Java Media Framework: RTP

Summary:

❑ RTP
  ○ RTP/RTCP Basics
  ○ Scenarios
❑ JMF RTP Implementation
  ○ Reception
  ○ Transmission
  ○ Code Examples

Sources:

❑ JMF 2.0 API Programmers Guide from Sun:
    http://java.sun.com/products/java-media/jmf/2.1/guide/

❑ RTP (RFC 1889) :
    http://data.uta.edu/~ramesh/multimedia/rfc1889.html

What is RTP?

❑ RTP is the Internet-standard protocol for the transport of real-time data, including audio and video. It can be used for media-on-demand as well as interactive services such as Internet telephony.

❑ RTP consists of a data and a control part. The latter is called RTCP.

  ○ The data part of RTP is a thin protocol providing support for applications with real-time properties such as
    • continuous media (e.g., audio and video),
    • timing reconstruction,
    • loss detection,
    • security and content identification.

  ○ RTCP provides support for real-time conferencing of groups of any size within an internet. This support includes:
    • Source identification and support for gateways like audio and video bridges
    • Multicast-to-unicast translators.
    • Quality-of-service feedback from receivers to the multicast group
    • Support for the synchronization of different media streams.

Basis for RTP

❑ RTP was designed around the concept of **Application Level Framing** (ALF), first described by Clark and Tennenhouse. The key argument underlying ALF is that there are many different ways an application might be able to cope with misordered or lost packets. These range from
  ○ ignoring the loss,
  ○ re-sending the missing data (either from a buffer or by regen-erating it), and
  ○ sending new data which supersedes the missing data.

❑ The application only has this choice if the transport protocol is dealing with data in ``Application Data Units'' (ADUs).
  ○ *An ADU contains data that can be processed out-of-order with respect to other ADUs.*

Basis for RTP

- ❑ The key property of a transport protocol for ADUs is that each ADU contains sufficient information to be processed by the receiver immediately.
    - ○ An example is a video stream, wherein the compressed video data in an ADU must be capable of being decompressed regardless of whether previous ADUs have been received. Additionally the ADU must contain ``header'' information detailing its position in the video image and the frame from which it came.
- ❑ Although an ADU need not be a packet, there are many applications for which a packet is a natural ADU. Such ALF applications have the great advantage that all packets that are received can be processed by the application immediately.
- ❑ The ALF philosophy only holds good if the RTP payload formats are also designed using an ALF philosophy. This implies
    - ○ smart, network aware end-points

RTP Use Scenarios

Scenario 1: *Simple Multicast Audio Conference*

- ❑ A working group of the IETF meets to discuss the latest protocol draft, using the IP multicast services of the Internet for voice communications.
- ❑ Through some allocation mechanism the working group chair obtains a multicast group address and pair of ports.
    - ○ One port is used for audio data, and the other is used for control (RTCP) packets. This address and port information is distributed to the intended participants.
- ❑ The audio conferencing application used by each conference participant sends audio data in small chunks of, say, 20 ms duration. Each chunk of audio data is preceded by an RTP header;

Scenario 1 (Contd.)

- ❑ RTP header and data are in turn contained in a UDP packet. The RTP header indicates what type of audio encoding (such as PCM, ADPCM or LPC) is contained in each packet so that senders can change the encoding during a conference,
    - ○ for example, to accommodate a new participant that is connected through a low-bandwidth link or react to indications of network congestion.
- ❑ The Internet, like other packet networks, occasionally loses and reorders packets and delays them by variable amounts of time.
    - ○ To cope with these impairments, the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source,
    - ○ In the example, chunks of audio are contiguously played out the speaker every 20 ms.

Scenario 1 (Contd.)

❒ Timing reconstruction is performed separately for each source of RTP packets in the conference.

❒ The sequence number can also be used by the receiver to estimate how many packets are being lost.

❒ Since members of the working group join and leave during the conference, it is useful to know who is participating at any moment and how well they are receiving the audio data.
   ○ Each instance of the audio application in the conference periodically multicasts a reception report plus the name of its user on the RTCP (control) port.
   ○ The reception report indicates how well the current speaker is being received and may be used to control adaptive encodings.

❒ In addition to the user name, other identifying information may also be included subject to control bandwidth limits.

❒ A site sends the RTCP BYE packet when it leaves the conference.

RTP Use Scenarios

Scenario 2: *Audio and Video Conference*

❒ If both audio and video media are used in a conference, they are transmitted as separate RTP sessions RTCP packets are transmitted for each medium using two different UDP port pairs and/or multicast addresses.

❒ There is no direct coupling at the RTP level between the audio and video sessions, except that a user participating in both sessions should use the same distinguished (canonical) name in the RTCP packets for both, so that the sessions can be associated.

❒ One motivation for this separation is to allow some participants in the conference to receive only one medium if they choose. Despite the separation, synchronized playback of a source's audio and video can be achieved using timing information carried in the RTCP packets for both sessions.

RTP Use Scenarios

Scenario 3: *Mixers and Translators*

❒ Previous scenarios assumed that all sites want to receive media data in the same format.
   ○ Consider the case where participants in one area are connected through a low-speed link to the majority of the conference participants who enjoy high-speed network access.

❒ Instead of forcing everyone to use a lower-bandwidth, reduced-quality audio encoding, an RTP-level relay called a *mixer* may be placed near the low-bandwidth area.
   ○ The mixer resynchronizes incoming audio packets to reconstruct the constant 20 ms spacing generated by the sender
   ○ mixes these reconstructed audio streams into a single stream,
   ○ translates the audio encoding to a lower-bandwidth one and forwards the lower-bandwidth packet stream across the low-speed link.
   ○ These packets might be unicast to a single recipient or multicast on a different address to multiple recipients.

❒ The RTP header includes a means for mixers to identify the sources that contributed to a mixed packet so that correct talker indication can be provided at the receivers.

Scenario 3: Translator

- ❏ Some of the intended participants in the audio conference may be connected with high bandwidth links but might not be directly reachable via IP multicast.
  - ○ For example, they might be behind an application-level firewall that will not let any IP packets pass.
- ❏ For these sites, mixing may not be necessary, in which case another type of RTP-level relay called a translator may be used.
- ❏ Two translators are installed, one on either side of the firewall, with the outside one tunneling all multicast packets received through a secure connection to the translator inside the firewall.
- ❏ The translator inside the firewall sends them again as multicast packets to a multicast group restricted to the site's internal network.

JMF RTP API
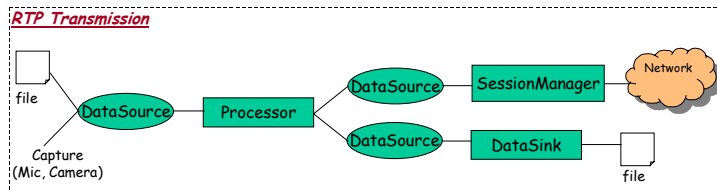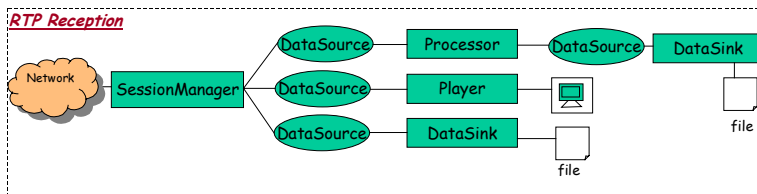
JMF RTP Reception

- ❏ **Reception**
  - ❏ The presentation of an incoming RTP stream is handled by a `Player`.
  - ❏ To receive and present a single stream from an RTP session, you can use a `MediaLocator` that describes the session to construct a Player. A media locator for an RTP session is of the form:

    rtp://address:port[:ssrc]/content-type/[ttl]

  - ❏ The `Player` is constructed and connected to the first stream in the session.
  - ❏ If there are multiple streams in the session that you want to present, you need to use a `SessionManager`. You can receive notification from the session manager whenever a stream is added to the session and construct a Player for each new stream.
  - ❏ Using a session manager also enables you to directly monitor and control the session.

JMF RTP Transmission

- **Transmission**

  A session manager can also be used to initialize and control a session so that you can stream data across the network. The data to be streamed is acquired from a Processor.

  For example, to create a send stream to transmit data from a live capture source, you would:

  1. Create, initialize, and start a `SessionManager` for the session.
  2. Construct a `Processor` using the appropriate capture `DataSource`.
  3. Set the output format of the `Processor` to an RTP-specific format. An appropriate RTP packetizer codec must be available for the data format you want to transmit.
  4. Retrieve the output `DataSource` from the `Processor`.
  5. Call `createSendStream` on the session manager and pass in the `DataSource`.

JMF RTP Transmission(Contd.)

- You control the transmission through the `SendStream` start and stop methods.
- When it is first started, the `SessionManager` behaves as a receiver (sends out RTCP receiver reports).
- As soon as a `SendStream` is created, it begins to send out RTCP sender reports and behaves as a sender host as long as one or more send streams exist.
- If all `SendStream`s are closed (not just stopped), the session manager reverts to being a passive receiver.