Multimedia Systems and Applications

Multimedia streaming

James Wang

What is multimedia streaming?

- **There are two ways to deliver multimedia content online:**
  - **Streaming:** When media is requested (eg an on-line song) it is sent to your machine and played back as soon as your computer begins to receive the media, however the media is not saved to your computer as it is played. If your wish to play the song again, the streaming process will have to repeat.
  - **Downloading:** When media is requested (eg an on-line song as above) it is first download it in it's entirety before you can play it back. It is saved to your computer for later playback without additional downloading.

2

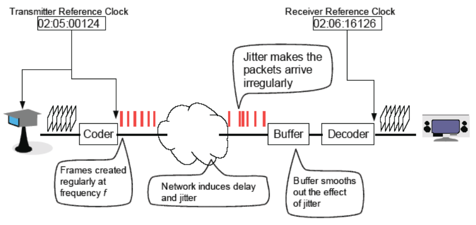
What is real-time?

- **Real-time communication: No noticeable delay**
- **Real-time application: An application able to have a sustained processing of a temporal data stream**
  - Without having queues building up
  - With reasonable sustained quality level
  - But - there is of course always delay to some extent!
- **Different communication scenarios may have different sensitiveness to delay.**

3

Real-time Scenario

- **Real-time multimedia streaming includes interaction, live broadcasting, etc.**


Transmitter Reference Clock
02:05:00124
Receiver Reference Clock
02:06:16126
Jitter makes the packets arrive irregularly
Coder
Buffer
Decoder
Frames created regularly at frequency *f*
Network induces delay and jitter
Buffer smooths out the effect of jitter

4

Real-time protocol design consideration

- **Skip retransmission**
  - The time it takes to send ACKs/NACKs upstream and wait for the lost packet may cost too much in delay.
- **Skip flow control**
  - Receiver should be a real-time application being able to cope with the (bounded) rate the sender transmits at.
- **No upstream communication**
  - if we have a broadcast scenario, no or little upstream communication can be had.

5

Challenges

- **Synchronization**
  - Sender and receiver needs to be synchronized to each other to avoid re-occurring buffer over-flows and under-runs
  - Essentially having the sender and receiver following a clock at exactly the same frequency.
- **Jitter**
  - Jitter is the varying of the packet-delay around the mean delay.
  - Jitter is usually caused by the network, but could also result from processing stages in the sender that varies in time.
  - The larger the jitter the larger the buffer at the receiver-side has to be.

6

A general view of the real-time protocols

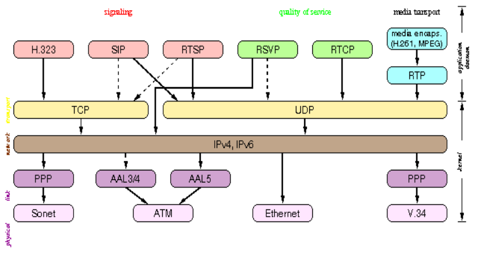
- **the protocols and their application field…**
  - stream description: SDP, SMIL..., describe the session and content
  - stream control: RTSP, remote control the session
  - media transport: RTP, send data and metadata
  - resource reservation (if any!): RSVP, DiffServ, make sure the communication path offers appropriate guarantees…

    …otherwise Best-Effort transmissions!

7

A general view… (cont')

- **and the result…**



8

RTP/RTCP protocols

- **IETF Audio/Video Transport Working Group**
  - RTPv1 RFC 1889 (January 1996)
  - RTPv2 RFC 3550 (July 2003)
- **Real-Time Protocol (RTP)**
  - a framework for real-time multimedia applications
  - does not define any QoS mechanism for real-time delivery!
- **Real-Time Control Protocol (RTCP)**
  - its companion control protocol
  - does not guarantee anything either!

