LayerP2P: A new way to Livestream

An Interdisciplinary Qualifying Project

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

Degree of Bachelor of Science

By: Matthew Aguiar

Project Advisor:

Professor Frederick Bianchi

Worcester Polytechnic Institute

Abstract

The entertainment and media industries are staples in modern society. Currently, there exists a multitude of internet platforms on which users communicate, share videos, and livestream every day. However, with real-time data rates rising, it is time to consider a new solution to the extreme networking demands of livestreaming. LayerP2P offers an incentivized approach with multiple video layers to produce much higher quality and stable streaming quality.

Acknowledgments

This project was influenced by the support and assistance of several individuals whom I wish to acknowledge. First, I would like to acknowledge my project advisor Frederick Bianchi for providing an opportunity for me to partake in this Interdisciplinary Qualifying Project. He has been invaluable to this project’s development with his thoughtful feedback and constructive criticism. I would also like to thank many of my professors at Worcester Polytechnic Institute including Yanhua Li for educating me on the underpinnings of computer networks.

Executive Summary

Contemporary networking infrastructure is based on the client-server networking paradigm. This model consists of client computers, such as the one you may be reading this on, that connect to large server computers. The client device will request content from the server such as HTML webpages, pictures, videos, etc. The issue is this centralized method of networking is becoming strained due to the extreme real-time data requirements of livestreaming. Given the rising popularity of livestreaming in the world, new networking methods that optimize for streaming are emerging.

Among the new networking strategies is P2P (Peer-to-Peer) networking. The philosophy behind P2P is it operates in a decentralized manner rather than a centralized one. Rather than use servers, which may easily become overwhelmed with livestreaming clients, P2P connects clients to each other. These computers are known as peers. Each peer on a P2P network sacrifices a portion of its resources as a sort of mini server. Resources for livestreaming include primarily upload bandwidth such that the video source will output better quality. However, there are three issues current P2P livestreaming services pose. First, they offer no incentives for users who contribute resources to the network. This in turn allows free-riders to selfishly consume content without providing anything in return. Second, current P2P networks lack the ability to adapt well when there is a bandwidth deficient scenario. For instance, if not enough users are actively turned on and connected, the amount of available bandwidth becomes restricted. Finally, they may suffer from severe quality degradation in the case of packet loss. Because the video is single layer, a loss of a packet means not only is the current frame of video degraded, but error propagation leads to successive frames of quality loss.

LayerP2P takes current P2P methods and builds on them to create the most flexible, high quality livestreaming network yet. First LayerP2P provides incentives to peers who contribute to the network. Those who contribute little to nothing will be left with very little download bandwidth for watching livestreams. As a result, this will motivate each user on the network to contribute something useful and make for a network rich in available resources. This concept extends to amend the issue with lack of adaptation in video quality in a bandwidth deficient environment. When the system demand exceeds the system supply, users with more contributions will be favored when adjusting video quality to accommodate the available bandwidth. Finally, LayerP2P offers layered video encoding for a graceful transition between video qualities. In layered video, each data packet contains L layers of video. If a packet becomes corrupted, it has several layers of video to choose from making video quality degradation less severe.

Further enhancements to LayerP2P include video smoothing and data scheduling. There are two methods of smoothing: amplitude reduction and frequency reduction. Amplitude reduction seeks to reduce extreme variations in which layer of video is chosen. It focuses on keeping the video quality in a stable range given a particular supply of bandwidth. Frequency bandwidth attempts to reduce how often the video changes layers to begin with. Data scheduling can be quite a challenge in LayerP2P due to the multiple layers of video per frame. Fortunately, Computer Scientist Xin Xiao poses a three-stage model to help with this.

Table of Contents

Table of Figures

Chapter 1: Introduction

Livestreaming has become a hallmark of the entertainment and social networking industries. Facebook, Twitter, Instagram, Twitch, YouTube, Vimeo, and many other platforms offer livestreaming services. With streams ranging from professional sports events to anyone with a personal device, there is no shortage of content for consumers to enjoy. However, with the newfound success of livestreaming comes a strain on available networking resources. Livestreaming is made possible through the transmission of real-time data; data that is encoded, processed, and transmitted continuously. In 2010, the amount of real-time data in the global datasphere was 0.5 zettabytes. While this is still over 100 trillion gigabytes, in 2025 it is projected the world will use 51 zettabytes of real-time data (O'Dea, 2021).

Due to the rising demand for livestreaming, the standard client-server networking model is becoming overworked. Traditional servers implement a centralized networking paradigm where personal devices connect to large server computers typically stored in warehouses. As the number of client devices grow combined with increasing amounts of livestreaming, servers are struggling to keep up. Fortunately, with the advent of P2P (Peer-to-Peer) networking, the struggles of transmitting livestreams are improving. Unlike client-server networking, P2P networks operate on a distributed basis rather than a centralized one. In a distributed network, there exists no servers. Instead, each device on the network is connected to each other and sacrifices a portion of its resources to be used as a mini server. Combining the resources of all devices on the network has proven to be a powerful method in transmitting data.

