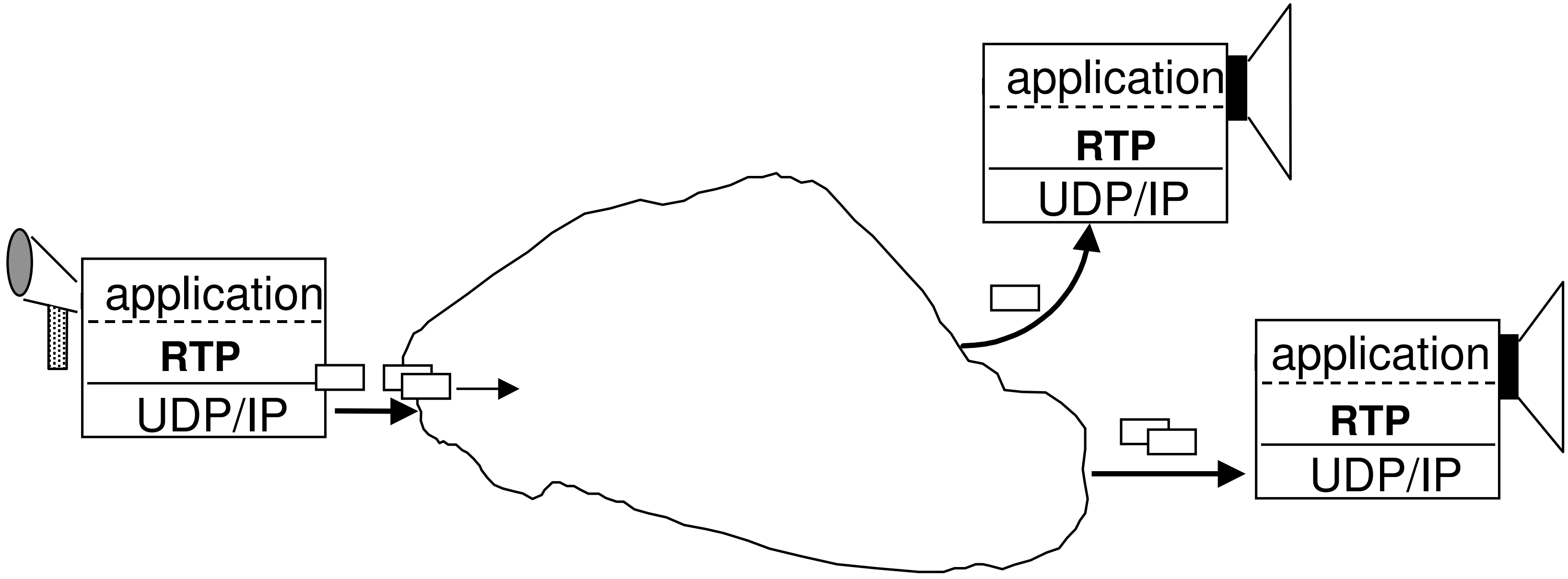