* http://www.ietf.org/rfc/rfc3550.txt

9

Design goals

- **Flexible: provide mechanisms, do not dictate algorithms!**

  ⇒ instantiations for H261, MPEG1/2/...
- **protocol neutral: UDP/IP, private, ATM networks...**
- **Scalable: unicast, multicast**
- **separate control/data: some functions may be taken over by conference control protocol (e.g. RTSP)**

10

RTP Terminologies

- **RTP Session**
  - The set of communicating entities which may via RTP and RTCP know of each other's identities.
- **Synchronization Source**
  - The origin of a stream where RTP-packets are generated. Identified by a 32-bit number: SSRC.
- **Translation**
  - The act of converting a stream to another format and/or rate.
- **Mixing**
  - The act of adding two or more streams together and encode the result into a new stream (with new SSRC).

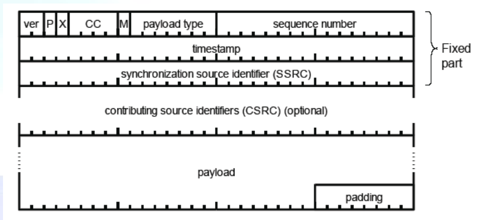
11

RTP data functions

- **content labeling**
  - source identification
  - loss detection
  - re-sequencing
- **timing**
  - intra-media synchronization: remove jitter with playout buffers
  - inter-media synchronization: lip-synchro between audio-video
- **Typically use**
  - uses UDP (TCP is not for real-time!!!)
  - no fixed UDP port (negotiated out of band)
  - UDP port for RTCP = UDP port for RTP + 1
  - usually one media per RTP session (i.e. port pair)

12

RTP header


ver
P
X
CC
M
payload type
sequence number
timestamp
synchronization source identifier (SSRC)
Fixed part
contributing source identifiers (CSRC) (optional)
payload
padding

- version (V)
- padding (P)
- extension (X)

CSRC count (CC)
marker (M)
payload type (PT)

13

Synchronization Source Identifier

- Every RTP packet is market with 32-bit number identifying a single timing and sequence number space
- SSRC provides a transport layer independent "tag". RTP may run over various protocols (Ipv4, Ipv6, ATM)
- SSRC must be unique among all hosts taking part in the RTP session
- SSRC is picked randomly, but sender must in case of a detected collision issue a BYE message over the control channel and change SSRC

14

Contributing Source Identifiers

- When a mixer creates a new stream out of others it may include the SSRCs of up to 15 of the contributing sources.
- CC-field denotes the number of source identifiers in CSRC field.
- Using CSRC is a safety measure against loops in a RTP streaming session.

15

Payload

- The payload type (7 bits) indicate the format of the media put into the payload area. Compared to the number of codecs out there one realizes that assigning the payload type from the dynamic range might be common. It is then up to the session handling mechanism to inform parties about the type used.
- A flow can only carry one payload type. But the type can change.
- Each payload type has a reference clock frequency associated with it.

16

RTP Payload Types (RFC 3551)

- http://www.ietf.org/rfc/rfc3551.txt

PT	A/V	Clock Freq	Channels	MIME-type	Description
3	audio	8000Hz	1	audio/gsm	3.2bps speech coder
10	audio	44100Hz	2	audio/l16	two-channel CD-style
11	audio	44100Hz	1	audio/l16	one-channel CD-style
14	audio	90kHz	special	audio/mpa	MPEG audio layer 1,2,3
26	video	90kHz	-	video/jpeg	Motion-JPEG
32	video	90kHz	-	video/mpv	MPEG-1,2 elementary stream
33	a/v	90kHz	special	video/mp2t	MPEG-2 transport stream
34	video	90kHz	-	video/h263	H.263 video telephony

35-71 unassigned
72-76 reserved
77-95 unassigned
96-127 dynamic

17

Timestamp

- Indicates the sampling instant of first octet of payload data.
- The mapping from transmitter's reference clock to timestamp is defined by payload type.
- Initial value is random
- Clock resolution should be enough to support synchronization and jitter measurements based on timestamp alone.
- Note that sender also transmits timestamps of "wall clock time" in RTCP sender reports. These are needed to synchronize two or more streams with each other.

