TS_0: 0: == real sampling time
    - TS_2: 23: < real sampling time
    - TS_3: 46: < real sampling time
    - TS_4: 69: < real sampling time
    - TS_5: 93: > real sampling time
  - TimeScale 44100, AU TimeStamps:
    - TS_x: x*1024: == real sampling time

Inter Stream Synchronization

- **Correct Playback of a set of media**
  - Audio + Video
  - Audio + Text, Video + Text
  - AV + application, …
- **Different sampling frequencies per media**
  - 25 FPS Video: one frame every 40 ms
  - AAC 44.1kHz: one frame every ~23.22 ms

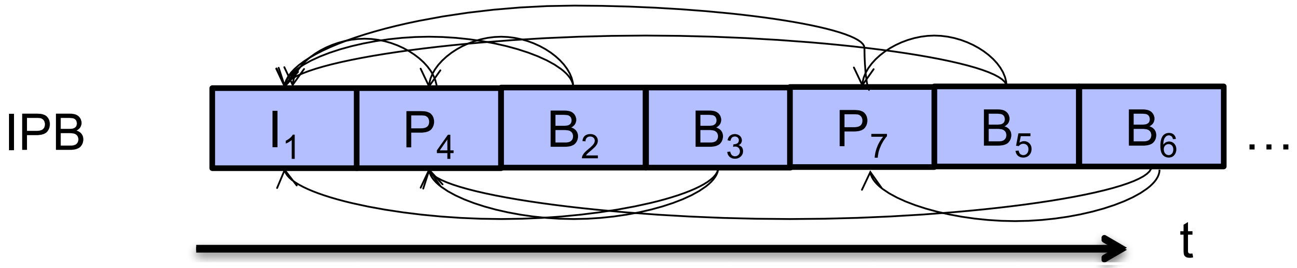


Inter Stream Synchronization Strategies

- **Defining a common time base**
  - Synchronization by comparing timestamps of each AU
  - Use a time scale introducing few rounding errors (least common multiple, …)
  - Synchronization done at the source
    - Not practical if more than one source
- **Correlate different time bases**
  - Identify a common synchronization time anchor T_{origin} for each media and its corresponding T_{media}
    - Implicit (0-based) in most file formats
    - Or using world clock in other cases (cf RTP)
  - Synchronize AUs by comparing $(TimeStamp_{AU} – T_{media})$ / $timescale_{Media}$

DTS: Decoding Time Stamp

- **Frames are not always decoded in the order of presentation**
  - Bi-directional video coding: B-frame
  - If same order, DTS ⇔ CTS



DTS	1	2	3	4	5	6	7
CTS	1	4	2	3	7	5	6
CTS-DTS	0	2	-1	-1	2	-1	-1
CTS'	2	5	3	4	8	6	7

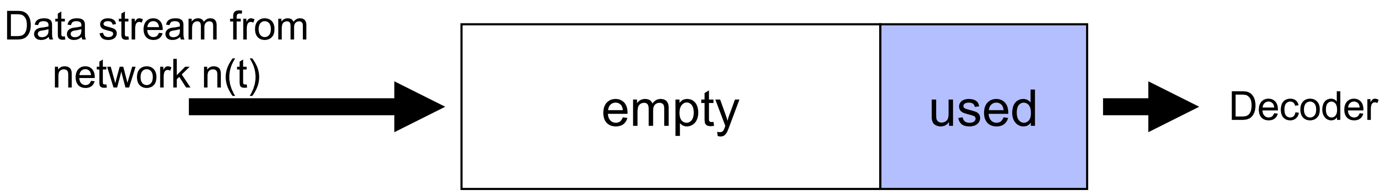
Cannot present before decode !

- **CTS needs adjustment to always have CTS>DTS**
  - This reflects the buffering, at the encoder side, of N frames for bi-directional prediction

TELECOM ParisTech

Synchronization and buffers

- **Constrained Systems**
  - Low, fixed memory available at the input of the decoder
  - Typical cases
    - Set-top Boxes MPEG-2 TS chips
    - Embedded video systems
- **Buffer Model**
  - allows for the encoder to control how much of the decoder's memory is available
    - Avoid buffer overflow
  - Terminology
    - MPEG-2: **V**ideo **B**uffer **V**erifier
    - AVC & HEVC : **H**ypothetical **R**eference **D**ecoder

