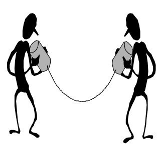
IEEE INFOCOM 2001

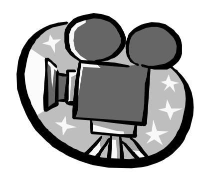
Tutorial T5

Video Over IP

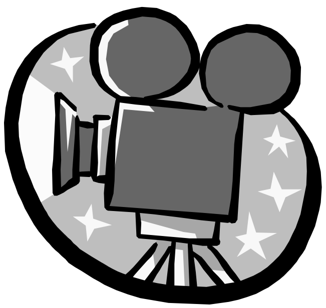
Magda El-Zarki (University of California at Irvine)

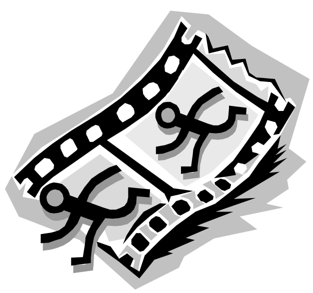
Monday, 23 April, 2001 - Morning

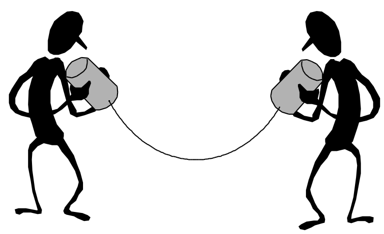




MPEG-4 over IP - Part 2



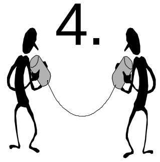


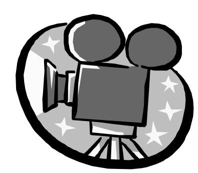


Magda El Zarki
Dept. of ICS
UC, Irvine
elzarki@uci.edu

Outline of Tutorial

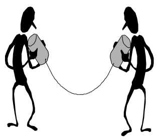
1. Part 1:
    1. Overview of Video Compression
    2. The MPEG suite
    3. Video Quality
2. <u>Part 2</u>:
    1. MPEG-4
3. Part 3:
    1. MPEG-4 Delivery over IP
4. Conclusions

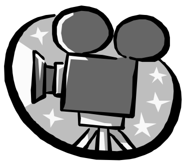




MPEG-4

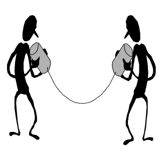
- 1999-2000: MPEG-4 Standard
  - Flexible Multimedia Communications
  - 5kbps - 50Mbps
  - Video object Compression
  - Audio object compression
  - Synthetic Audio/Speech and Video
  - Systems: multiplexing and flexible composition

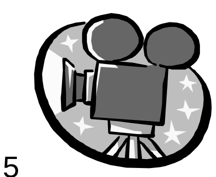




MPEG-4 Goal

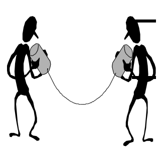
- To become THE standard for streaming AV media on the Internet and wireless networks
  – Better Audio/Video compression
  – Robustness against packet loss
  – Scalability of bit rate vs. quality

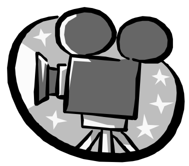




What’s new in MPEG-4

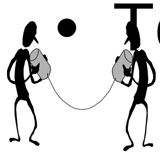
- Content-based Interactivity
  - Does not deem video frame as a whole anymore
  - Code each Audio-Visual (AV) object (AVO) into independent stream
  - More freedom to flexibly **interact** with what is ***within*** scenes
- Support integration of natural and synthetic AV media (“Virtual Playground”)
- Universal Access
  - Robust, independent of environment
  - Content-based scalability based on client’s request
  - Dynamically adaptive to available network bandwidth

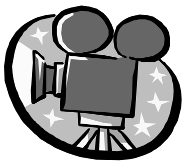




Audio-Visual Objects (AVOs)

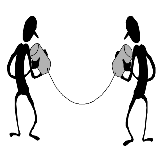
- Natural video sequence with/without shape information
- General 3D animation
- Still image
- Natural audio
- Structured Audio such as MIDI
- Text to speech (TTS)

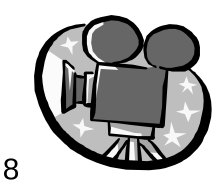




Pros of Content based Coding

- Improves reusability and coding efficiency of individual components
  - Apply different coding algorithms on different AVOs
- Allows content-based scalability
  - High resolution only on interesting part
  - Streaming object, pre downloaded object and local object can work together
  - Object based QoS support





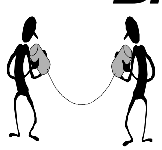
Integration of Natural and Synthetic Content

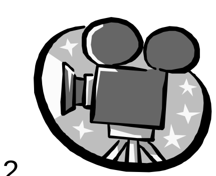
Augmented Reality

Tele Presence

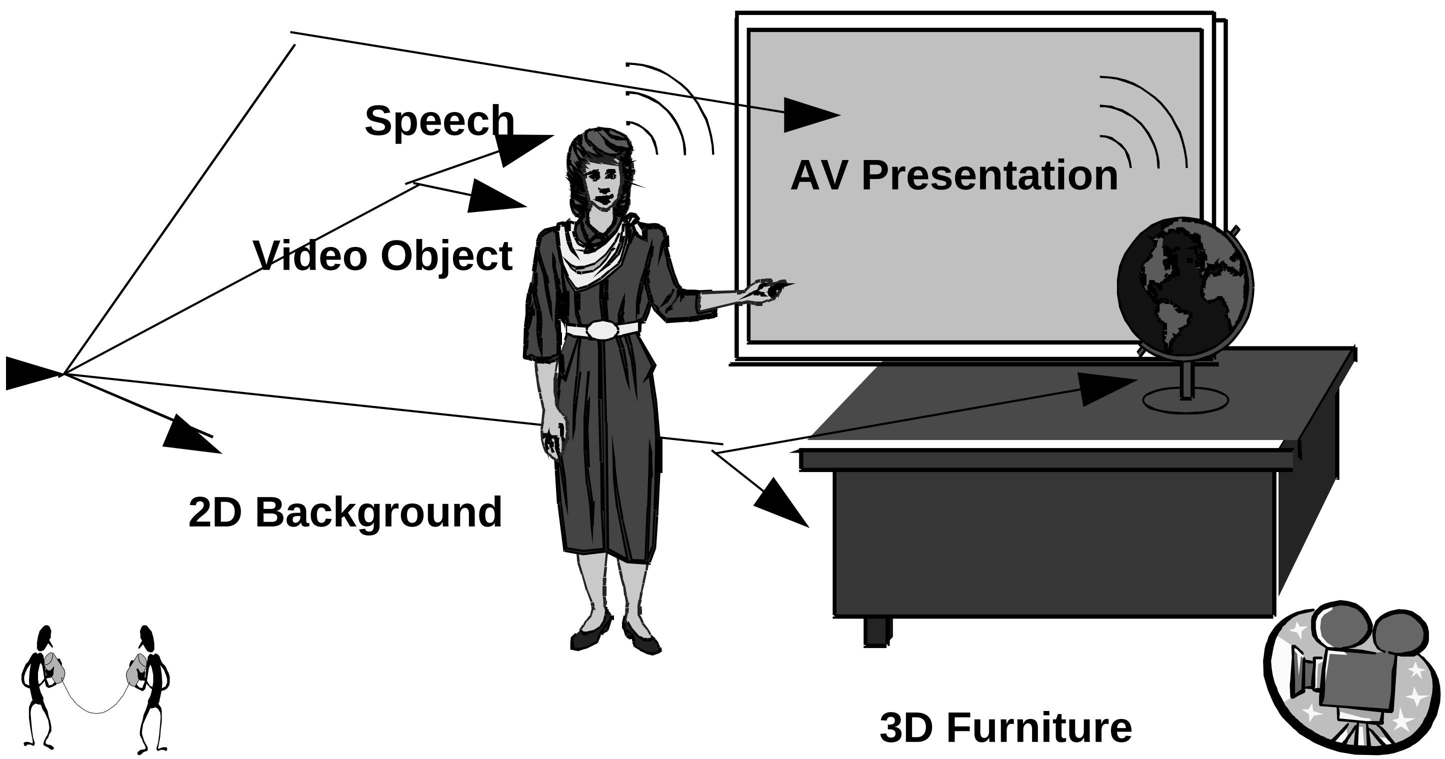
Scene-Composition

- A ‘**compositor**’ composes objects in a scene (A&V, 2&3D) creating a composite scene
- A scene description defines how AVOs appear on End User screen (composition view)
- With the scene description an End User can
  - change the position of individual video object
  - Zoom in/out interesting object
  - Choose different audio track (language, music)
  - Turn on/off individual object
- ***Bi***nary ***F***ormat for ***S***cene Description : ‘**BIFS**’

