RTP (and RTCP): the Real-Time Transport Protocol

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 23

'Roadmap' for Multimedia Networking

1. Introduction
   - why QoS?
   - what are the problems?
2. Basic operations
   - jitter buffers (at hosts)
   - task scheduling (at hosts)
   - packet shaping (at hosts)
   - packet dropping (at routers)
   - packet scheduling (at routers)
3. Types of service
   - Integrated Services (IntServ) and Resource Reservation Protocol (RSVP)
   - **Differentiated Services (DiffServ)**

Lecture 17

Lecture 16

Lecture 18

Lecture 19

Lectures 20 and 21

Lecture 22

"Roadmap" (cont'd)

4. Application-level feedback and control
   - Real-time Protocol (RTP), Real-time Control Protocol (RTCP)
   - Real-time Streaming Protocol (RTSP)

Today's Lecture

5. Application signaling and device control
   - Session Announcement Protocol (SAP)
   - Session Description Protocol (SDP)
   - Session Initiation Protocol (SIP)
   - Media Gateway Control Protocol (MGCP)
6. Routing
   - Multi-protocol Label Switching (MPLS)
   - multicasting

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Transporting Multimedia Over IP

- TCP overhead + delay is <u>unacceptable</u> for much multimedia
  - preferable: use UDP for transport
- Problem: UDP is connectionless, unreliable
- RTP: the Real-Time Transport Protocol
  - IETF RFC 1889
- RTP = *application-level framing*
  - application controls recovery, in-order delivery, playback timing
  - synchronization of sender and receiver
  - negotiation between sender and receiver allows application to adapt to changing network or receiver conditions

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Functions

1. Timing reconstruction
   - e.g., control of playout (jitter) buffer
2. Synchronization of different media types
   - e.g., audio + video
3. Packet sequencing
4. Loss detection
5. Content identification
6. QoS feedback and rate adaptation
   - e.g., reduce frame rate if loss rate is too high

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

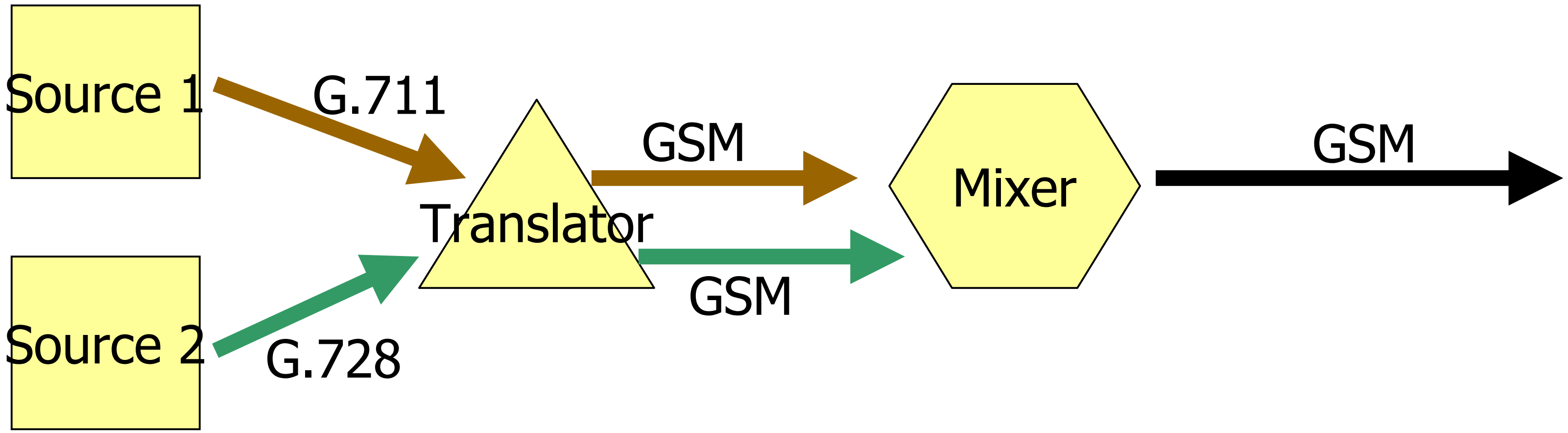
RTP Non-Functions

- It does not…
  - guarantee reliable delivery of packets
  - guarantee QoS

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Translators and Mixers

- Participants in a multimedia session may use different media formats or compression standards
  - a *translator* (also called a *media gateway*) converts from one media format to another
- Audio from multiple senders can be mixed into a single audio stream
  - a *mixer* combines media streams



Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Header

# Bits	Purpose
2	Version field
1	Padding Flag (1 = packet has been "padded", last byte of packet specifies how many padding bytes there are)
1	Extension Flag (1 = extension header follows RTP header
4	Number of Sources mixed (max of 15)
1	Application Marker Bit (indicates frame start, or beginning of "talkspurt")
7	Payload Type (audio or video encoding method)

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Header (cont'd)

# Bits	Purpose
16	Sequence Number (max of 65535); gaps → packet loss
32	Timestamp (format specified by the application)
32	Synchronization Source Identifier (SSRC) original sender of message, or mixer if used "identifiers" are randomly generated; conflicts can be detected and resolved
N*32	Contributor Source Identifiers (CSRC) N = Number of Sources – 1

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Timestamps

- Initial value is random number
- Resolution is payload-dependent, specified as part of the A/V profile standard
  - resolution must be at least = sampling rate of the media type (e.g., 8KHz for compressed voice)
- Timestamp in an RTP packet is time of the first sample in the packet
- Several consecutive RTP packets may have equal Timestamps if they are (logically) generated at once
  - e.g., belong to the same video frame
- Consecutive RTP packets may contain Timestamps that are not monotonic if the data is not transmitted in the order it was sampled
  - e.g., MPEG interpolated video frames

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Ports and Profiles

- A port-pair for RTP and RTCP
  - e.g., 5004 for RTP, 5005 for RTCP
  - each media stream has own RTP connection
- RFC 1990 specifes *A/V Profiles*
  - default Payload Types for common audio and video compression standards
  - mapping of media encoding to payload format
  - default packet rate

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Summary: RTP

- Adds a new header to each packet
  - minimum of 12 bytes long
  - most important info: payload type, sequence number, and timestamp

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTCP – the RTP Control Protocol

- Closely tied to RTP; used for negotiation between senders and receivers
- Reports the quality of the connection between sender and receivers
  - the frequency of reporting between senders and receivers is carefully controlled to prevent excessive overhead
- RTCP message types
  1. Sender Report – current time and amount of data sent so far
  2. Receiver Report – feedback about what has been received so far
  3. Source Description – useful information about the source
  4. Bye – source is disconnecting
  5. Application-specific