18

Sequence Number

- Each packet has a 16-bit sequence number
- Increases with one for each new sent packet.
- Starts with a random value
- Used to detect packet loss. Need not be related to play-out order
- Basic RTP does not provide error correction

19

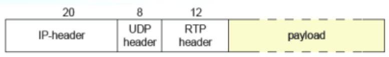
Marker

- Marks this packet as special in a sense defined by payload type.
  - May be set in speech formats for indicating talk starts again after a period of silence (or comfort noise)
  - Most video formats specify the marker bit to be set to “1” in the last packet for a frame. This gives a simple indication that it is time to display.

20

RTP Transmission

- RTP is designed to run over any packet switched network. On Internet we use UDP/IP.



- The overhead per RTP packet will be 40 bytes. This is sometimes seen as too excessive and several header compression techniques have been suggested:
  - RObust Header Compression (ROHC) – RFC3095
  - Enhanced Compressed RTP (CRTP)- RFC3545
  - IP Header Compression – RFC2507 (LuTH)

21

How to Mix Audio and Video?

- Same session, different SSRC
  - Does not support different QoS profiles
  - Impossible in a multicast scenario to join just the audio stream (for instance)
- Different sessions, same SSRC
  - works ok. SSRC:s totally unrelated
- Different sessions, different SSRC
  - works ok. SSRC:s totally unrelated

22

Real-time Control Protocol (RTCP)

- periodic transmission of control packets
- several functions
  - feedback on the quality of data distribution
  - let everybody evaluate the number of participants
  - persistent transport-level canonical name for a source, CNAME
    - usually: user@host
    - will not change, even if SSRC does!
    - provides binding across multiple media tools for a single user
  - Carry NTP timestamps enabling inter-media synchronization.
  - Optional minimal session control.

23

Five RTCP packets

SR	Sender Reports
	tx and rx statistics from active senders
RR	Receiver Reports
	rx statistics from other participants, or from active senders if more than 31 sources
SDES	source description, including CNAME
BYE	explicit leave
APP	application specific extensions

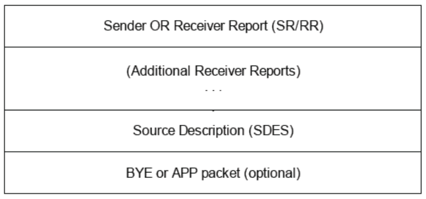
24

RTCP generalities

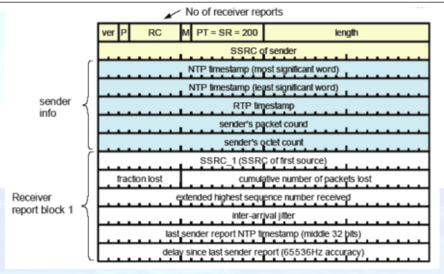
- **distribution**
  - use same distribution mechanisms as data packets (n→m multicast)
  - multiple RTCP packets can be concatenated by translators/mixers
    - ⇒ compound RTCP packet
- **scalability with session size**
  - RTCP traffic should not exceed 5% of total session bandwidth
    - requires an evaluation of number of participants
    - RTCP tx interval = f(number of participants)
  - at least 25% of RTCP bandwidth is for source reports
    - let new receivers quickly know CNAME of sources!