Although P2P networks improve the performance of livestreams, it is still a relatively new technology with limitations of its own. Electrical and Computer engineers Zhengye Liu, Yanming Shen and Keith W. Ross highlight the three downfalls of contemporary P2P networks. First, they argue that P2P networks lack incentives for users who contribute large amounts of upload bandwidth, allowing free-riders who contribute no bandwidth to freely reap the benefits. Second, in a bandwidth deficient environment, possibly caused due to a low number of users actively connected, video quality may be severely degraded. Finally, if a data packet is lost, it not only affects the current video frame, but successive ones as well due to error propagation (Liu, 2009, pg. 3).

In order to fix the three major downfalls with P2P livestreaming, this paper will explore a new way to livestream called LayerP2P. This technology uses a similar distributed networking paradigm as normal P2P networks. However, it provides incentives for users who contribute resources and introduces layered video streaming to reduce severe video degradation. It is not without its own limitations though. This paper will also explore the issues with fine video smoothing and data scheduling that hinder LayerP2P and how methods may be applied to help.

Chapter 2: Background

The client-server model has been and remains the most prominent method of networking. It is composed of two key components: clients and servers. The clients on the network are made up of devices attempting to access remote content; content that is hosted on a server computer far away. These devices include personal computers, phones, tablets, home security cameras and other network-oriented systems. Server computers are traditionally large macro-computers composed of hundreds to even thousands of smaller ones. This enables them to attain maximum computing power and networking bandwidth to provide content such as HTML webpages, photos, videos, livestreams, etc. An illustration of the client-server networking model is shown in **Figure 1** below.

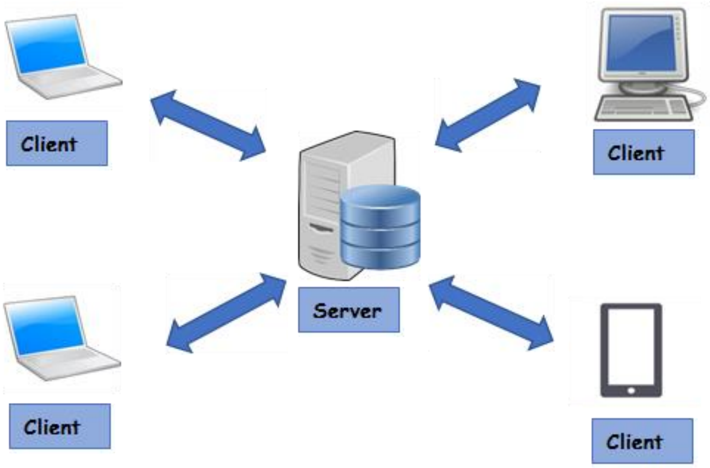


Figure 1 – Visual representation of client-server relationship (Omnisci)

The client-server model has many strengths such as its ability to store large amounts of data. For example, Google Drive allows each of its user accounts 15 gigabytes of remote data storage for free. Google is estimated to have around 4.037 billion accounts as of 2021 (Tankovska, 2021). However, this model has become increasingly strained with livestreams which use large amounts of real-time data. With increasing numbers of livestreams occurring, the amount of available bandwidth is dropping, leading to quality loss. The reason behind this bandwidth struggle is due to the client-server's centralized nature. **Figure 1** on the previous page illustrates this well as there are four clients connected to the same central server. In reality this client to server ratio would be much greater. Combining the disproportionate number of clients to servers with the demanding real-time data requirements of livestreaming yields a deficiency in bandwidth and thus, degradation of video quality.

P2P networks seek to reduce bandwidth deficiency by operating on a distributed basis rather than a centralized one. Instead of a few large servers, P2P networks use all client computers as both clients and servers. This means each device, or peer, on the network is connected to each other and may sacrifice a portion of its resources to use as network bandwidth and storage. **Figure 2** on the next page provides a visual representation of a P2P network where each peer is connected and there is a global tracker. The tracker simply contains tables of which peers are connected to each other, making it easier to route data from one peer to the next (Thampi, 2004, pg. 12).

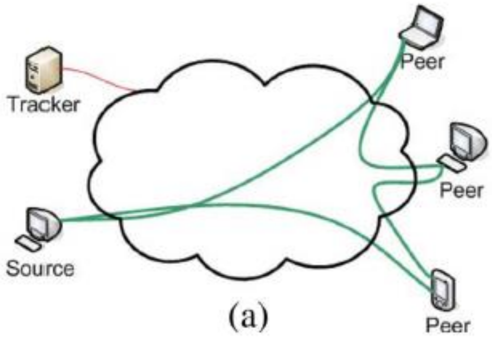


Figure 2 – Peers connected via LayerP2P to each other and a tracker (Liu, 2009, pg. 3)

It is important to note that while P2P may conceptually amend the issue of low bandwidth, in practice it is still difficult to achieve a network with enough devices to provide sufficient bandwidth for all users. Later, in the **Findings** chapter, this paper will discuss the current issues posing livestream P2P networks and how a new technology, LayerP2P attempts to fix them (Thampi, 2004, pg. 12).

