International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 06 | Sep-2015 www.irjet.net p-ISSN: 2395-0072

Quality Control in Video Streaming

Darshan Modi

Project Associate (Computer Science), Information and Library Network Centre (INFLIBNET), Gandhinagar, Gujarat, India

---

Abstract – ***Video Streaming technology has become main-stream as the availability of Internet and high bandwidth for the large companies. Organizations, research institutes and other things are regularly using video streaming technology on a day-to-day basis. This reacts its need in the modern era. The need to stream video in the varieties of applications, video streaming comes into the picture. In this paper, I focus on open source video streaming technology that tends to provide the functionalities like dynamic streaming and video-on-demand. The main aim of this paper to establish a video streaming technology and to improve the quality of video by reducing frame drops.***

Key Words: *Video Streaming, RTMP, RTMPS, RTMPT, RTMPE, x264*

1. INTRODUCTION

Video streaming is a word used while video streams are delivered in real time environment. You do not have to download a whole file containing video data, then watch it, but rather transfer the stream in format, and start watching it after some time to assure a smooth experience. In many ways, streaming video is something like television, where frames are presented to the user while delivered by the provider. The opposite to download a file in, before watching it, can be compared to a CD or DVD. Television is sometimes implemented as digital video streams in modern areas where advanced network infrastructure with fibre-optical cables is built. In this paper, we are concerned with open source video streaming technology which refers to the transport of media in real time and also aims to ensure quality of video while publishing.

Video streaming is typically classified into mainly two categories: On demand and Live. In on demand streaming, the request for a video comes via a usable interface, which initiates the stream. Live streaming is like television, where a video is presented constantly to the users, and users may select an alternative to tune in.

Streaming video is generally transmitted over data network. This means a unidirectional transmission to the viewer, in which both the client and server took part for uninterrupted motion. The client waits and buffers for a few seconds of data before it starts playing, which is helpful in reducing delays in packet delivery. With non-streaming video, some portion of the video file must be downloaded over the Internet before a video can start playing. It allows for live video transmission over the Internet, and video data transmission rate varies depending on the user's current connection status. The obtainable bandwidth to people's home has been a revolution also.

Due to real time nature of video streaming, there are issues with bandwidth, delay and loss. To achieve smooth quality of streaming, these issues must be resolved. The rest of paper is organized as follows. First we discuss about the video streaming architecture. Then we discuss about the implementation strategy. After that we discuss how the quality of video can be achieved by reducing frame drops. At the end, we summarize the discussion.

2. VIDEO STREAMING ARCHITECTURE

Fig 1 shows the architecture for video streaming. It consists of the web camera to capture live video, media encoder to encode the raw data, streaming server to stream the encoded videos to the client. User can watch the videos by accessing the server over the network. Video encoding [2] is used to achieve transmission efficiency, because raw data takes a large amount of bandwidth. Streaming server is used for the distribution of video stream to the end users. The streaming server must be able to deliver the videos to the end user when user requests for the videos. The streaming video is delivered to the end users over networks.

2.1 Video encoder

Normally video captured from web cameras or IP cameras are in raw format. This must be compressed to achieve the efficiency in transmission. Compression is used to reduce and remove redundant data from video so that a digital video can be efficiently sent over a network and stored on disks. With these techniques, some amount of reduction in

file size can be achieved. In case of live distribution, encoding of video can be done in real time, while for on-demand request it may not be done in real time. Basically, video compression can be divided into two approaches: scalable and non-scalable video coding. A non-scalable video encoder generates one compressed stream while a scalable video encoder compresses a raw video sequence into multiple sub-streams. One of the stream, provides visual quality is the base stream.

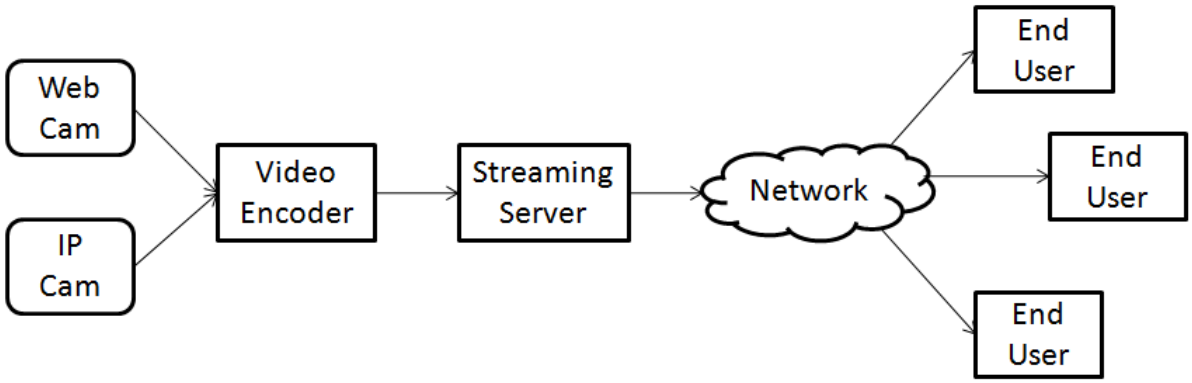

Fig -1: Video Streaming Architecture

2.2 Streaming Server

To supply quality streaming services, streaming servers are required to process multimedia data under timing constraints in order to prevent artefacts during the playback. It also retrieves the data in synchronization fashion. For example while delivering lecture, audio and video must be synchronized. The client can also demand for a particular video. The operating system must satisfy the requirements for the streaming servers. It must be able to handle streaming application loads.

2.2.1 Protocols for Streaming Video

In internet, TCP is most dominant protocol for data transfer [4]. TCP is used for video streaming but it does not provide reliable and good quality of services. It must be able to handle data rate variability and end to end delay retransmissions. TCP is not used in streaming because it handles the packet loss with the help of retransmission which can also causes the jitter in the video.

UDP is another protocol for video streaming. It takes care of packet loss and delay n another manner. It allows the packet to drop if it has been timeout or damaged [4]. But the problem with the UDP is some firewall blocks due to security reasons so it cannot be used.

RTP (Real-time Transport Protocol) is another protocol for streaming [4]. RTP is an Internet standard protocol for the transport of real-time media, audio and video. There two parts, data & control part which is called RTCP. The data part of RTP supports real-time transmission video and audio. The RTCP (RTP control protocol) part checks identification at source. RTP is built on the top of UDP and provide functionality or transport of media. But there is no guarantee for QOS.

RSVP (Resource ReSerVation Protocol) is designed for streaming applications[4].It is suitable for streaming since it allows a requisition which are transmitting information over a directed system to ask for assets at very hub and endeavour to make an asset booking for stream. This characteristic is utilized to guarantee the craved QOS with a dependable association. An alternate playing point is the scalability. The disadvantage of it is, receivers experience random packet loss for small reservations if some routers do not allow filter reservations.

RTMP (Real Time Messaging protocol) is a TCP-based protocol which maintains continuous connections and allows low-latency communication [3]. To provide streams and transmit as much data as possible, it divides it into different fragments. In RTMP, the fragment sizes are 64-bytes for audio, and 128 bytes for video. The RTMP defines many virtual channels on which packets may be sent and received, and which operate independently of each other. During a typical RTMP session, many channels may be active continuously at any time. When data is encoded, a header is generated. The header specifies, amongst other matters, the id of the channel on which it is to be sent and the size of the packet's payload. This header is then followed by the actual payload content, which is broken into pieces according to the currently agreed-upon fragment size before it is sent. The header is never fragmented, and its size does not count. Only the actual packet payload is fragmented. The next section covers the implementation details.

3. IMPLEMENTATION APPROACH

Video Streaming is the type of media that is delivered and introduced to the end user while being provided by the provider without interruption. Implementation of video streaming technology requires setting up a streaming server, encoder to encode or compress raw data. RTMP (Real Time Messaging Protocol) is used to establish a streaming server. RTMP is developed by macromedia to stream audio, video and data over internet [1]. Now it is owned by Adobe. RTMP protocol has many variants like RTMPS, RTMPE and RTMPT. RTMPS is RTMP over tsl/ssl connection. RTMPE is encrypted version of RTMP by security mechanism and RTMPT is encapsulated within HTTP to bypass through proxy and firewalls. The reason behind using RTMP, protocols like TCP or UDP breaks data into packets. They resend the lost or damaged packets and allow randomly ordered packets to be reassembled at a later point of time. This is convenient for downloading files or progressive download. But with streaming video, data need to be in real time and all the pieces in the right order. Another advantage of RTMP is, it can do dynamic streaming where video quality changes according to the end user's bandwidth. Another important part of video streaming is to capture the live stream and encode it. To achieve the transmission efficiency, video must be encoded. FFMPEG [14] is a video and audio converter tool that can get audio and video from a live source. It changes over between different rates and resizes video data. FFMPEG encoder is used to capture video from the live source, encode the video and publish it to the streaming server. FFmpeg supports x264[16]. x264 is a H.264/MPEG-4 AVC encoder. It is used for providing higher quality. It supports two modes to control the quality of the video and bit-rate. One is Constant rate factor and the other is Two pass ABR.

In constant rate factor (CRF) mode, it allows to maintain certain quality output when the file size is not important. This helps to achieve maximum compression. A lower value means the higher quality but in that case the file size is more compared to the original file. We can also apply the different encoding preset to achieve the speed in the encoding. FFmpeg provides different presets like ultrafast, superfast, very fast, faster, fast, medium, slow, slower and very slow. While two pass ABR mode allows maintaining certain quality output and certain file size. The encoded stream is now available at streaming server. From there, it will be delivered to the end users across the network. In addition to that, we must satisfy the quality requirements of the video. Next section covers about the, quality related aspects with regards to the streaming video.

4. PROPOSED APPROACH

Streaming video is real time in nature so we have to deal with the video quality. In our setup, while publishing stream or video to the streaming server, there were frame drops. This frame drops increases continuously. In real time streaming environment, if there is delay or some frame drops then the output would be disastrous. So the video streaming setup must be able to handle the frame drops. The reason behind frame drops is encoding performance. This is due encoder is not able to encode the frames at input frame rate. In other, we can say that if encoding performance cannot keep up, then the frames will be dropped. In order to solve this problem, I have suggested three possible solutions increase the encoding the speed or set the real time buffer or adjust the input and output frame rate. By increasing the encoding speed, we can avoid frame drops. So encoder will be able to encode the video at the input frame rate. This will be done using different presets. But we have to compromise on video quality if we will increase the speed of encoding. We can also reduce frame drops by setting the real time buffer. But the frames continue to drop. Another option is to set the input and output frame rate to control the frame drops. By specifying higher output frame rate compared to input frame rate and setting encoding speed as medium, we can avoid dropping frames. By using this mechanism good level of video quality can be achieved. Following this, the next section covers simulation results.

5. SIMULATION RESULTS

The following table shows the output of the same video file encoded using different encoding presets/speeds and using different constant rate factor (CRF) values in CRF mode.

Encoding Speed	Bit-rate(kbps)			File-Size(Mb)		
CRF	0	20	51	0	20	51
Slow	4788.9	673.81	35.50	22.89	3.74	0.77
Medium	4833.9	652.4	35.90	23.6	3.7	0.79
Fast	4839.9	652.5	36.99	23.12	3.65	0.80
Very fast	5920.4	621.0	33.01	28.8	3.5	0.78
Ultrafast	11050	1893	57.45	53.2	9.64	0.89

Fig-2: Comparison of different encoding Speed

The initial file was of 2.88 Mb. All the encoding was done on an Intel Xeon Core Processor running at 2.0 GHz with 4GB of RAM. If we do not set any frame rate and use the encoding speed/preset as a medium, there will be frame drops. The erroneous output is shown in below figure.

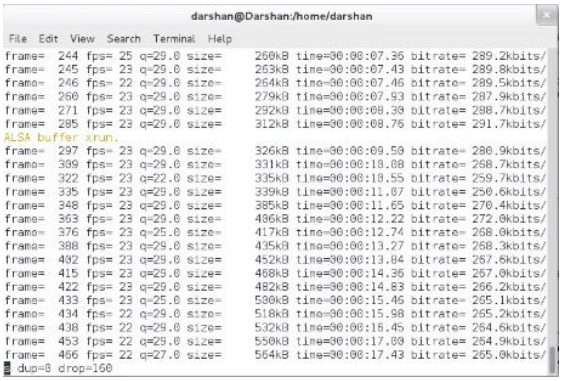

Fig-3: Frame Drops while streaming

We can avoid dropping frames by specifying the higher output frame rate compare to input frame rate and setting the encoding speed as medium. In this way the encoder will able to encode the frames at the input frame rate and publish it to the server. This will be able to reduce frame drops. The following output shows that there will no frame drops if we will set input and output frame rate accurately.

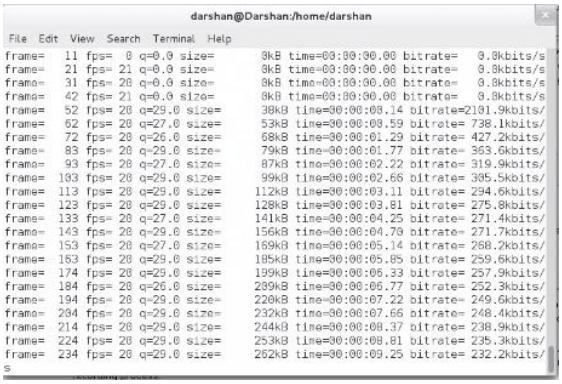

Fig-4: Reduced Frame Drops

6. CONCLUSIONS

In this paper, I have established open-source video streaming technology. I have also described how the video streams are delivered to the end user. The article proposes how the streams/files are encoded and published in real time environment. To provide enhance the video performance, I have reduced frame drops while publishing video to server. The simulation results are promising and shows the effectiveness in proposed work.

REFERENCES

[1] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha, "Streaming Video over the Internet: Approaches", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, MARCH 2001

[2] H. Parmar, M. Thornburgh, "Real Time Messaging Protocol", Adobe, December 21, 2012. December 21, 2012

[3] Huifang Sun, Anthony Vetro, Jun Xin, "An Overview of Scalable Video Streaming" TR2007-007 February. 2007

[4] Dan, D. Sitaram and P. Shahabuddin. Dynamic Batching Policies for an On-Demand Video Server, Multimedia Systems, 4(3): 112-121, June 1996.

[5] Kien A. Hua, Ying Cai and Simon Sheu. Patching: A Multicast Technique for True Video-on-Demand Services, In Proc. of ACM Multimedia, pp 191-200, Bristol, U.K, September 1998.

[6] Santosh Kulkarni ," Bandwidth Efficient Video-on-demand Algorithm (BEVA)" , Telstra Research Labs, Clayton, VIC 3168, Australia.

[7] Hanan Luss,"Optimal Content Distribution in Video-on-Demand Tree Networks", IEEE Transactions on systems,man, and CyberneticsPART A: Systems and Humans, Vol 40, No. 1, January 2010

[8] http://www.panasonic.com/in/business/security-systems/i-Pro-BB-BL-Series/dome-network-cameras/wv-ns202a.html

[9] http://www.nuuo.com/ProductNode.php?node=5&t=20130912065943#1

[10] http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol

[11] http://wiki.nginx.org/Main

[12] https://github.com/arut/nginx-rtmp-module

[13] http://www.mpeg.org/mpeg.html#Description

[14] https://trac.mpeg.org/wiki/x264EncodingGuide