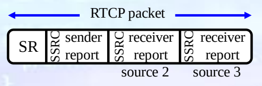
RTCP Compound Packets

- **Several RTCP packets are concatenated to form one compound packet. These will at least contain two individual packets (SR/RR and a SDES).**

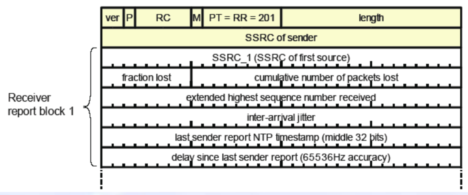

Sender OR Receiver Report (SR/RR)
(Additional Receiver Reports)
. . .
Source Description (SDES)
BYE or APP packet (optional)

SR RTCP packets


No of receiver reports
ver
P
RC
M
PT = SR = 200
length
SSRC of sender
NTP timestamp (most significant word)
NTP timestamp (least significant word)
RTP timestamp
sender's packet count
sender's octet count
sender info
SSRC_1 (SSRC of first source)
fraction lost
cumulative number of packets lost
extended highest sequence number received
inter-arrival jitter
last sender report NTP timestamp (middle 32 bits)
delay since last sender report (65536Hz accuracy)
Receiver report block 1

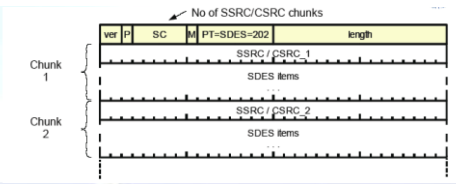


RR RTCP packets

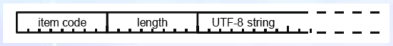

ver
P
RC
M
PT = RR = 201
length
SSRC of sender
SSRC_1 (SSRC of first source)
fraction lost
cumulative number of packets lost
extended highest sequence number received
inter-arrival jitter
last sender report NTP timestamp (middle 32 bits)
delay since last sender report (65536Hz accuracy)
Receiver report block 1

- **When participant in an RTP-session only receives flow we skip the sender information block. Note that a receiver need to have an SSRC anyway!**
- **A problem here is to which address and port do we send our RTCP messages!**
  - If it's a multicast session on UDP(IP#, port) we send them to UDP(IP#,port+1),
  - but if it's a unicast session the RTCP upstream IP# and port has to be configured by a higher level session protocol.

RTCP Source Description (SDES)

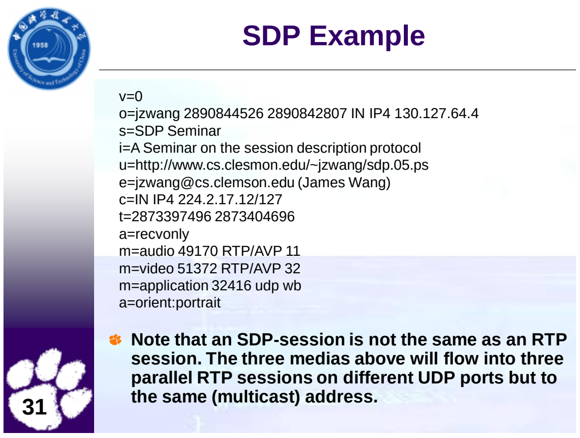


- **SDES information blocks must be sent by every sender. It gives some textual information regarding the flow. The flow might possibly be the result of mixing in which case information for each mixed flow is present.**
- **An SDES item looks like**



Session Description Protocol (SDP)

- **http://www.ietf.org/rfc/rfc2327.txt**
- **SDP conveys information needed to join one or more multimedia sessions**
  - Session name and purpose
  - Time(s) when session is active
  - What media is involved
  - Information on how to receive those media (addresses, ports, formats etc.)
  - Optionally bandwidth requirements
  - Optionally how to contact person responsible

