

# Lecture 6: Internet Streaming Media

Reji Mathew & Jian Zhang  
NICTA & CSE UNSW  
COMP9519 Multimedia Systems  
S2 2009



## Introduction

### The story so far ....

- Video Coding – MPEG-4
- Audio Coding – u-Law, AMR
- Packetization and Transport of media – RTP
- Transport Control / QoS Feedback - RTCP

### Lecture Outline

- Literature Review
  - Client Architecture : An Example
- Next
  - Further Protocols Enabling Streaming
  - SDP - Description
  - RTSP – Signaling
  - SIP



COMP9519 Multimedia Systems – Lecture 7 – Slide 2 – R. Mathew & J. Zhang



## Client Architecture : An Example

### • Reference

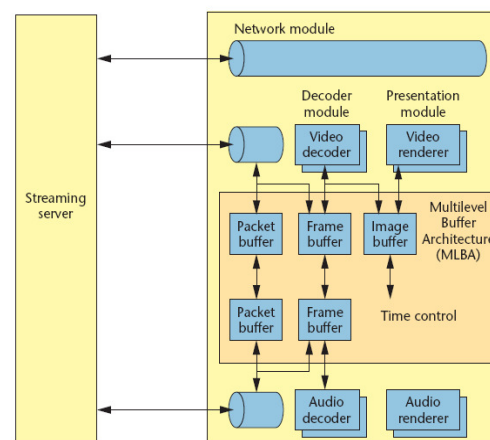
- “Client architecture for MPEG-4 streaming”, Haifeng Xu Diamand, J. Luthra, A. IEEE Multimedia, April-June 2004, Vol 11, Issue: 2, pages 16-23
- Review of
  - Streaming Video, RTP, MPEG-4
- Client Architecture for streaming MPEG-4 video
  - Multilevel buffer system



COMP9519 Multimedia Systems – Lecture 7 – Slide 3 – R. Mathew & J. Zhang



## Client Architecture : An Example



Multiple buffer architecture to allow efficient control of media processing and presentation.

Three Modules

- Network Module : Packet buffer
- Codec Module : Frame Buffer
- Presentation Module : Image Buffer

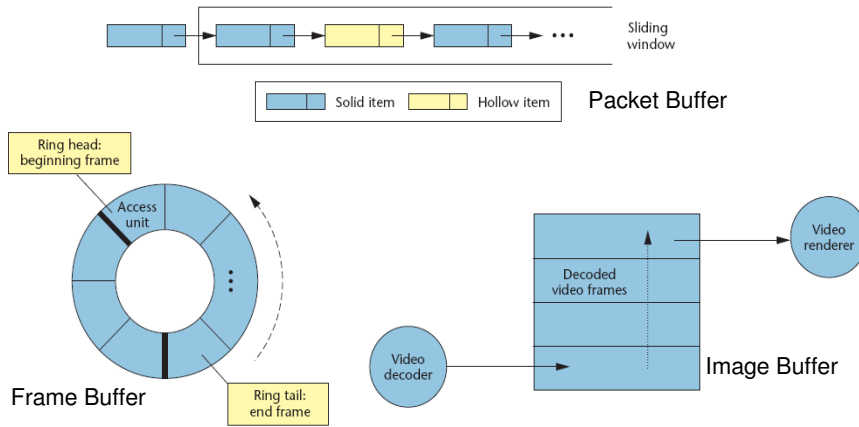
“Client architecture for MPEG-4 streaming”, Haifeng Xu Diamand, J. Luthra, A. IEEE Multimedia, April-June 2004, Vol 11, Issue: 2, pages 16-23



COMP9519 Multimedia Systems – Lecture 7 – Slide 4 – R. Mathew & J. Zhang



## Client Architecture : Multiple Buffers



"Client architecture for MPEG-4 streaming", Haifeng Xu, Diamand, J., Luthra, A.  
IEEE Multimedia, April-June 2004, Vol 11, Issue: 2, pages 16-23



COMP9519 Multimedia Systems – Lecture 7 – Slide 5 – R. Mathew & J. Zhang



## Client Architecture : Multiple Buffers



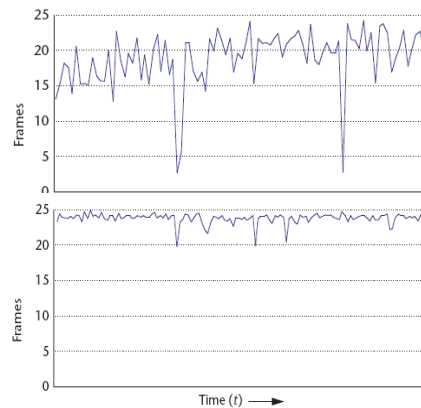
- Packet Buffer
  - Reorder packets
  - Identify missing packets (allow for easy insertion)
  - Ignores delayed packets (outside moving window)
- Frame Buffer
  - Stores video packets (smoothing out network jitter)
  - Allows dropping of frame (when lacking CPU resources)
  - Identify frames to be dropped (B, P)
- Image Buffer
  - Stores decoded frames for rendering
  - Smoothing decoders speed