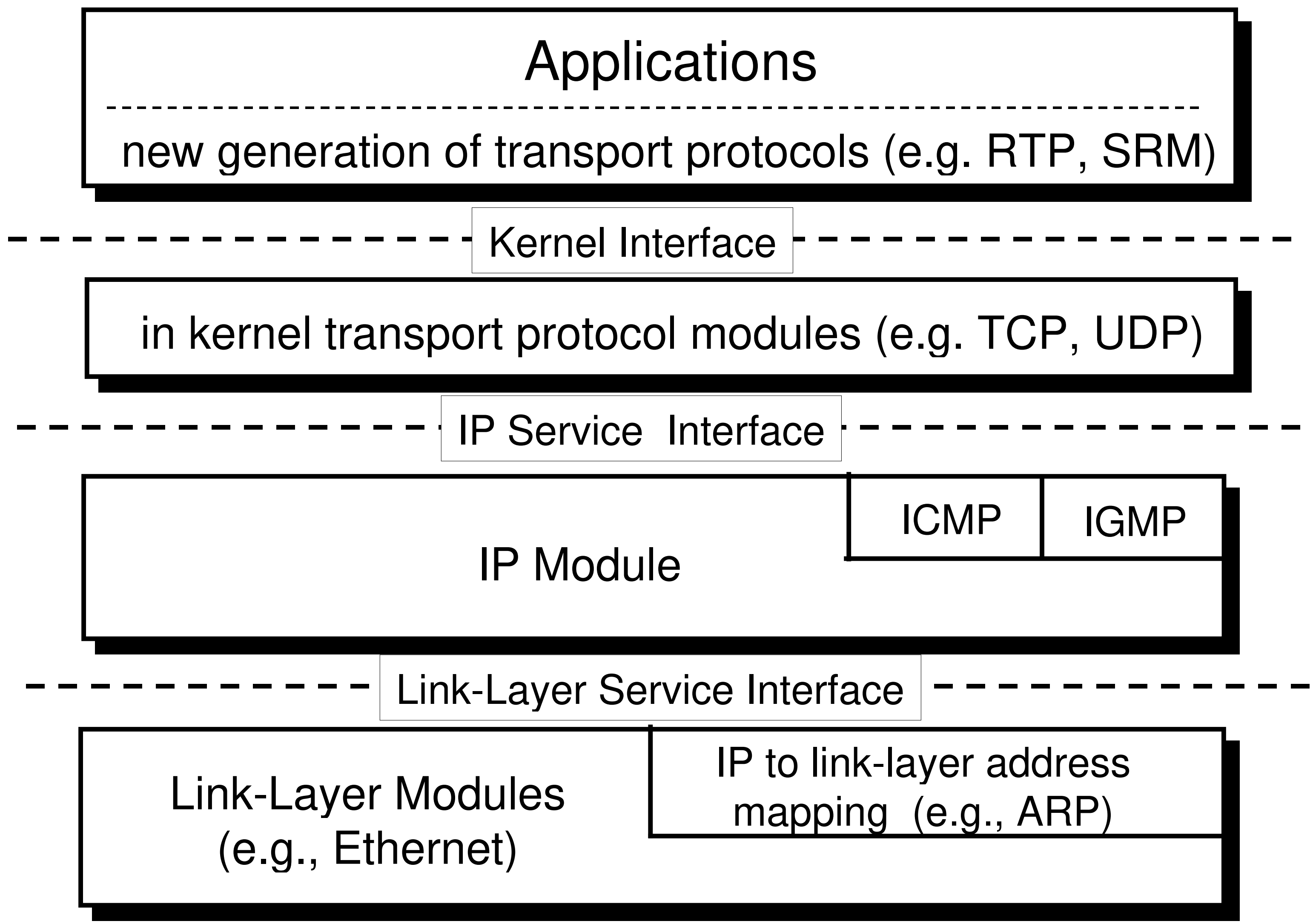
Real-Time Transport: RTP