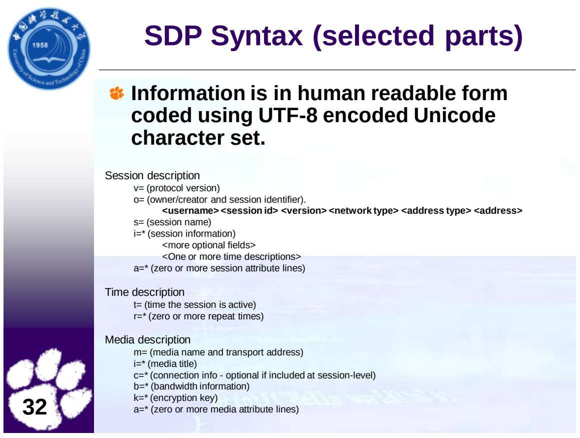


SDP Example

```
v=0
o=jzwang 2890844526 2890842807 IN IP4 130.127.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.clesmon.edu/~jzwang/sdp.05.ps
e=jzwang@cs.clemson.edu (James Wang)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 11
m=video 51372 RTP/AVP 32
m=application 32416 udp wb
a=orient:portrait
```

- **Note that an SDP-session is not the same as an RTP session. The three medias above will flow into three parallel RTP sessions on different UDP ports but to the same (multicast) address.**

31



SDP Syntax (selected parts)

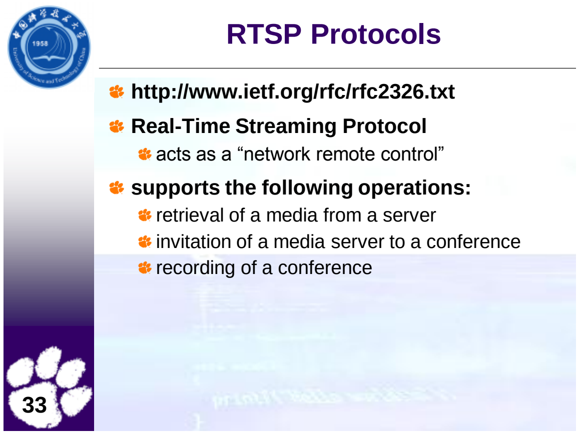
- **Information is in human readable form coded using UTF-8 encoded Unicode character set.**

```
Session description
        v= (protocol version)
        o= (owner/creator and session identifier).
                <username> <session id> <version> <network type> <address type> <address>
        s= (session name)
        i=* (session information)
                <more optional fields>
                <One or more time descriptions>
        a=* (zero or more session attribute lines)

Time description
        t= (time the session is active)
        r=* (zero or more repeat times)

Media description
        m= (media name and transport address)
        i=* (media title)
        c=* (connection info - optional if included at session-level)
        b=* (bandwidth information)
        k=* (encryption key)
        a=* (zero or more media attribute lines)
```

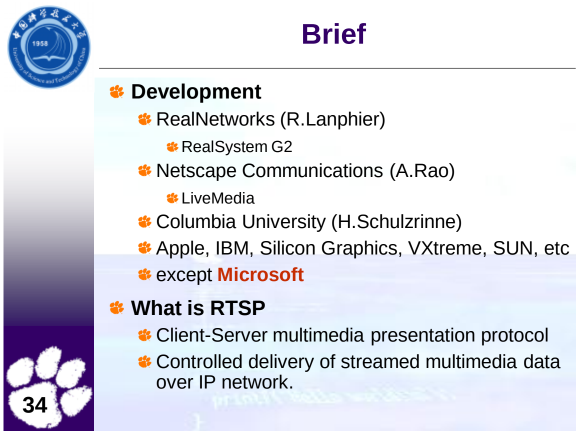
32



RTSP Protocols

- **http://www.ietf.org/rfc/rfc2326.txt**
- **Real-Time Streaming Protocol**
  - acts as a "network remote control"
- **supports the following operations:**
  - retrieval of a media from a server
  - invitation of a media server to a conference
  - recording of a conference

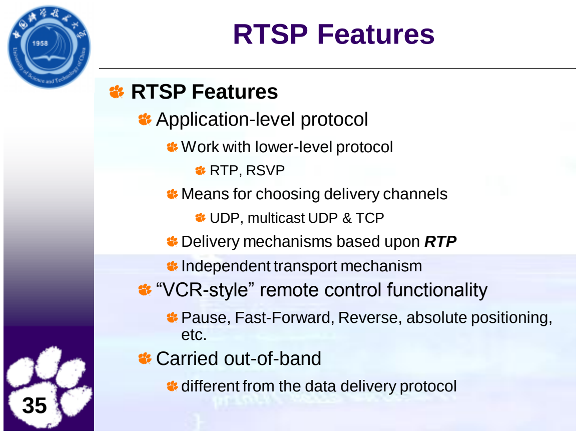
33



Brief

- **Development**
  - RealNetworks (R.Lanphier)
    - RealSystem G2
  - Netscape Communications (A.Rao)
    - LiveMedia
  - Columbia University (H.Schulzrinne)
  - Apple, IBM, Silicon Graphics, VXtreme, SUN, etc
  - except **Microsoft**
- **What is RTSP**
  - Client-Server multimedia presentation protocol
  - Controlled delivery of streamed multimedia data over IP network.

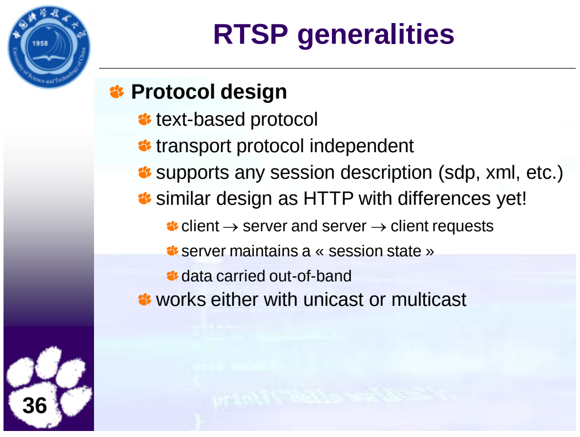
34



RTSP Features

- **RTSP Features**
  - Application-level protocol
    - Work with lower-level protocol
      - RTP, RSVP
    - Means for choosing delivery channels
      - UDP, multicast UDP & TCP
    - Delivery mechanisms based upon ***RTP***
    - Independent transport mechanism
  - "VCR-style" remote control functionality
    - Pause, Fast-Forward, Reverse, absolute positioning, etc.
  - Carried out-of-band
    - different from the data delivery protocol

35



RTSP generalities

- **Protocol design**
  - text-based protocol
  - transport protocol independent
  - supports any session description (sdp, xml, etc.)
  - similar design as HTTP with differences yet!
    - client → server and server → client requests
    - server maintains a « session state »
    - data carried out-of-band
  - works either with unicast or multicast

36

RTSP vs. HTTP

- **similar to HTTP/1.1**
  - works for streamed multimedia data (audio, video)
  - use URLs
  - RTSP aims to provide the same services on streamed audio and video just as HTTP does for text and graphics.
  - Number of new methods
  - Different protocol identifier
  - Maintain state by default (stateless nature of HTTP)
  - Carried out-of-band
  - ISO 10646 (UTF-8)
    - ISO 8859-1, HTML internationalization efforts

37

Protocol Supports

- **Retrieval of media from media server**
  - the client can request a presentation description, and ask the server to setup a session to send the requested data
- **Invitation of a media server to a conference**
  - The media server can be invited to the conference to play back media or to record a presentation
- **Addition of media to an existing presentation**
  - The server or the client can notify each other about any additional media becoming available

38

Protocol Properties

- **Extendable**
  - New methods & Parameters can be easily added
- **Easy to parse**
  - standard HTTP or MIME parsers
- **Secure**
  - re-use web security mechanisms
- **Transport-independent**
  - such as UDP, RDP, and TCP
- **Multi-server capable**
  - Each media stream reside on a different server

39

Protocol Properties (cont.)

- **Control of recording devices**
  - control both recording and playback devices
- **Separation of stream control and conference initiation**
  - SIP or H.323 may be used to invite a server to conference
- **Suitable for professional applications**
  - allow remote digital editing
  - frame-level accuracy
- **Presentation description neutral**

40

Protocol Properties (cont.)

- **Proxy and firewall friendly**
  - readily handled by both application and transport-layer firewalls
- **HTTP-friendly**
  - reuse HTTP concepts
- **Appropriate server control**
- **Transport negotiation**
  - negotiate the transport method
- **Capability negotiation**
  - if seeking is not allowed, the user interface must be able to disallow moving a sliding position indicator

41

Overall Operation

- **Each presentation & media stream**
  - individually controllable by RTSP
  - identified by an RTSP URL
- **Presentation & properties of the media**
  - defined by presentation description file
  - Presentation description
    - using HTTP or other means such as email
    - describe on or more presentations
    - contain several media streams
    - encodings, languages, other parameters
- **Several media streams can be located on different servers**
- **network destination address and port need to be determined.**

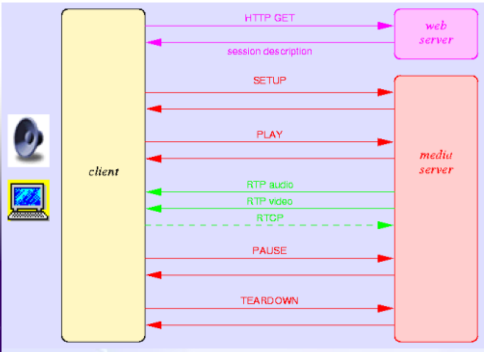
42

Modes of Operation

- **Unicast :**
  - the media is transmitted to the source of the RTSP request, with the port number chosen by the client
- **Multicast, Server chooses address :**
  - the media server picks the multicast address & port
- **Multicast, Client chooses address :**
  - the multicast address, port and encryption key are given by the conference description

43

RTSP Operation



44

RTSP Operation : Web integration

- **Web page with "program guide"**
- **Contains pointer to presentation description**

```
<session>
    <group>
        <track src="rtsp://audio.mtv.com/movie">
        <track src="rtsp://video.mtv.com/movie">
    </group>
</session>
```

- **RTSP sets up and controls delivery**
- **RSVP reserves resources**
- **RTP delivers data**

45

Metafile Example

```
<title>Twister</title>
<session>
      <group language=en lipsync>
            <switch>
              <track type=audio
                   e="PCMU/8000/1"
                   src = "rtsp://audio.example.com/twister/audio.en/lofi">
              <track type=audio
                   e="DVI4/16000/2" pt="90 DVI4/8000/1"
                   src="rtsp://audio.example.com/twister/audio.en/hifi">
            </switch>
          <track type="video/jpeg"
                   src="rtsp://video.example.com/twister/video">
       </group>
</session>
```

46

RTSP methods

- **Major methods**

SETUP:	server allocates resources for a stream and starts an RTSP session
PLAY:	starts data tx on a stream
PAUSE:	temporarily halts a stream
TEARDOWN:	free resources of the stream, no RTSP session on server any more

- **Additional methods**

OPTIONS:	get available methods
ANNOUNCE:	change description of media object
DESCRIBE:	get low level descr. of media object
RECORD:	server starts recording a stream
REDIRECT:	redirect client to new server
SET_PARAMETER:	device or encoding control

47

RTSP Session

```
C->W :  DESCRIBE /twister HTTP/1.1
        host : www.ibm.com
        Accept : application/rtsl; application/sdp
W->C :  200 OK
        Content-Type : application/rtsl

        <session>
        ...

C->A :  SETUP rtsp://audio.ibm.com/twister/au.en/lofi RTSP/1.0
        Transport : rtp/udp; compression; port=3056;
             destination=224.2.0.1; mode=PLAY
A->C :  RTSP/1.0 200 1 OK
        Session : 1234
```

48

RTSP Session (cont.)

```
C->A :    PLAY rtsp://audio.ibm.com/twister/au.en/lofi RTSP/1.0
          Session : 1234
          Range : smpte=0:10:00-0:12;
               time=19970123T143720Z, pmpte15-

C->A :    PAUSE rtsp://audio.ibm.com/twister/au.en/lofi RTSP/1.0
          Session : 1234
          Range : smpte=0:11:00
C->A :    TEARDOWN rtsp://audio.ibm.com/twister/au.en/lofi
RTSP/1.0
          Session : 1234
A->C :    200 3 OK
```

49

Session Initiation Protocol

- All telephone calls and video conference calls take place over the Internet
- People are identified by names or e-mail addresses, rather than by phone numbers.
- You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using.
- Sponsored by IETF.

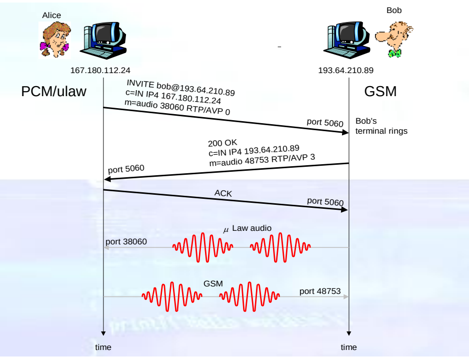
50

SIP Services

- **Setting up a call**
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms to end call.
- **Determine current IP address of callee.**
  - Maps mnemonic identifier to current IP address
- **Call management**
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

51

Call to a known IP address



52

Setting up a call

- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address & preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- Default SIP port number is 5060.

53

Setting up a call (more)

- **Codec negotiation:**
  - Suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply and list encoders he can use.
  - Alice can then send a new INVITE message, advertising an appropriate encoder.
- **Rejecting the call**
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden".
- **Media can be sent over RTP or some other protocol.**

54

Example of SIP message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Note:
1. HTTP message syntax
2. sdp = session description protocol
3. Call-ID is unique for every call.

- Here we don't know Bob's IP address. Intermediate SIP servers will be necessary.
- Alice sends and receives SIP messages using the SIP default port number 506.
- Alice specifies in Via: header that SIP client sends and receives SIP messages over UDP

55

Name translation and user location

- **Caller wants to call callee, but only has callee's name or e-mail address.**
- **Need to get IP address of callee's current host:**
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, PDA, car device)
- **Result can be based on:**
  - time of day (work, home)
  - caller (don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

56

SIP Registrar

- **When Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server (similar function needed by Instant Messaging)**
- Register Message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

57

SIP Proxy

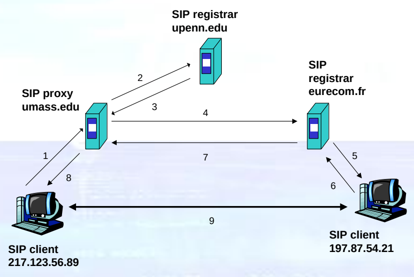
- **Alice send's invite message to her proxy server**
  - contains address sip:bob@domain.com
- **Proxy responsible for routing SIP messages to callee**
  - possibly through multiple proxies.
- **Callee sends response back through the same set of proxies.**
- **Proxy returns SIP response message to Alice**
  - contains Bob's IP address

Note: proxy is analogous to local DNS server

58

Example

Caller jim@umass.edu places a call to keith@upenn.edu



59

Message Routing

(1) Jim sends INVITE message to umass SIP proxy.
(2) Proxy forwards request to upenn registrar server.
(3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr
(4) umass proxy sends INVITE to eurecom registrar.
(5) eurecom regristrar forwards INVITE to 197.87.54.21, which is running keith's SIP client.
(6-8) SIP response sent back
(9) media sent directly between clients.

Note: also a SIP ack message, which is not shown.

60