COMP9519 Multimedia Systems – Lecture 7 – Slide 6 – R. Mathew & J. Zhang



## Client Architecture : Rendering



Rendered frames rates with and without the image buffer

"Client architecture for MPEG-4 streaming", Haifeng Xu, Diamand, J., Luthra, A.  
IEEE Multimedia, April-June 2004, Vol 11, Issue: 2, pages 16-23



COMP9519 Multimedia Systems – Lecture 7 – Slide 7 – R. Mathew & J. Zhang



## Client Architecture : Conclusion



- The MPEG-4 MLBA subsystem facilitates three player-related activities:
  - precise A/V synchronization,
  - client-based QoS management, and
  - improved rendering performance through an image buffer.

"Client architecture for MPEG-4 streaming", Haifeng Xu, Diamand, J., Luthra, A.  
IEEE Multimedia, April-June 2004, Vol 11, Issue: 2, pages 16-23



COMP9519 Multimedia Systems – Lecture 7 – Slide 8 – R. Mathew & J. Zhang



## Lecture Outline

- Session Description Protocol
  - How to describe a multimedia session ?
  - SDP example
- RTSP – Internet VCR controls
  - Stop, Pause, Play, Fast Forward over the internet
  - RTSP Methods
  - Signal Timing Diagram
  - Protocol stack for a total streaming system
- SIP – Quick Overview
- MPEG-4 File Format
  - What's so good about MP4?
  - File Format and Structure
  - Hinting

## Introduction

- Example : An existing live multicast session
  - Video and Audio streams
  - Transport - RTP/UDP/IP, Control – RTCP/UDP/IP
- A new client wanting to join the multicast session
  - Needs to know
    - multicast IP address and port
    - Media streams in a session (e.g. video only or audio + video)
    - Payload format (e.g. MPEG-4 video, AMR audio)
    - Initialization data for video and audio decoders
    - Transport protocol used
    - Other information ....
- Need a way to describe a multimedia session
  - To enable new clients to easily join the session

## SDP

- Session Description Protocol (SDP)
  - IETF – RFC2327
    - [www.ietf.org/rfc/rfc2327.txt](http://www.ietf.org/rfc/rfc2327.txt)
  - For describing multimedia sessions
    - To communicate the existence of a session
    - To convey sufficient information to join a session
  - Simple text format
  - Defined to be general purpose
    - Can be used for a wide range of network environments
    - And applications

## SDP

- SDP includes
  - Session name and purpose
  - Time the session is active
  - The media comprising the session
  - Information to receive media (addresses, ports, formats)
  - Information about bandwidth to be used
  - Contact information of a person responsible for the session
- SDP is used by other signaling / initiation protocols
  - SIP : Session Initiation Protocol
  - RTSP : Real-time Streaming Protocol

# SDP

session level description

- v= (protocol version)
- o= (owner and session identifier)
- s= (session name)
- i=\* (session information)
- u=\* (URI of description)
- e=\* (email address)
- p=\* (phone number)
- c=\* (connection information)
- b=\* (bandwidth information)
- z=\* (time zone adjustments)
- k=\* (encryption key)
- a=\* (zero or more session attribute)
- t= (time the session is active)
- r=\* (zero or more repeat times)

SDP session description consists of a number of lines of text of the form

<type>=<value>

<type> is always exactly one character and is case-significant.

media level description

- m= (media & transport address)
- i=\* (media title)
- c=\* (connection information)
- b=\* (bandwidth information)
- k=\* (encryption key)
- a=\* (zero or more media attribute)



# SDP

```

v=0
o=NICTA 2890844526 2890842807 IN IP4 129.94.135.201
s=
i=
u=
e=
c=IN IP4 2
t=0 0
a=recv
m=vide
a=rtpm
a=fm
a=ori
    
```

Annotations:

- Version number**: v=0
- Originator information**: o=NICTA 2890844526 2890842807 IN IP4 129.94.135.201
- Session Information**: s=
- Session name**: i=
- URI for more information about the session**: u=
- Contact person e-mail**: e=
- Connection Details**: c=IN IP4 2
- Session level attribute**: t=0 0
- Media attribute (rtp map)**: a=recv
- Media attribute (format specific parameters)**: m=vide
- Media attribute (orientation)**: a=rtpm, a=fm, a=ori

Additional notes:

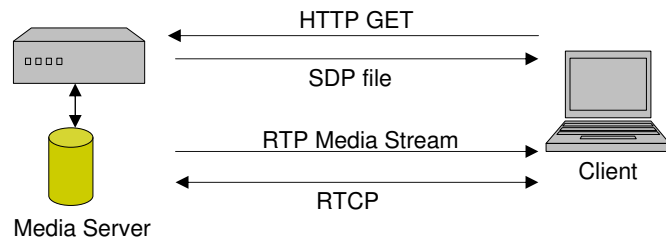
- increased when a modification is made to the session data. Recommended that an NTP timestamp is used
- was created
- only used in some applications, example – landscape or portrait



# System Overview

- Example – Using SDP to join a multicast session
  - Request SDP file via HTTP
  - Retrieve information from downloaded SDP file
  - Receive RTP streams on SDP specified address & port
  - Decode and display specified media

<http://www.nicta.com/lecture.sdp>



# System Overview

- Example – Traffic Monitoring
  - Continuous multicast streaming of video
  - A client can receive the stream
    - by downloading SDP via http (web browser)
    - Provide SDP file to QuickTime player
    - Player initializes and waits for stream data
    - Easily support multiple client

• [Demo Link](#)



## Lecture Outline

- Session Description Protocol
  - How to describe a multimedia session ?
  - SDP example
- RTSP – Internet “Remote” VCR controls
  - Stop, Pause, Play, Fast Forward over the internet
  - RTSP Methods
  - Signal Timing Diagram
  - Protocol stack for a total streaming system
- SIP – Quick Overview
- MPEG-4 File Format
  - What’s so good about MP4?
  - File Format and Structure
  - Hinting

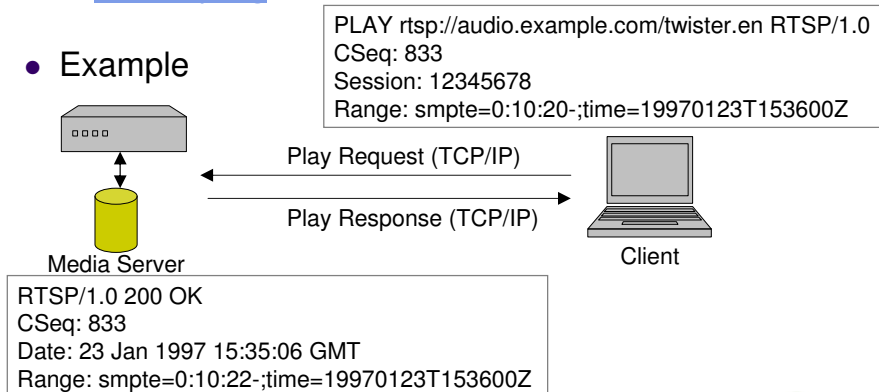
## RTSP

- Now we can describe and transport media streams
- But how to control these streams
  - Start, Stop, Pause, Fast Forward, Rewind
  - “internet VCR”
- Solution – RTSP
  - Real Time Streaming Protocol
  - Establishes and controls one or more continuous media streams - such as audio and video
- Similar in syntax and operation to HTTP/1.1
  - Client –Server protocol
  - Text based
- IETF RFC 2326

## RTSP

- More Info
  - [www.ietf.org/rfc/rfc2326.txt](http://www.ietf.org/rfc/rfc2326.txt)
  - [www.rtsp.org](http://www.rtsp.org)

### Example



## RTSP

### Protocol Properties

- Media Transport Independent
  - RTSP does not depend on the transport mechanism used to carry the continuous media
  - Media transport can be via RTP or some other mechanism
- Transport Layer Independent:
  - RTSP may use either an unreliable datagram protocol (UDP) or a reliable stream protocol such as TCP
- Control of recording devices:
  - Can control both recording only and playback only devices,
  - As well as devices that can alternate between the two modes

# RTSP

- Protocol Properties

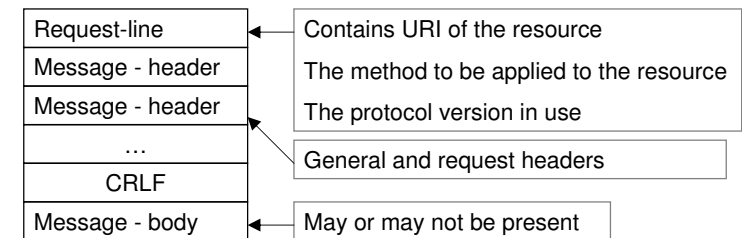
- Extendable:
  - New methods and parameters can be easily added to RTSP.
- Easy to parse
- Multi-server capable:
  - Each media stream within a presentation can reside on a different server.
- HTTP-friendly:
  - RTSP reuses HTTP concepts, so that the existing infrastructure can be reused.