RTP operation: a basic picture

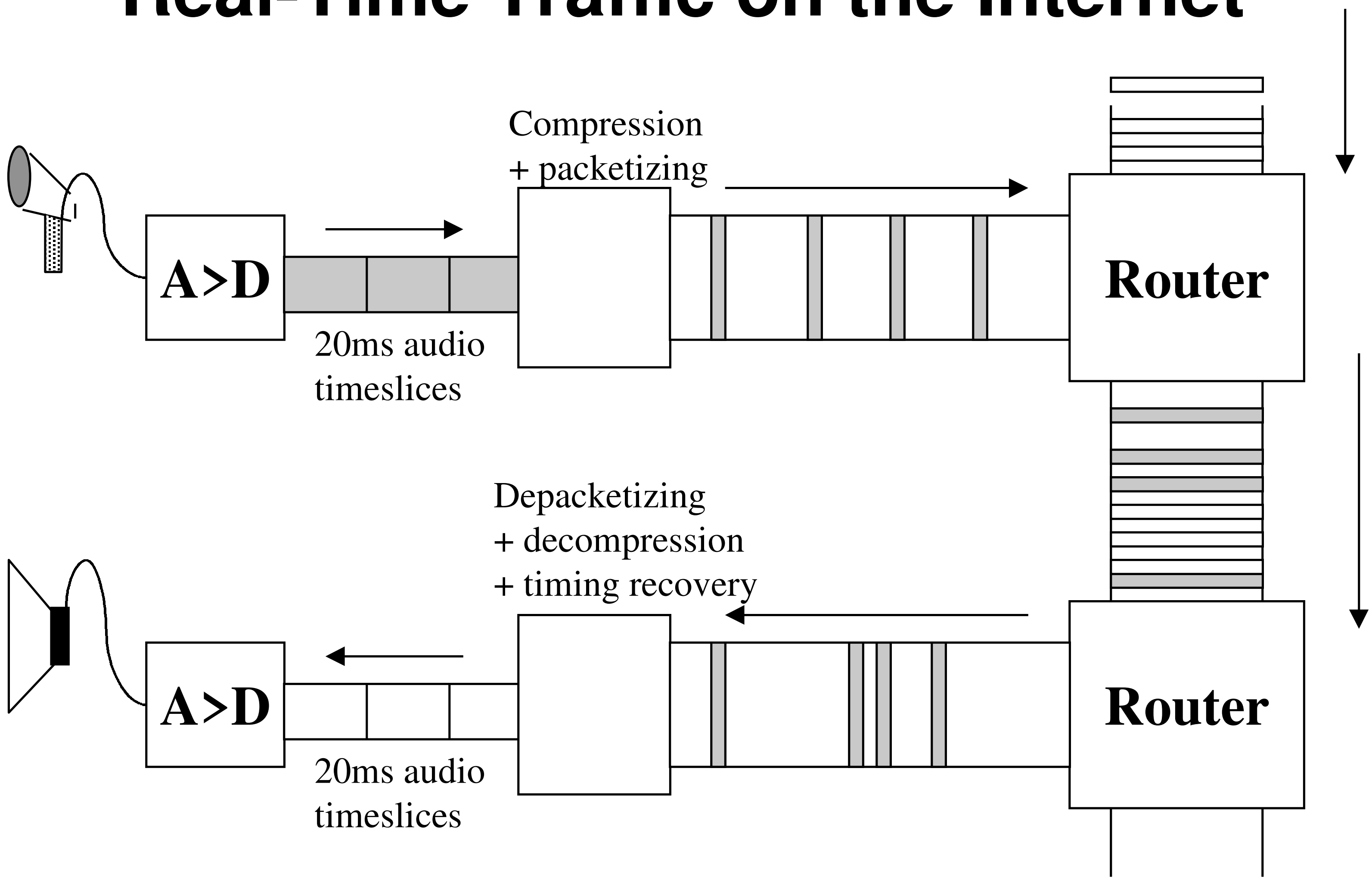
- end-to-end delivery for temporally sensitive data
- supports both unicast & multicast
- may run on top of other network/transport protocols



Protocol Layers Within a Host



Real-Time Traffic on the Internet



What RTP Does

- Data source and payload type identification
- Data sequencing
- Timing and cross-media synchronization
- Delivery monitoring
  - user information, tools for fault diagnosis, feedback on the quality of data distribution
- Providing tools for handling network and host heterogeneity

What RTP Does Not Do

- Network resource reservations
- Quality of service guarantees
- Reliable data delivery
- Multicast address alloction