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

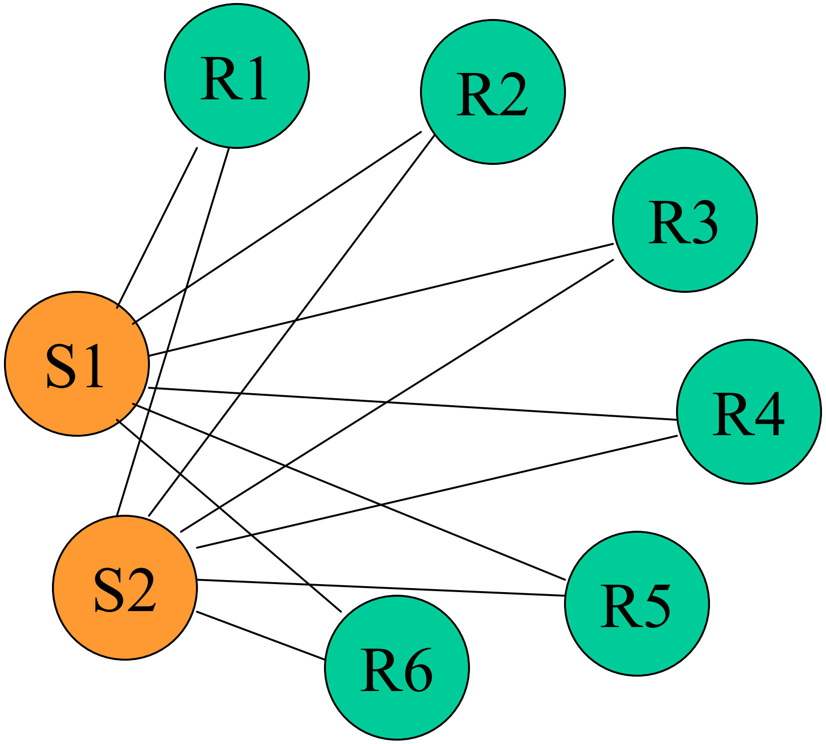
RTCP Sender Report Message

# Bits	Purpose
2+1	Version+Padding Flag
5	Number of Receiver Blocks in packet
8	Packet Type (=200 for sender report)
16	Packet Length
32	Sender SRC ID
64	NTP Timestamp (standardized representation of actual time of day)
32	RTP Timestamp (application specific)
32	Number of Packets Sent (so far)
32	Number of Bytes Sent (so far)
M*24*8	Receiver Report Blocks, M = # of remote sources

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Receiver Report Blocks

- There is one Receiver Report Block for each receiver which is reporting to this sender
- Receiver Report Block confirms feedback from receiver, and also distributes it to <u>other</u> receivers



Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sender Report (cont'd) → Receiver Report Blocks

# Bits	Purpose
32	SSRC of source
8	Fraction of Blocks Lost Since Last Report (relative to 255)
24	Cumulative Number of Packets Lost
32	Highest Packet Sequence Number Received RTP sequence numbers wrap around after 2^{16}-1 this is "extended sequence number"
32	Interarrival Jitter Estimate
32	Time of Last Report From This Receiver NTP format, just the middle 4 bytes
32	Time Since Last Report From This Receiver in units of 1/65536 seconds

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTCP Receiver Report Message

# Bits	Purpose
2+1	Version+Padding Indicator
5	Number of Source Report Blocks in packet
8	Packet Type (=201 for sender report)
16	Packet Length
32	SSRC
M*24	Source Report Blocks, M = # of remote senders

- Source report blocks have same format and info as receiver report blocks
- There is one Source Report Block for each sender about which the receiver is reporting

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTCP Source Description Message

# Bits	Purpose
2+1	Version+Padding Indicator
5	Number of Source Descriptors in packet
8	Packet Type (=202 for source description pkt)
16	Packet Length
M*x	Source Descriptors, M = # of remote sources length (x) is application dependent

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Source Descriptors

- Each Source Descriptor contains
  - Source ID (32 bits)
  - sequence of TLV (tag, length, value)-encoded fields
- Source description fields
  - each TLV must be aligned on 32-bit boundary
  - T and L are 8 bits each
  - some fields have been standardized
    - examples: name, email, phone number, location, …

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Additional RTCP Messages…

- Bye message
  - SSRCs of sources
  - optionally, "reason for leaving" (LV-encoded)
- Application-Specific message
  - allows RTCP extensions easily
- Message combining
  - can put multiple messages into one "compound" packet to save overhead

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTCP Traffic Volume

- RTCP traffic designed to be no more than 5% of the media traffic (RTP)
  - 1.25% allocated to senders, and 3.75% allocated to receivers
  - randomized response; as number of receivers increases, frequence of response per receiver decreases
- Minimum packet transmission frequency is 5 seconds
- (Can also explicitly indicate the bandwidth for RTCP)
  - Using b=RS:xxxx and b=RR:yyyy extension parameters in SDP
    - RS = bandwidth for sender report
    - RR = bandwidth for receiver report (in bits/sec)

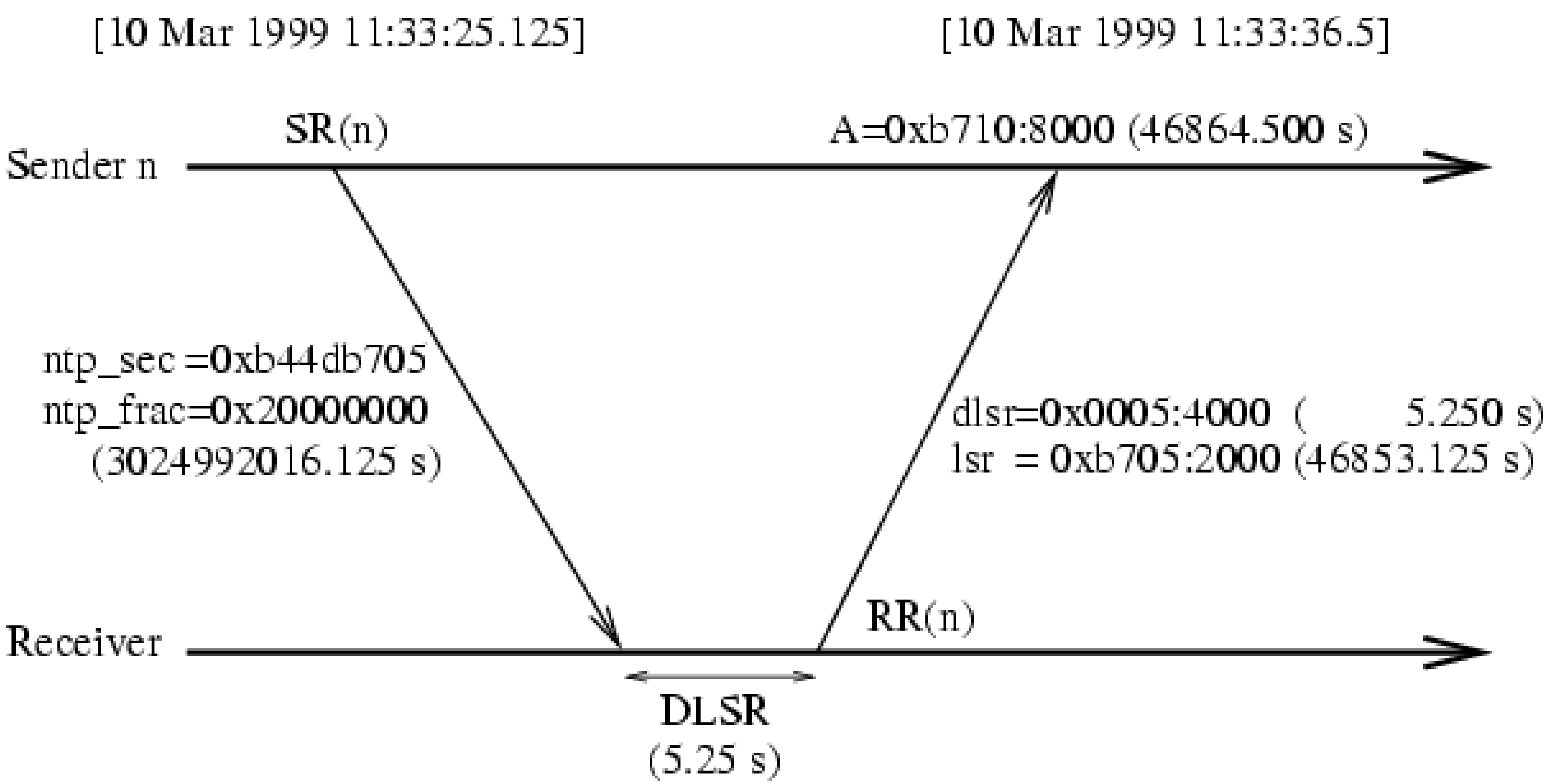
Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP For Multicasting

- RTP designed for multi-party applications
- Feedback from each receiver is multicast to all participants
  - allows easy monitoring of quality by an external entity
  - makes controlling the total amount of feedback to sender from all receivers simpler

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Computing Round-Trip Delay



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)

Copyright

RTP Jitter Calculations

- Calculating jitter (statistical variance of the RTP data interarrival times) to be inserted in the interarrival jitter field of Receiver Reports
  - `s` points to state for the source, `r` points to state for the receiver, `rr` is receiver report
  - `jitter` field of the reception report is integer, jitter estimate is floating point
- Inputs
  - `r->ts` , the timestamp from the incoming packet
  - `arrival` , the current time in the same units
- Outputs
  - `s->transit` holds the relative transit time for the previous packet
  - `s->jitter` holds the estimated jitter

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Jitter Calculations (cont'd)

- As each data packet arrives, jitter is estimated:

```
int transit = arrival - r->ts;
int d = | transit - s->transit |;
s->transit = transit;
s->jitter += (1./16.) * ((double)d - s->jitter);
rr->jitter = (u_int32) s->jitter;
```

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Jitter Calculation Example

Initially, s->transit = 0, s->jitter = 0

arrival = 1000, r->ts = 700
transit = 300
d = 300 – 0 = 300
s->transit = 300
s->jitter = 0 + 1/16*(300-0) = 18.75
rr->jitter = 18

arrival=1800,r->ts=1700
transit = 100
d = 200
s->transit = 100
s->jitter = 18.75 + 1/16*(200 – 18.75) = 30.01
rr->jitter = 30

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

RTP Types for DTMF Digits, Tones, and Events

- Problem: audio compression may distort DTMF digits and other signals beyond safe recognition
  - voice message systems, flashook, etc.
- Rather than transmit as compressed audio, recognize at the sending point and transmit as a well-defined event
  - Must convey duration and volume as part of the payload

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Header Compression: Motivation

- Every voice packet has overhead…
  - IP = 20 bytes
  - UDP = 8 bytes
  - RTP = 12 bytes
  - Total = 40 bytes
- With 20 bytes of payload, 66% of the packet is overhead!
  - this is particularly a problem with slow links (e.g., access network)
- Why not put more payload per packet?
  - more payload bytes = more packetization delay
- Goal: reduce packet overhead
  - transparent to application
  - transparent to IP, UDP, RTP
  - solution: compress at the link layer

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Header Compression: Principles

- Many of the fields in the headers do not change
  - same in packet i, i+1, i+2, …
  - transmit only the first time!
- Many of the fields change by a constant increment between packets
  - transmit the first time, and indicate increment value to use
- Many fields values change by only small amount
  - use differential coding on these fields
- Actual (transmitted) header: index #, and indicator to "use the normal prediction"

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: Identifying Each RTP Session

- Identifying a "session context"
  - IP source and destination addresses
  - UDP source and destination ports
  - RTP session source ID (SSRC)
- Hash the session context to a unique session index #
- Substitute for (IP+UDP+RTP) header…
  - session index #
  - indicator to use "the normal prediction"

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: IP Header

Field	No Change	Fixed Increment	Other
Version	X		
Header Length	X		
TOS	X		
Total Length	X		Redundant, let link layer handle
Identification Number		X	
Flags	X		
Fragment Offset	X		
TTL	X		
Protocol	X		
Header Checksum			Redundant, let link layer handle
Source IP Address	x		
Destination IP Address			

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: UDP Header

Field	No Change	Fixed Increment	Other
Source Port Number	X		
Destination Port Number	X		
UDP Length			Redundant, let link layer handle
UDP Checksum			Optional, but if present cannot be predicted or compressed

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: RTP Header

- No change: version, padding flag, extension flag, number of sources, payload type, SSRC, contributor source identifiers

Field	Fixed Increment	Other
Market Bit		Needed
Sequence Number	X	
Timestampl	X	

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: Increment Encoding

- Default increments are stored in the session context table
- Updated when an explicit value is sent
  - encoding of these values may be negotiated previously

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: Sending Uncompressed Headers

- Used to automatically "refresh" the synchronization of sender and receiver
  - or, explicitly requested by receiver when sychronization is lost
- Contains complete, normal header
  - must add session ID # and sequence number
  - use existing fields!
- Keep a "negative cache" of sessions that won't compress well
  - don't keep trying to compress; just give up!

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

HC: Results

- Amount of compression (typical): down to 2-4 bytes!
- Success rate of prediction (typical): 95-98%!
- How trigger removal from session context table?

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sources of Info

- Books
  - Thomas, “IPNg and the TCP/IP Protocols”, 1996 (chapter 11)
  - Douskalis, “IP Telephony”, 2000 (chapter 2)
  - Davidson, “Voice over IP Fundamentals”, 2000 (chapters 8 and 9)
- Web
  - RTP: A Transport Protocol for Real-Time Applications
  - Compressing TCP/IP Headers for Low-Speed Serial Links

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)