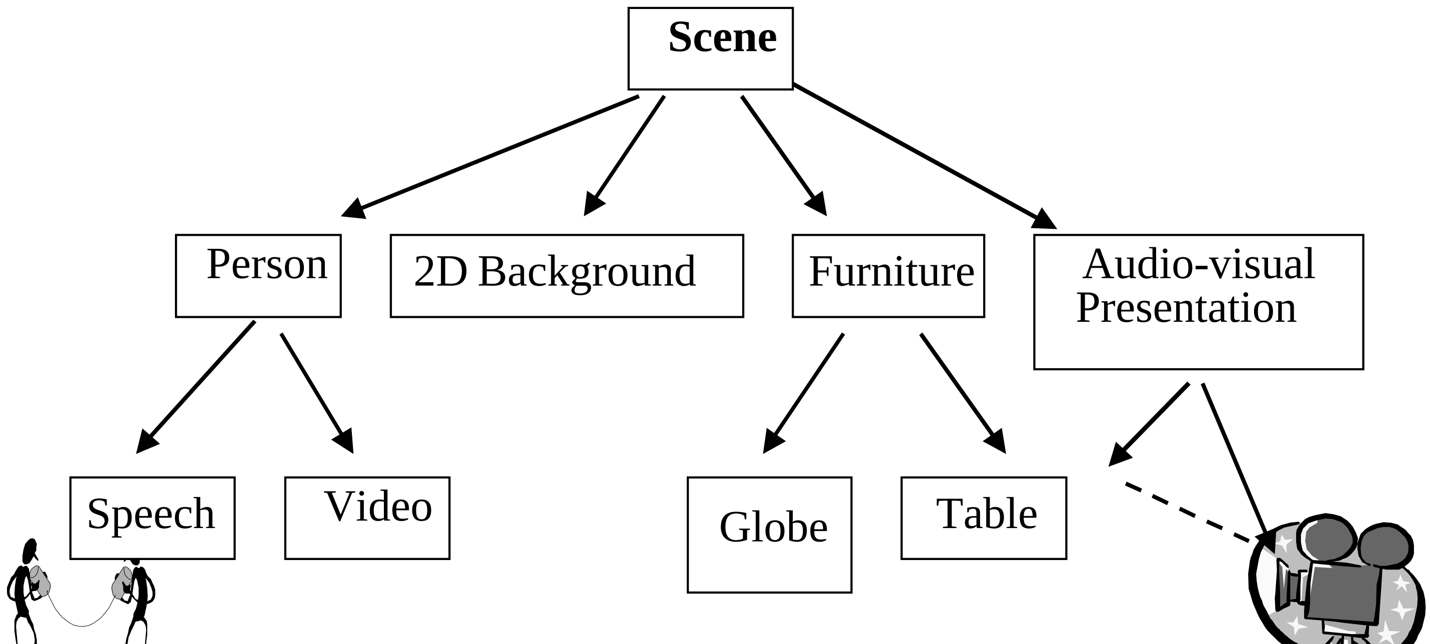




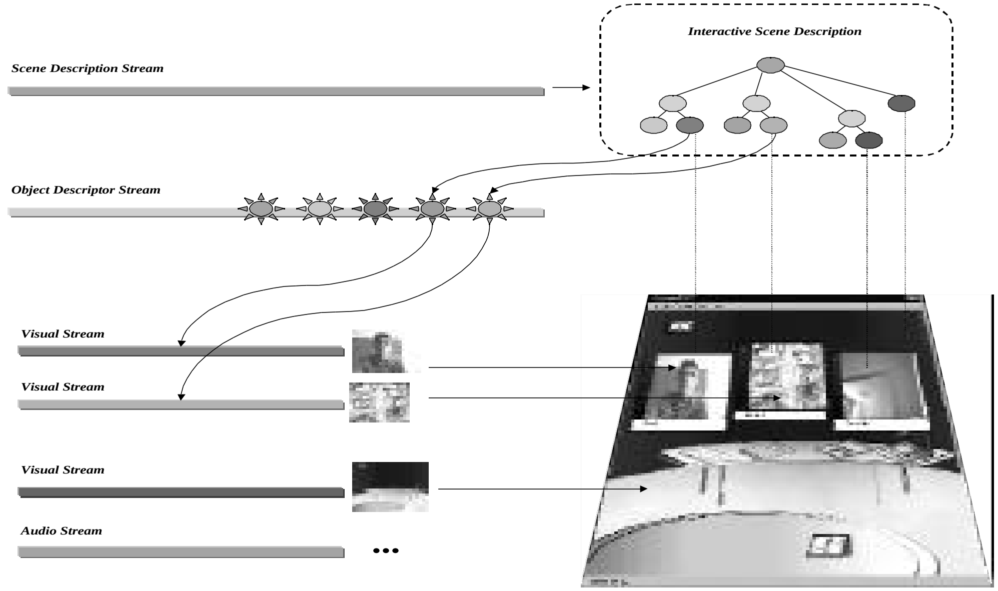
AVO Scene Composition

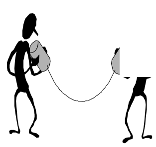


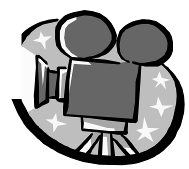
Composition of Scenes



MPEG-4 Streams

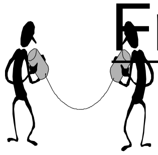


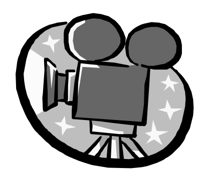




MPEG-4 Standard

- Part 1: System
- Part 2: Visual
- Part 3: Audio
- Part 4: Conformance Testing
- Part 5: Reference Software
- Part 6: Delivery Multimedia Integration Framework (DMIF)



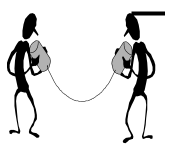


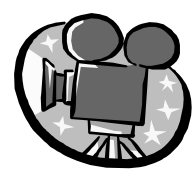
Part 1:System and file format

- ISO 14496-1

System Spec.

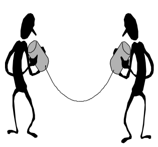
- Inherited from MPEG-1 & MPEG-2
  - Overall Architecture
  - Multiplexing
  - Synchronization
- MPEG-4 Specific
  - Scene Description
  - Content Description
  - Programmability

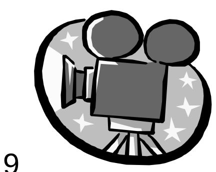




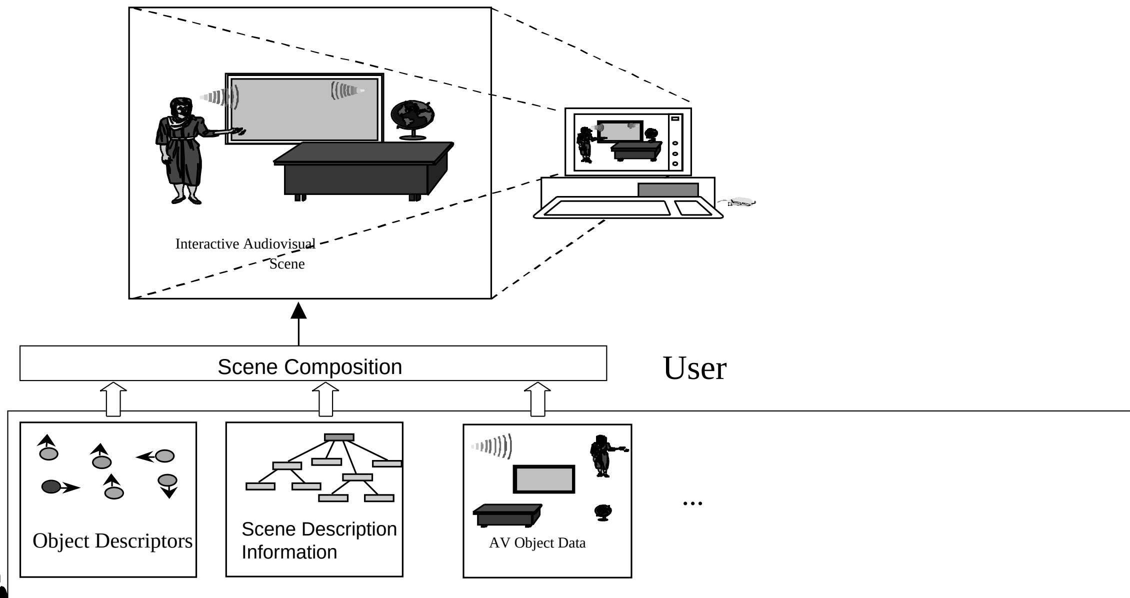
MPEG-4 System

- System Decoder Model
- Synchronization (Sync Layer)
- Management & Transport of streams

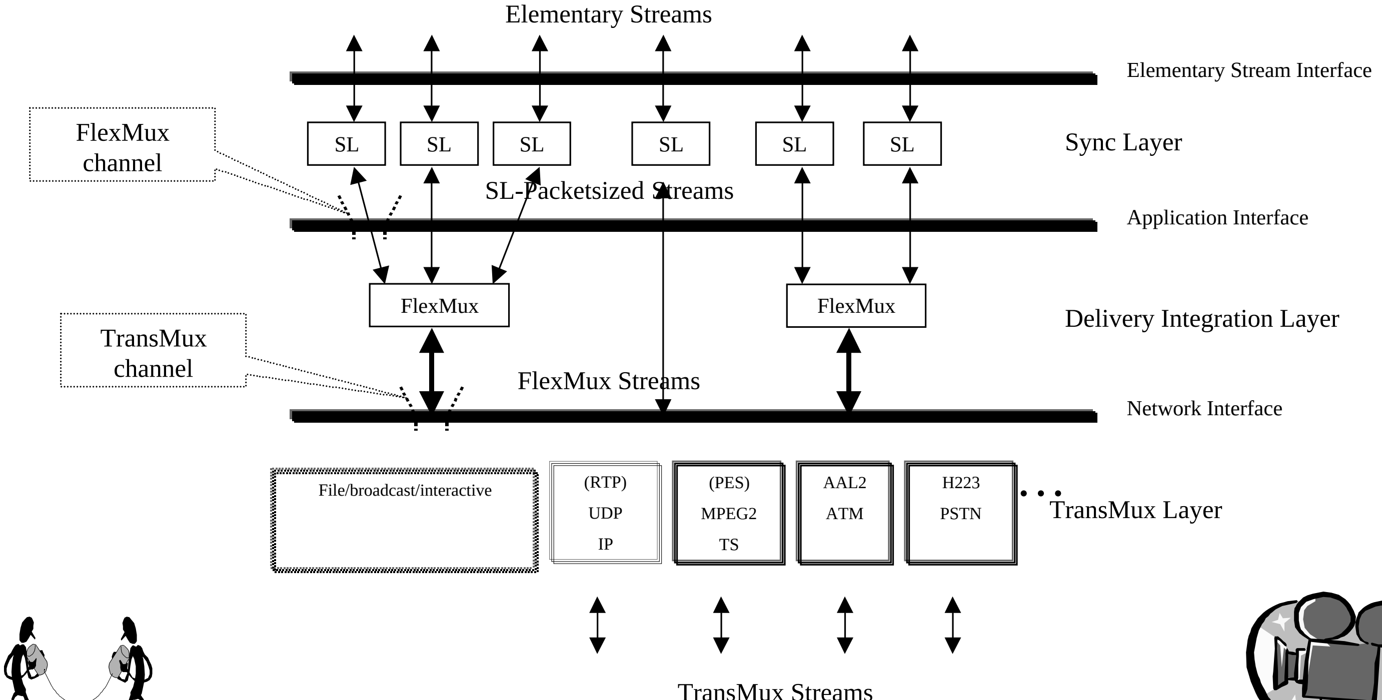




MPEG 4 Decoder Model

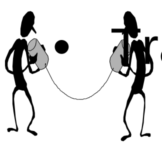


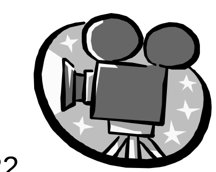
Sync Layer & Transport System

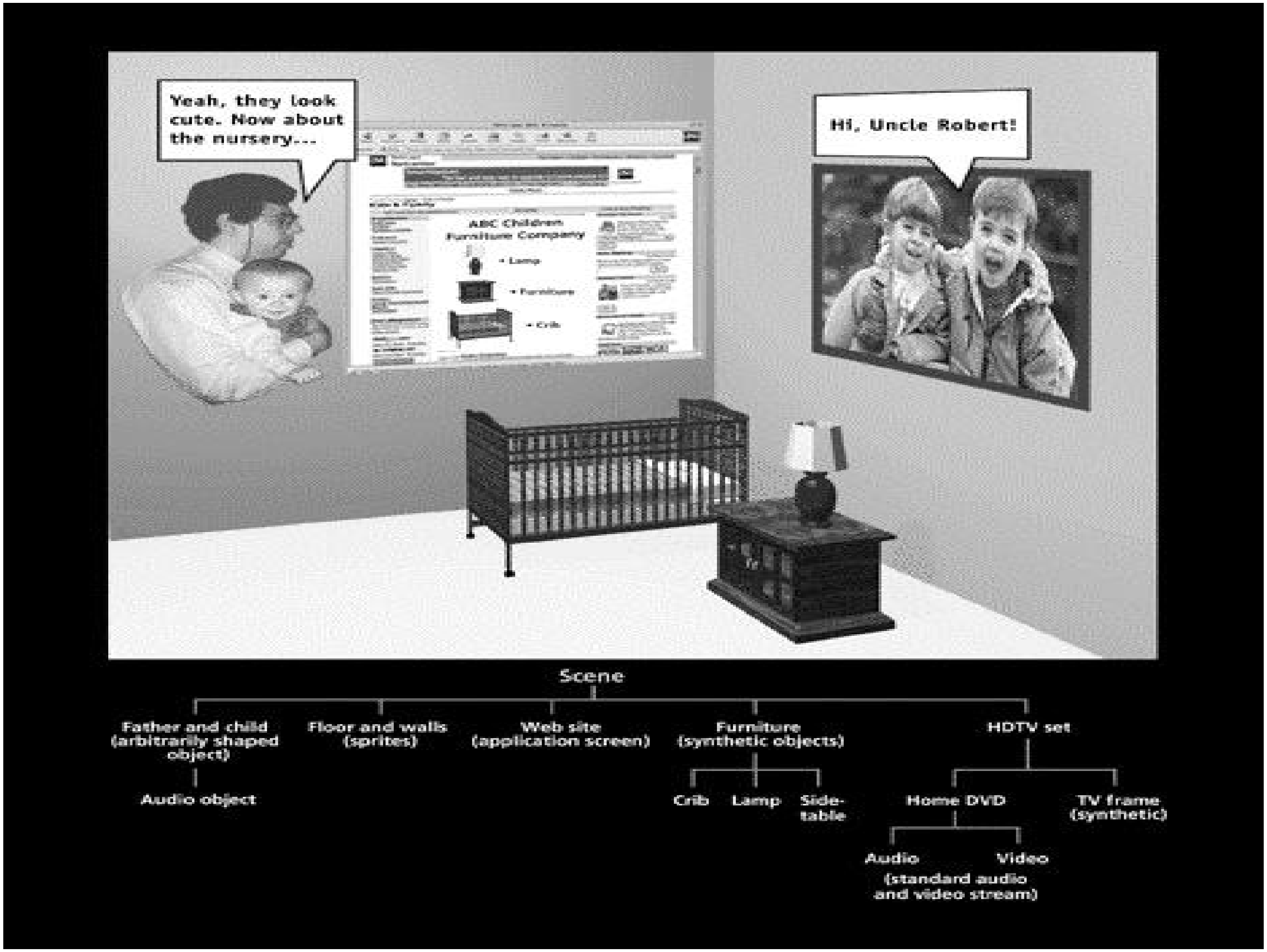


Scene Description (SD)

- Provide spatial-temporal relationship between audio-visual objects
- Tree structure – VRML based
- Contains pointers to object descriptors
- Provides Interactivity for End User with AV objects
  - Trigger event: e.g., start a video stream
  - Turn on/off individual object
  - Change object's position or client's viewing position
  - Select desired language if multi-language tracks exist
- Provides a Binary encoding of the scene (BIFS)
- Transported in the scene descriptor stream



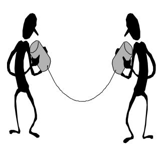


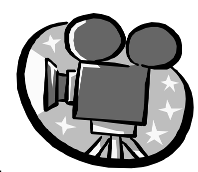


Yeah, they look cute. Now about the nursery...
Hi, Uncle Robert!
ABC Children Furniture Company
Lamp
Furniture
Crib
Scene
Father and child (arbitrarily shaped object)
Audio object
Floor and walls (sprites)
Web site (application screen)
Furniture (synthetic objects)
Crib
Lamp
Side-table
HDTV set
Home DVD
Audio
Video
(standard audio and video stream)
TV frame (synthetic)

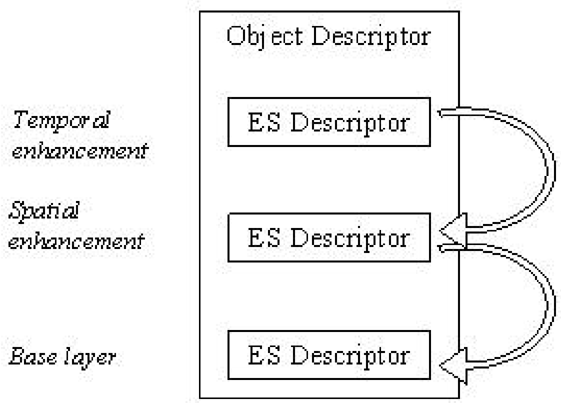
Object Descriptor (OD)

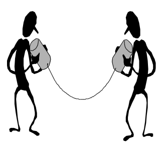
- Groups a set of Elementary streams associated with a particular object as a single entity (e.g. base and enhancement layers)
- Transported in object descriptor stream
- Object descriptors can be updated dynamically over time

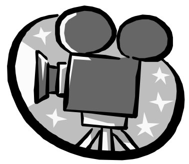




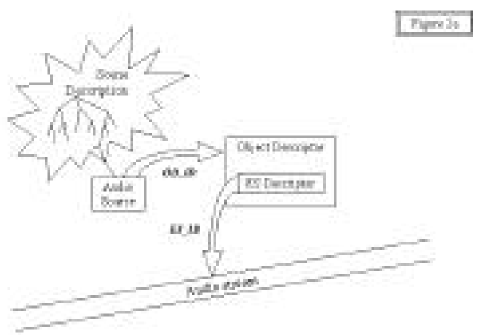
Generic Sample of an Object Descriptor

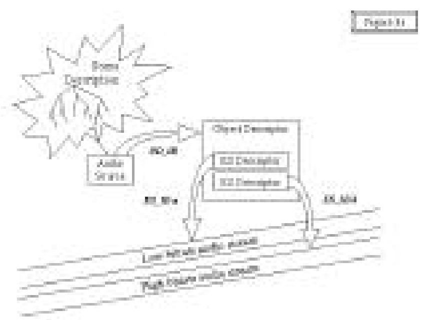


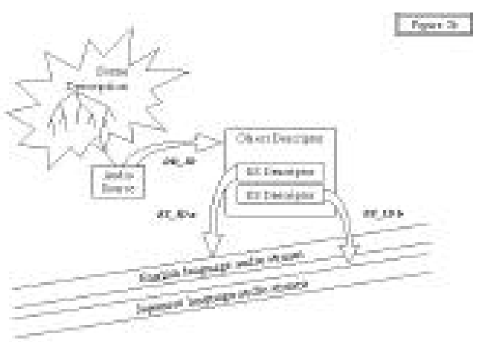


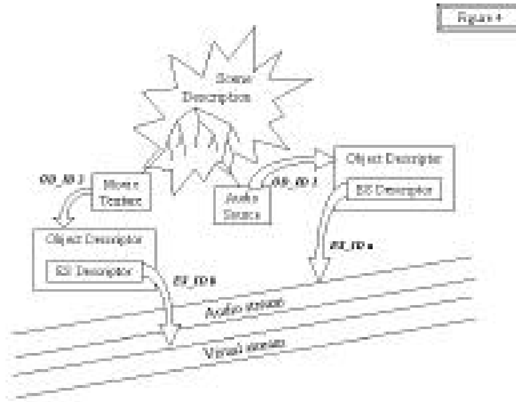


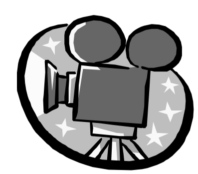
Examples of Object Descriptors





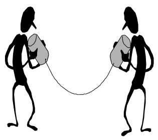


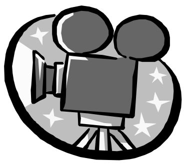




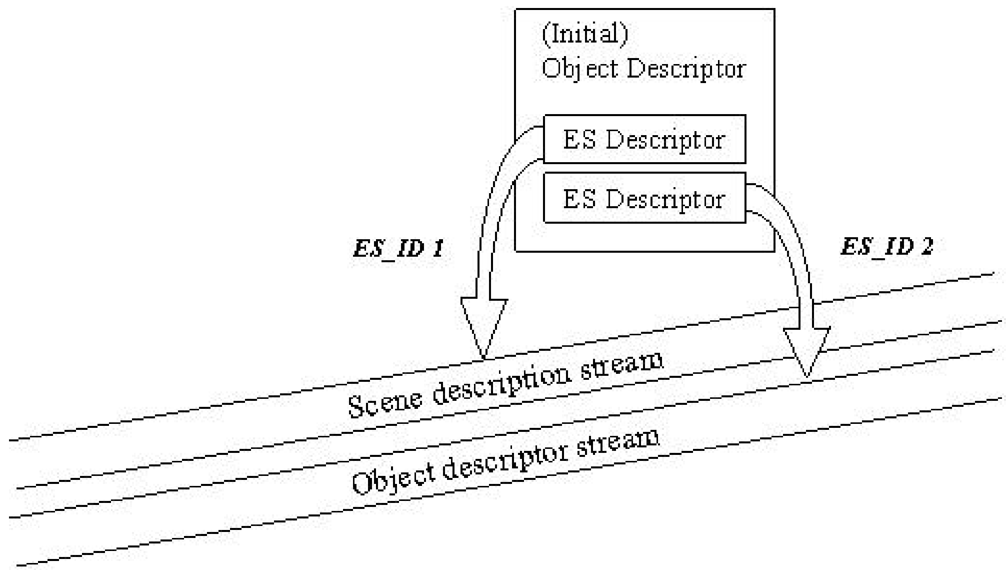
Initial Object Descriptor

- Identify SD & OD stream
- Convey profile & level information
- Usually transferred during session initialization phase



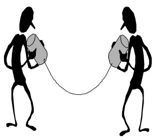


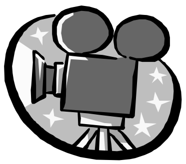
Initial Object Descriptor



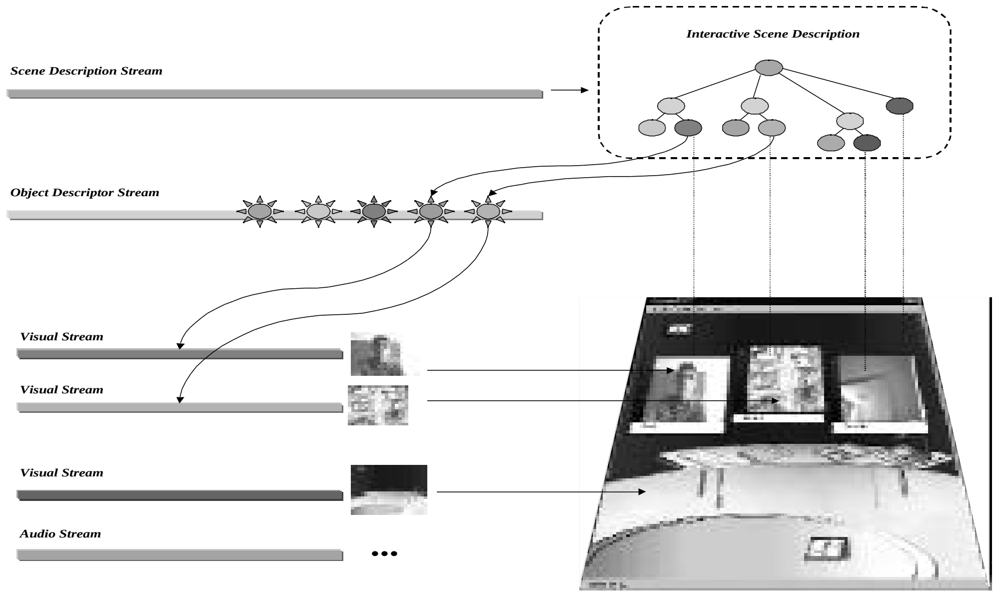
ES-Descriptors

- Each describes one Elementary Stream (ES) (audio stream, video stream, etc.)
- Includes configuration information for dedicated stream decoder (DecoderConfig)
- Contains sync layer configuration information for this stream (SLConfig)
- Conveys QoS Requirements to transport channel (***optional*** QoS descriptor)

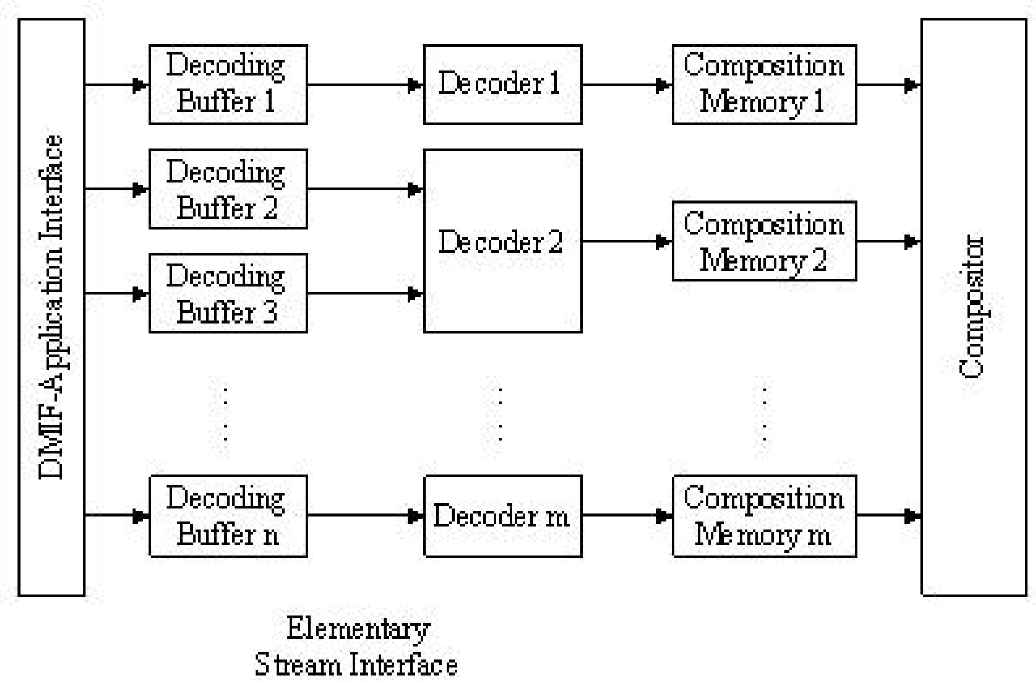




Example

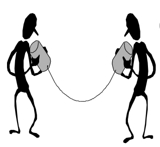

Interactive Scene Description
Scene Description Stream
Object Descriptor Stream
Visual Stream
Visual Stream
Visual Stream
Audio Stream
...

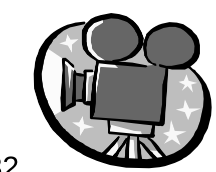
System Decoder Model



Synchronization Layer (SL)

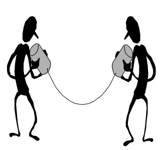
- The sole mechanism for implementing timing and synchronization in MPEG-4
- Packetizes ES into access units (AU) that correspond to complete video object planes (VOP)
  - AU is the smallest unit carrying time info
  - AU determines decoder buffer size
- Adds header with time stamp and packet sequence information
- Contains no packet length indication (delivery layer handles this) -> SL-packet stream is not self-contained (unlike MPEG-2 PES)

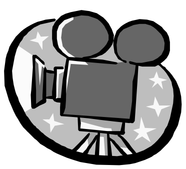




SL Time Stamp

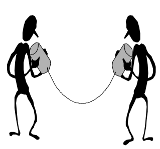
- Highly configurable
  - Explicit clock reference and time stamp
  - Implicit time stamp (rate based)
  - Without clock information, suitable for slide show presentation
  - May refer to another ES for clock reference if using the same time base
  - May contain only clock references for several other ES that then do not carry timestamps

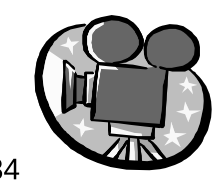




Time Stamp Types

- Decoding time stamp (DTS)
- Composition time stamp (CTS)
- Object time base (OTB)
  - Valid for an individual or a set of elementary streams
  - Each audio-visual object has its own OTB
  - Not universal
- Object clock reference time stamp (OCR)



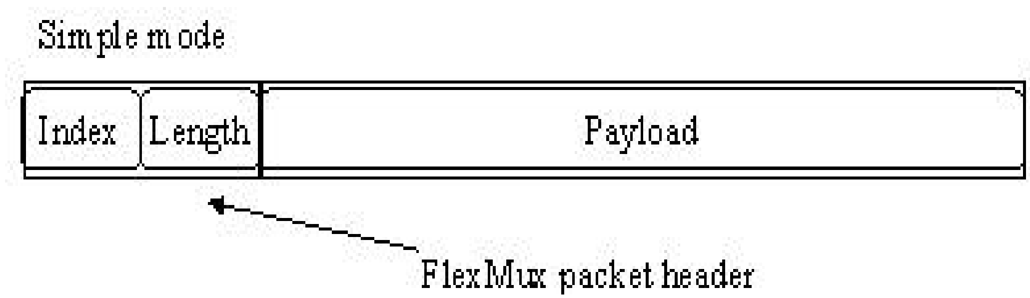


FlexMux Layer

- Used to multiplex "similar" streams to reduce the overall number of ***individual*** streams
- Suitable for low bitrate, low delay ES such as SD, OD, animation & speech
- Not a true transport level protocol, used when:
  - Underlying transport layer does not define multiplex, e.g., if H.223 is used, no need for FlexMux
  - Existing multiplex may not fit MPEG-4 requirements (low delay, short overhead)

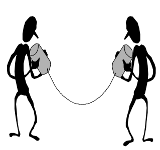
Simple Mode FlexMux

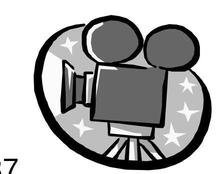
- 8 bit index for FlexMux Channel (FMC)
- 8 bit payload length



MPEG-4 File Format: .MP4

- Based on QuickTime
- Independent of delivery protocol
- ***Streamable*** file format (as opposed to streaming format)
  - One file format fits the need for both local storage and network streaming



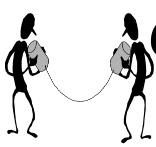


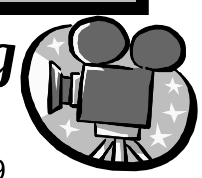
Part 2: Visual

- ISO 14496-2

MPEG-4 Video

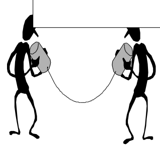
Baseline	**Extended**
Compression	**Content-based Coding**
Error Resilience	**Still Texture Coding**
Scalability	
Conventional coding	***Object coding***

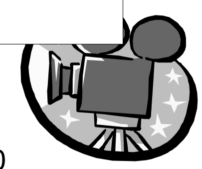




MPEG-4 Video Standard

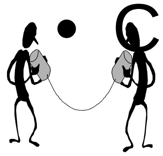
- MPEG-4 Video is Compatible to Baseline H.263
- And almost Compatible to MPEG-1
- And almost compatible to MPEG-2

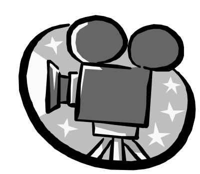




MPEG-4 Video Support

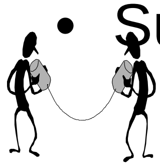
- Progressive & interlaced scan
- SQCIF/QCIF/CIF/4*CIF/CCIR 601, 8*8 ~ 2048*2048
- Y/C_b/C_r/Alpha
- Depth 4~12 bit, typical 8 bit
- Until now 4:2:0, future 4:2:2 and 4:4:4
- Continuous variable frame rate

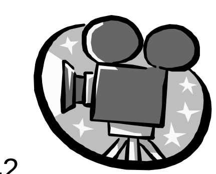




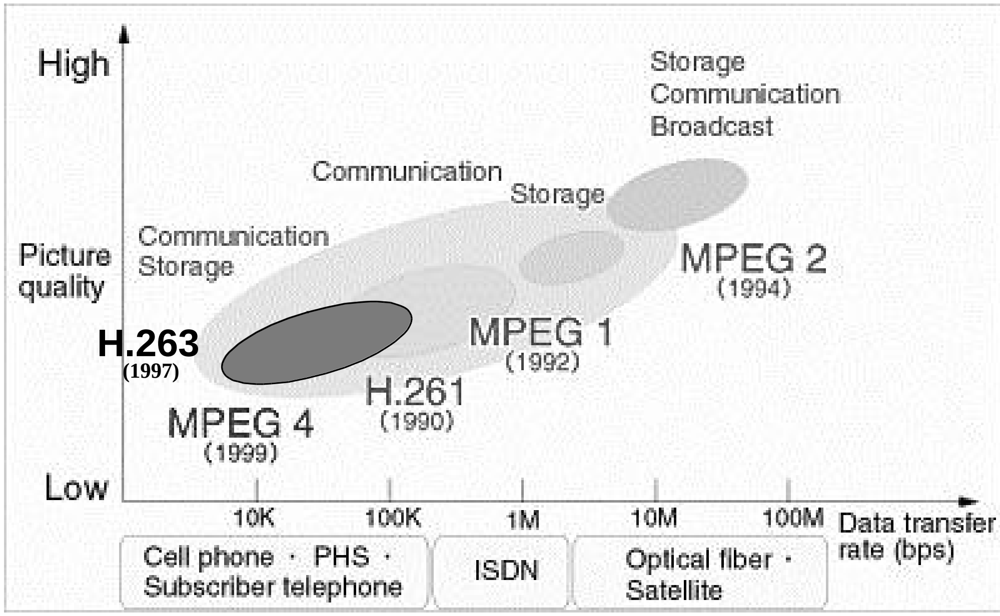
MPEG-4 Application Bit rates

- Low (<64kbps)
  - Low bandwidth wireless
- Intermediate (64-384kbps)
  - Low bandwidth Internet
- High (384k-4Mbps and in future up to 50Mbps)
  - High bandwidth Internet
  - Cable Modem, ADSL
- Supports CBR and VBR



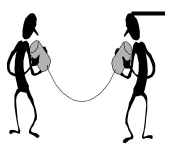


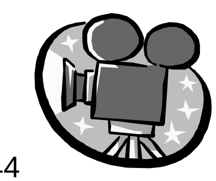
Coding Bit rates



MPEG-4 Video Principle

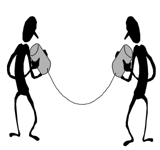
- Retained from predecessor
  - Block based natural video coding
  - Motion estimation and compensation
  - Variable length coding
- MPEG-4 Specific
  - Object based coding
  - Arbitrary shape coding
  - 2D & 3D mesh and texture coding

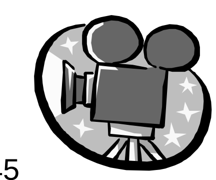




Profile & Level

- For encoder, profile & level give the upper bound of complexity of the bit stream
- For decoding hardware, profile & level give minimum performance constraints
- For decoding software, profile & level also imply resource availability to be monitored at run time

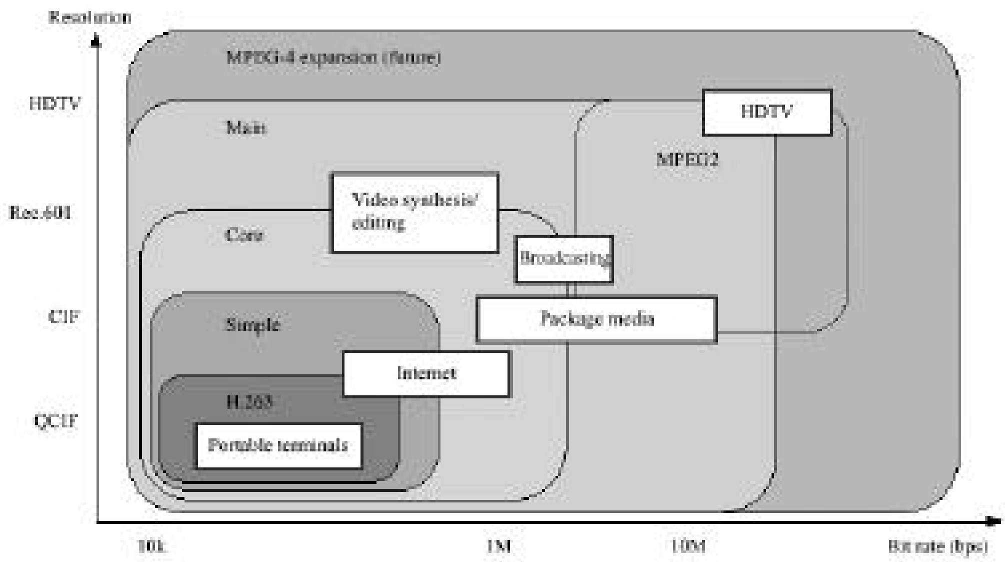




Levels for Natural Video Profile

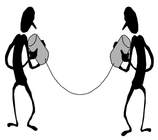
Profile and Level		Typical scene size	Bitrate (bit/sec)	Maximum number of objects	Total mblk memory (mblk units)
Simple Profile	L1	QCIF	64 k	4	198
	L2	CIF	128 k	4	792
	L3	CIF	384 k	4	792
Core Profile	L1	QCIF	384 k	4	594
	L2	CIF	2 M	16	2376
Main Profile	L2	CIF	2 M	16	2376
	L3	ITU-R 601	15 M	32	9720
	L4	1920x1088	38.4 M	32	48960

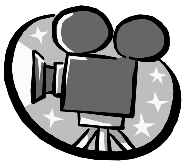
Application Mapping onto Profiles



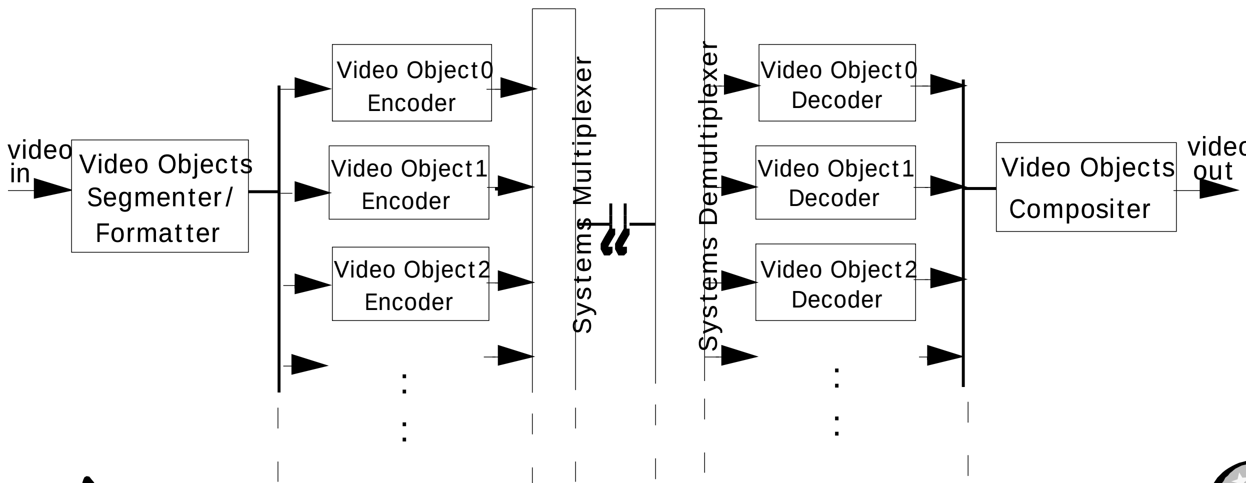
Video Coding Hierarchy

- Video Object Sequence (VS): Contains 2D/3D, natural/synthetic objects - the complete MPEG-4 visual scene
- Video Object (VO): one object in the scene, in its simplest form could be a rectangular frame, or it can be an arbitrarily shaped object corresponding to a sprite or a background scene
- Video Object Layer (VOL): provides support for scalable coding
  – Full MPEG-4 header VOL
  – Short header VOL (compatible with baseline H.263)
- Video Object Plane (VOP): Time sample of a video object(e.g. a rectangular frame, or a sprite). Contains motion, shape and texture information.