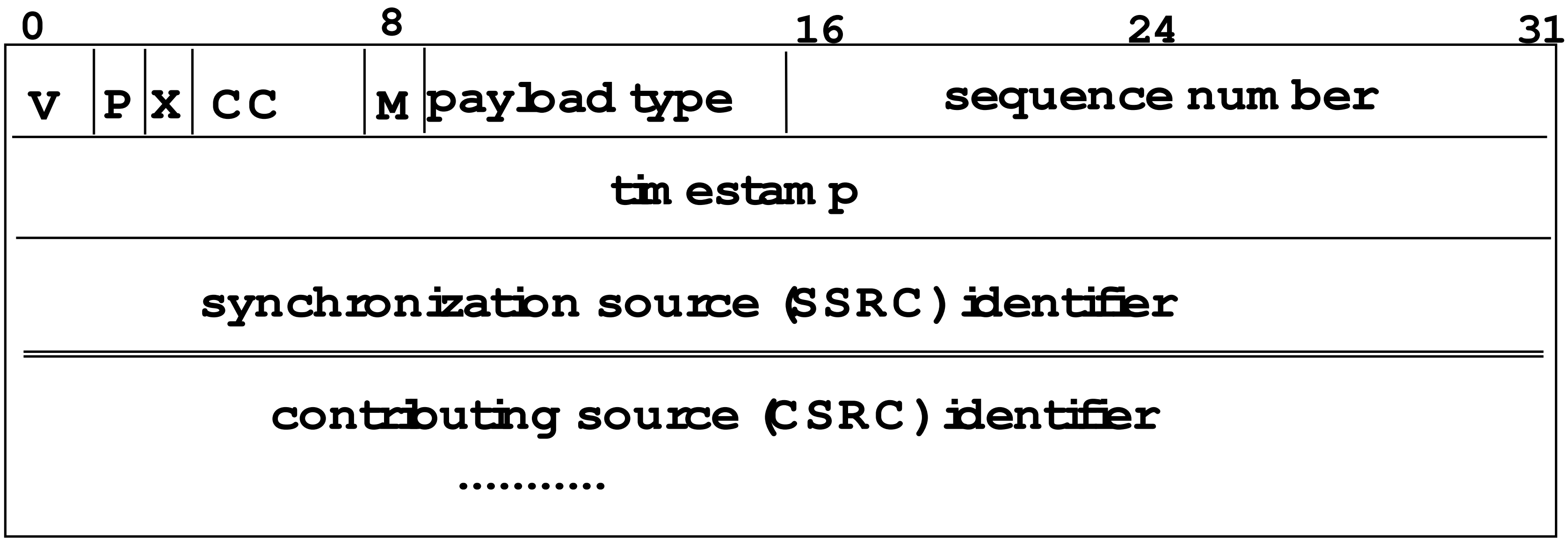
RTP Terminology

- RTP session: defined by a pair of transport addresses

  network address + { port(RTP) must be even#, port(RTCP) next higher # }

- RTP payload: the data transported in a RTP packet
- SSRC: synchronization source, identifies a unique RTP packet sender
  - The packet IP source address does not identify the sender, which means users can move, and transcoders can be employed
- CSRC: contributing source. If streams are mixed, CSRC identifies the sources that were mixed together

RTP Header Format

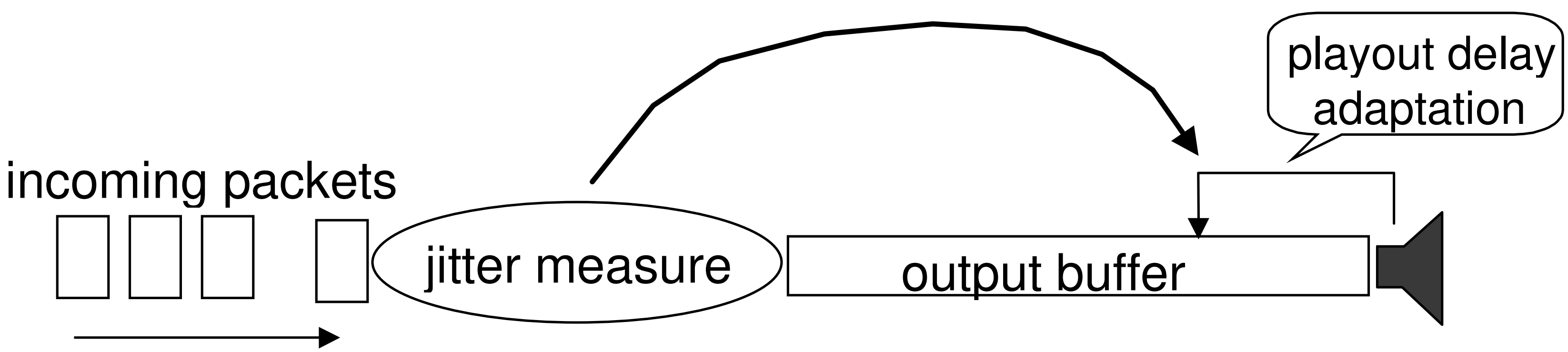


- V: version number
- P: flag to indicate whether any padding bytes present
- X: header extension flag
- CC: CSRC count
- M: marker (media dependent, defined in RTP profile)
- timestamp: sampling instant of the first byte, from media encoding clock

RTP Usage Scenarios

Example 1: An audio conference session

- each packet carries
  - a SSRC
  - a sequence number for data ordering and loss detection
  - a timestamp for inter-packet gap calculation
- Question: how long to hold the data before playing out?



An Audio Conference Session (cont.)

