Optimizing Parameters In Hybrid Clustered Peer Assisted DASH-SVC System using Particle Swarm Optimization

A B. Tech Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Arpit Gupta & Priyanshu Singh
(170101012 & 170101049)

under the guidance of

Dr. Pinaki Mitra

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Optimizing Parameters In Hybrid Clustered Peer Assisted DASH-SVC System using Particle Swarm Optimization**" is a bonafide work of **Arpit Gupta & Priyanshu Singh** (**Roll No. 170101012 & 170101049**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Dr. Pinaki Mitra**

Assistant/Associate

Professor,

Nov, 2021 Department of Computer Science & Engineering,

Guwahati. Indian Institute of Technology Guwahati, Assam.

Abstract

Video traffic over the internet has been increasing at a staggering rate over the years. As per a study conducted by Cisco, every second, a million minutes(17,000 hours) of video content will cross the internet by 2021. To combat such high demands, solely relying on CDN technology will not be enough as they have problems related to a single point of failure and limited bandwidth. So, P2P solutions have to be developed.

Over the years, many developments have been made to provide better P2P networks, like creating P2P-CDN hybrids and DASH standards to provide the client with video quality based on its network connection. We will be taking one such work that uses all these technologies and optimize its parameters further using Particle Swarm Optimization, to improve the QoE of users.

Contents

Chapter 1

Introduction

1.1 Quality Of Experience (QoE)

It is the measure of satisfaction of users with the service or network connection. It has historically emerged from QoS. Quality of Service (QoS) measures network parameters like throughput, which refers to the amount of data transferred within a timeframe from source to destination, packet loss, delay, and jitter, which refers to the variance in the delay. These parameters measure the network performance and are most of the times not related to user experience.

On the other hand, QoE can be measured either subjectively using mean opinion scores, which is the mean of the scores assigned by a subject to a system according to his own opinions regarding performance, PSNR, which is the peak signal-to-noise ratio, or using mathematical expressions, e.g., the QoE for video streaming [ea] can be calculated by

$$QoE_{video} = 4.23 - 0.0672T_i - 0.742F_r - 0.106T_r \tag{1.1}$$

where T_i, F_r and T_r are initial playback delay, average buffering and rebuffering frequency respectively.

QoE Estimation To give relatively reliable QoE prediction but avoid the necessity of doing subjective test, researchers develop objective quality models. Objective quality models compute a metric as a function of QoS parameters and external factors [CWZ15]. Two most basic objective quality models are Mean Squared Error (MSE) and Peak-Signalto-Noise Ratio (PSNR). MSE can be calculated as

$$MSE = \frac{1}{N}\sum_{i}(y_i - x_i)^2 \tag{1.2}$$

in which x_i is ith sample of the original signal and y_i is the ith sample of the distorted signal. PSNR is defined as

$$PSNR = 10\log_{10}\frac{MAX}{MSE} \tag{1.3}$$

where MAX is the maximum signal energy.

1.2 CDN-P2P Hybrids

CDN refers to the Content Distribution Network. It consists of geographically distributed servers to provide content in the form of HTML pages, Javascript files, or videos to end users.

P2P refers to Peer to Peer network. It consists of multiple peers/users connected together. In P2P, peers also participate in data distribution i.e. any peer can download data in chunks from multiple peers who already have this chunk, thus reducing latency.

A CDN-P2P hybrid consists of both the main server and a network of peers for data distribution. A peer can ask for data chunks from other peers as well as the main server.

There are two types of CDN-P2P hybrid networks:

- **Peer Assisted CDN(PAC):** In this hybrid network, peers are served mainly by CDNs, and other peers are just there to support, thus reducing stress on CDN or work as a backup in case of CDN failure.
- **CDN-Assisted P2P(CAP):** In this type of network, most content is provided by peers and CDN is only there to support starving peers.

1.3 Peer-Assisted DASH

DASH [Led] refers to Dynamically Adaptive Streaming over HTTP. In this standard, a media file is broken into several chunks, and each of these chunks is encoded into different resolutions and bit rates. The client can then download these chunks from the CDNs based on their network connection. Since the logic of which chunk to download solely lies with the client, it reduces the main server's load. This method works very well because now the client does not have to download the complete file at once but can decide which chunk to download (quality-wise) dynamically during the session. All the information about these chunks, like their size, bandwidth required, and URLs, is stored in **Media Presentation Description(MPD)** file.

