Lecture15

Java Media Framework IV

Processing Individual Frames

- The JMF's BufferToImage and ImageToBuffer classes can be used to obtain frame images from a video and to construct a video from a sequence of images respectively.
- In obtaining a frame from a video, we first need to construct a realized player and then obtain FramePositioningControl object to seek the player at the required frame.
- Also a FrameGrabbingControl object is needed to obtain the frame in the form of a Buffer object.
  - Note that the following setHint() method of Manager is needed to obtain FrameGrabbingControl object from a Player.
  - Manager.setHint(Manager.PLUGIN_PLAYER, new Boolean(true));
- After BufferToImage object is created with a correct VideoFormat object, the createImage(Buffer b) method of BufferToImage can be used to convert the Buffer object to a java.awt.Image object.

Getting the Grabber and Positioner

```
Manager.setHint(Manager.PLUGIN_PLAYER, new Boolean(true));
Player p = Manager.createRealizedPlayer(videoSource);
FramePositioningControl positioner = (FramePositioningControl)
    p.getControl("javax.media.control.FramePositioningControl");
if (positioner == null) {
    System.err.println("Error in Getting Positioning Control");
    System.exit(1);
}
FrameGrabbingControl grabber = (FrameGrabbingControl)
    p.getControl("javax.media.control.FrameGrabbingControl");
if (grabber == null) {
    System.err.println("Error in Getting Grabber Control");
    System.exit(1);
}
```

Rendering the Frames

```
public void render(Graphics2D g2d) {
    Buffer buffer; VideoFormat vFormat; BufferToImage bToImage;
    Image captured;
    for (int i = 1;i < NUMBER_OF_FRAMES;i++) {
          positioner.seek(i);
          buffer = grabber.grabFrame();
          vFormat = (VideoFormat) buffer.getFormat();
          bToImage = new BufferToImage(vFormat);
          captured = bToImage.createImage(buffer);
          g2d.drawImage(captured,0,0,null);
          // Wait if needed
          synchronized (this) {
             try {
                    wait(100);
             } catch (InterruptedException ie) {}
          }
    }
}
```

Multimedia Communications and JMF

- When media content is streamed to a client in real-time, the client can begin to play the stream without having to wait for the complete stream to download.
- The term streaming media is often used to refer to both this technique of delivering content over the network in real-time and the real-time media content that's delivered.
- Transmitting media data across the net in real-time requires high network throughput. It's easier to compensate for lost data than to compensate for large delays in receiving the data.
- This is very different from accessing static data such as a file, where the most important thing is that all of the data arrive at its destination.
- The HTTP and FTP protocols are based on the Transmission Control Protocol (TCP).
  - TCP is a transport-layer protocol1 designed for reliable data communications on low-bandwidth, high-error rate networks. When a packet is lost or corrupted, it's retransmitted. The overhead of guaranteeing reliable data transfer slows the overall transmission rate.
- For this reason, underlying protocols other than TCP are typically used for streaming media. One that's commonly used is the User Datagram Protocol (UDP).
  - UDP is an unreliable protocol; it does not guarantee that each packet will reach its destination. There's also no guarantee that the packets will arrive in the order that they were sent.
  - Like TCP, UDP is a general transport-layer protocol–a lower-level networking protocol on top of which more application-specific protocols are built

Real-Time Transport Protocol (RTP)

- The Internet standard for transporting real-time data such as audio and video is the Real-Time Transport Protocol (RTP).
  - RTP provides end-to-end network delivery services for the transmission of real-time data.
  - RTP is network and transport-protocol independent, though it is often used over UDP.
  - RTP can be used over both unicast and multicast network services.
    - Over a unicast network service, separate copies of the data are sent from the source to each destination.
    - Over a multicast network service, the data is sent from the source only once and the network is responsible for transmitting the data to multiple locations. Multicasting is more efficient for many multimedia applications, such as video conferences. The standard Internet Protocol (IP) supports multicasting.