- Playout delay must be long enough so that once a talk spurt starts playout, all data in the spurt will (most likely) be available in time
  - wait time = delay jitter = longest-delay - shortest delay
- wait for at least the jitter time long before playing the first packet in a talk spurt
- play out the whole talk spurt at the medium's clock
  - a real implementation may be more complex
    - e.g. being more conservative or more aggressive based on previous observation

Delay Jitter Measurement

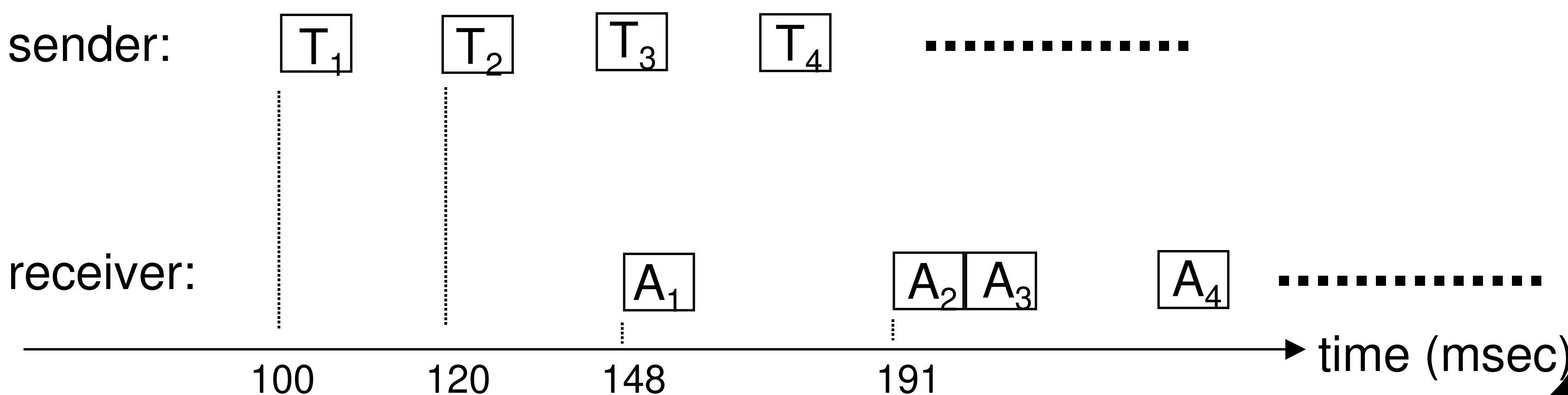
T_i : timestamp of packet i

A_i : packet i's arrival time at receiver

$J_i = | (A_i - A_{i-1}) - (T_i - T_{i-1}) | = |(A_i - T_i) - (A_{i-1} - T_{i-1}) |$

mean jitter: $M_i = (1\text{-}g)\ M_{i-1} + g \times J_i = M_{i-1} + g\ (J_i - M_{i-1})$

an example: $J_2 = | (A_2 - A_1) - (T_2 - T_1) | = 43 - 20 = 23$ msec



RTCP: Realtime Transport Control Protocol

- RTCP SDES messages are sent by all session members
- RTCP SR messages are sent be all data senders
  - provide the relationship between media timestamps and realtime.
- RTCP RR messages are sent by all receivers
  - provide feedback to the senders on reception quality.
- messages can be combined into compound packets
- Their rate depends on the membership

A RTCP Sender Report Message

RTCP header	V=2	P	RC	PT=SR=200	length
	SSRC of RTCP sender				
sender info	NTP timestamp (8 bytes)				
	RTP timestamp				
	sender's packet count				
	sender's byte count				
	Receiver report blocks … … …				

RTCP Receiver Report Message

RTCP header	V	P	RC	PT=SR=200	length
	SSRC of RTCP packet sender				
report1	SSRC_1				
	frac. lost			cumulative number of packets lost	
	extended highest sequence number received				
	inter-arrival jitter				
	last SR timestamp				
	delay since last LSR				
report2	SSRC_2				
	......				

