Real-Time MultiMedia
How to deal different file types

Jason Spencer

April 2018

---

1. Explain in detail, referring to the RTP/RTSP protocols where applicable as well as the Java classes in your answer, how you would go about modifying your solution in order to implement a client that can support an additional media encoding besides MJPEG. [10 Marks]

   The payload type descibes the content-type encoding. So we would need to descibe in the header lines in, RTSPPackets.java, the different encodings we would like to see. According to Schulzrinne (1998), payload type (PT): 7 bits. Identifies the format of the RTP payload and its interpretation by the application.
   Content-Encoding e req. SET_PARAMETER
   Content-Encoding e req. DESCRIBE, ANNOUNCE
   Content-Type e req. SET_PARAMETER, ANNOUNCE
   Content-Type r req. entity

   As we can see above the client is required (req) to set the parameter (SET_PARAMETER) (I.E. This method requests to set the value of a parameter for a media file specified by the URI). It will tell the server to give them a media file of *this* encoding (Schulzrinne, 1998).

   ANNOUNCE goes from the client to the server and vica versa. From the client to server, ANNOUNCE posts the description of a media file found through the request URL to a server (Schulzrinne, 1998). From the server to client, ANNOUNCE updates the session information to the client in real-time (Schulzrinne, 1998).

   The Session Description Protocol (SDP) is separated into three main sections session, timing, and media descriptions (Schulzrinne, 1998). SDP can be used to describe the media streams or presentations in RTSP. This usage is limited to usage is limited to defining the encoding/encodings and access (Schulzrinne, 1998).

   An SDP file describes how a client should transcribe SDP content (in this case we are talking about the content-encoding) returned in reply to a DESCRIBE request.

   The payload type are specified in the "m=" field of the header fields.

   Format-specific parameters are relayed using the "fmtp" media characteristic (Schulzrinne, 1998). According to, Schulzrinne (1998), "The syntax of the "fmtp" attribute is specific to the encoding(s) that the attribute refers to".

   As can bee seen above i would need to change the RSTPPacket.java file to allow for different encodings. I would also add

   static int MJPEG_TYPE = 26;
   with the other media encoding that the client woould like to request from the, server.

2. Imagine that, mid-session, either the client or the server wishes to change the parameters of the media streaming. Explain under what circumstances that may be appropriate, and how such an intervention could be factored into the project, making reference to the RTSP protocol where applicable as well as the Java classes in your answer. [15 Marks]

scale

The client may have missed information or seen/heard part of the media file before and therefor would want to rewind/fastford the file to any point in thie file. in order to do this i would add two listener classes (namely:fastForward, and Rewind). I would also need to add a header for scale the client could send to the server a request with the apropriate scale change so the server could process the request.

A scale value of 1 indicates normal play. If a request with a scale of anything that isnt 1 is sent the return will be the rate in respect to the *normal play* (Schulzrinne, 1998).

speed

if a client wants to speed up or slow down the media file (say they are watch a video explaining some mathamatical precinipal they may like they online professor to talk slowly to give them more time to follow) and we should allow for this. In order to do this we need to work with the speed field in the header field, changing this field changes the bandwidth used for data delivery (Schulzrinne, 1998). Therefore i would need to add a header field into the RTSPPacket for speed. I would also add a listener class to know when the client wants to change this information (Schulzrinne, 1998).

Device Control

A client may want to use mulitple types of devices to watch the streaming services and we should be able to support all these devices without comprimising the quality of the service.

We need a header sent from the client to the server descibing the type of device (ie the dimentions) or if it is pc/ mobile. from this we we be able to code in the sendRequest in RSTPClient specify that we want a media that fits the format of our device (Schulzrinne, 1998).

Quality

The client should be able to control the quality of the video. I would first add a listener class. i would record where the timestamp is and what the sequence number is, i would then change the state to INIT then call setup with the new quality parameters - from this i would call play with the given timestamp and sequence number so that the media could carry on from where it left off. Some media streaming services rewind a bit in case some information is lost and although this is not required id suggest doing this for viewing pleasure and to assure the client they have not missed anything.

References

Schulzrinne, H. 1998 (June). *Real Time Streaming Protocol.*