# RTSP

- Protocol Operation

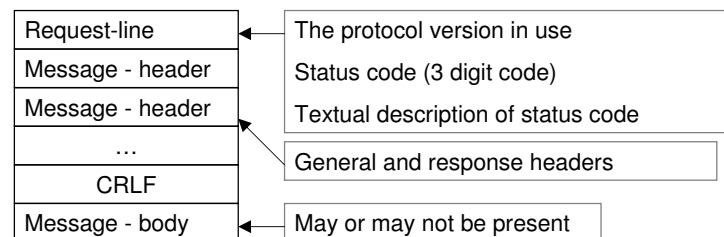
- Text based messages between client and server
- Messages can be :
  - Requests
  - Responses
- Request Messages



# RTSP

- Protocol Operation

- Response Messages

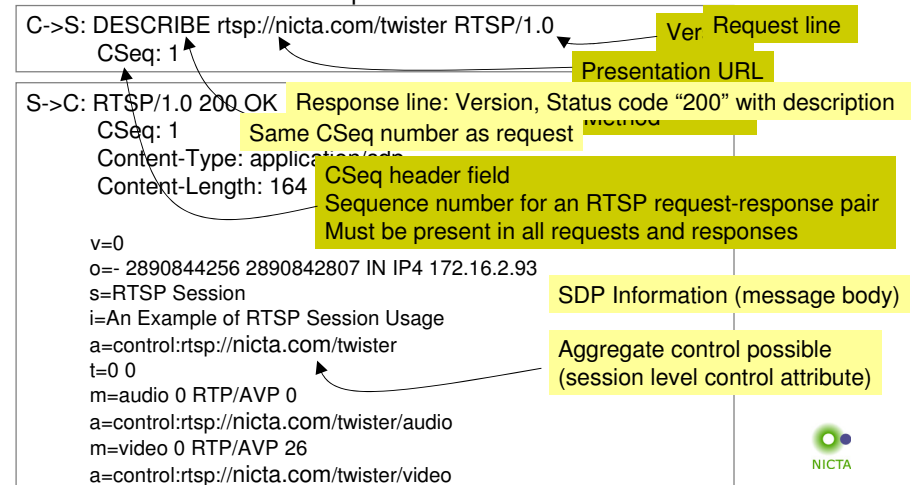


- Next : examples of Request and Response messages
  - Corresponding to five important request methods
  - Methods : Describe, Setup, Play, Pause, Teardown
  - Unicast example



# RTSP : Describe Method

- Retrieves the description of a presentation, identified by the request URL, from a server. The DESCRIBE reply-response pair constitutes the media initialization phase of RTSP.



## RTSP : Setup Method



- Specifies the transport mechanism to be used for the streamed media

```
C->S: SETUP rtsp://nicta.com/twister/audio RTSP/1.0
CSeq: 2
Transport: RTP/AVP;unicast;client_port=8000-8001

S->C: RTSP/1.0 200 OK
CSeq: 2
Transport: RTP/AVP;unicast;client_port=8000-8001; server_port=9000-9001
Session: 12345678

C->S: SETUP rtsp://nicta.com/twister/video RTSP/1.0
CSeq: 3
Transport: RTP/AVP;unicast;client_port=8002-8003
Session: 12345678

S->C: RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP;unicast;client_port=8002-8003; server_port=9004-9005
Session: 12345678
```

unicast RTP/RTCP port pair on which the client has chosen to receive media data and control information

RTSP session ID

unicast RTP/RTCP port pair on which the server has chosen to send media data and control information

Setup operates on each stream separately

Same session ID used



## RTSP : Play Method



- Tells the server to start sending data via the mechanism specified in SETUP request/response messages

```
C->S: PLAY rtsp://nicta.com/twister RTSP/1.0
CSeq: 4
Range: npt=0-
Session: 12345678

S->C: RTSP/1.0 200 OK
CSeq: 4
Session: 12345678
RTP-Info: url=rtsp://foo/twister/video;
seq=9810092;rtptime=3450012
```

Aggregate control of all media streams in the presentation. The PLAY request starts streaming both audio and video streams.

Range header used to specify time ranges (start / stop) for playback

Normal play time (NPT) indicates the stream absolute position relative to the beginning of the presentation.



## RTSP : Pause Method



- Causes the stream delivery to be interrupted (halted) temporarily.

```
C->S: PAUSE rtsp://nicta.com/twister RTSP/1.0
CSeq: 5
Session: 12345678
```

Aggregate control of all media streams in the presentation. The PAUSE request halts the streaming of both audio and video streams.

```
S->C: RTSP/1.0 200 OK
CSeq: 5
Session: 12345678
```