Peer-Assisted DASH integrates the P2P network and DASH standard, i.e., peers can download chunks of media files with varying resolution or bit-rate from other peers. Since this method differs from conventional DASH in using peers upload bandwidth, so the MPD file structure is changed. The MPD file now also includes the URLs of peers where these chunks can be downloading.

1.4 H.264/Scalable Video Coding (SVC)

Varying connection qualities of various devices demand different qualities of video streams in a P2P network. One of the methods to achieve this can be **simulcast**; in this method, multiple single bit-streams can be transferred from a peer to the server, which can then send appropriate quality bit-stream to the requester. However, this method incurs a high bit rate thus, a single stream solution is required.

H.264/SVC [SMW07] is a concept that allows the encoding and decoding of multiple quality streams into a single bit-stream. This method provides a way to encode a high quality video bit stream containing one or more low quality bit-streams, which can be decoded. Thus we can think of this bit-stream as consisting of multiple layers, and as one keeps on removing these layers, we get low quality bit-stream. The bit-streams produced by this method are called scalable. There are various modes of scalability; some of them are spatial, temporal and quality. Spatial, temporal, and quality scalability describes the case in which a subset of bit-stream represents low resolution, low frame rate, and low signal to noise ratio, respectively.

1.5 Particle Swarm Optimization

It is a method used to find a candidate solution for an optimization problem iteratively. It solves the problem by having a population of candidate solutions (particles). These particles move around in the search space, with each particle having a position and velocity. The particle's velocity is composed of its speed, its past best solution, and the group's best solution. These parameters, like the group's best solution, are also updated as the particles move. These updates keep on happening until a termination criterion is met, which can be as simple as $number of iterations$.

Dynamic Particle Swarm Optimization is quite similar to Simple Particle Swarm Optimization, with one difference being the factors that constitute velocity are changed dynamically with iterations.

$$v_i = \omega \cdot v_i + c_1.rand(0,1) \cdot (p_i - x_i) + c_2.rand(0,1) \cdot (g - x_i) \tag{1.4}$$

$$\omega = 1.3 \cdot e^{\frac{-iter \cdot \pi}{2 \cdot iter\_max}} + 0.1 \tag{1.5}$$

$$c_1 = 2.5 \cdot e^{\frac{-iter \cdot \pi}{2 \cdot iter\_max}} \tag{1.6}$$

$$c_2 = -2.5 \cdot e^{\frac{-iter \cdot \pi}{2 \cdot iter\_max}} + 3 \tag{1.7}$$

For e.g. in above equation for velocity(v_i) update for particle i, p_i denotes the previous best position/solution for this particle, g denotes the best position among all particles, ω denotes the inertial weight and c_1 and c_2 are the accelerations. In DPSO the parameters ω, c_1 and c_2 are dynamically updated by iteration count.

Chapter 2

Review of Prior Works

We are trying to improve upon the paper "Hybrid Clustered Peer-Assisted DASH-SVC System" [DHF+15] by R. Dubin, Ofer Hadar, Yariv Freifeld and others, which was published in 2015.

The authors introduced a Hybrid CDN-P2P-SVN-DASH system to improve users' QoE by providing them with video chunks that are compatible with their network connections. They proposed a hierarchical structure to distribute the peers into multiple clusters, with each cluster having a sub-tracker as its leader. Which peer would act as sub-tracker is decided by the central tracker. For distribution of chunks among peers, the authors used PA-DASH integrated with H.264/SVC technology, which helps deliver appropriate resolution (spatial/temporal) chunks to the peers according to their network condition.

2.1 Algorithm

The algorithm proposed is mainly divided into two parts. The first is to decide from where and of what quality should next chunk be downloaded and the second is to subscribe itself to a cluster.