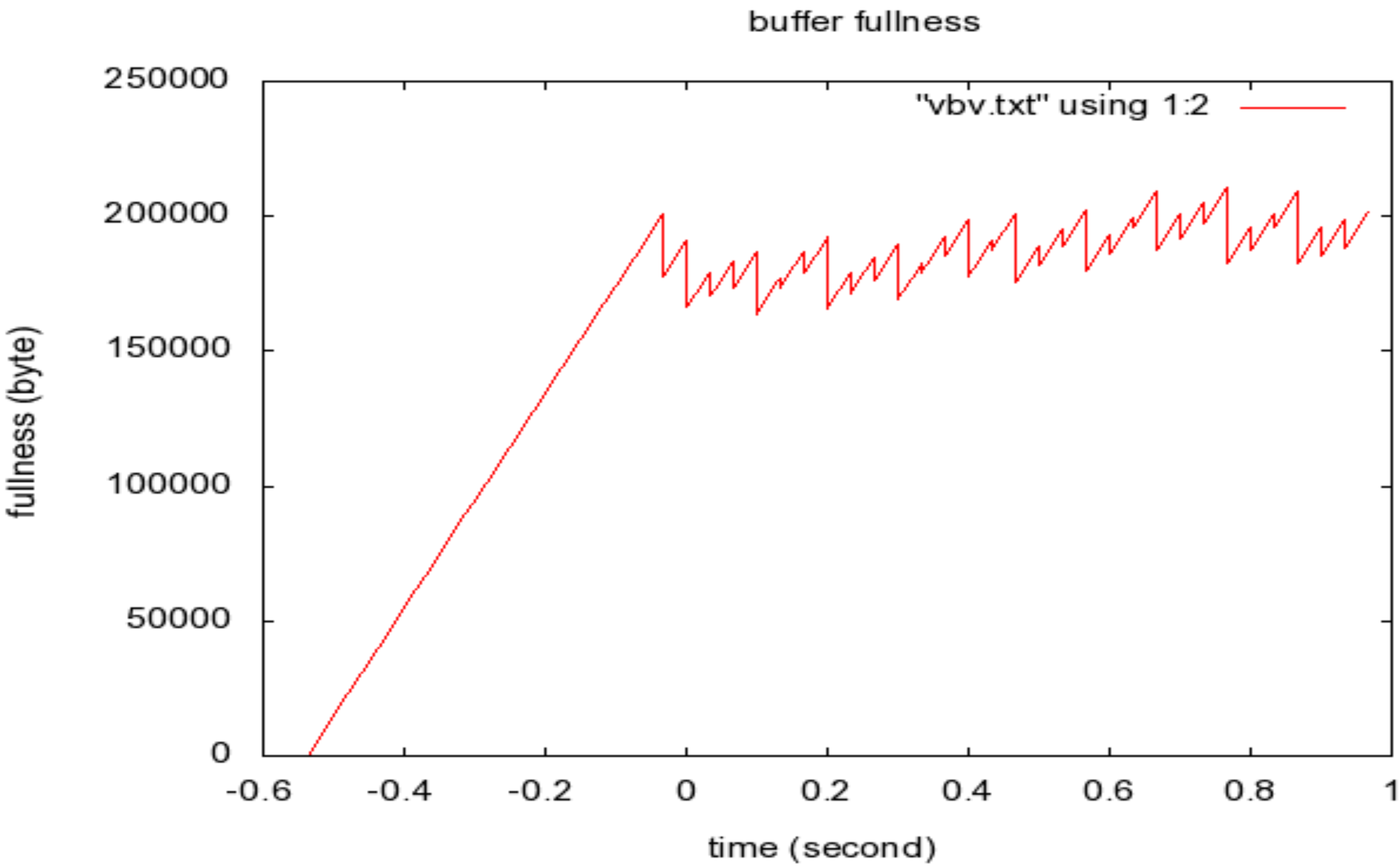


Buffer Fullness:

$$x(t) = \frac{\int_0^t (n(u) - c(u))\,du}{s}$$

- x(t) = 0 may be an issue if decoder expects new data
  - Starvation or buffer underflow
- x(t) = 1 is an issue if other data arrives from network
  - « buffer overflow » causing data loss