## RTSP : Teardown Method



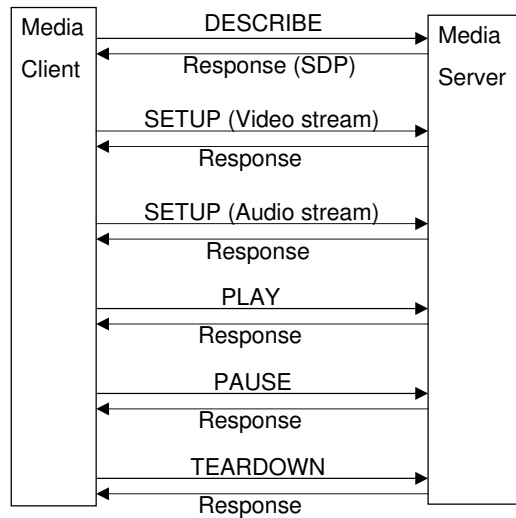
- Stops the stream delivery for the given URI, freeing the resources associated with it

```
C->S: TEARDOWN rtsp://nicta.com/twister RTSP/1.0
CSeq: 892
Session: 12345678
```

```
S->C: RTSP/1.0 200 OK
CSeq: 892
```



## RTSP : Signal Timing Diagram

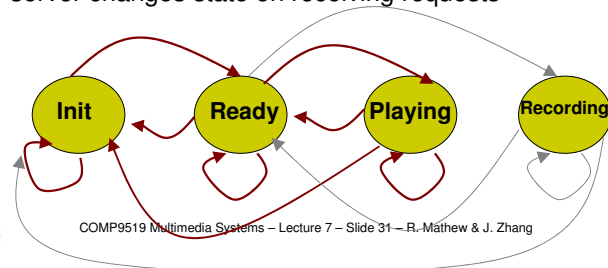


## Summary of RTSP Methods

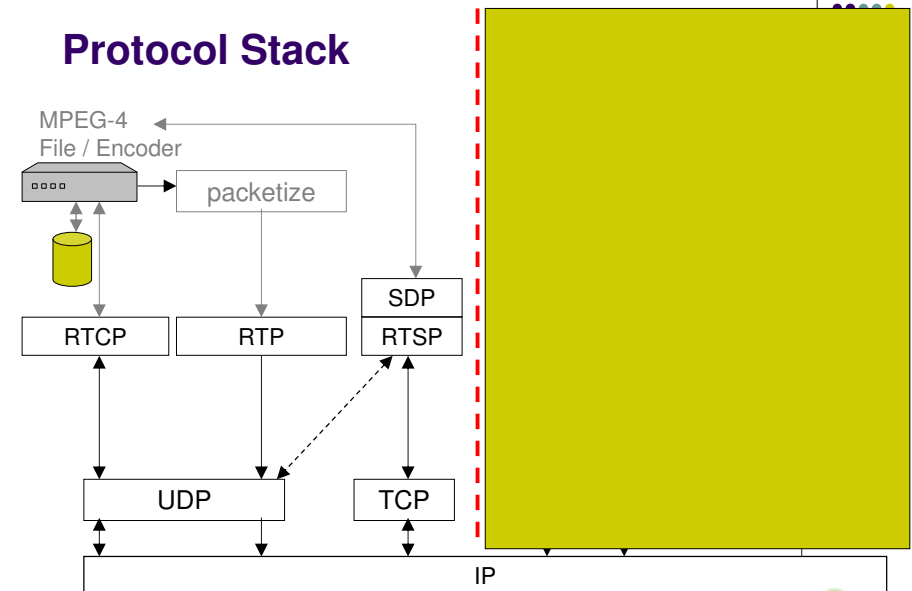
- **OPTIONS :** get available methods
- **SETUP :** establish transport
- **ANNOUNCE :** change description of media object
- **DESCRIBE :** get description of media
- **PLAY :** start playback, reposition
- **RECORD :** start recording
- **REDIRECT :** redirect client to new server
- **PAUSE :** pause delivery, while keeping state
- **SET PARAMETER :** device or encoding control
- **TEARDOWN :** remove state / close

## RTSP : Comparison with HTTP

- Similar to HTTP, but some important differences for RTSP are :
  - Data delivery takes place out-of-band in a different protocol
    - e.g. RTP
  - Both the media client and media server can issue requests
    - Client > Server : SETUP, PLAY, ....
    - Server > Client : ANNOUNCE, REDIRECT, ....
  - Requests are not stateless : client & server state machines
    - client changes state on receipt of replies to requests
    - server changes state on receiving requests



## Protocol Stack



## Lecture Outline

- Session Description Protocol
  - How to describe a multimedia session ?
  - SDP example
- RTSP – Internet VCR controls
  - Stop, Pause, Play, Fast Forward over the internet
  - RTSP Methods
  - Signal Timing Diagram
  - Protocol stack for a total streaming system
- SIP – Quick Overview
- MPEG-4 File Format
  - What's so good about MP4?
  - File Format and Structure
  - Hinting

## SIP

- Session Initiation Protocol (SIP)
  - Another session control protocol – diff SDP and RTSP??
  - IETF RFC 3261 [www.ietf.org/rfc/rfc3261.txt](http://www.ietf.org/rfc/rfc3261.txt)
  - Protocol that can establish, modify and terminate multimedia sessions.
  - Applications – IP telephony, multimedia conferences
  - SIP, like RTSP,
    - Uses text-based request/response transaction model
      - Requests contain Methods and Header fields
      - Responses include 3 digit status codes (eg “200 OK”)
    - Is Transport layer independent
    - Uses other protocols for media delivery (eg RTP).

