The Multimedia Control Protocol RTCP

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
October 5, 1999

http://www.cs.unc.edu/~jeffay/courses/comp249f99

The Multimedia Transport Protocol RTP

Outline

◆ RTP concepts
  » Entities and abstractions

◆ Protocol definition
  » Header format and packet structure

◆ Developing interoperable applications with RTP
  » RTP profiles

◆ Quality-of-service monitoring and reporting
  » Real-time control protocol RTCP
The Real-Time Control Protocol RTCP

Overview

- Senders & receivers periodically generate reports of various session statistics and multicast to the group
- RTCP enables...
  - Diagnosis of faults in the multicast distribution tree
  - Congestion control
  - Third party performance monitoring & logging
  - (Simple) conference control

The Real-Time Control Protocol RTCP

Message types

- Sender reports (SR)
  - cumulative frame & byte counts
  - wall clock/timestamp values
- Receiver reports (RR)
  - frame loss/Frame delivery rate
- Source description (SDES) items
  - useful ASCII text strings (user & host name of participant, e-mail address, notes, ...)
- “Bye” message
  - used to update participant’s SSRC tables
The Real-Time Control Protocol RTCP

Mechanics

- RTCP messages are “stackable”
  - To amortize header overhead, multiple RTCP messages can be combined and sent in a *compound RTCP message*.

- RTCP messages are always sent in (at least) pairs
  - Messages must always contain a sender/receiver report and a source description message containing the *canonical name* (CNAME) of the participant.

- RTCP messages are sent periodically with a period set to ensure that control messages consume no more than 5% of the session bandwidth
  - Much of the contents of sender & receiver reports are included so that participants can compute the RTCP sending interval.

The Real-Time Control Protocol RTCP

Message encapsulation

![Diagram of RTCP message encapsulation]

- **UDP Message**
  - **IP header**
  - **UDP header**
  - **RTCP header**
  - **Report Block**
    - **RTP/AVP Session Description (SDES) Report**
    - **RTCP Sender/Receiver Report**
    - **Optional Reports**

- **Compound RTCP Message**
  - **RTCP header**
  - **Reception Report**
    - Reception Report \(_1\)
    - Reception Report \(_n\)**
  - **CNAME**
  - **RTCP header**
The Real-Time Control Protocol RTCP

Common sender/receiver report message header

- All report messages have the same 8 byte header
  - version number (same as RTP)
  - padding indicator
  - reception report count (5 bits)
  - RTCP message type (8 bits)
  - RTCP message length (16 bits)
  - SSRC for the sender of this report (32 bits)

The Real-Time Control Protocol RTCP

Sender reports — packet format

- NTP timestamp (two 32-bit words)
- RTP timestamp
- Sender’s cumulative packet count
- Sender’s cumulative byte count
- Reception Report Block 1
- Reception Report Block 2
- :
The Real-Time Control Protocol RTCP

Reception report blocks

- Each sender and receiver report should contain a reception report block for each synchronization source heard from since the last RTCP report.

- Contents:
  - source identifier for the block (SSRC)
  - fraction of RTP packets from this source lost since the last report
  - cumulative number of lost packets
  - extended highest sequence number received
  - estimated average RTP packet interarrival time jitter
  - last SR timestamp received from this source
  - delay since receiving the last SR report from this source

RTCP Reception Report Blocks

Loss calculation

- The number of lost packets is expressed as a fraction

\[
\text{fraction lost} = \frac{\text{number of packets lost}}{\text{number of packets expected}}
\]

- where:

\[
\text{nbr of packets lost} = \text{nbr packets expected} - \text{nbr packets received}
\]
\[
\text{number of packets expected} = \text{EHSNR} - \text{initial sequence number}
\]

\[
\text{EHSNR} = \text{extended highest sequence number received}
\]
\[
= \text{number of sequence number cycles} \times 2^{16}
\]
\[
+ \text{last sequence number received}
\]
Jitter calculation

◆ Interarrival time jitter is an estimate of the statistical variance of RTP data packet interarrival times
  » the smoothed mean absolute value of the difference between the sending interval at a source and the interarrival time at a receiver

\[
\text{jitter}_{\text{new}} = \text{jitter}_{\text{old}} + \frac{\text{instantaneous jitter} - \text{jitter}_{\text{old}}}{16}
\]

\[
\text{instantaneous jitter} = |(\text{rec}_i - \text{rec}_{i-1}) - (\text{sent}_i - \text{sent}_{i-1})|
\]

Round trip time calculation

◆ The last-SR-timestamp received and delay-since-receiving-last-SR-report fields in the reception report block are used to compute an estimate of the round-trip time from the receiver to a synchronization source

\[
\text{estimated round-trip time} = \text{RR received} - \text{SR sent} - \text{delay}
\]

\[
\text{RR received} = \text{time a source received this reception report}
\]

\[
\text{SR sent} = \text{last SR timestamp received field}
\]

\[
\text{delay} = \text{delay since last SR report field}
\]
The Real-Time Control Protocol RTCP

**Receiver reports**

<table>
<thead>
<tr>
<th>v</th>
<th>p</th>
<th>RR count</th>
<th>packet type=201</th>
<th>message length</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSRC of report sender**
- **SSRC of first source heard from**
- **fraction lost**
- **cumulative number of lost packets**
- **extended highest sequence number received**
- **estimate RTP packet interarrival time jitter**
- **timestamp of last SR report received**
- **elapsed time since last SR report received**

**Reception Report 2**

---

The Real-Time Control Protocol RTCP

**Source description item (SDES) messages**

<table>
<thead>
<tr>
<th>v</th>
<th>p</th>
<th>chunk cnt</th>
<th>packet type=202</th>
<th>message length</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSRC or first CSRC**

<table>
<thead>
<tr>
<th>SDES type</th>
<th>SDES length</th>
<th>SDES item</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSRC or second CSRC**

<table>
<thead>
<tr>
<th>SDES type</th>
<th>SDES length</th>
<th>SDES item</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**An SDES message consists of one or more “chunks” of source description items**

- **CNAME** (*user@host*)
- **PHONE**
- **NOTE**
- **NAME**
- **LOC**
- **EMAIL**
- **TOOL**
Source Description Item (SDES) Messages

BYE message

The BYE message is used by participants to update their SSRC tables
» participants can optionally say why they are leaving
  (a cheap way of reporting errors in real-time)

The Real-Time Control Protocol RTCP

Scalability issues

- RTP data transmission is inherently scalable
- RTCP message transmissions are not!
  » RTCP message bandwidth must be controlled
- Generic RTCP transmission guidelines:
  » RTCP messages should consume no more than 5% of session bandwidth
  » 25% of RTCP bandwidth should be allocated to senders
- The challenge: Each session participant must independently compute an RTCP sending interval that ensures transmission guidelines are met
Computing the RTCP Sending Interval

General principles

- Participants keep a running estimate of the session size
- For a given session bandwidth, session size, and RTCP packet size, compute a transmission delay
- Randomize the delay to avoid synchronization effects
  » Randomize uniformly in the range \([0.5, 1.5] \times \text{computed delay}\)
- As other participants join and leave the group, adjust the delay accordingly

Computing the RTCP Sending Interval

Algorithm

- Case 1: \textit{number of senders} < 25\% of session membership
  » Compute the average expected RTCP message interarrival time

\[
\begin{align*}
\text{average IAT of} & \quad \text{sender reports} = \frac{\text{average RTCP packet size}}{0.25 \times \text{RTCP bandwidth}} \\
\text{average IAT of} & \quad \text{receiver reports} = \frac{\text{average RTCP packet size}}{0.75 \times \text{RTCP bandwidth}}
\end{align*}
\]

» Average transmission delay is

\textit{number of peers} \times \text{average SR/RR IAT}
Computing the RTCP Sending Interval Algorithm

» **Case 1:** *number of senders < 25% of session membership*
  - Compute the “deterministic interval” \( T_d = \text{MAX}(T_{\text{min}}, \text{ave delay}) \)
    - where \( T_{\text{min}} = 2.5 \text{ secs} \) if the session is starting, and \( T_{\text{min}} = 5 \text{ secs} \) otherwise
  - Then delay for a duration \( T = \frac{(0.5 + \text{random()}) \times T_d}{e - 1.5} \)

Computing the RTCP Sending Interval Algorithm

» **Case 2:** *number of senders > 25% of session membership*
  - Average expected RTCP message interarrival time is simply
    \[\frac{\text{average RTCP packet size}}{\text{RTCP bandwidth}}\]
  - The deterministic interval is \( T_d = \text{MAX}(T_{\text{min}}, n \times \text{ave RTCP IAT}) \)
    - where \( n \) is the total number of session participants
  - \( T_{\text{min}} \) and the computed delay \( T \) are as before
Computing the RTCP Sending Interval

Example

- For a 32 kbps DVI4 audio conference with 100 participants
  » senders transmit every

\[
\text{number of senders} \times \frac{\text{average RTCP packet size}}{0.25 \times \text{RTCP bandwidth}} = \frac{5 \times 100 \text{ bytes} \times 8 \text{ bits/byte}}{0.25 \times 32 \text{ kbps} \times 0.05} = 10 \text{ secs}
\]

» receivers transmit every

\[
\text{number of receivers} \times \frac{\text{average RTCP packet size}}{0.75 \times \text{RTCP bandwidth}} = \frac{95 \times 100 \text{ bytes} \times 8 \text{ bits/byte}}{0.75 \times 32 \text{ kbps} \times 0.05} = 63 \text{ secs}
\]

RTCP Scalability Issues

BYE floods

Cumulative Number of RTCP Packets Sent

Packets sent with “timer reconsideration”

Session ends

Time (secs)
RTCP Scalability Issues
Timer reconsideration

- We want to slow down the RTCP transmission process when the session size is growing and speed up the process when the session size is shrinking

- Timer reconsideration:
  - While waiting to send an RTCP message, update session size estimate
  - When transmission timer expires, recompute $T$. If
    \[
    \text{time of last RTCP transmission} + T \leq \text{current time}
    \]
    then transmit an RTCP message and calculate another delay,
  else, reschedule transmission for time
    \[
    \text{time of last RTCP transmission} + T
    \]

RTCP Scalability Issues
Reverse reconsideration

- On a receipt of a BYE message, if session size has decreased since last RTCP delay was computed, then next message will be sent at time:
  \[
  \text{current time} + \frac{\text{new session size}}{\text{previous session size}} \times \left( \frac{\text{next RTCP transmission time} - \text{current time}}{\text{latest RTCP transmission time} - \text{current time}} \right)
  \]
  - We also update the time of the last (logical) RTCP transmission

\[
\text{current time} - \frac{\text{new session size}}{\text{previous session size}} \times \left( \frac{\text{current time} - \text{time of last RTCP transmission}}{\text{latest RTCP transmission time} - \text{current time}} \right)
\]
On a receipt of a BYE message, if session size has decreased since last RTCP delay was computed, then next message will be sent at time

\[
\text{current time} + \frac{\text{new session size}}{\text{previous session size}} \times \left( \text{next RTCP transmission time} - \text{current time} \right)
\]

» We also update the time of the last (logical) RTCP transmission

\[
\text{current time} - \frac{\text{new session size}}{\text{previous session size}} \times \left( \text{current time} - \text{time of last RTCP} \right)
\]

For BYE message we “splurge” and allow bursts of RTCP messages

» Set session size to 1, average packet size to BYE message size and compute \( T \) as before

» Only increment the session size on receipt of BYE messages

In the worst case BYE messages consume 5% of session bandwidth

» Other RTCP traffic consumes 5% of session bandwidth…

» Hence worst case is an additional 5% of session bandwidth consumed
Translators that do not modify RTP data packets typically will not modify RTCP packets.

Translators that modify data packets must modify RTCP packets to so that the reported statistics reflect the performance of the modified stream.

---

**The Real-Time Control Protocol RTCP**

**Sender reports**

<table>
<thead>
<tr>
<th>v=2</th>
<th>RR count</th>
<th>packet type=200</th>
<th>message length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSRC of report sender**
- **NTP timestamp** (two 32-bit words)
- **RTP timestamp**
- **Sender’s cumulative packet count**
- **Sender’s cumulative byte count**
- **Reception Report Block 1**
- **Reception Report Block 2**
- ...

---

For the diagram, the flow of data and messages is depicted as follows:

- **RTP data** flows from the source to the translator.
- **SR messages** flow from the translator to the receiver.
- **RR messages** flow from the receiver back to the translator.

---

**The Real-Time Control Protocol RTCP**

RTCP message processing in translators & mixers
The Real-Time Control Protocol RTCP

Receiver reports

<table>
<thead>
<tr>
<th>v = 2</th>
<th>p</th>
<th>RR count</th>
<th>packet type=201</th>
<th>message length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SSRC of report sender

SSRC of first source heard from

fraction lost
cumulative number of lost packets

extended highest sequence number received

estimate RTP packet interarrival time jitter

timestamp of last SR report received

elapsed time since last SR report received

Reception Report 2

The Real-Time Control Protocol RTCP

RTCP message processing in translators & mixers

- Since mixers are synchronization sources, they generate their own RCTP packets
  - Mixers generate SR sender information in exactly the same way sources do
  - Mixers generate reception report blocks for sources in exactly the same way receivers do

- But all messages are only sent to one “cloud”