Project 5 Streaming Video with RTSP and RTP

Implement a streaming video server and client that communicate using the Real-Time Streaming Protocol (RTSP) and send data using the Real-time Transfer Protocol (RTP).

Classes

There are 4 classes in the assignment.

Client
This class implements the client and the user interface which you use to send RTSP commands and which is used to display the video. The controls you need to implement are: the actions that are taken when the buttons are pressed.

Server
This class implements the server which responds to the RTSP requests and streams back the video.

RTPpacket
This class is used to handle the RTP packets.

VideoStream
This class is used to read video data from the file on disk.

1. The Client

Your first task is to implement the RTSP on the client side. To do this, you need to complete the functions that are called when the user clicks on the buttons in the user interface. For each button in the interface there is a handler function in the code. You will need to implement the following actions in each handler function.

When the client starts, it also opens the RTSP socket to the server. Use this socket for sending all RTSP requests.

SETUP

- Create a socket for receiving RTP data and set the timeout on the socket to 5 milliseconds.
- Send SETUP request to server. You will need to insert the Transport header in which you specify the port for the RTP data socket you just created.
- Read reply from server and parse the Session header in the response to get the session ID.
PLAY

- Send PLAY request. You must insert the Session header and use the session ID returned in the SETUP response. You must not put the Transport header in this request.
- Read server's response.

PAUSE

- Send PAUSE request. You must insert the Session header and use the session ID returned in the SETUP response. You must not put the Transport header in this request.
- Read server's response.

TEARDOWN

1. Send TEARDOWN request. You must insert the Session header and use the session ID returned in the SETUP response. You must not put the Transport header in this request.
2. Read server's response.

2. The Server

When the server receives the PLAY-request from the client, it starts a timer which is triggered every 100 ms. At these times the server will read one video frame from the file and send it to the client. The server creates an RTPpacket-object which is the RTP-encapsulation of the video frame.

The server calls the first constructor of the class RTPpacket to perform the encapsulation. Your task is to write this function. You will need to do the following: (the letters in parenthesis refer to the fields in the RTP packet format below)

1. Set the RTP-version field (V). You must set this to 2.
2. Set padding (P), extension (X), number of contributing sources (CC), and marker (M) fields. These are all set to zero in this lab.
3. Set payload type field (PT). In this lab we use MJPEG and the type for that is 26.
4. Set the sequence number. The server gives this the sequence number as the Framenb argument to the constructor.
5. Set the timestamp. The server gives this number as the Time argument to the constructor.
6. Set the source identifier (SSRC). This field identifies the server. You can pick any integer value you like.
Because we have no other contributing sources (field CC == 0), the CSRC-field does not exist. The length of the packet header is therefore 12 bytes, or the first three lines from the diagram below.

```
+----------------+----------------+----------------+----------------+
|    V=2  |   P  |   X  |    CC   |   M  |  PT  | sequence number |
|----------------+----------------+----------------+----------------+
|                |               |               |                |
|         +----------------+----------------+----------------+----------------+
|         |          timestamp          |
|         |----------------+----------------+----------------+----------------+
|         |                |    synchronization source (SSRC) identifier    |
|         |                |                |                |
|         +----------------+----------------+----------------+----------------+
|            | contributing source (CSRC) identifiers |
|            |                |                |
+----------------+----------------+----------------+----------------+
```

You must fill in the header in the array header of the RTPpacket-class. You will also need to copy the payload (given as argument data) to the variable payload. The length of the payload is given in the argument data_length.

For more details on RTP, please see RFC 1889.

**Project Deliverables**

i. Video Streaming Client application (including source code).
ii. Video Streaming Server application (including source code).
iii. Video file used in the application demonstration.
iv. A project summary paper that explains the design of your solution, how to compile, and how to execute the solution.
v. Demonstration to professor that your solution works.