RTCP RR's give

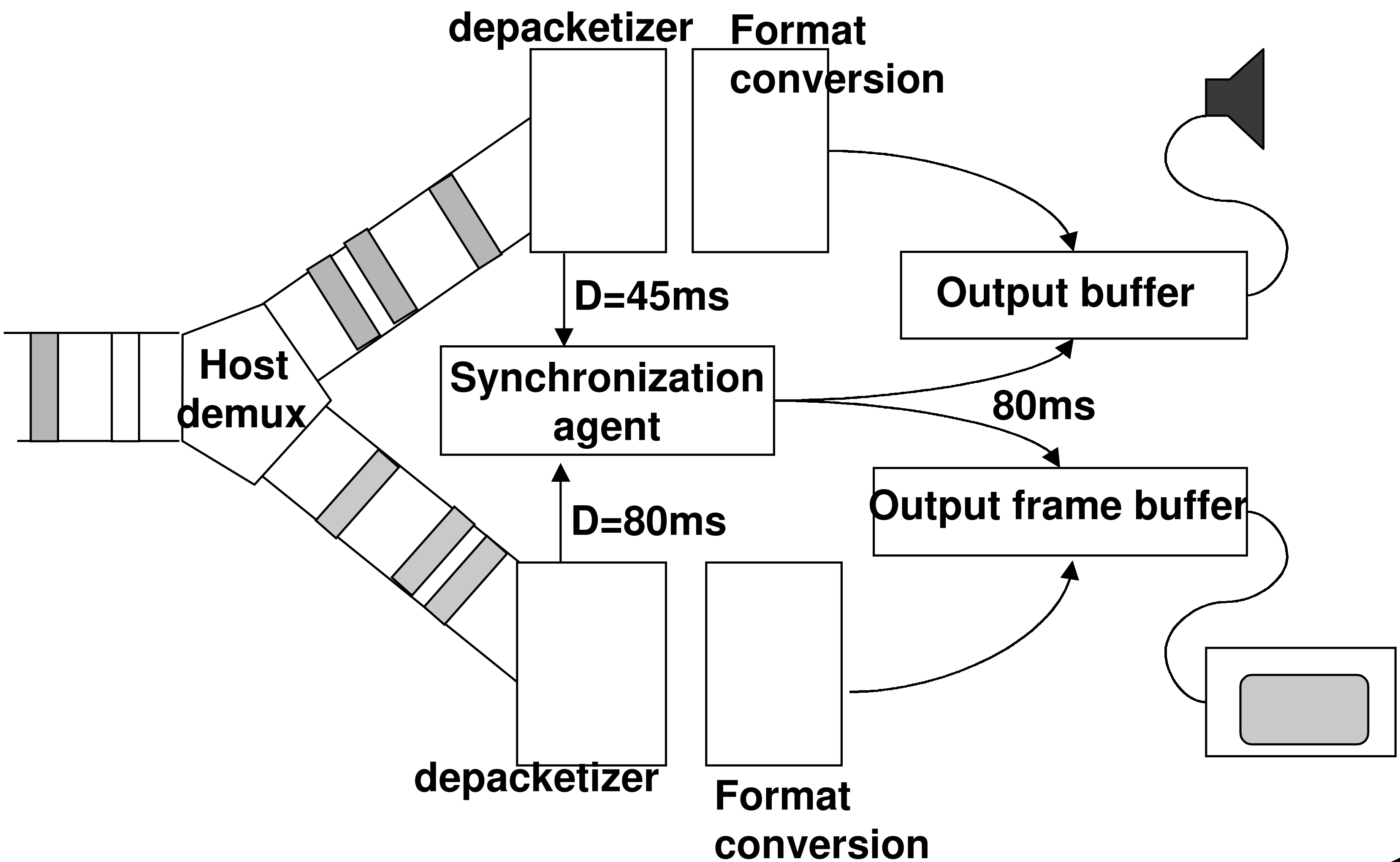
- **loss feedback**
- **jitter feedback**
- **RTT feedback**

RTP Usage Scenario: Inter-stream synchronization

Example 2: An audio & video conference application

- each medium is transmitted in a separate RTP session
  - provides flexibility for participants to receive one stream only
- RTCP provides cross-media clock mapping information
  - sessions from the same participant associated by CNAME
  - once media timestamps are related to realtime, inter-stream sychronization can be performed by adjusting the palyout delays

Inter-stream Synchronization



SSRC Identifier Allocation

- A 32-bit end identifier, unique within the session
- Chosen randomly
  - probability of collision, $P = 1 - \exp(-N^2 / 2^{(L+1)})$
- Collision detection:
  - data packets with same SSRC but different source transport addresses
  - RTCP packets with same SSRC but different CNAMEs
- Collision resolution:
  - receivers may choose to keep packets from one source and drop those from the others
  - sender notices collision and chooses another SSRC value