RTP Services

- RTP enables you to identify the type of data being transmitted, determine what order the packets of data should be presented in, and synchronize media streams from different sources.
- RTP data packets are not guaranteed to arrive in the order that they were sent–in fact, they're not guaranteed to arrive at all. It's up to the receiver to reconstruct the sender's packet sequence and detect lost packets using the information provided in the packet header.
- While RTP does not provide any mechanism to ensure timely delivery or provide other quality of service guarantees, it is augmented by a control protocol (RTCP) that enables you to monitor the quality of the data distribution. RTCP also provides control and identification mechanisms for RTP transmissions.

RTP Architecture

- An RTP session is an association among a set of applications communicating with RTP. A session is identified by a network address and a pair of ports. One port is used for the media data and the other is used for control (RTCP) data.
- A participant is a single machine, host, or user participating in the session. Participation in a session can consist of passive reception of data (receiver), active transmission of data (sender), or both.
- Each media type is transmitted in a different session. For example, if both audio and video are used in a conference, one session is used to transmit the audio data and a separate session is used to transmit the video data.
  - This enables participants to choose which media types they want to receive–for example, someone who has a low-bandwidth network connection might only want to receive the audio portion of a conference.

Data Packets

- The media data for a session is transmitted as a series of packets.
- A series of data packets that originate from a particular source is referred to as an RTP stream.
- Each RTP data packet in a stream contains two parts, a structured header and the actual data (the packet's payload).

Control Packets

- In addition to the media data for a session, control data (RTCP) packets are sent periodically to all of the participants in the session.
- RTCP packets can contain information about the quality of service for the session participants, information about the source of the media being transmitted on the data port, and statistics pertaining to the data that has been transmitted so far.
- RTCP packets are "stackable" and are sent as a compound packet that contains at least two packets, a report packet and a source description packet.
  - The first packet in a compound RTCP packet has to be a report packet, even if no data has been sent or received–in which case, an empty receiver report is sent.

Control Packets

There are several types of RTCP packets:

- Sender Report – A participant that has recently sent data packets issues a sender report. The sender report (SR) contains the total number of packets and bytes sent as well as information that can be used to synchronize media streams from different sessions.
- Receiver Report - Session participants periodically issue receiver reports for all of the sources from which they are receiving data packets. A receiver report (RR) contains information about the number of packets lost, the highest sequence number received, and a timestamp that can be used to estimate the round-trip delay between a sender and the receiver.
- Source Description – All compound RTCP packets must include a source description (SDES) element that contains the canonical name (CNAME) that identifies the source. Additional information might be included in the source description, such as the source's name, email address, phone number, geographic location, application name, or a message describing the current state of the source.
- Bye - When a source is no longer active, it sends an RTCP BYE packet. The BYE notice can include the reason that the source is leaving the session.
- Application-specific

RTP Applications

- RTP applications are often divided into those that need to be able to receive data from the network (RTP Clients) and those that need to be able to transmit data across the network (RTP Servers).
- Some applications do both–for example, conferencing applications capture and transmit data at the same time that they're receiving data from the network.

Receiving Media Streams From the Network

- Being able to receive RTP streams is necessary for several types of applications. For example:
  - Conferencing applications need to be able to receive a media stream from an RTP session and render it on the console.
  - A telephone answering machine application needs to be able to receive a media stream from an RTP session and store it in a file.
  - An application that records a conversation or conference must be able to receive a media stream from an RTP session and both render it on the console and store it in a file.

Transmitting Media Streams Across the Network

- RTP server applications transmit captured or stored media streams across the network. For example-
  - In a conferencing application, a media stream might be captured from a video camera and sent out on one or more RTP sessions.

RTP with JMF

- We will not be doing substantial JMF programming with RTP. But it must be mentioned that JMF provides following facilities for sophisticated RTP applications:
  - Managing sessions, participants, and media streams (for example, starting, finishing, adding, and so on).
  - Monitoring (through listener interfaces) of RTP events.
  - Gathering and generating statistics (for example, quality of connection)

Example

- We will simply write a program which output a RTP stream of type H.263 on to a Datasink connected to a port.
- Please see the file RTPStreamFromFile.java