2.1.1 Peer and Chunk Selection Algorithm

The steps of the algorithm are as follows:

- Whenever a peer first starts to download the stream or after coming from re-buffer mode, it will start by downloading the base layer to reduce start up delay.
- Otherwise, the peer gets the download time for each possible layer from every peer possible and then selects the best combination of quality and download time. Such that it is possible to complete the download before the current buffer empties.
- The time to download the chunk from a particular peer is calculated from the historical statistics it gathered. For e.g. for RTT it is given by

$$PeerRTT = (1 - \alpha) * receivedRTT + \alpha * oldRTT \tag{2.1}$$

2.1.2 Clustering Algorithm

The steps for this algorithm are as follows:

- The first time the peer connects to the network, the main tracker assigns it a temporary cluster based on its RTT and geographical location.
- If the number of peers in the temporary cluster reaches a threshold value of $Cluster\_size$, the main tracker classifies these peers as a new cluster and sends a message $classify\_new\_cluster$ to all the peers.
- Upon receiving the message, each peer will calculate the radius RTT (maximum RTT) from each peer in the cluster and median RTT also from each peer possible. Using these values, each peer calculates its $Centdian$.
- Then, based on $Centdian$ main tracker will assign a peer to be a sub-tracker for this

cluster. *Centdian* is given by

$$Centdian = \lambda * radiusRTT + (1 - \lambda) * medianRTT \tag{2.2}$$

2.2 Conclusion

The authors first compared the average quality and average start delay as a function of distance from CDN in the No P2P and Pure P2P (no clusters) cases. Peers in Pure P2P network received better quality streams and had low start delays. Also, as the peer's distance from CDN increases, its average quality decreases, and average start delay increases in both No and Pure P2P cases.

As the cluster size increases, the average quality and average start delay of each peer improves. However, around 80% of performance improvement is achieved at a cluster size of 15. Increasing it further will give diminishing returns and also increase the load on the sub tracker.

Chapter 3

Proposed Algorithm

From section chapter 2, the authors tried to obtain a better average QoE across the peers. And had static α and λ values for equations 2.2 and 2.3.

We propose to update these parameters α and λ using Dynamic Particle Swarm Optimization (DPSO).

- λ: We consider different λ values for each cluster. Initially, when a cluster is first formed, a random λ is used; then, each time a new peer is added to this cluster, we can optimize λ. For applying DPSO, we need the previous best λ for this cluster and the overall best across all clusters based on QoE. We can take the average QoE of all peers in a particular cluster as the cluster's QoE. Thus, as new peers keep on adding to this cluster, we obtain a better sub tracker. The termination condition can be taken as the maximum number of peers allowed in a cluster.

- α: We can consider different α values for each peer. α is updated every time a peer receives any message (data response, connection response). For applying DPSO, we need previous best and global best α based on QoE. For local best, we can take α corresponding to the previous best QoE of this peer, and for global best, we can take α corresponding to the best QoE of any peer across the network. Termination

condition, in this case, can be the number of iterations or number of times updates are allowed for a peer.

We observe that the average QoE increases if the subtracker is close to the peers giving out more packets, i.e. the peers having the lesser M/M/1 queue length [4.2]. We introduce another weight parameter γ, which takes into account the queue length factor. Therefore the definition of Centdian is modified to:

$$Centdian = \lambda * radiusRTT + (1 - \lambda) * medianRTT + \gamma * queueLEN \tag{3.1}$$

The parameter γ is found using DPSO, similar to the parameters α and λ.

Chapter 4

Experimental Results

4.1 Simulation Idea

We initially take a fixed number of peers whose coordinates are according to a uniform distribution in the two-dimensional plane for simulation of the proposed algorithm. These peers are then clustered randomly into a fixed number of clusters, each cluster having a cluster centre (subtracker) which is also a peer. Each cluster contains a fixed number of packets that are randomly distributed among the peers in the cluster, and this information of distribution is present with the cluster centre. The distribution is such that each packet is present with atleast one peer. The simulation's objective is to record how much time is taken by each peer to get all the packets it does not have, which is a direct measure of the QoE of the peer. Our proposed algorithm aims to find better cluster centres and increase the overall QoE of peers in the cluster. The simulation is done in C++ using multi-threading.

4.2 Code Architecture

Each peer is a separate thread in the program that progressively queries the subtracker for the packets it does not have. The subtracker thread has complete information about

the distribution of packets in peers, which it stores in a map. Upon receiving the query from a peer for a particular packet, the subtracker responds by giving the nearest peer containing the requested packet to the requesting peer. Since all the peers are running simultaneously and multiple peers can query the same subtracker simultaneously, a lock is used in the subtracker to avoid contention. Sending/receiving of packets is modelled using M/M/1 queue object associated with each peer. Hence the time taken for sending/receiving packets is taken as the time taken in M/M/1 queue plus the time taken for the propagation between the two peers. After initial random clustering, the cluster centres are updated using DPSO as discussed in the Proposed Algorithm by optimizing parameters λ, α and γ. The QoE calculation of a peer considers the total time taken to get all the required packets and the standard deviation of times in getting the successive packets.