## SIP

- SIP Methods
  - INVITE, ACK, CANCEL : for setting up sessions
  - BYE : for terminating sessions
  - REGISTER : for registering contact information
  - OPTIONS : for querying servers about capabilities
- INVITE
  - User agent client initiates a session with INVITE
  - Invitation can be accepted by a user agent server
  - SIP invitations convey session descriptions that allow participants to agree on a set of compatible media types
  - Offer/answer model
- ACK
  - Response confirmation to INVITE

## SIP

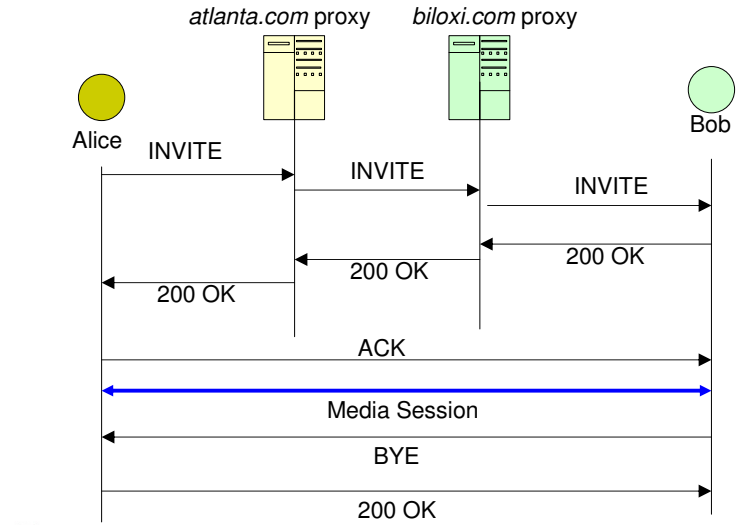
- INVITE Example

```
INVITE sip:bob@biloxi.com SIP/2.0
Via:SIP/2.0/UDPpc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
(Alice's SDP not shown)
```

# SIP

- SIP supports user mobility
  - With proxy servers to help route requests to user's current location
  - Registration function allowing users to upload current locations for use by proxy servers

# SIP



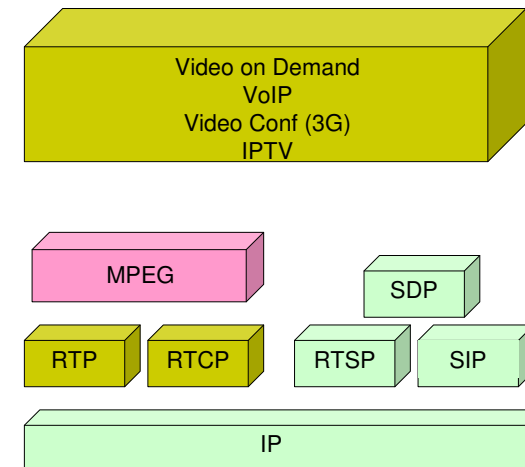
# SIP

## Response Example

```

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
    ;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
    ;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
    ;branch=z9hG4bK776asdhs ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
(Bob's SDP not shown)
    
```

# Streaming Applications



## Streaming Media Players



- Quick Time Player
- Real Player
- Microsoft Media player
- Packet Video
  
- Players for various platforms
  - PC
  - Pocket PC
  - 3G Phones

## Lecture Outline



- Session Description Protocol
  - How to describe a multimedia session ?
  - SDP example
  
- RTSP – Internet VCR controls
  - Stop, Pause, Play, Fast Forward over the internet
  - RTSP Methods
  - Signal Timing Diagram
  - Protocol stack for a total streaming system
  
- SIP – Quick Overview
  
- MPEG-4 File Format
  - What's so good about MP4?
  - File Format and Structure
  - Hinting

## File Format



- Now we can code and stream video and audio
- Next we look at file formats for exchanging multimedia
- MPEG-1 & MPEG-2 content typically exchanged as files that represent a stream ready to be delivered
  - Embedded absolute time stamps
  - Fragmentation of media for some preferred transport
  - Random Access could be difficult
- Need a new file format that can be used when
  - Storing captured media
  - Editing stored media
  - Combining multiple files
  - Delivering as File Download or Streaming
  - A “life-cycle” file format

## File Format



- MPEG-4 file format : MP4
  - “life-cycle” file format defined as part of MPEG-4
  - Based on Apple’s QuickTime format (“mov files”)
  