General Coding Diagram

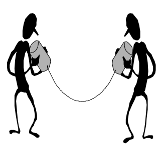


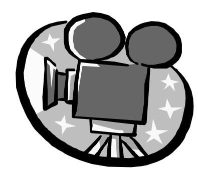
VOP Motion Coding

- Macro Block (16x16) and Block (8x8)
- I-VOP, P-VOP and B-VOP
- Half pixel precision Motion Estimation

VOP Texture Coding

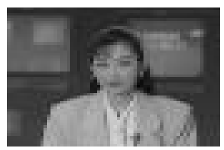
- DCT
- Quantization
- Zigzag Scan
- Variable Length Coding

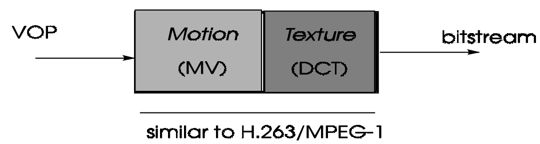




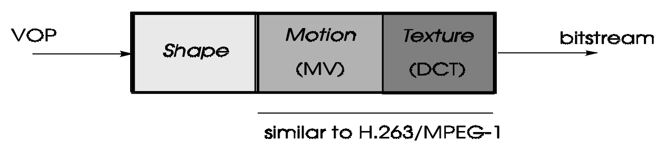
MPEG-4 Coder

MPEG-4 Core Coder

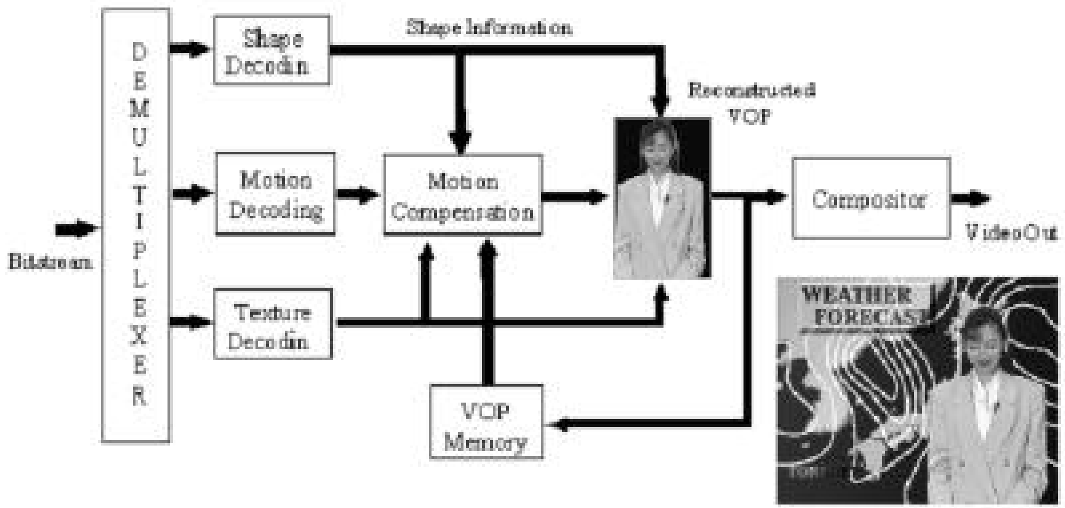




Extended MPEG-4 Core Coder



Decoding Diagram



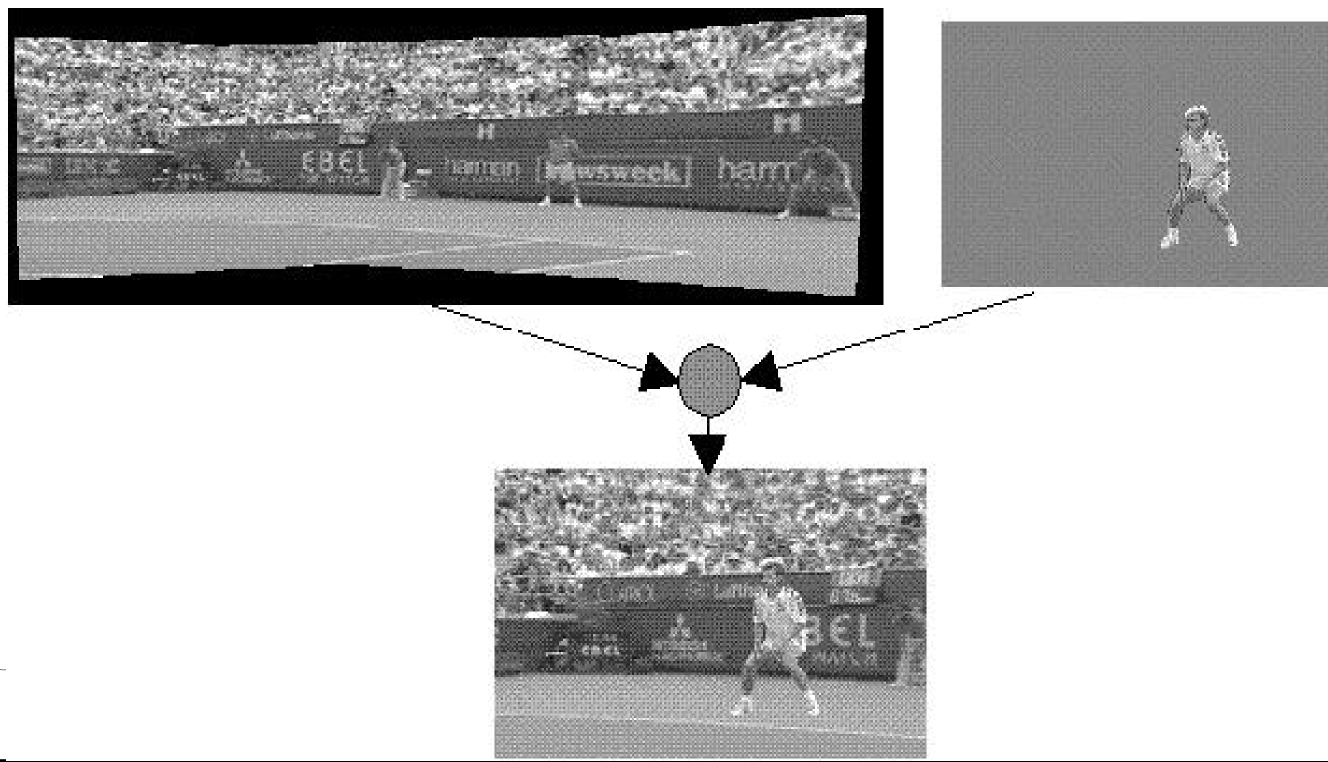
Content Based Functionalities

- Shape Coding
- Sprites
- Scalability (content based)
- Error resilience
- Scalable Texture Coding

Shape Coding

- The shape of an object can be represented in binary form or as a gray-level (or alpha channel - luminance) information.
- Alpha channel provides superior quality but is more expensive to encode.
- Both coding techniques use 16x16 blocks (MB)

Sprites



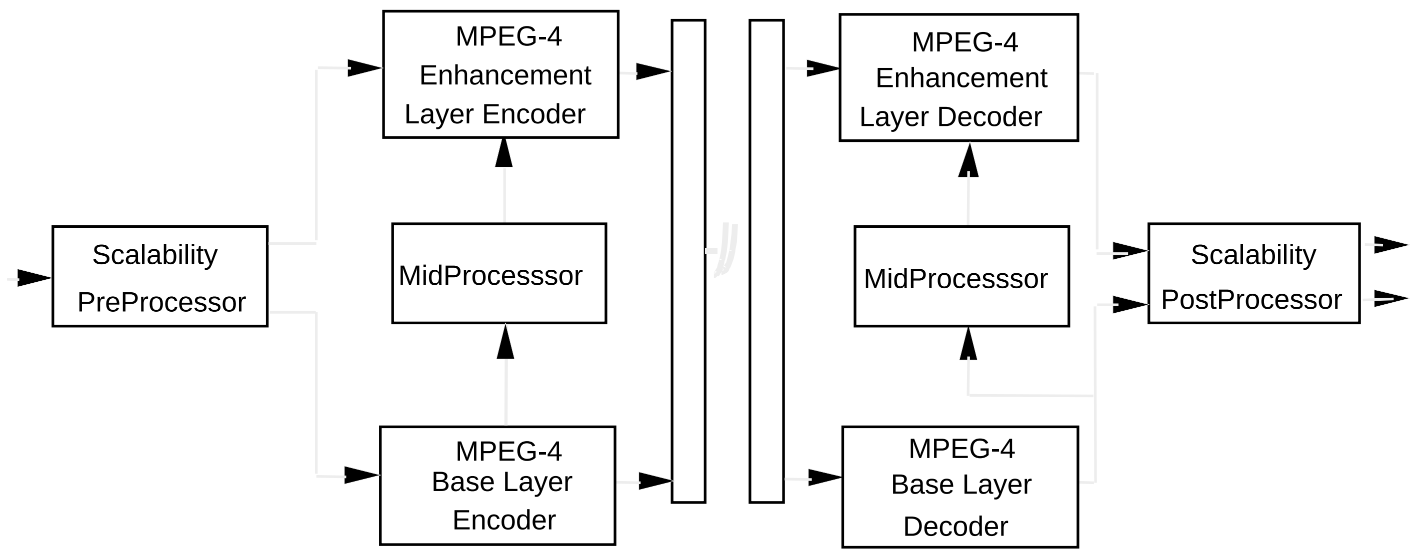
Scalable Coding

- Object based spatial scalability
  – Texture spatial & Shape spatial
- Object based temporal scalability
  – Partition VOP into layers
- Both provide resilience to transmission errors

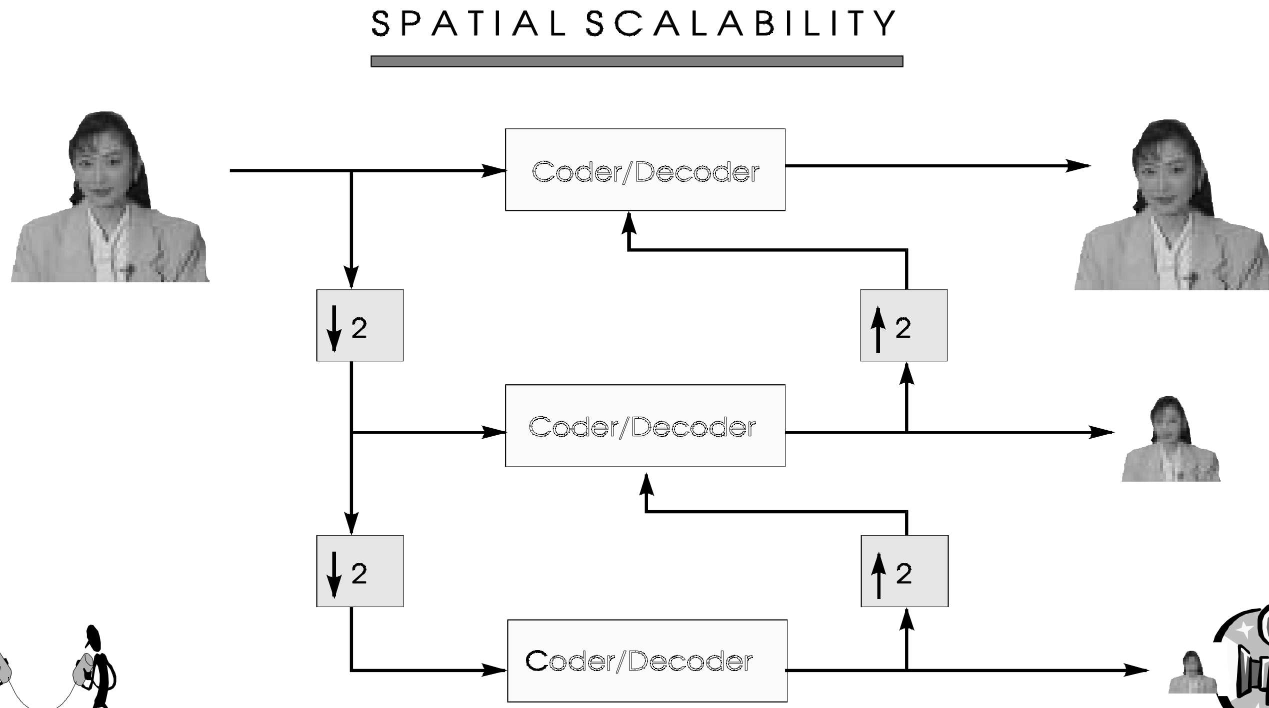
Scalable Coding

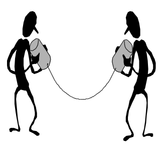
- Each layer is coded into an individual ES with unique ES_ID
- All layers belonging to the same object (i.e,. all ES that refer to the same object) are placed in the same Object Descriptor with its unique OD_ID

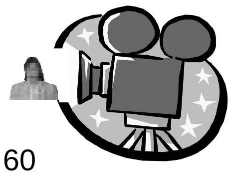
Scalable Coding Diagram



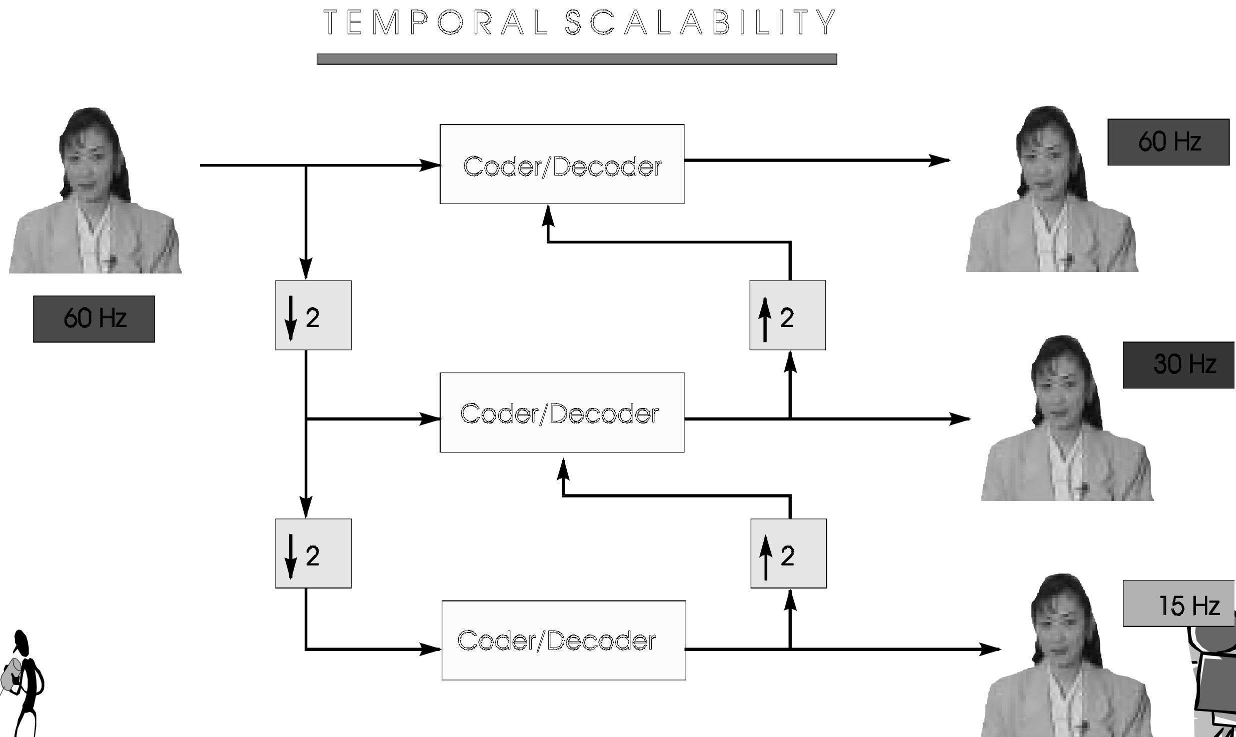
Spatial Scalability

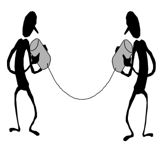


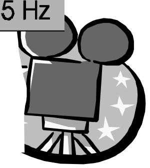




Temporal Scalability

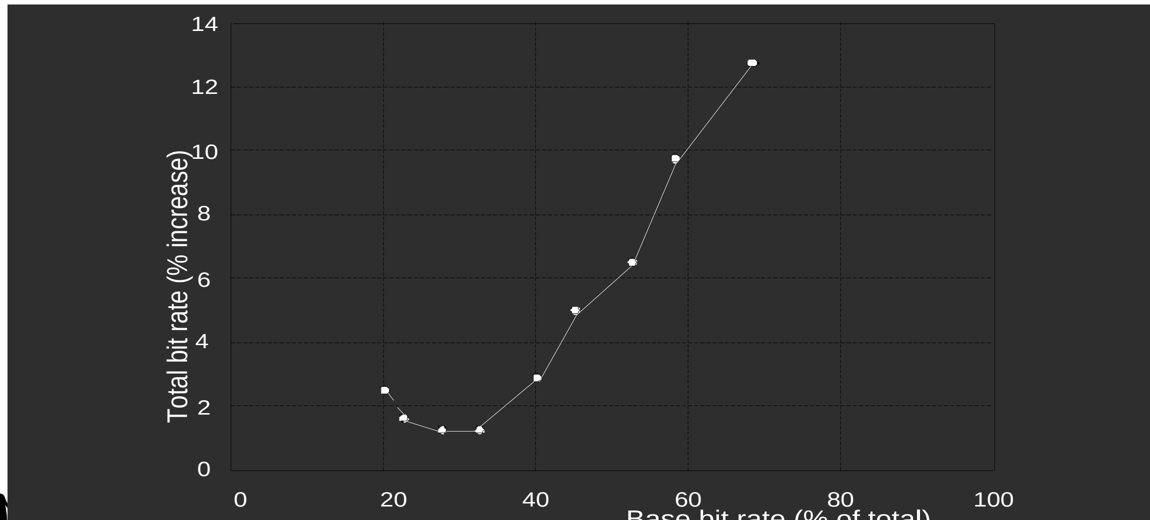






Tradeoff of Layered Coding

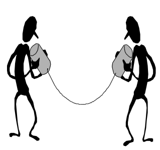
Layered coding schemes incur an increase in bit rate or decrease in video quality in comparison to a single-layer codec of equivalent quality.

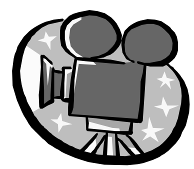


Percentage increase of total bit rate versus
percentage base layer of total bit rate

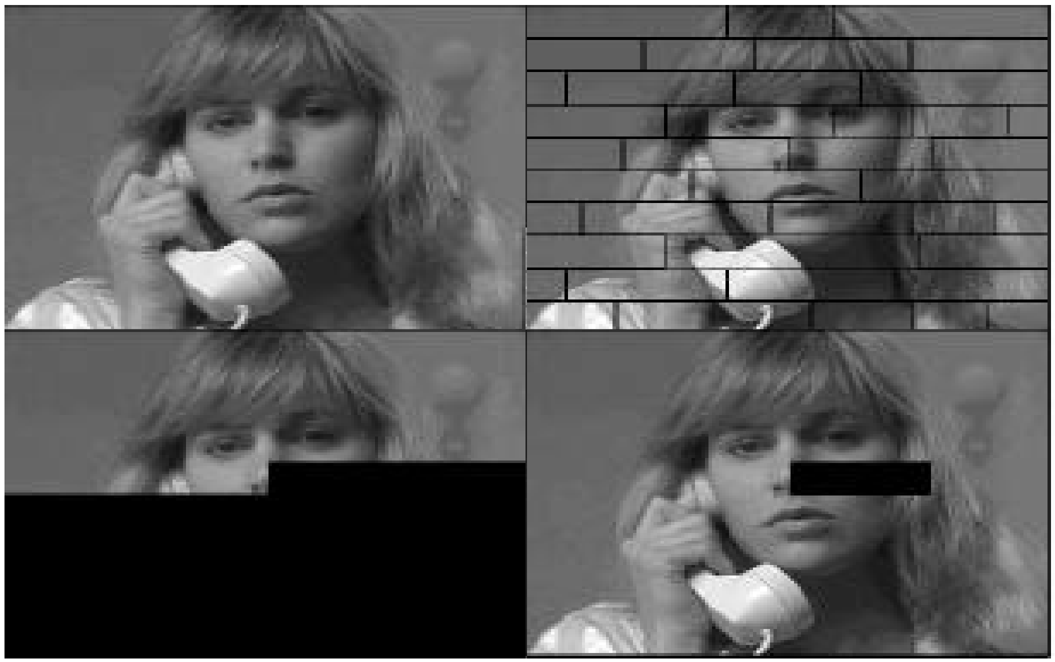
MPEG4 Error Resilience Tools

- Re-synchronization
- HEC
- Data partition
- RVLC
- AIR
- NEWPRED



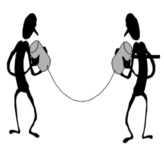


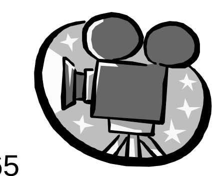
Resynchronization



Resynchronization Marker

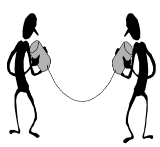
- Must define a Video Packet (VP)
  - Fixed #of bits instead of fixed #of MBs (recall slices of MPEG-2)
- Insert extra headers into the stream (in each VP) that will allow the decoder to re-sync in case of losses.
- Associated Overhead
  - Resync marker
  - MB number
  - Quant_scale

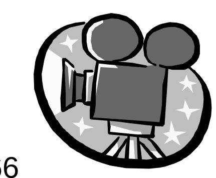




Header Extension Coding (HEC)

- Optional and of variable length
- Adds additional VOP header info into each VP
- Can correct VOP header when transmission errors occur

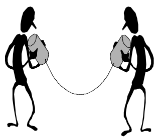


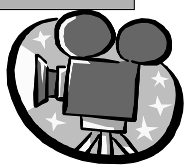


Data Partition

- Signaled at Video Object Layer (VOL)
- To localize error for error concealment at decoder
- Separate motion information from texture information

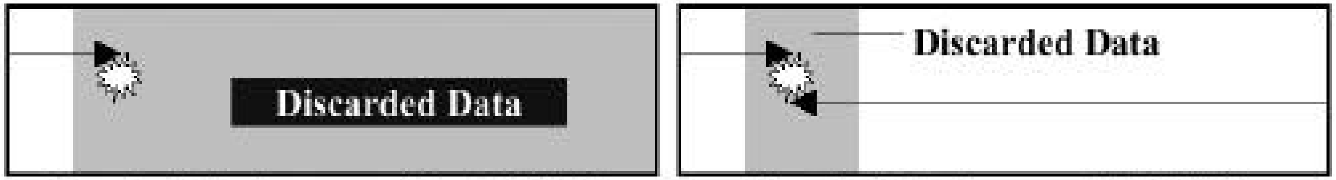
Resync Marker	macroblock_number	quant_scale	HEC	Motion &Header Information	Motion Marker	Texture Information	Resync Marker





Reversible VLC (RVLC)

- Using RVLC table instead of VLC to code texture
- RVLC allows both forward and backward texture decoding
- Signaled at VOL