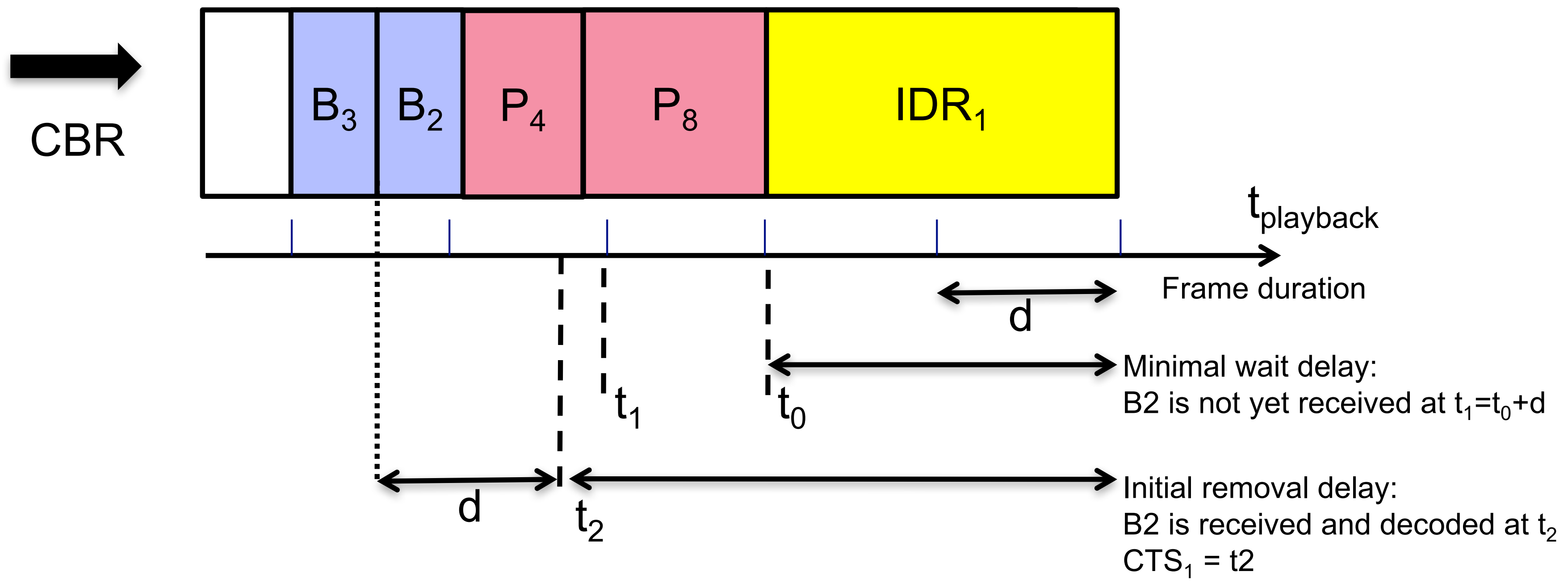
Buffer Fullness example



http://codesequoia.wordpress.com/2010/04/19/buffering-delay-and-mpeg-2-transport-stream/

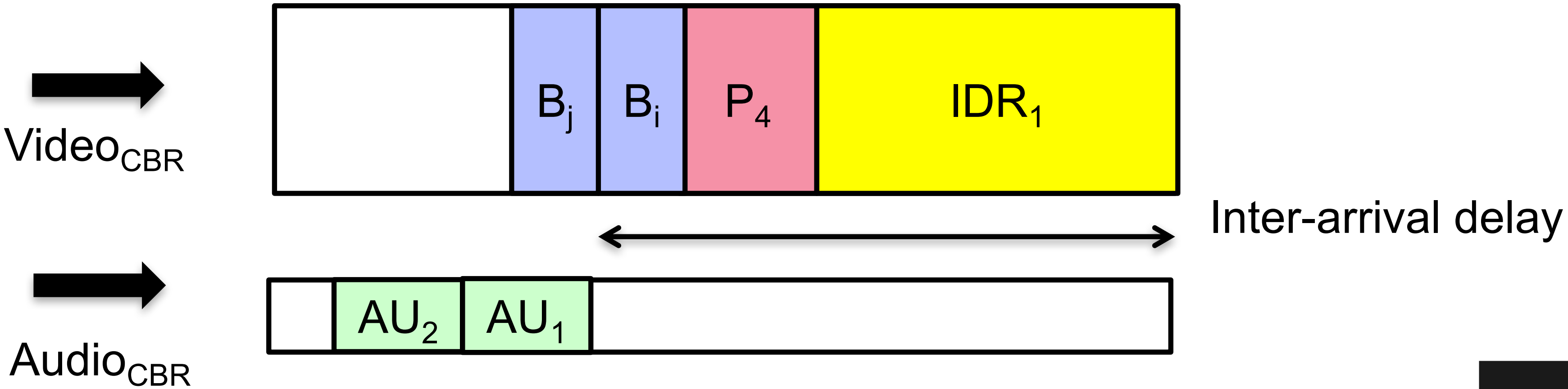
Buffer in compression

- **CBR = Constant Bit Rate**
  - However frames have different sizes: S(I) > S(P) > S(B)
  - CBR ⇔ decoder input rate constant but leveled across different frames
  - However frames are usually at constant frame rate ??
- **Decoding Buffer solves the issue**



Buffer levels

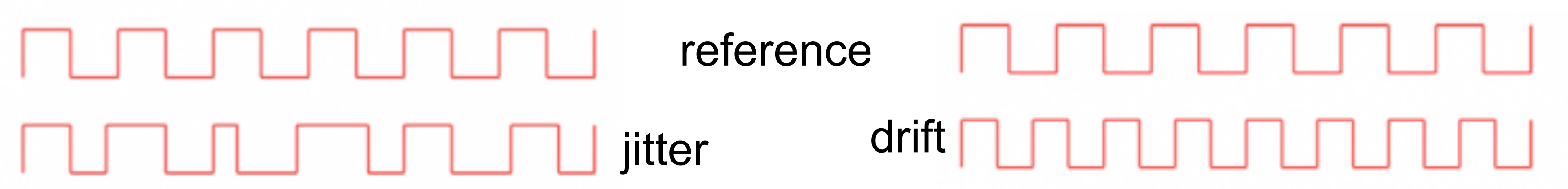
- **Trade-off between memory costs and quality**
  - MPEG-2 Video: 0,7 sec
  - AVC: max 4 to 10 sec
    - 1 to 2 sec for TV broadcast
- **Inter-Stream Synchronization**
  - First bytes of video frame sent much before the DTS to fill-up the buffer
  - However audio frames don't have this buffer management issue (low frame size and ~constant)
    - Decoding as soon as received



TELECOM ParisTech

Trusting a clock

- **Hardware clock not reliable**



- **System clock jitter**
  - Frames not displayed at regular interval
- **System clock drift**
  - $Clock_{DEC}(t) < Clock_{ENC}(t)$
    - AUs are not removed fast enough: buffer overflow, frame loss
    - Longer playback
  - $Clock_{DEC}(t) > Clock_{ENC}(t)$
    - AUs are removed too fast : buffer under-run, frame freeze
    - Shorter playback
  - Problematic if multiple devices play the content
    - Conferencing systems
    - Second screen applications
    - Multiple audio systems

Trusting a clock

- **Checking system clock**
  - Different devices use different time servers
    - Not always in sync !
  - UTC Drift can be important on most hardware
    - A few seconds/day is quite common
    - Check your device UTC sync at http://time.is
- **Use audio hardware rather than system clock**
  - But audio hardware clocks drift too …

Example: Mobile device clock drifts

Device	Drift in parts/million (ppm)	Drift (ms/min)
Apple iPad 2 Wi-Fi	+11.96	+0.72
Apple iPhone 6+	+416.84	+25.01
Asus Nexus 7 2012	+2.92	+0.18
LG Nexus 4 (rev. 10)	+6.74	+0.40
LG Nexus 5	+6.44	+0.39
Samsung Galaxy Nexus	+79.67	+4.78
Samsung Galaxy Note 10.1	+17.03	+1.02
Samsung Galaxy S	+4.33	+0.26
Samsung Galaxy S2	+273.93	+16.44

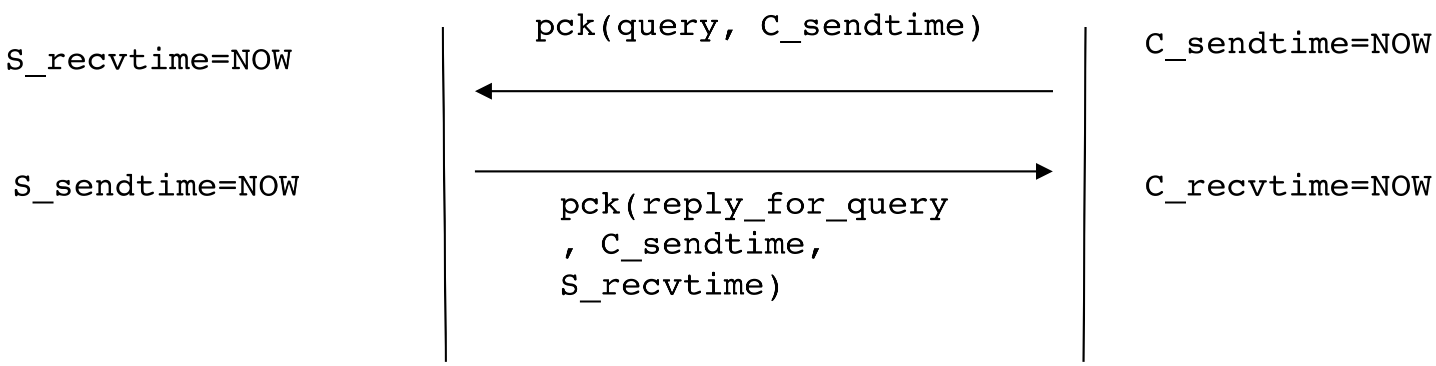
http://protyposis.net/clockdrift/high-precision-audio-drift-measurements-with-gps/

Trusting a clock

- **Send encoder clock**
  - MPEG-2 TS "Program Clock Reference"
  - Works fine if no jitter
- **Locally rebuild the clock**
  - Get anchor time
    - Source/server Time T_S for given media time T_M
    - usually using NTP
      - But not error-prone
  - Use system or audio clock once anchor is found
  - Estimate drift
    - compare to world clock (UTC, GPS) on regular basis
- **Assume no drift at all at the encoder side !**

Querying clocks

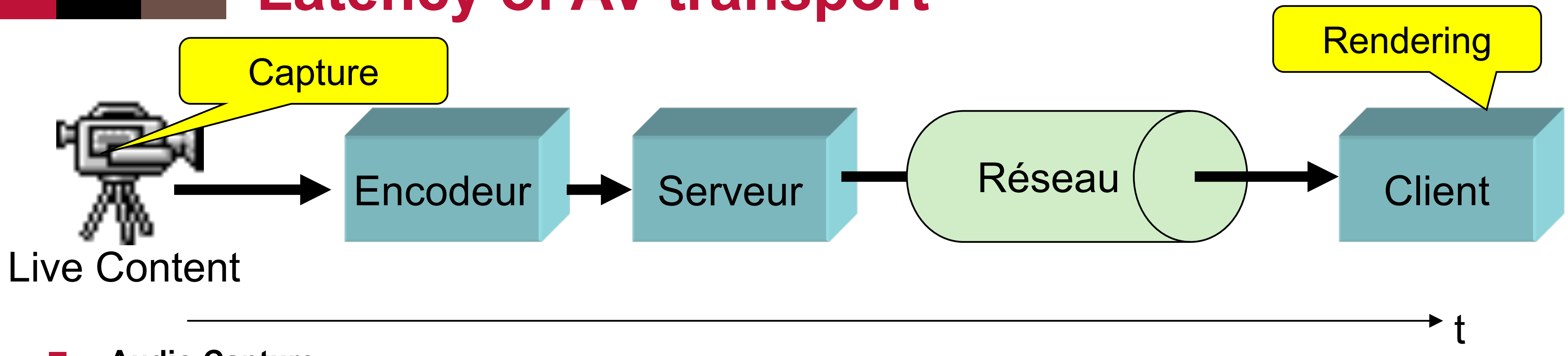
- **Usage**
  - Correcting local clock drifts
  - Synchronizing multiple devices
- **Where**
  - NTP (Network Time Protocol)
    - Used to keep your device on time
  - RTP/RTCP
    - Used to estimate streams jitter
  - DVB-CSS for multi screen synchronization
- **Principles**




```
delay = (S_recvtime - C_sendtime + C_recvtime - S_sendtime) / 2
```

- Symmetrical delays are assumed
- A few exchanges are usually required

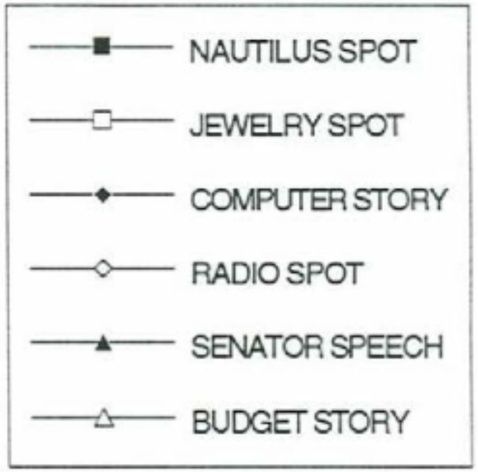
Latency of AV transport

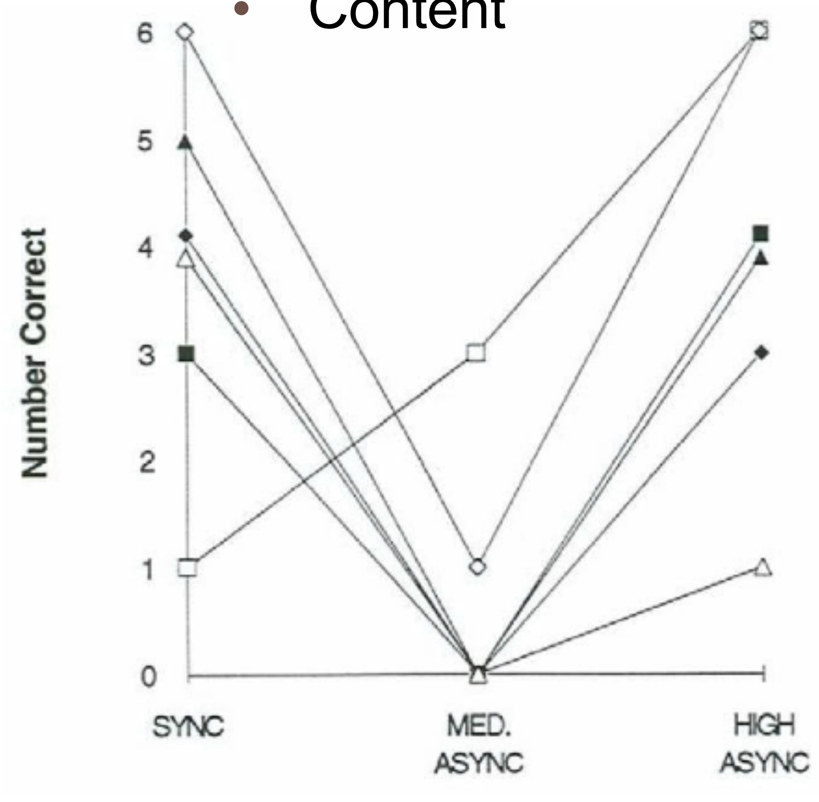


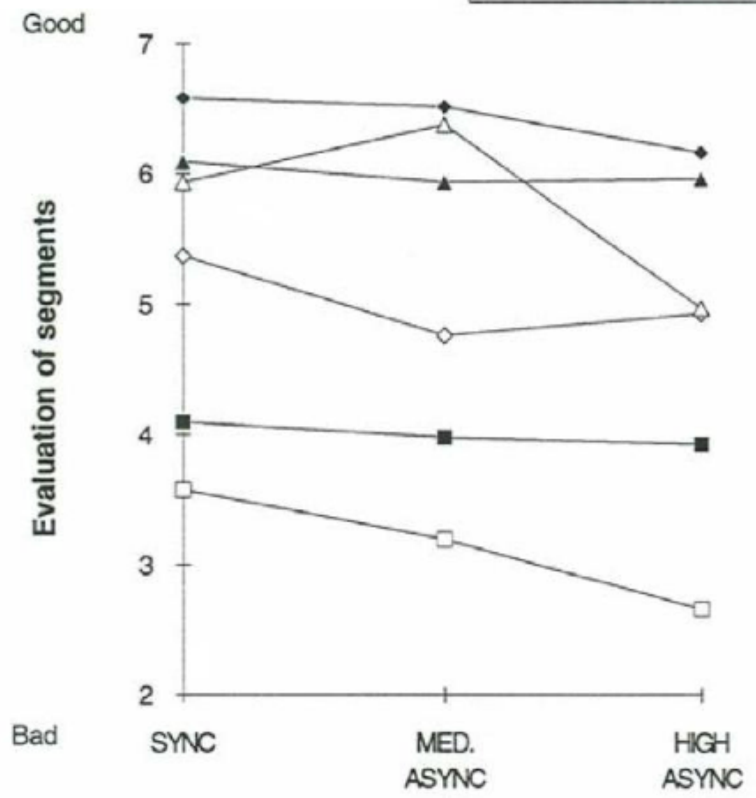
- **Audio Capture**
  - Sound card captures K (>1) samples per cycle
- **Pre-encoding**
  - Audio encoder consumes N (!= K) samples per cycle
    - Additional delay to setup for complete audio chunk for the encoder
- **Encoding**
  - Not instantaneous
  - Video encoding using bi-directional coding
    - Additional delay to buffer P (resp. B) frames before coding B (resp. hierarchical B) frames
- **Packetization**
  - More or less complex, depending on the transport format
- **Networking**
  - Delivery time > 0
  - Over IP:
    - Potentially different path (and delays) for audio and video streams
    - Packets may arrive out of order
    - Jitter: variation of the delivery time
- **Client side**
  - De-jittering buffers
  - Decoding buffers
  - Audio card feeding

Synchronization errors

- **Sync loss perception depends on**
  - People
  - Content







Effects of Audio-Video Asynchrony on Viewer’s Memory, Evaluation of Content and Detection Ability
Standford University

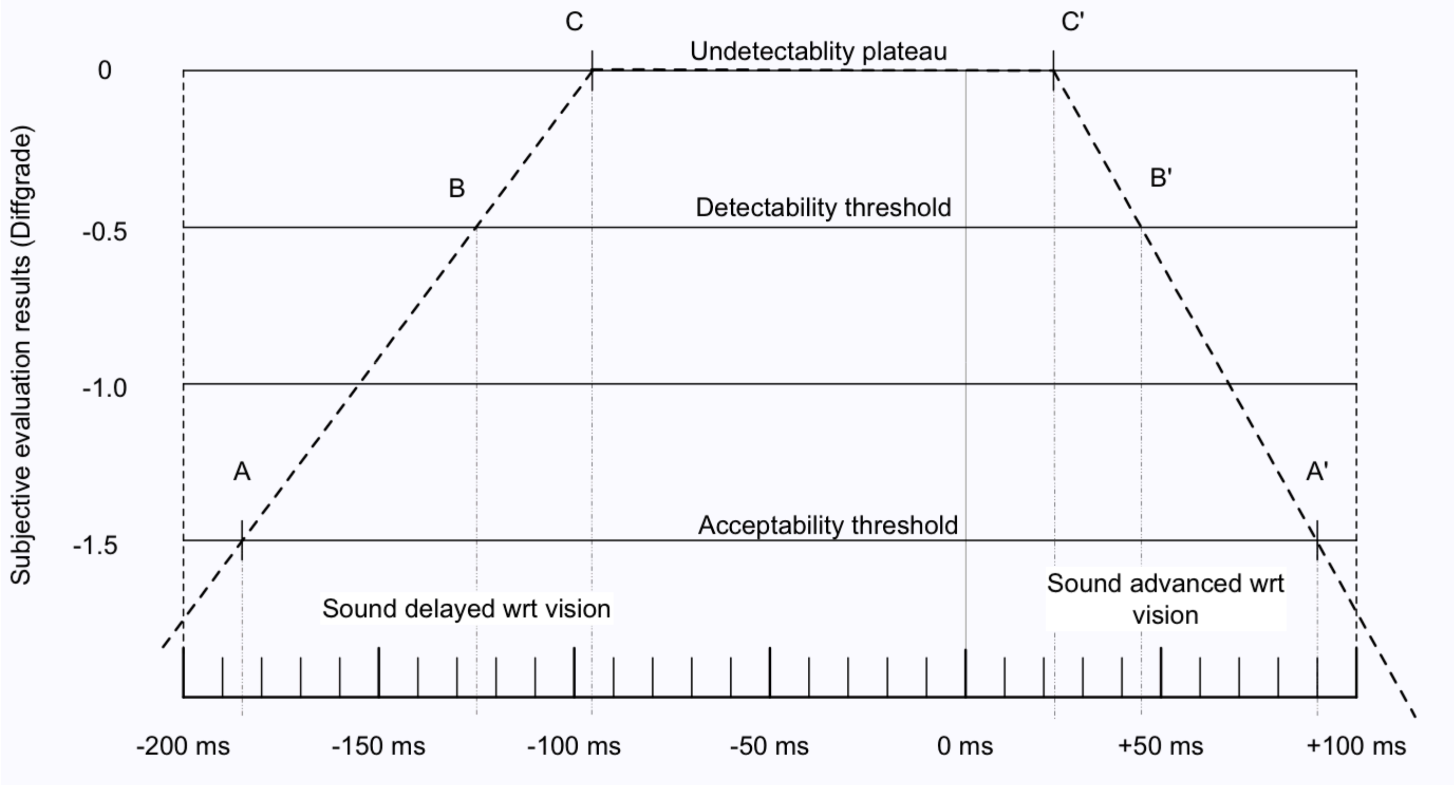
Impact of Synchronization Errors

- **Various effects**
  - Negative evaluation of the content
  - Async content better remembered
- **Regardless of delay perception**
- **Human AV sync perception**
  - Used to sound being late / visual (speed of sound vs speed of light)
  - Discomfort if sound ahead of the video

AV Synchronization in practice

Acceptability Window

- (audio – video) timing
- +90ms to -185ms



ITU-R BT.1359 Figure 2

AV Synchronization in practice

- **Digital TV**
  - EBU Recommendation R37
    - Advance audio max 40 ms
    - Delay audio max 60 ms
  - ATSC IS-191
    - Advance audio max 45 ms
    - Delay audio max 75 ms
- **Video conferencing**
    - Similar numbers
    - Sensibility depends on image resolution and quality!!

Precise Synchronization: advanced cases

- **New service types**
  - Scalable coding
    - Per layer (eg SVC/SHVC): switch SD -> HD, SNR
    - Multiple Description Coding (MDC)
  - 3D Services
    - Views are coded and transported independently
- **New needs for synchronization:**
  - Decoding synchronization : no errors allowed !
  - If delay on one stream
    - Broken decoding (multi-layer coding)
    - Broken Reconstruction (misalignment of left/right views)

Bidirectional Synchronization

- **Delays involved**
  - Downlink and uplink delay
  - (capture + coding + transport + decoding + rendering) * 2
- **Depends on application**
  - Medical, video-surveillance, remote working, gaming
- **Traditional Conferencing**
  - ~ 200 up to 400 ms round-trip
    - 100 up to 200 ms
- **« Hard-interactivity »**
  - <100 ms round-trip
    - 50 ms delay

AV Transport

Exercices

Raw Video

- Compute the bitrate of an uncompressed HD stream, 25 fps, interlaced, 4:2:0, 8 bit ?
- What would the bitrate in progressive mode ?

Raw Video

- **Compute the bitrate of an uncompressed UHD stream, 50 fps, progressive, 4:2:0, 8 bit ? 4:2:2, 8 bit ?**

Raw Video

- Compute the raw bitrate of an uncompressed UHD stream, 50 fps, progressive, 4:2:0, 8 bit ? 4:2:2, 8 bit ?

Raw Video

- **Compute the raw bitrate of an uncompressed UHD stream, 50 fps, progressive, 4:2:0, 10 bit ? What assumption do you make on byte alignment?**

Raw Audio

- **Compute the bitrate of**
  - an uncompressed audio stream stereo, 44100 Hz, 16bit ?
  - an uncompressed audio stream 5.1, 48000 Hz, 16bit ?

Raw Audio

- **How many bytes of uncompressed data does an AAC access unit contain for a stereo stream 44100 Hz, 16bit**

Raw Audio

- **What is the duration of an AAC access unit of a stereo stream 44100 Hz, 24bit, 5.1 channels ?**

AU	DTS	CTS	Size	RAP	Offset
1	0	2000	249941	1	50480
2	2000	8000	31166	0	300421
3	4000	4000	3579	0	331587
4	6000	6000	3926	0	335166
5	8000	14000	82795	0	339092
6	10000	10000	4753	0	421887
7	12000	12000	4847	0	426640
8	14000	20000	57419	0	431487
9	16000	16000	4317	0	488906
10	18000	18000	4453	0	493223
11	20000	26000	77765	0	505763
12	22000	22000	5318	0	583528
13	24000	24000	5592	0	588846
14	26000	32000	59532	0	594438
15	28000	28000	4156	0	653970
16	30000	30000	4742	0	658126
17	32000	38000	59661	0	662868
18	34000	34000	5227	0	722529
19	36000	36000	5399	0	727756
20	38000	44000	92156	0	733155
21	40000	40000	6582	0	833141
22	42000	42000	6674	0	839723
23	44000	50000	62309	0	846397
24	46000	46000	6445	0	908706
25	48000	48000	6665	0	915151
26	50000	56000	68618	0	921816
27	52000	52000	5192	0	990434
28	54000	54000	6355	0	995626
29	56000	62000	70309	0	1001981
30	58000	58000	7022	0	1072290
31	60000	60000	7330	0	1087001
32	62000	66000	53970	0	1094331
33	64000	64000	5577	0	1148301
34	66000	68000	474588	1	1153878
35	68000	74000	27197	0	1628466
36	70000	70000	4705	0	1655663
37	72000	72000	5690	0	1660368
38	74000	80000	42121	0	1666058
39	76000	76000	5755	0	1708179
40	78000	78000	6342	0	1713934
41	80000	86000	79436	0	1728031
42	82000	82000	8731	0	1807467

- What is the framerate of the video ?
- What is the average gop length in second ?
- Comment the CTS value of the first AU
- Comment the GOP pattern

Timescale 50000

TELECOM ParisTech

AU VIDEO	DTS	CTS	Size	RAP		AU AUDIO	DTS	CTS	Size	RAP
1	0	2000	249941	1		1	0	0	334	1
2	2000	8000	31166	0		2	1024	1024	492	1
3	4000	4000	3579	0		3	2048	2048	456	1
4	6000	6000	3926	0		4	3072	3072	400	1
5	8000	14000	82795	0		5	4096	4096	457	1
6	10000	10000	4753	0		6	5120	5120	444	1
7	12000	12000	4847	0		7	6144	6144	502	1
8	14000	20000	57419	0		8	7168	7168	471	1
9	16000	16000	4317	0		9	8192	8192	414	1

Timescale 50000

Timescale 44100

- Comment on audio timescale
- How would you describe the AV synchronization for this content ?

TELECOM ParisTech