  - Due to MP4’s flexibility and extensibility
    - Industry acceptance for exchange of MPEG-4 video and audio
  
    - Basis for Motion JPEG 2000 file format
    - Basis for 3GPP file format used in MMS
      - Multimedia Messaging Service

## MP4 : Main File Format Concepts



- The Media data is stored separately from Meta data
  - Media data : Audio, Video samples
  - Metadata : Data describing the media  
Examples : timing info,  
number of bytes required for a frame
- Timing information specified by relative numbers (durations) rather than absolute numbers
  - Allows editing to be easier – eg insertion of a new frame
- Able to store media data distributed over several files
  - Use URLs to point to media data stored at various locations

## MP4 : Main File Format Concepts



- The Media data is stored separately from Meta data
- Timing information specified by relative numbers (durations) rather than absolute numbers
- Able to store media data distributed over several files
- Locating media data by means of data offsets and length information
  - Metadata tables mapping media sample number to location in a file
- Support streaming protocols through optional hint tracks
  - Metadata information for packetization and header data
  - Example – hints for RTP streaming stored as a separate track

## MP4: File Structure

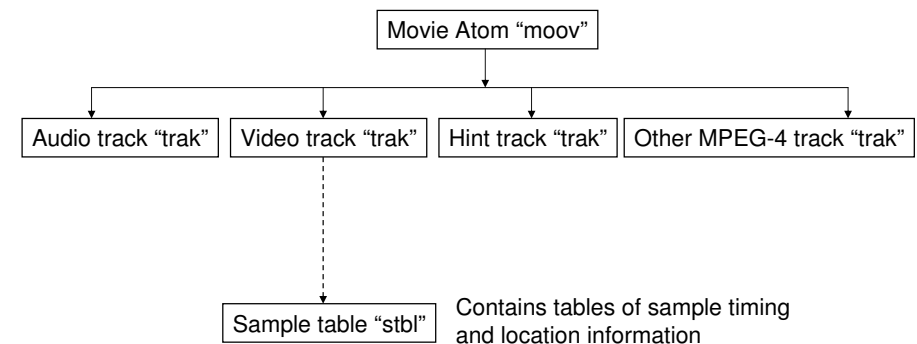


- The MP4 file format is composed of object-oriented structures called 'atoms'.
  - A unique tag and a length identify each atom.
- Metadata atoms :
  - A hierarchy of metadata giving information such as
    - index points, durations, and pointers to the media data.
  - This collection of atoms is contained in an atom called the 'movie atom'.
- Media Data Atoms :
  - The media data is contained in one or more 'mdat' or media data atoms,
  - or could be located outside the MP4 file; referenced via URL's.

## MP4: File Structure

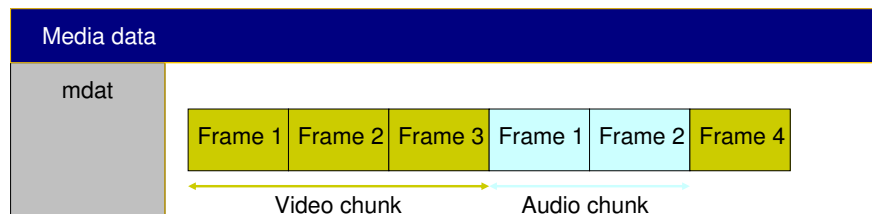


- Metadata Hierarchy [5]



## MP4: File Structure

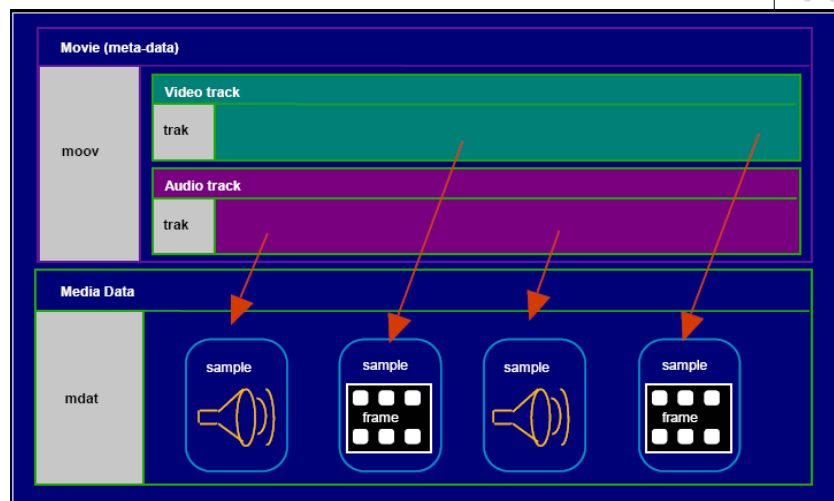
- Media data “*mdat*” Atom
  - Contains interleaved, time ordered video and audio samples referred to as Access Units (AU)
  - Several frames from the same track are often stored contiguously in “*chunks*”.



