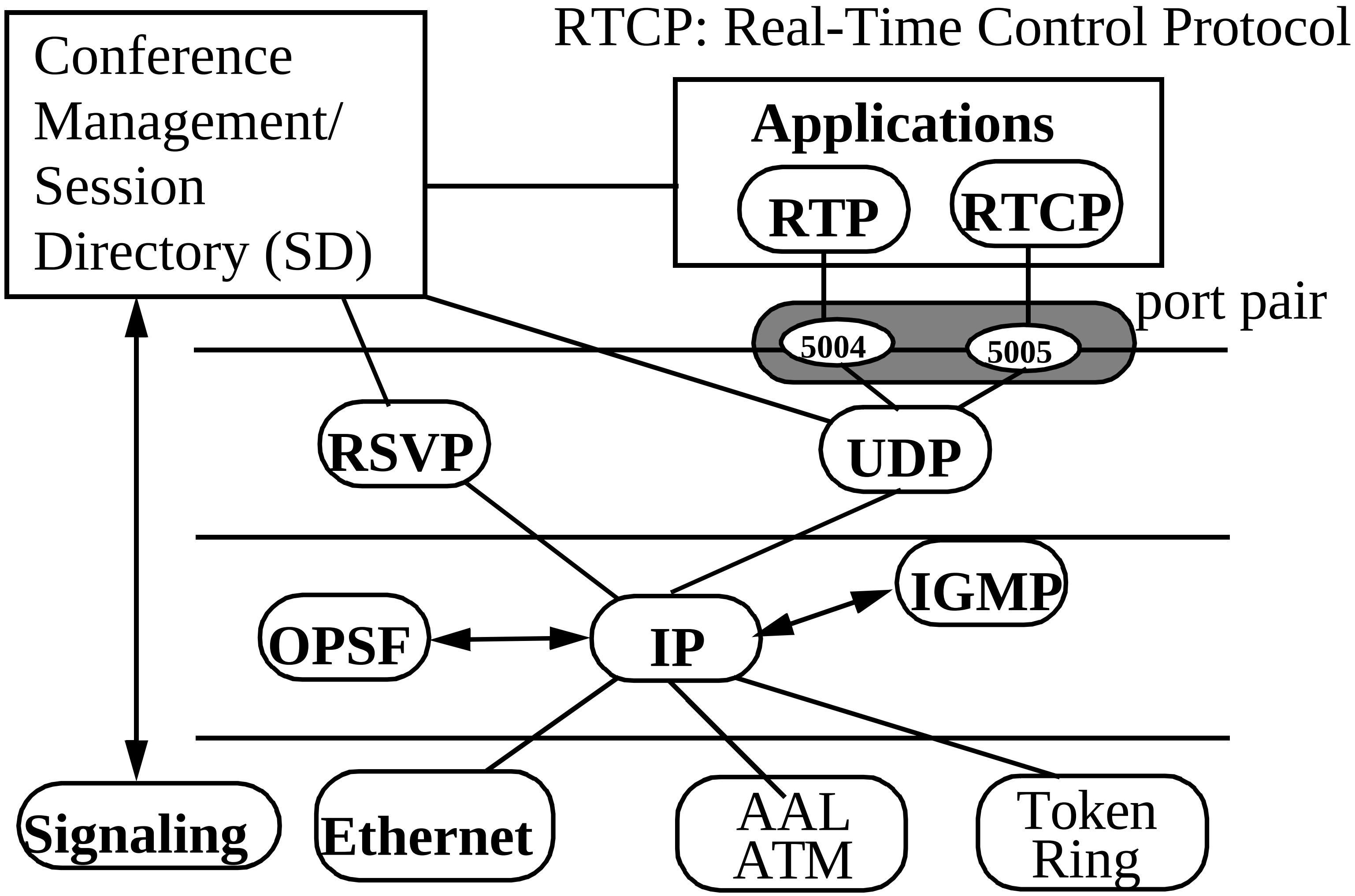
RTP: A Transport Protocol for Real-Time Applications



RTP

- A Standard Track Protocol
- RFC 1889: RTP
  - provide end-to-end transport support for real-time audio, video, simulation data
  - augmented by RTCP for monitoring, QoS feedback, awareness
- RFC 1890: RTP profile for audio and video conferences with minimal control
  - for different applications, different “profiles” are specified
  - which include mapping of payload type to encoding specification
- Product of IETF Audio Video Transport Working Group (AVT WG)
- Part of the puzzle for Internet integrated services.
  - RSVP, OS,...

RTP Goals

- Light weight, efficient
  - apply Application Layer Framing (ALF) or integrated layer processing design principle [Clark and Tannenhouse90]
  - typical integrated with the applications (VAT, VIC)
- Scalable support multicast from 2 to O(1000?)
  - divide control bandwidth among receiver’s reports
- Separation of control and data.
  - some function of RTCP may be taken over by conference control protocols or session protocols.
- Secure
  - support encryption

RTP Functions

- segmentation and reassembly, and checksum done by UDP
- resequencing (if needed) supported by sequence number field
- loss detection for quality estimation
- support jitter estimation
  - with NTP timestamp (64 bits), and RTP timestamp (32 bits).
- intra-media synchronization
  - remove delay jitter through playout buffer
  - drifting sampling clocks
- inter-media synchronization (lip sync between audio and video)
  - use Canonical Name (CNAME) to associate different media streams
- QoS feedback and rate adaptation for reporting
- source/talker identification
- support mixers and translators

RTP Mixer and Translators

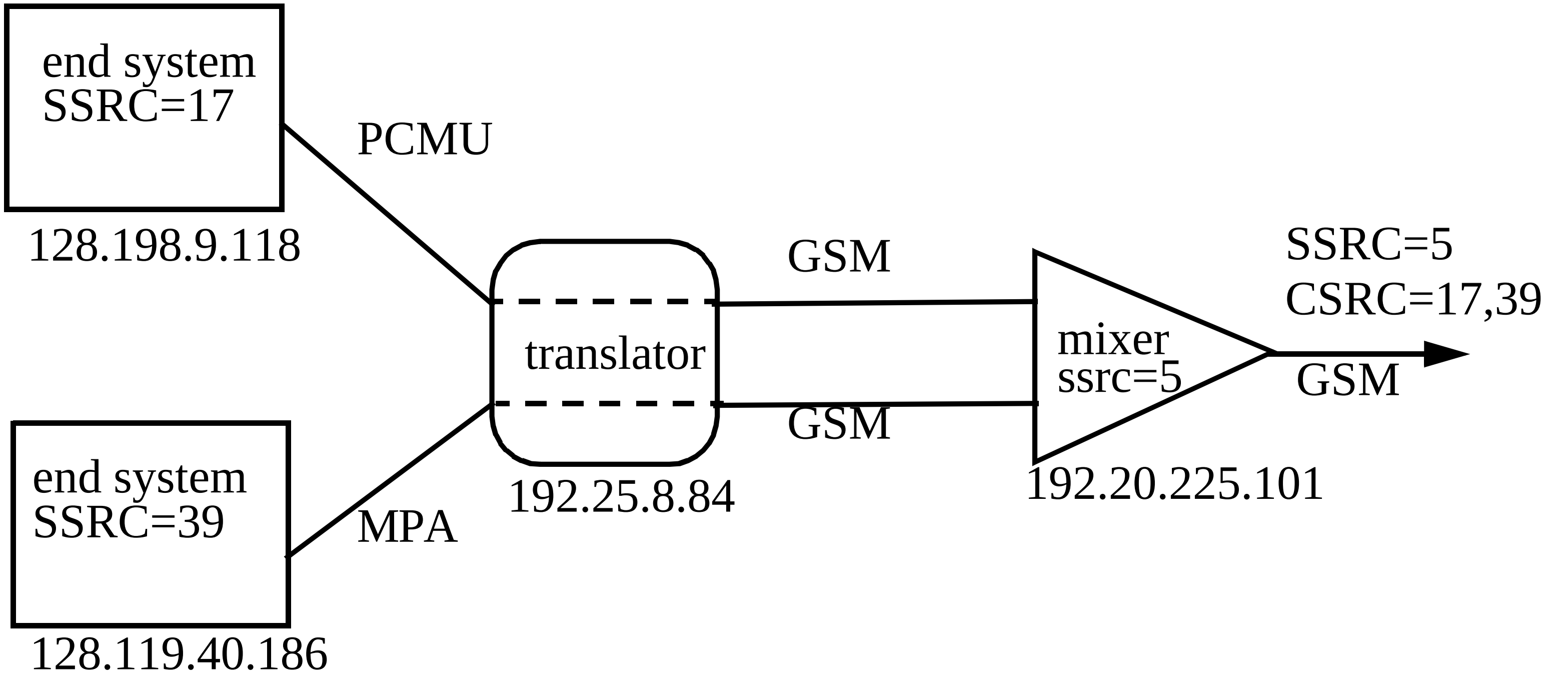
Mixer

- Merge several media streams (of the same types) into one new stream (possible with new encoding)
- Reduce bandwidth of the network (esp. for low speed dial up networks)
- appears as new source with new ID.

Translator

- Single media stream
- may convert encoding (high speed to low speed, or different encoding schemes)
- can be used for protocol translation (ATM-IP) or firewall

Example of RTP Mixer and Translators



RTP Packet Header Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Header Extension                  |           |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            Payload (audio, video, ... )                       |
|                             ....                    0x00| bytes|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- P: Padding (for encryption) last byte of the packet contains padding count
- M: marker bit, indicate frame, and beginning of talkspurt

  — used to allow delay adjustment
- CSRC: Contributing Source ID.

- CC: Content source count (for mixers)
- PT: Payload Type, indicating encoding method; may change during session
- Sequence number: for detecting packet lost or resequencing packets.
- Timestamp: the sampling instant of the first octet of data in RTP packet.
  - clock frequency depends on the format (typical 8kHz audio, 90kHz video)
  - sampling clock is used (not the system clock)
  - audio RTP pkt sent per 20msec with 8kHz, will increment timestamp by 160.
- SSRC: Synchronization Source ID
  - Each source picks at random
  - There are collision resolution mechanisms for detecting/resolving multiple source choosing the same ID.

RTP Timestamp

- Increment by 1 for each sample (160 for 20 ms packets @8000 Hz)
  - audio packet are typical sent at 20 ms during
  - all data sampled during the period are sent in one packet
  - there are 160 samples in a packet
- random starting value (for security, encryption reason)
- constant rate for each audio type
  - 8000 Hz for PCMU (μ-law)
  - 44100 Hz for L16
- 90kHz for video
- several video frames may have the same timestamp
- gaps indicating silence
- typical 20 to 100 (e.g., lecture or less sensitive applications) ms for audio

RTP in a Network

- on top UDP, no fixed port
- RTCP port (always odd) = PTP port (always even)+ 1
- native ATM: RTP fit directly into AAL 5 frame
- encapsulation for other using length fields
- Typical one media (audio, video,...) per port pair
- exception: bundled MPEG multimedia stream

RTCP Msg Type

- Stackable packets (memory alignment 32 bit boundary)
- SR: Sender Report
  - Byte Sent Field for estimating the rate
  - Timestamp Fields for synchronizing media
- RR: Reception Report
  - number of pkts sent and expected for lost, jitter, and round-trip delay estimation
- SDES: source description.
  - CNAME (canonical name=user@host) identifies user across media
  - Name (full name)
  - email
  - Location (address),....
- BYE: Explicit leave
- APP: Extension for application specific data (none yet)

RTCP Compound Packet

```
if encrypted: random 32-bit integer
    |
    |[------- packet -------][----------- packet -----------][-packet-]
    |
    |             receiver reports          chunk        chunk
    V                                    item  item     item  item
   --------------------------------------------------------------------
   |R[SR|# sender #site#site][SDES|# CNAME PHONE |#CNAME LOC][BYE##why]
   |R[  |# report #  1 #  2 ][    |#             |#         ][   ##   ]
   |R[  |#        #    #    ][    |#             |#         ][   ##   ]
   |R[  |#        #    #    ][    |#             |#         ][   ##   ]
   --------------------------------------------------------------------
  |<------------------  UDP packet (compound packet) --------------->|
```

RTCP Announcement Interval Computation

Goals:

- Estimate current number and IDs of participants dynamically
- Use SDES field to indicate who is talking
- QoS feedback report adjust to sender rate
- built in scaling control (up to O(1000))
- control bandwidth is by default 5% of total bandwidth
- Adjusting the reporting rate according to the number of participants.

RTCP bandwidth Scaling

- Every participant periodically multicast RTCP packet to same group as data
- Every will know who is out there.
- Session bandwidth
  - single audio stream
  - sum of concurrent report streams
- Sender period (~2-5 min)
  T = (# of sender/(0.25*0.05*session bandwidth))*avg RTCP packet size
- Receiver period
  T = (# of receivers/(0.75*0.05*session bandwidth))*avg RTCP packet size
- Next RTCP packet sending time=
  last RTCP packet sending time + max (5Ts, 30min) - random(0.5...1.5)
- randomization prevent “bunching” all report after network partition heals

RTCP Sender Reports (SR) Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    RC   |   PT=SR=200   |             length            | header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SSRC of sender                        |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|              NTP timestamp, most significant word             | sender
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ info
|             NTP timestamp, least significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP timestamp                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     sender's packet count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      sender's octet count                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first source)                 | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
| fraction lost |       cumulative number of packets lost       |   1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_2 (SSRC of second source)                | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                  profile-specific extensions                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- NTP: Network Time Protocol, it reports the wallclock time in unsigned 64 bit fixed point number (to the left of second 32 bits) since 0h January 1 1900.
- RTP timestamp are unsigned 32 bit fixed point number (to the left of second 16 bits) depends on source and offset with random interval (security reason?)
- NTP timestamp and RTP timestamp are computed at the same time for correlation when a sample is created.
- Fraction Lost: 8 bit fixed point on the left edge, report the faction of RTP data packet from SSRC_n lost since the previous SR or RR was sent.
- LSR: middle 32 bits of SSRC_n's SR NTP timestamp.
- DLSR: 32 bit, delay btw receiving of last SR and the sending of this RR.

Interarrival Jitter Calculation

- Let Si be the RTP timestamp from packet i.
- Let Ri be the time of arrival in RTP timestamp unit for packet i.
- PS@Sender(i,j) =Packet Spacing between packet i and j at sender site
- PS@Receiver(i,j)=Packet Spacing between packet i and j at sender site
- Difference in packet spacing between packet i and j
  D(i,j)= PS@Receiver(i,j) - PS@Sender(i, j)
  = (Rj -Ri) - (Sj-Si)
- Interarrival jitter, J, is calculated continuously as each data packet i is received from source SSRC_n.
- J=J+(|D(i-1, i)| - J)/16.
- This algorithm is called optimal first-order estimator.
- The gain parameter, 1/16, gives a good noise reduction ratio while maintaining a reason rate of convergence [Cadzow87].

Example of Round-Trip Time Computation

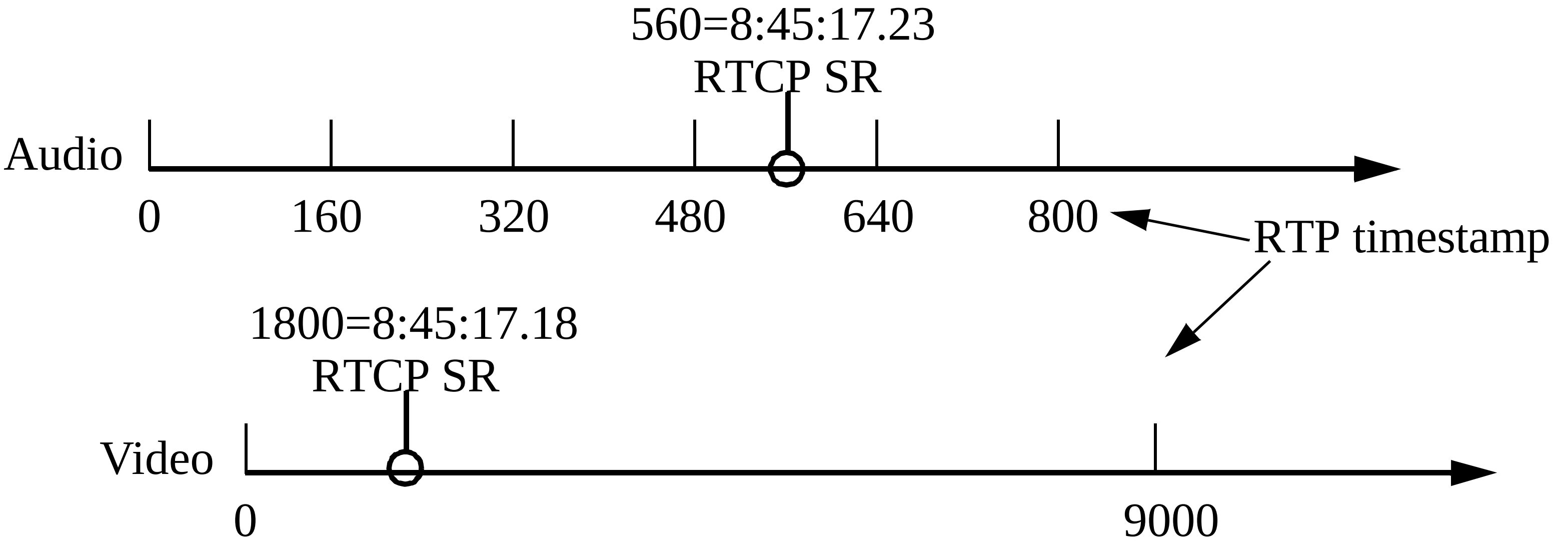
```
[10 Nov 1995 11:33:25.125]           [10 Nov 1995 11:33:36.5]
   n                 SR(n)              A=b710:8000 (46864.500 s)
   ----------------------------------------------------------------->
                      v                 ^
   ntp_sec =0xb44db705 v               ^ dlsr=0x0005.4000 (    5.250s)
   ntp_frac=0x20000000  v             ^  lsr =0xb705:2000 (46853.125s)
     (3024992016.125 s)  v           ^
   r                      v         ^ RR(n)
   ----------------------------------------------------------------->
                          |<-DLSR->|
                           (5.250 s)

   A     0xb710:8000 (46864.500 s)
   DLSR -0x0005:4000 (    5.250 s)
   LSR  -0xb705:2000 (46853.125 s)
   -------------------------------
   delay 0x   6:2000 (    6.125 s)
```

- n: sender, r: receiver.
- A: SSRC_n’s recording time when RR sent by SSRC_r is received.
- The round trip propagation delay = A - LSR - DLSR.

Intermedia Synchronization

Synchronization between different media stream (audio, video, white board data,...)

- Be aware that timestamps in different streams offset with different random intervals
- may not tick at nominal rate (due to system overhead)
- Every SR correlates the “real” time (via NTP) with RTP timestamp values are computed when a sample was generated.



Source Description (SDES) RTCP Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    SC   |  PT=SDES=202  |             length            | header
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                          SSRC/CSRC_1                          | chunk
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   1
|                           SDES items                          |
|                              ...                              |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                          SSRC/CSRC_2                          | chunk
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   2
|                           SDES items                          |
|                              ...                              |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

- The SDES packet is a three-level structure composed of a header and zero or more chunks, each of which is composed of items describing the source identified in that chunk. The items are described individually in subsequent sections.
- source count (SC): 5 bits,
  - — The number of SSRC/CSRC chunks contained in this SDES packet.

SDES ITEMS

- 8 bit type field, 8 bit octet count, and text not null terminated.
- list of SDES items in each chunk is terminated by null octets (up to the next 32 bit boundary

```
CNAME: Canonical end-point identifier SDES item
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    CNAME=1    |     length    | user and domain name         ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Because the randomly allocated SSRC identifier may change if a conflict is discovered or if a program is restarted, the CNAME item is required to provide the binding from the SSRC identifier to an identifier for the source that remains constant.
- Like the SSRC identifier, the CNAME identifier should also be unique among all participants within one RTP session.
- To provide a binding across multiple media tools used by one participant in a set of related RTP sessions, the CNAME should be fixed for that participant.
- suggestion name: email address (for multiuser system) ip address or domain name for system with no user name.

SDES items

- Name=2, Common name of the source
- Email=3,
- Phone=4
- Loc=5, Mailing address
- Tool=6, name and version no. of the application that generates the stream
- Note=7
- Priv=8, define experimental or application specific extension

BYE and APP RTCP Packet

- BYE: Goodbye RTCP packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    SC   |   PT=BYE=203  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                              ...                              :
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|     length    |               reason for leaving             ... (opt)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- APP: Application-defined RTCP packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P| subtype |   PT=APP=204  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name (ASCII)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   application-dependent data                ...
```

Problems with RTP
Selecting the Same SSRC Number

- Two sources may pick the same SSRC.
- Detect by correlated CNAME and SSRC.
- Probability is low if you pick the one not used so far
- Send BYE for old SSRC and pick a new ID.

Looping

- Forwarding packets to the same multicast group (via translators)
- Look similar to SSRC selection collision
  but changing SSRC does not help

Not Suitable for Massive Receiving?

- For low speed audio multicast, it may takes long hours to know all the receivers.
  - — useless for QoS feedback
  - — who care about receivers (Nielson!)

RFC 1890: RTP Profile for Audio and Video Conference with Minimal Control

Audio Encoding-Independent Recommendations

- Set the marker bit on the first packet of a talkspurt (after the silence period)
- RTP clock rate = the number of sampling periods per second.
- For N-channel encoding, each sampling period generates N samples.
- channels are numbered left-to-right, starting from one. followed AIFF-C format.

channels	description	channel					
		1	2	3	4	5	6
2	stereo	l	r				
3		l	r	c			
4	quadrophonic	Fl	Fr	Rl	Rr		
4		l	c	r	S		
5		Fl	Fr	Fc	Sl	Sr	
6		l	lc	c	r	rc	S

- Sampling frequencies: 8000,11025, 16000, 22050, 24000, 32000, 44100, 48000

- Apple Macintosh computers have native sampling rates of 22254.54 and 11127.27 can be converted to 22050 and 11025 by dropping 4 or 2 samples in 20 ms frame.
- Default audio packetization interval is 20 ms.
- A receive should accept packets representing between 0 and 200 ms of audio data.

Audio Encoding Formats

encoding	sample/frame	bits/sample	ms/frame	
1016	frame	N/A	30	DOD 4.8kbps CELP3.2
DVI4	sample	4		IMA ADPCM (3 bit)
G721	sample	4		ITU
G722	sample	8		ITU 7kHz, 64kbps
G728	frame	N/A	2.5	16kbps
GSM	frame	N/A	20	European GSM 6.10
L8	sample	8		
L16	sample	16		
LPC	frame	N/A	20	Motorola PARC
MPA	frame	N/A		MPEG-I or II audio
PCMA	sample	8		G711 A-Law
PCMU	sample	8		G711 m-Law
VDVI	sample	var.bit rate version of DVI4		

Payload Type for Standard Audio/Video Encodings

PT	encoding	audio/video	clock rate	channels	bitrates
	name	(A/V)	(Hz)	(audio)	
0	PCMU	A	8000	1	64k
1	1016	A	8000	1	
2	G721	A	8000	1	
3	GSM	A	8000	1	13.2k
4	unassigned	A	8000	1	
5	DVI4	A	8000	1	32k
6	DVI4	A	16000	1	64k
7	LPC	A	8000	1	
8	PCMA	A	8000	1	
9	G722	A	8000	1	
10	L16	A	44100	2	1.4112M
11	L16	A	44100	1	
12	unassigned	A			
13	unassigned	A			
14	MPA	A	90000	(see text)	L3 128k
15	G728	A	8000	1	
16--23	unassigned	A			
24	unassigned	V			
25	CelB	V	90000	SUN Cell-B	
26	JPEG	V	90000	ISO 10918-1/2	

27	unassigned	V		
28	nv	V	90000	XERO PARK nv4
29	unassigned	V		
30	unassigned	V		
31	H261	V	90000	ITU-T
32	MPV	V	90000	MPEG-I/II video
33	MP2T	AV	90000	MPEG-II transport
34--71	unassigned	?		
72--76	reserved	N/A	N/A	N/A
77--95	unassigned	?		
96--127	dynamic	?		

- Audio application should at a minimum be able to send/receive type 0 and 5.
- 90k timestamp frequency is the same as MPEG presentation timestamp frequency. This frequency yield exact integer timestamp increments for
  - HDTV (30 frame rate; 60 field rate)
  - PAL (25 frame rate; 50 field rate)
  - NTSC (29.97 frame rate; 59.94 field rate)
- 90kHz gives sufficiency resolution