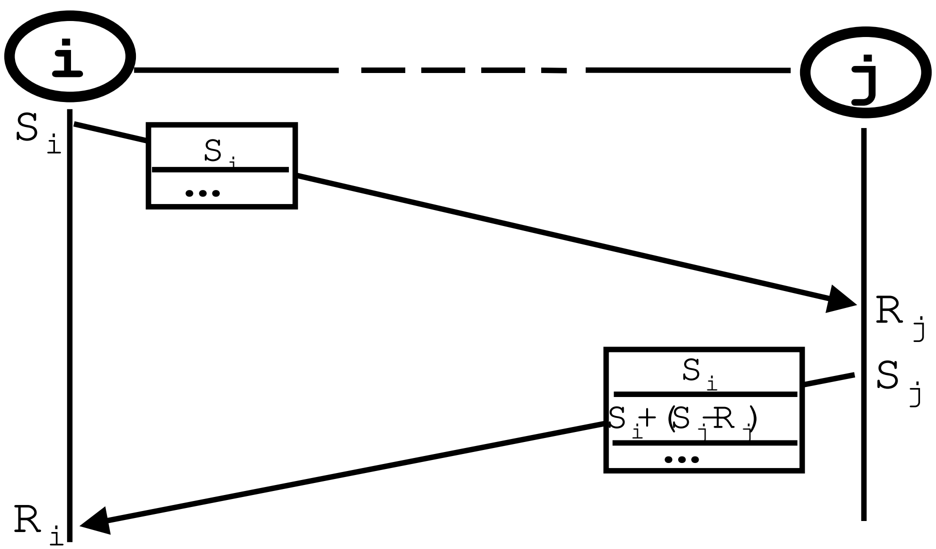
Analyzing Sender and Receiver Reports

Collect information from all participants

- jitter: transient congestion;
  loss: persistent congestion
- facilitate problem diagnosis: global or local?
- help senders decide whether / how to adjust transmissions accordingly
- round-trip-time between participants

Round-Trip Time Estimation among Participants

RTT = arrival-time - timestamp-of-last-report - delay-at-other-site



data record at node J:

	sent	recv	distance
i:	S_i	R_j	d_{ij}

$$T_{ij} = R_i - S_i - (S_j - R_j)$$
$$d_{ij} = T_{ij}/2$$

Scaling RTCP Transmission

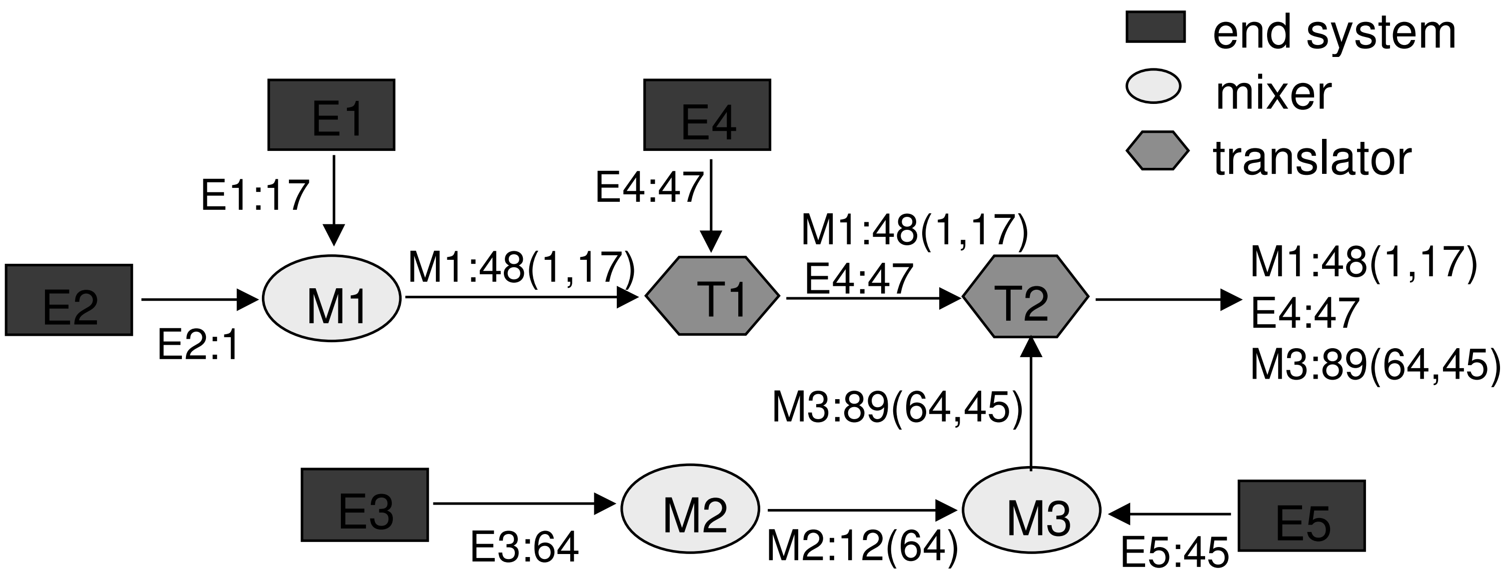
- goal: to limit RTCP's bandwidth consumption to a small fraction of data rate
- means: automatically adjusting RTCP packet interval by participant count
  - interval = avg_RTCP_size x N / RTCP_bandwidth
  - the interval between RTCP packets varies randomly over **[0.5, 1.5]** times the calculated interval
- senders collectively get at least 1/4 of the total
  - allow new members to receive sender information quickly