## MP4: File Structure

- Basic structure of MP4 file
  - Metadata “*moov*” atom
    - Each stream has a track “*trak*” atom – which contain tables
    - Tables with info on AU’s
      - Timing, size, location (offsets), synchronization points
      - Tables used to point to AU in “*mdat*”
  - Media Data “*mdat*” atom
    - Contains AU of audio and video
  - Putting it together
    - See next slide

## MP4: File Structure [4]



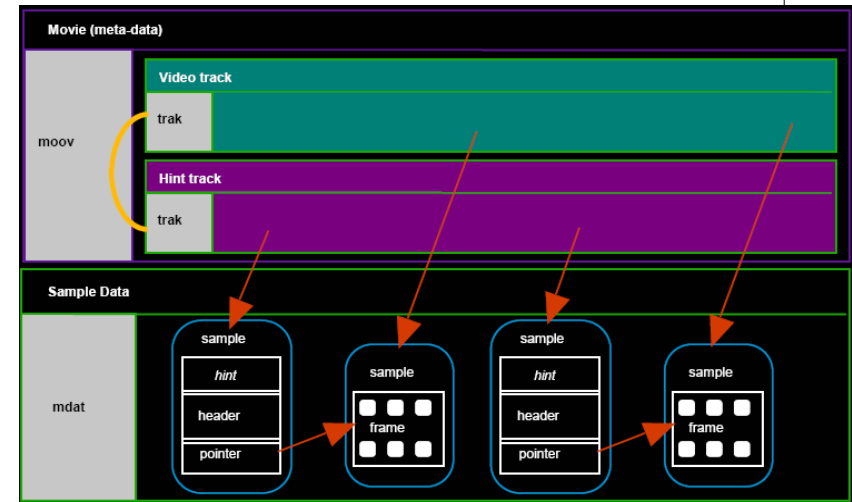
## MP4: Hinting

- MP4 file format is a streamable format, as opposed to a streaming format.
- The file format facilitates streaming without ever being streamed directly
  - File format does NOT define an on-the-wire protocol, and is never actually streamed over a transmission medium.
  - Metadata in the file known as ‘hint tracks’ provide instructions, telling a server application how to deliver the media data over a particular delivery protocol.
  - There can be multiple hint tracks for one presentation, describing how to deliver over various delivery protocols.

## MP4: Hinting

- Remember for streaming, need to format data into packets, with each packet containing a
  - Header, and
  - Samples or Media data
  - Example – RTP for MPEG-4
- Hint Track
  - Tables pointing to header information
    - The header information is stored in “mdat”
    - Example – RTP header information
  - Then points to appropriate fragment of sample data
    - Example – one video frame for an RTP packet
  - See next slide

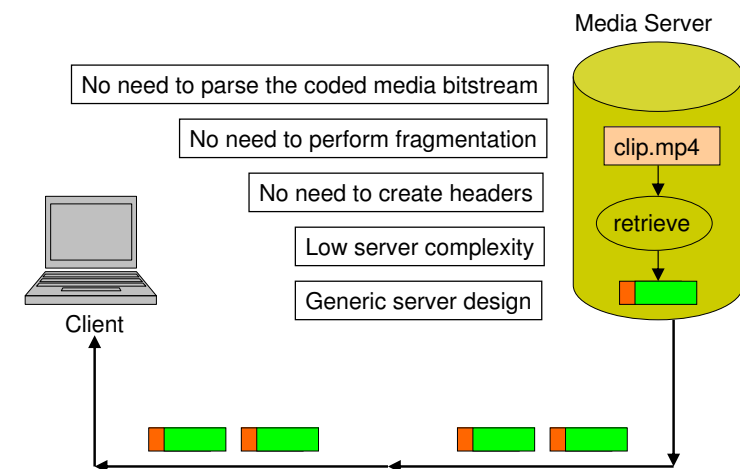
## MP4: Hinting [4]



## MP4: Hinting

- What to do to accommodate different transport protocols ?
  - Multiple hint tracks for one media
- Advantage to the media server ?
  - The server can be “dumb”
  - No knowledge of transport protocol required
    - Header information already stored in the MP4 file
  - No need to do media fragmentation
    - Pointers to appropriate media fragments already created
    - No knowledge of media codec required

## MP4: Hinting



## References and Further Reading



1. IETF RFC 2327 “SDP: Session Description Protocol”
2. IETF RFC 2326 “Real Time Streaming Protocol (RTSP)”
3. IETF RFC 3261 “SIP: Session Initiation Protocol”
4. David Singer, QuickTime Engineering “Quick Time etc. Introduction for UC Davis”, Feb 2004
5. MPEG-4 Overview, ISO/IEC JTC1/SC29/WG11 N4668  
[www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm](http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm)