ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(3), DOI: 10.17485/ijst/2016/v9i3/78864, January, 2016 ISSN (Online) : 0974-5645

Multi Server based Cloud-Assisted Real-Time Transrating for HTTP Live Streaming

G . Rohini[1*] and A. Srinivasan[2]

[1]Department of Information Technology, Adhiyamaan College of Engineering, Krishnagiri, Hosur – 635109, Tamil Nadu, India; rohiniphd789@gmail.com
[2]Department of Computer Science and Engineering, MNM Jain Engineering College, Thorapakkam, Chennai - 600097, Tamil Nadu, India; asrini30@gmail.com

Abstract

Background/Objective: The main objective of this work is to reduce the load balancing problems and problems like frozen play in the HTTP live streaming. **Methods/Statistical Analysis:** Cloud based http live streaming system used in the previous researches provides improved client satisfaction level but the main issue is the load balancing where the multiple clients' requests need to be processed in the single HLS server. The burden of HLS server would be increased in case of presence of multiple user requests. To overcome this problem multi server based HTTP live streaming approach is proposed. **Findings:** The multiple clients' requests would be handled by providing the appropriate services to them based on PSO approach which is used to calculate the weight function. An efficient particle swarm optimization is used for access the multimedia content from appropriate severs when client send the request to multiple servers. The multi-server concept can eliminate the high load variation problem by changing the load request based on user requirements dynamically. **Applications/Improvements:** Experimental results prove that the proposed methodology provides better result than the existing approach in terms bandwidth utilization and Peak signal-to-noise ratio.

Keywords: Bit Rate, HTTP Live Streaming, Media Segment

1. Introduction

RTP is an IP-based protocol which is supported for real time data transmission such as audio and video streams. Real-time Transport Protocol (RTP) cannot guarantee for timely delivery, it needs lower layer support[1].

Real-Time Control Protocol is proposed to work in conjunction with Real-time transport protocol. During RTP session, participants send RTCP packets in regular interval of time to convey feedback on quality of data delivery. Real-Time Control Protocol[2] provides Quality of service monitoring services.

The Real-Time Streaming Protocol (RTSP)[3] is used for sent the Multimedia data across the network in the form of streams. It is better Instead of storing large multimedia files and then playing back. If the client connects to the server, it holds the client status until client disconnect from the streaming server. Once a session between the client and the server has been established, the server sends the media as a continuous stream of packets over either UDP[4] or TCP[5] transport. In order to increase client media quality service, adaptive streaming has been introduced. In adaptive streaming, according to the client bandwidth condition the streaming server provide various stream modes. Then client select any one mode for efficient streaming based on the hardware capability. However, the client bandwidth changes during the play it is difficult to maintain media quality.

In order to improve the media quality HTTP live streaming is used. The client send request to server for download media content[6]. The severs provides the multimedia content according to the current bandwidth condition of client. It is used to compute possible changes in the bandwidth in the future.

* Author for correspondence

Streaming is the process of playing the audio and video file still it downloading. Video streaming[7] refers to the real time transmission of stored video or live video. There are two types for transmission of stored video across the Internet available. One is download mode and another one is streaming mode.

The Real-time Transport Protocol (RTP) is used to support live streaming and broadcast, video on Demand applications. It is designed for carry data over UDP/IP but can be used in conjunction with other transport protocols, such as TCP and DCCP. In order to monitor the end to end delay RTP protocol used RTP Control Protocol (RTCP). It describes Sender Report (SR) and Receiver Report (RR) for session monitor. SR has harmonization information for media play out and RR provides observed session characteristics (loss, jitter). RTP protocol does not give guarantee for timely delivery[8].

RTSP is a Real-Time Streaming Protocol which is suitable for efficient real data streaming. It sometime uses RTP protocol. Real-Time Streaming Protocol requires three methods. There are SETUP, PLAY and TEARDOWN. At first a client send the SETUP request to server for initialization. In order to start the streaming process, the client again sends the PLAY request. Finally streaming process is closed by send TEARDOWN request.

In order to increase client media quality service, adaptive streaming has been introduced. In adaptive streaming, according to the client bandwidth condition the streaming server provide various stream modes. Then client select any one mode for efficient streaming based on the hardware capability. However, the client bandwidth changes during the play it is difficult to maintain media quality.

HTTP live streaming protocol[9] is used for maintain the media quality. If the client sends request to the server for streaming multimedia content, the server can be select the bit rate for streaming based on bandwidth condition of client. Real time task scheduling has been presented in[10].

2. Methodologies

2.1 Proposed Method

HTTP live streaming is a media streaming protocol which is introduced for real-time streaming or pre-coded media streaming. HTTP live streaming protocol divides the media file into sequence of media segments. The client player chooses a suitable streaming bit rate based on the network bandwidth condition.

To provide high media quality for the user's network environment, the cloud-assisted real-time transrating system is introduced on HLS. A cloud service provider offers required facilities to clients who take minimum cost to request, process and to compute multimedia services. In order to reduce the probable load on a single server, the Multimedia segments are spread in different servers; the cloud access system is designed among multiple HLS servers. By using cloud computing the client will get different transrating media clips into different transrators based on their needs.

Each server has three main components. There are bandwidth recorder, segmenter transrating subsystem and segment redirecting subsystem. The bandwidth recorder is used to record the bandwidth condition of users who has a connection to the server. The segmenter transrating subsystem is used to analyse the current status of client based on the results of bandwidth recorder, and also compute which scheme is used for calculate the coding rate for next media segment and to perform transrating. The segment redirecting subsystem direct the HTTP media segment needs of client in the play list of media segments recording transrating.

The server computes the bandwidth condition for transcoding while it obtains the bandwidth connection to the client. However, some times the network condition becomes unsteady. To solve this problem we describe three modes which are used to tackle various network conditions. The mode transition state machine is used for balance bandwidth evaluation error. The three models are Normal Mode, Active Mode, and Conservative Mode.

Normal Mode: The server uses current down loading media segment bandwidth as a bit rate for transcoding. The target bit rate of the next media segment is computed by using this formula,

$$B_{next} = B_{avg} = \frac{\sum_{i=1}^{N_p} B_i}{N_p} \quad (1)$$

N_p - Number of packets

B_i- Bandwidth

B_{avg} - average bit rate for segment download

The bandwidth value is computed by using TCPdump which is used to record packets in the annular queue.

Active mode: In order to evaluate the bandwidth variation, the server compares last bandwidth condition with currently evaluated bandwidth condition.

$$B_{next} = \begin{Bmatrix} B_{end} + \alpha * T_{remained}, \text{if } a > 0 \\ B_{avg}, \quad \text{otherwise} \end{Bmatrix} \quad (2)$$

Where,
α- slope of change trend
B_{end} - Bit rate of end segment
$T_{remained}$ - Remaining time to next download.

Conservative Mode: In order to evaluate the bandwidth variation, the server compares last bandwidth condition with currently evaluated bandwidth condition. The mode computation is described below,

$$B_{next} = \begin{Bmatrix} B_{end} + \alpha * T_{remained}, \text{if } a, < 0 \\ B_{avg}, \quad \text{otherwise} \end{Bmatrix} \quad (3)$$

The bandwidth errors are used to enable the state switching based on different errors. The errors are normal error, overestimated error and under estimated error.

The gap between current bandwidth and the previously computed bandwidth is placed within Threshold bandwidth (Tb) range, it is known as Normal Error and also represents a Type B Error. If the current bandwidth is less than the previously computed bandwidth, it is known as Overestimated Error. It is also known as Type C error. When the current bandwidth is greater than the previously computed bandwidth, and it will be larger than Tb, it is called an underestimated Error, as represented by Type A Error. Here Tb standard deviation of bandwidth for downloading a media segment in a steady state.

At present, the server set a normal mode as an initial state. If Type A Error or Type C Error happen more than two repeated times, the mode transition state machine enters the Active or Conservative mode. It accepts equivalent decision making approaches in this state. The network bandwidth state is overestimated means, Type C Error occurs in the Active state. In Conservative mode Type A Error occur two repeated times it will be returns to Normal mode.

3. Multi-Server based Load Balancing Approach

The single server based cloud-assisted real-time transrating is used for HTTP live streaming. It cause server overloading problem. To overcome this problem, multi-server based load balancing approach is proposed. In that approach, it is difficult to analyse which server is suitable for access the multimedia content. The Particle swarm optimization algorithm is used to solve this burden.

In order to reduce the probable load on single server, multiple servers are used to store the multimedia content. The multimedia contents are same in all servers. Client sends a request to cloud for access multimedia service. Here Particle swarm optimization algorithm is used to calculate appropriate server for streaming process. The Weight value is calculated for compute cost function which is used to transmitting multimedia content between server i and client j. The proximity between client j and server i is computed for select appropriate server. The objective function is

$$w_{ij}^{t} = d_{ij}^{t}.l_{ij}^{t} \quad (4)$$

d_{ij}^{t} Proximity between server i and client j.
l_{ij}^{t} - Traffic load of the link between server i and client j.

$$l_{ij}^{t} = \sum_{k} \in_{k} u_{ij}^{t} C_{i} \quad (5)$$

u_{ij}^{t}–server i utilization ratio due to client j at time t
C_{i} - server Capacity

Based on the weight value the appropriate server is selected for access the multimedia content.

Algorithm

Step 1: Initialize particle.
Step 2: For each particle.
Step 4: Do
Step 6: Calculate fitness function (weight value)
Step 7: If calculated fitness value is better than current value take it as local best.
Step 8: Choose the particle with the highest a fitness function (weight value) as global best.
Step 9: Compute particle velocity

V [t+1] = v[t] + c1 * rand (1) * (pbest[t] - present[t]) + c2 * rand (2) * (gbest[t] - present[t])

// v [t+1] is the particle velocity,
// present [t] is the current particle (solution).
// pbest[t] and gbest[t] are defined as stated before.
// rand (1), rand (2) is a random number between (0, 1)
// c1, c2 are learning factors.

Step 10: Update particle position

Present [t+1] = present[t] + v [t+1].

Step 11: While maximum iteration until the best fitness value occurs.
Step 12: Return appropriate server with better media quality.
Step 13: End.

4. Experimental Results

4.1 Peak Signal-to-Noise Ratio (PSNR)

Peak signal-to-Noise Ratio (PSNR) is used to represent the index of media quality analysis and substitutes the average MSE of frames in the PSNR computing equation to obtain the PSNR value of the media segment, expressed as,

$$[\mathrm{PSNR}]_{(s)} = -10 * \log_{10} \sum_{i=1}^{N} \frac{\mathrm{MSE}_i}{\frac{\mathrm{MAX}^2}{N}} \quad (6)$$

Where

N - Number of frames.

The numerical results of PSNR comparison are given in Table 1.

Table 1. PSNR values.

Number of Segments	Single Server based HLS	Multi Server based HLS
5	5	10
10	10	15
15	15	20
20	20	25
25	30	35
30	35	40

The graphical representation of these values described below in Figure 1,

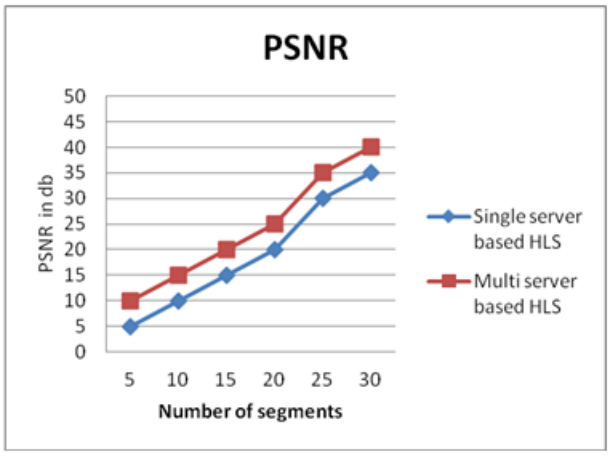

Figure 1. PSNR comparison.

In this graph, x axis will be number of segments and y axis will be PSNR value. The higher PSNR achieves better image quality. At all segments, the proposed methodology has better PSNR values. When number of segments is 25, single server based HLS has PSNR of 30 while the multi-server based HLS has 35. From the above graph it can be proved that the proposed methodology provides better PSNR result than the existing approach.

4.2 Bandwidth Utilization

The bandwidth is describe a data transfer rate, the amount of data that can be carried from one point to another in a given time period (usually a second).

Bandwidth for downloading multimedia segment is allocated for user according to the network condition. The numerical results are given in Table 2.

Table 2. Bandwidth utilization

Number of segments	Single server based HLS	Multi server based HLS
5	20	15
10	30	20
15	40	32
20	50	45
25	65	60
30	80	70

The graphical representation of these values described below in Figure 2.

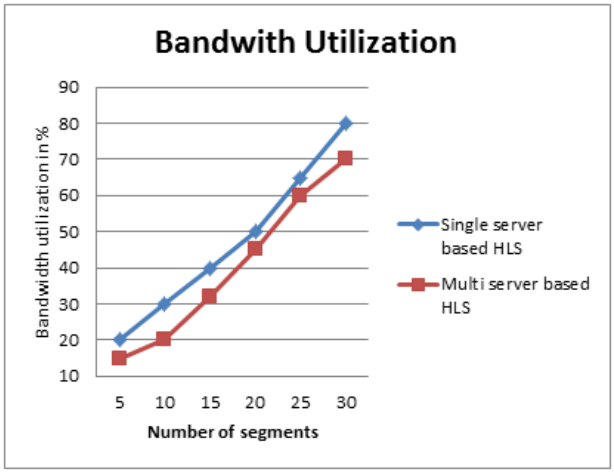

Figure 2. Bandwidth utilization.

In this graph, X axis will be number of segments and Y axis will be bandwidth utilization. At all segments, the proposed methodology has better values. When number of segments is 25, single server based HLS has bandwidth utilization of 65 while the multi-server based HLS has 60. From the above graph it can be proved that the proposed methodology provides lower bandwidth utilization result than the existing approach.

5. Conclusion

The proposed multi server based HTTP live streaming approach is introduced for reduce probable load on single server. The multiple clients' requests would handle by offering appropriate multimedia services to them based on PSO approach which is used to calculate the weight function. The PSO algorithm calculates the weight function based on the network proximity between server and client, traffic load of the link between server and client. The multi-server concept can eliminate the high load variation problem by changing the load request based on user requirements dynamically.

7. References

1. Singh V, Ahsan S, Ott J. MPRTP: Multipath considerations for real-time media. Proceedings of the 4thACM Multimedia Systems Conference; 2013. p. 190–201.
2. Johanson M. An RTP to HTTP Video Gateway. Proceedings of the 10th International Conference on World Wide Web (WWW'01); 2001. p. 499–503.
3. Yeung SF, Lui JCS, Yau DKY. Secure Real-Time Streaming Protocol (RTSP) for hierarchical proxy caching. International Journal of Network Security. 2007; 7(3):1–20.
4. Wang M, Li B. Network coding in live peer-to-peer streaming. IEEE Transactions on Multimedia. 2007; 9(8):1554–67.
5. Stockhammer T. Dynamic adaptive streaming over HTTP–standards and design principles. Proceedings of the 2nd Annual ACM Conference on Multimedia Systems; 2011. p. 133–44.
6. Goel S. Cloud-based mobile video streaming techniques. Global Journal of Computer Science and Technology. 2012; 1227:1–6.
7. Iyyanar P, Chitra M, Sabarinath P. Effective and secure scheme for video streaming using SRTP. International Journal of Machine Learning and Computing. 2012; 2(6):855–9.
8. Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A transport protocol for real-time applications. IETF RFC 3550. 2003; 311–4.
9. Lai CF, Chao HC, Lai YX, Wan J. Cloud-assisted real-time transrating for HTTP live streaming. IEEE Wireless Communications. 2013; 20(3):62–70.
10. Sivakumar P, Vinod B, Sandhya Devi RS, Jayasakthi Rajkumar ER. Real-Time Task Scheduling for Distributed Embedded System using MATLAB Toolboxes. Indian Journal of Science and Technology. 2015; 8(15).