4.3 Results

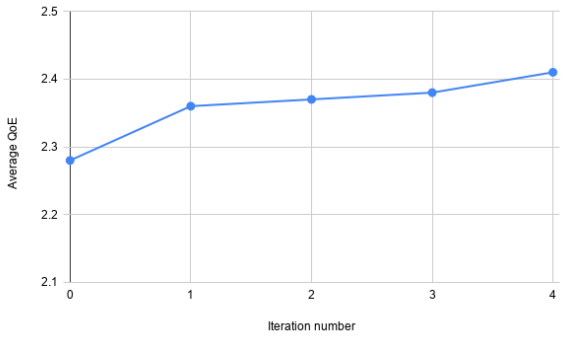

Fig. 4.1 Average QoE vs iterations (1000 peers)

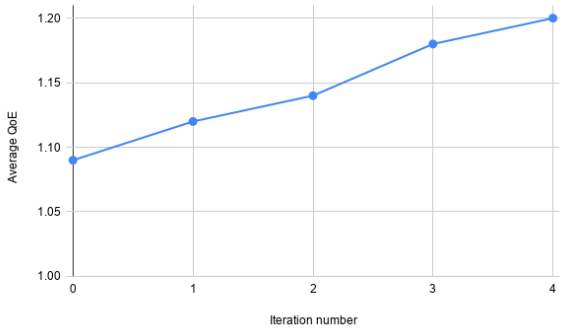

Fig. 4.2 Average QoE vs iterations (2500 peers)

The figures given above show the Average QoE plot vs no of iterations of the DPSO algorithm for 1000 peers and 2500 peers. Both the above simulations have used initial cluster numbers as 20, max coordinates in x and y as 1000 and number of packets as 100. The plot clearly shows an increase in QoE achieved by optimizing network parameters α,

λ and γ. The following equation calculates the QoE of a peer

$$QoE = \frac{1}{avg.time + SD} * 1000 \tag{4.1}$$

where avg. time is the average time taken to get all the packets a peer does not have, and SD is the standard deviation of these times.

Chapter 5

Conclusion and Future Work

The presented work above aims to improve the Hybrid Clustered Peer Assisted DASH-SVC System by optimizing network parameters presented in [DHF+15] using Particle Swarm Optimization. The simulation is done in C++, and the RTT between the peers was modelled using the M/M/1 queue object and their geographical distance. The results show an increase in average QoE for different network topologies when applying Dynamic Particle Swarm Optimization.

One of the areas for improvement is finding a better metric for QoE calculation than the one proposed. The current metric is intuitive but quite simple and is only an approximate reflection of the Quality of Service. A better formula for peer QoE would look like this

$$QoE = \frac{1}{a_1 * avgtime + a_2 * SD} * 1000 \tag{5.1}$$

where a_1 and a_2 are constants that are to be found empirically. Although the given simulation runs only for a fixed number of peers and clusters, it can be scaled to include simulation cases where new peers are continuously added to the network. One of the simple ways to implement this would be to add the new peer to its geographically closest cluster. This algorithm would become similar to the incremental clustering methods when new points are continuously added to the dataset. Some specific cluster size can be set to accommodate

these extra points, and upon exceeding that limit, reclustering of the complete network would happen using DPSO, as discussed in this report.

References

[CWZ15] Y. Chen, K. Wu, and Q. Zhang. From qos to qoe: A tutorial on video quality assessment. *IEEE Communications Surveys Tutorials*, 17(2):1126–1165, 2015.

[DHF+15] R. Dubin, Ofer Hadar, Yariv Freifeld, Aviv Ruham, Amit Dvir, Nissim Harel, and Refael Barkan. Hybrid clustered peer-assisted dash-svc system. 10 2015.

[ea] S. Y. Yoon et al. *Mobile data service QoE analytics and optimization.* 2015 IEEE International Conference on Communication Workshop (ICCW), London, 2015, pp. 1699-1704.

[Led] Christopher & Timmerer Christian. (2012). Lederer, Stefan & Mueller. *Towards Peer-Assisted Dynamic Adaptive Streaming over HTTP.* 2012 19th International Packet Video Workshop, PV 2012. 1-6. 10.1109/PV.2012.6229730.

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.