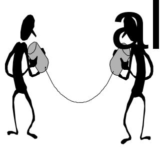


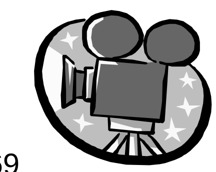
Result of single error without RVLCs

Result of single error with RVLCs

Problems with RVLC

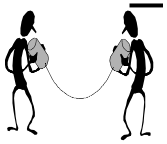
- Calculation Overload
  - Encoder side: different table used, not much of a problem
  - Decoder side: complicated algorithm used for discarding error bit
- Not impressive quality improvement compared to simple error concealment algorithm such as MB copying

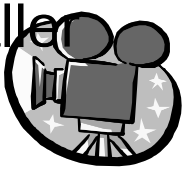




Performance Issues

- Depends on video sequence characteristics
  - Picture size
  - Frame rate
  - Target stream bit rate
- Also depends on network condition
  - For wired network, VP can be larger
  - For wireless network, VP should be smaller



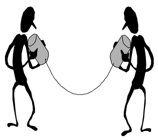


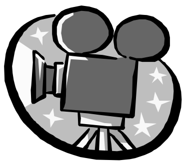
Adaptive Intra Refresh (AIR)

- Used for Rectangular VOP
- Extract motion area and encode it in intra mode more frequently, i.e, only use intra mode on motion portions of scene
- Transmission error can be recovered from sooner

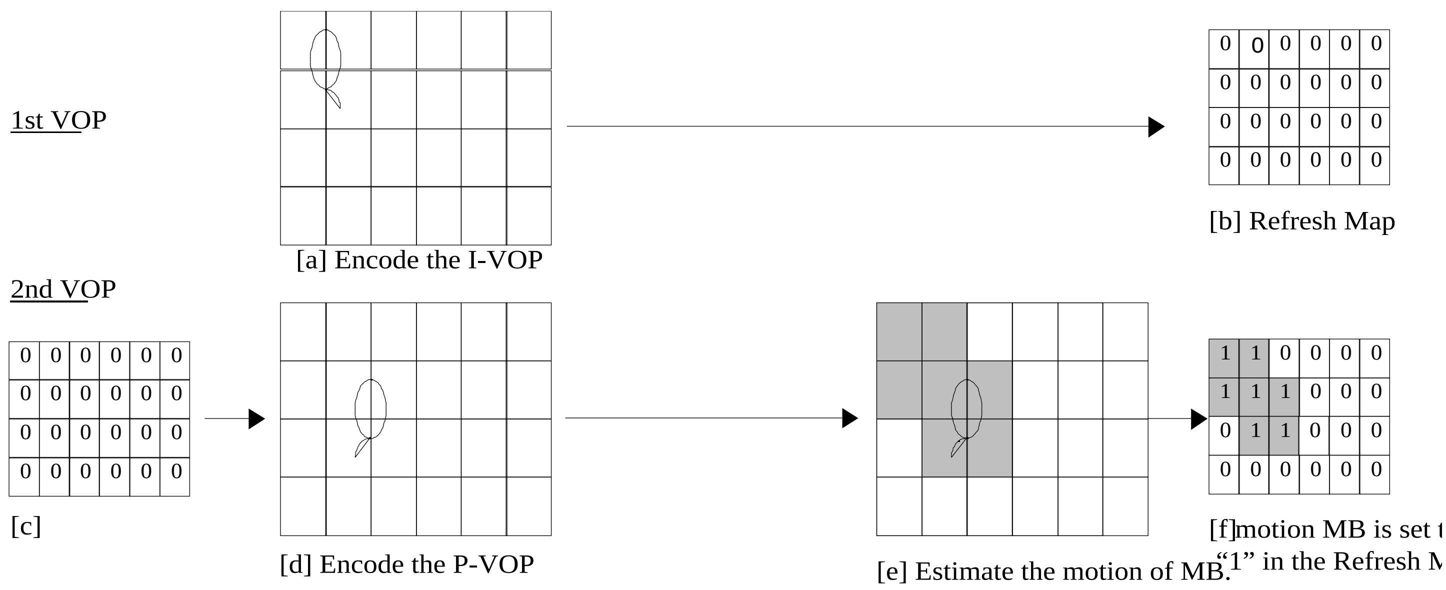
AIR (2)

- #of intra MB per VOP
  - Fixed and pre determined
  - Depends on bit rate, frame rate and so on
- Encoder estimates motion of each MB
  - Motion estimation is based on Sum of the absolute differential (SAD)
  - Compare SAD with threshold (SAD_th)
  - Results are recorded to the Refresh Map MB by MB

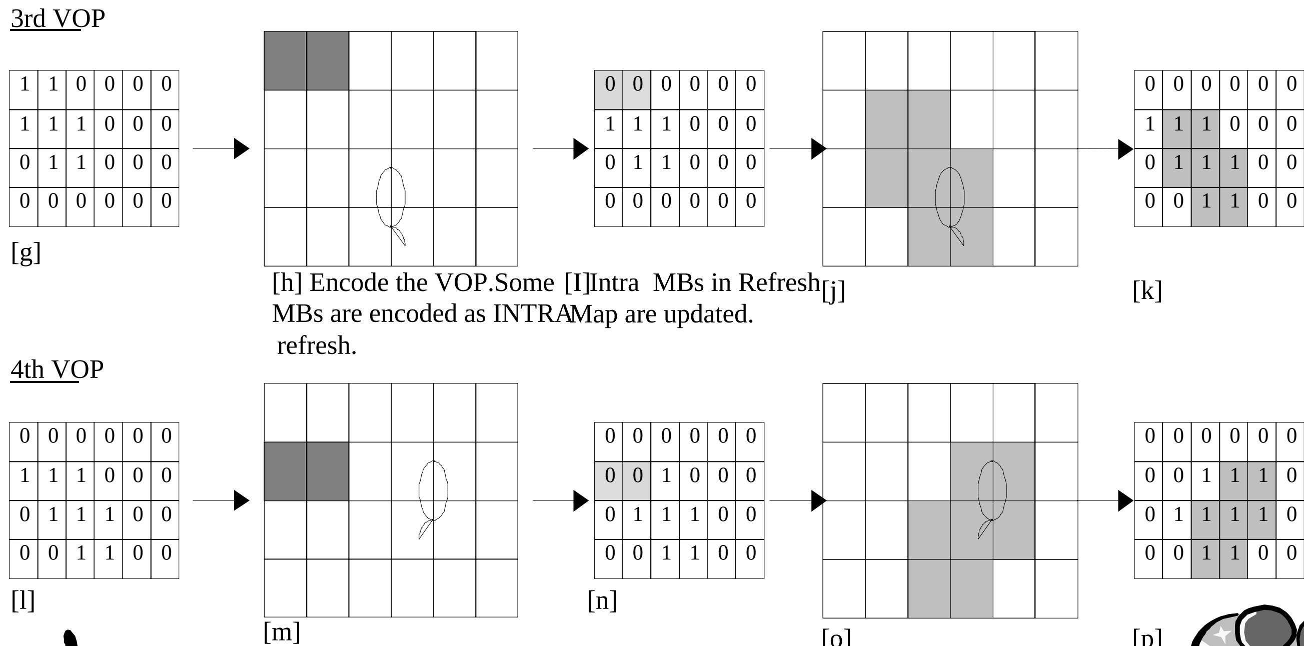




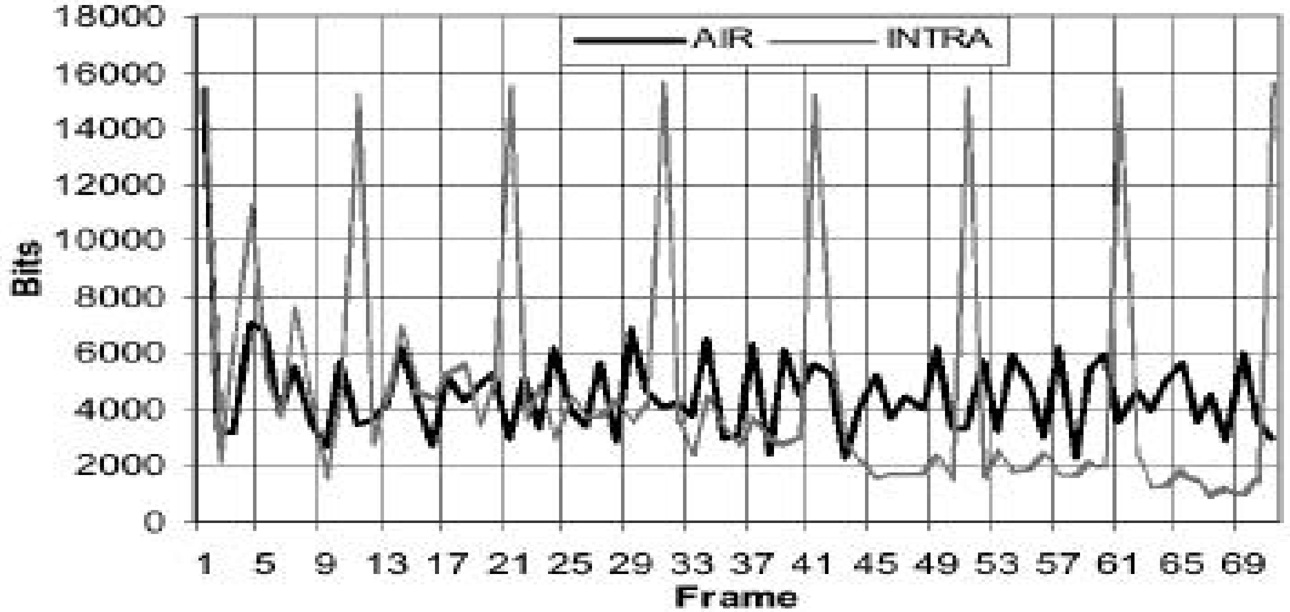
AIR Example 1/2



AIR Example 2/2

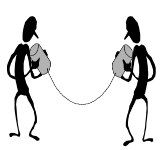


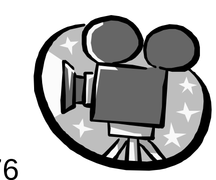
Throughput of AIR



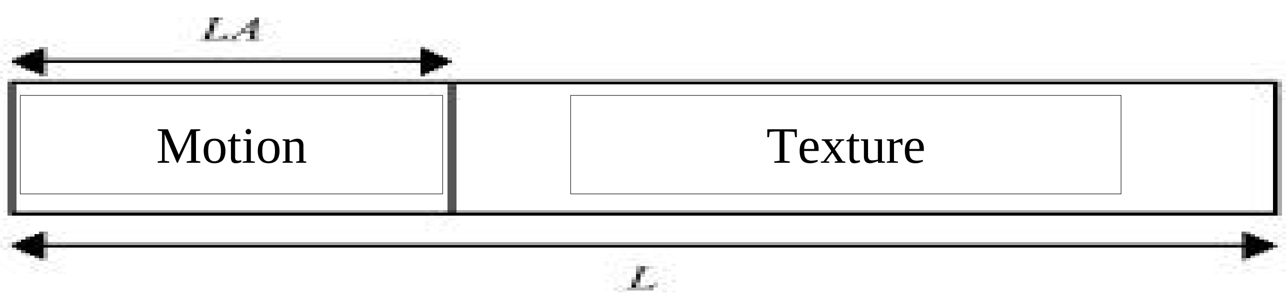
Problems related to AIR

- Calculation complexity at encoder:
  - Not much of a problem since SAD is calculated for motion estimation at encoder
- Choice of # of MB to refresh every VOP
- Choice of SAD_th
- Decoder side
  - Needs to do nothing for AIR





A proposal for choice of # of MB

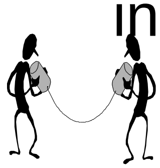


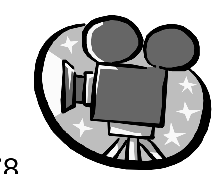
A	MBs/frm
<0.1	4
0.1-0.15	8
0.15-0.2	12

A	MBs/frm
0.2-0.3	16
0.3-0.35	8
>0.35	4

NEWPRED (1)

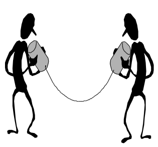
- One VP coded as one NP segment, signaled at VOL
- Decoder sends upstream message to encoder to indicate which NP segment was correctly decoded
- Encoder uses only correctly decoded part for prediction in inter-frame coding
- Prevents temporal error propagation w/o the insertion of Intra MB

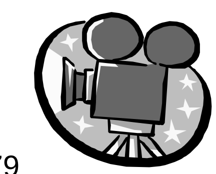




NEWPRED (2)

- Upstream channel needed
- Requires framing mechanism for synchronization messages (e.g. 14496-1)
- Strategy for selecting reference picture is not defined in standard yet
- Additional decoding buffer needed based on
  - Reference picture selecting algorithm at encoder
  - Transmission delay





Other Mechanisms

- Split data partitioned stream into two sub-streams
- Need Unequal Error Protection support
- Not defined in standard

Channel Coding

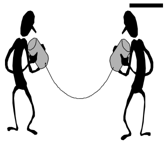
- Forward Error Control (FEC)
- Automatic repeat request (ARQ)

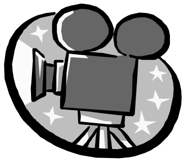
FEC

- Used for error correction at data link layer
- Add redundancy code to localize and correct errors
- Cannot handle burst error
- Coding schemes: BCH, Reed Solomon, turbo codes and Hamming codes

ARQ (1)

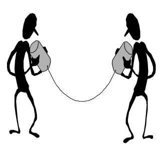
- Ask for retransmission when incorrect packet is received
  - Error detection: CRC code
  - Explicit (NACK) vs. Implicit (ACK)
- Protocol Examples
  - Stop and wait
  - Go back n
  - Selective repeat

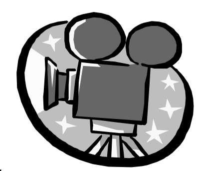




ARQ (2)

- Maybe suitable for one way real time video applications such as VoD
- Delay might be too long for interactive services such as video conferencing

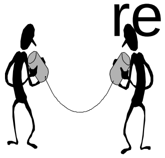


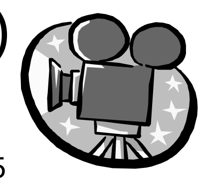


Hybrid FEC/ARQ Algorithms

h stands for redundancy factor for FEC

- ARQ (hI=hP=hB=0)
- Pure FEC (hI=hP=hB>0)
- Hybrid FEC/ARQ (hI=hP=hB>0)
- Hybrid FEC/ARQ with priority dependent redundancy (hI:hP:hB=1:0.25:0.05)
- Adaptive FEC/ARQ with priority dependent redundancy (hI,hP,hB changes adaptively)



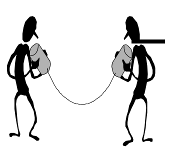


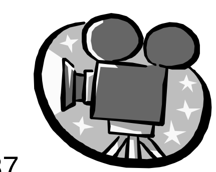
FEC/ARQ

- Needed?
  - MPEG4 Error Resilience tools can be deemed as Passive error control at the application layer (end to end)
  - Some errors do not generate invalid syntax
  - FEC/ARQ: Active error control at the data link layer
- FEC/ARQ can be more effective in some cases as it is closer to the source of error.

Open Issues

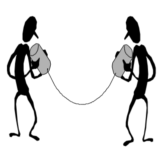
- Trade off between MPEG-4 ER and FEC/ARQ
  - Packet size dependent, bit rate dependent, etc.
- Perceptual quality?
  - E.g., 0.5*bw correctly received video vs. 0.8*bw received video with errors
  - Optimal point

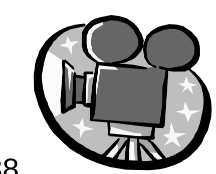




Error Concealment

- Many techniques exist: spatial, temporal, spectral, hybrid
- For low delay, low bitrate applications it has been shown that a simple MB replacement strategy works well.
  - Copy blocks from previous frame





Some Error Concealment Results:

(a). Unconcealed Image; (b). Frequency concealment (FC); (c). The 16th frame after initial FC on the first image; (d). Spatial Concealment

Left: Unconcealed Image. Middle: Concealed by simple motion vector estimation, simply averaging the top and bottom mvs. Right: Concealed with more motion vectors. All the adjacent mvs are used directly or indirectly.

Part 6: DMIF

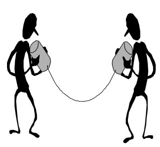
- Delivery Multimedia Integration Framework (DMIF) (ISO 14496-6)

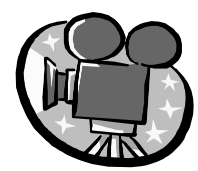
Why DMIF?

- MPEG-1 & MPEG-2 System Specification
  – Focus on specific storage/transport usage
  – Synchronization and transportation combined in one part
  – inappropriate or inefficient when delivery medium changes

DMIF Goals

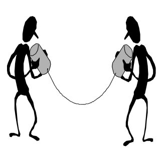
- Define a session level protocol for stream management
- Hide delivery technology details from DMIF user
- Ensure interoperability between end systems
  - Find universal solutions for various scenarios (local retrieval, remote interaction, broadcast or multicast)

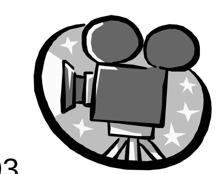




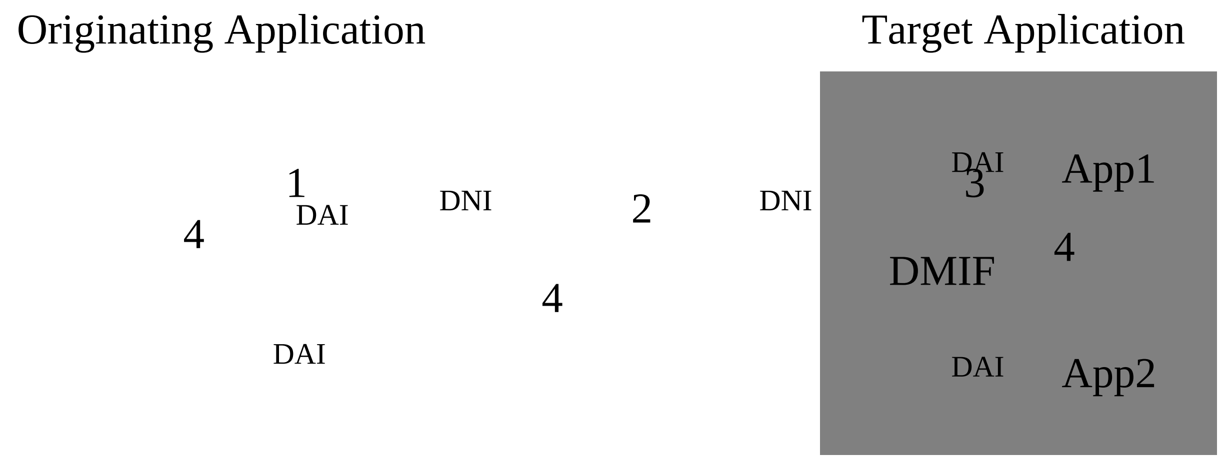
DMIF Principle

- Session Protocol for real-time stream management
- Similar to FTP
  - FTP returns Data where DMIF returns pointer to where to get data
- Application accesses data through the DMIF Application Interface (DAI)
  - The DAI is QoS aware and accesses multimedia content in a uniform manner
- The DMIF Network Interface (DNI) indicates the actions that DMIF must take with respect to the network
  - The DNI also enables the exchange of parameters that DMIF peers need to communicate





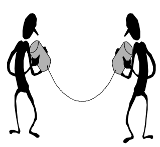
DMIF Model

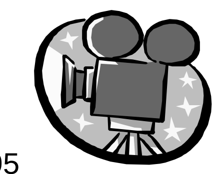


Control Plane Connectivity
User Plane Connectivity

Client-Server Modeling (1)

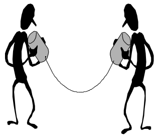
- Server-side Supports:
  - Encoding
  - Buffering of Sync. Layer (SL)-Packetized Streams
  - Scheduling of Objects

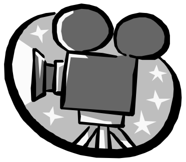




Client-Server Modeling (2)

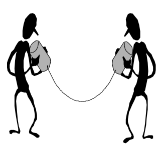
- Client-side Supports:
  - Creation of buffers and decoders
  - Reception and binding of information with the appropriate decoder
  - Composition and rendering
  - Handling user events

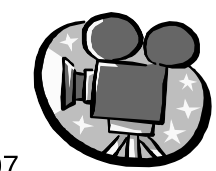




Client-Server Modeling (3)

- Both support:
  - Management of control and data flows
  - Communication between each other





MPEG-4 Layers

media aware
delivery unaware
ISO/IEC 14496-2 Visual
ISO/IEC 14496-3 Audio

Compression Layer

Elementary Stream Interface (ESI)

media unaware
delivery unaware
ISO/IEC 14496-1 Systems

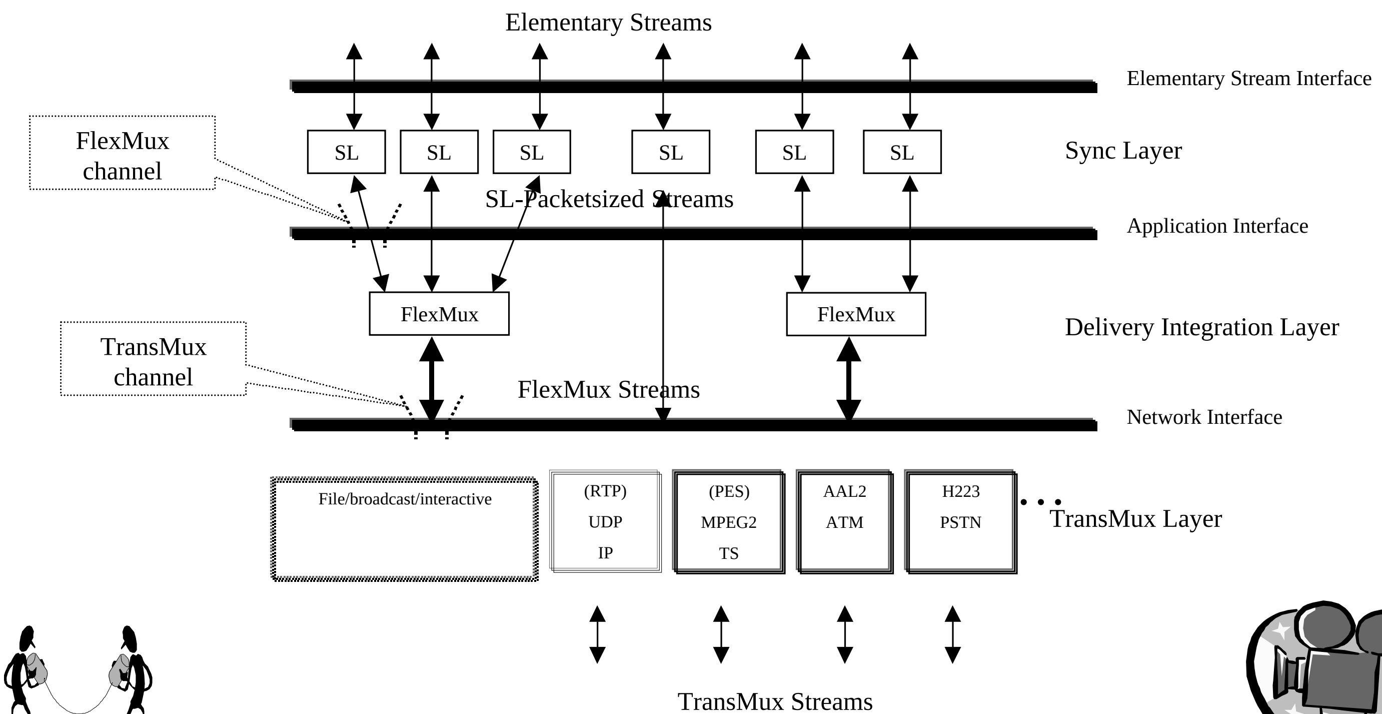
Sync Layer

DMIF Application Interface (DAI)

media unaware
delivery aware
ISO/IEC 14496-6 DMIF

Delivery Layer

MPEG-4 User Plane



MPEG-4 User Plane

- FlexMux layer interleaves one or more elementary streams
- TransMux layer interleaves one or more flexmux streams
- FlexMux can be bypassed
  - One ES maps to one TransMux stream