RTP Translators and Mixers

- translator and mixer: a middleman to facilitate
  - firewall crossing
  - media encoding translation
  - bandwidth reduction for slow links
- Translator: passes data streams of different sources separately
  - keep SSRC's intact
- Mixer: combines all passing streams to one
  - packets carry the mixer's SSRC
  - contributing sources are identified by CSRC list

RTP Translators and Mixers (cont.)

- each RTP entity connects to one translator or mixer
- output data stream from a mixer is identified by the mixer's SSRC

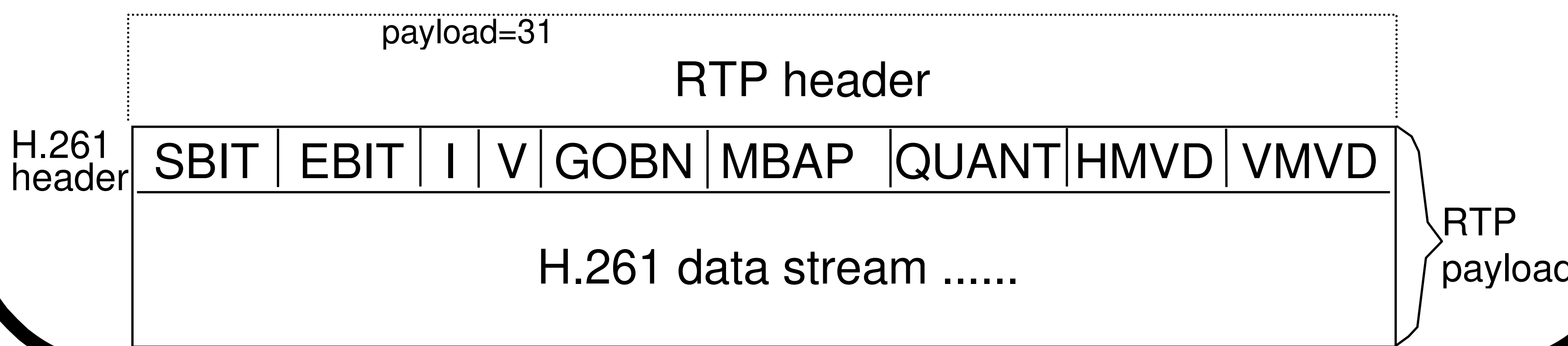


RTP Components II: profile and payload format specifications

- profile: defines interpretation of, and extensions to, RTP header fields that are tailored to specific environment or application classes
- default mapping of payload types to payload formats
  - possible extensions
    - additional fixed fields after SSRC
    - variable length extension after CSRC field
- payload format specification
  - defines data encoding of each specific payload type

An example: using RTP for H.261 video

- payload type = 31, video only, clock rate 90KHz
  - defined in RFC1890, “RTP Profile for Audio and Video Conference with Minimal Control”
- RFC2032 defines RTP payload for H.261
  - marker bit = 1 in the last packet of a video
  - RTP payload format: H.261 header followed by data
  - also defines two H.261-specific RTCP packets



Security Support in RTP

- Support confidentiality through encryption
  - use profile to specify security services and algorithms
  - DES in CBC mode as the default encryption algorithm
- May define additional payload types for encrypted encodings.
- Expect support for authentication and message integrity from lower layers.

RTP Status and Information

- IETF proposed standard; moving to draft standard
  - RFC 1889: RTP protocol specification
  - RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Support
- Audio-Video Transport Working Group
  - mailing list: **`rem-conf@es.net`**
  - Group homepage

    `http://www.ietf.org/html.charters/avt-charter.html`

    (contains pointers to RTP related documents such as RTP profile and payload specifications for various media types)