Chapter 3: Methodology

This paper is broken into six key components for analyzing why LayerP2P technology is important and how it enhances the user experience. First, it is important to understand traditional networking standards and why they are less than satisfactory for livestreaming demands. The **Background** chapter has already discussed this.

The second component involves investigating the limitations of contemporary P2P networks. While standard P2P networks improve upon the client-server networking model there remain difficulties with various aspects of the technology. Once the limitations of P2P have been properly fleshed out, it becomes clear what needs to improve and leads to possible solutions.

After laying out the limitations of P2P networking, the most important part of the investigation turns to identifying specific solutions that would improve the technology. This paper explores a new method called LayerP2P which employs several methods for optimizing the traditional P2P model for livestreaming.

Once LayerP2Ps improvements have been stated and analyzed, it is essential to once again evaluate any limitations that remain. It is common in technology for new problems to arise when old ones are solved. LayerP2P is no different. Thus, the fourth component of this paper revolves around identifying the new drawbacks LayerP2P brings forth.

Naturally, once the limitations of LayerP2P have been exposed, it is important to look at even more solutions to further patch the technology. Specifically, data scheduling and video smoothing will be explored as methods for further improving LayerP2P.

Finally, after LayerP2P has been thoroughly examined, this paper will discuss some of the potential impacts this technology will have on society. As interesting as the inner workings of computers may be, in the end, it always comes down to the end user. Technology is produced for people not engineers.

Chapter 4: Findings

Current day P2P livestreaming services include CoolStreaming, PPLive, PPStream and UUSee. Although these platforms are showing potential to be powerful livestreaming services, they contain three key limitations. First, they lack incentives for peers that contribute network resources such as bandwidth. In a P2P network, users have a choice as to how much of their system they are willing to sacrifice to other peers. The issue regarding livestreaming is that all peers currently receive the same video quality regardless of how many resources they provide. Thus, peers who provide nothing and solely consume content are called free-riders. P2P livestreaming systems also lack adaptation of bandwidth availability. In a client-server network, servers are expected to be powered on 24 hours, providing constant service at the server's maximum bandwidth. However, available bandwidth on P2P networks can fluctuate greatly as the number of computers powered on varies called *peer churn* (Liu, 2009, pg. 3). As a result, in a situation of insufficient bandwidth for livestream producers and consumers, video quality is significantly degraded. Finally, packet-loss is a symptom of any network congested with users. In the case of single layered video, as is the case in current P2P services, packet loss may lead to severe quality degradation due to error propagation. In other words, once one packet is corrupted, each successive one also becomes corrupted. Fortunately, LayerP2P poses solutions to each of these three limitations.

Layered video encoding is the primary mechanic of LayerP2P rather than traditional single-layered encoding. From a high-level perspective, the system works by encoding the video source into *L* layers where each layer is sliced into data packets called *layer chunks* (*LCs*). These layer chunks are then distributed over a mesh overlay. In simple terms a mesh overlay is a map of all devices currently viewing a video. The mesh is built when the video source device divides the video bit-stream into layered chunks and disperses them at random to its peers. Thus, when a device on the network wishes to view a livestream, it obtains a list of peers in the mesh already watching the stream via a global tracker. **Figure 2** in the background chapter illustrates the use of a tracker. **Figure 3** below conveys a mesh-overlay.

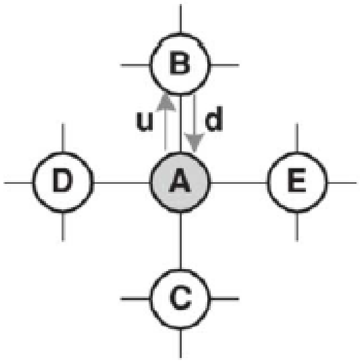


Figure 3 – Visual aid of mesh-overlay. *A* represents the video source (Liu, 2009, pg. 3).

LayerP2P's first goal is to incorporate incentives using a tit-for-tat strategy that rewards users for contributing their system resources, namely upload bandwidth for streaming. This strategy is adapted from another P2P network, BitTorrent, which focuses on file sharing. In this system, each peer measures its download rates from its neighboring peers (Liu, 2009, pg. 4). A peer will form a mutual relationship with its neighbors by providing a larger fraction of its upload rate. This way, a peer with greater contributions of upload bandwidth is more likely to be rewarded with more layer chunks, thus enhancing their video quality. Free-riders are peers who seek to consume content without contributing any bandwidth or resources to the network. Unfortunately, this cannot be prevented. However, the concept of an incentivized P2P network discourages this behavior, as any device with little to no contributions are provided with few layered chunks, causing poor video quality.

Another key component of LayerP2P is its ability to dynamically distribute bandwidth when the average upload bandwidth of the network falls below the full video rate. Once again, the network will do this using its tit-for-tat strategy to ensure users with the most contribution will receive the most bandwidth. Unfortunately, the lack of users currently on the network is a downfall of any P2P system and cannot be fully resolved. Therefore, it is important to get as many customers on the network as possible. As the network grows the more computing power it has.

The most important component of LayerP2P is how this technology transitions between video qualities gracefully. **Figure 4** shows a model of a typical layered video buffer.

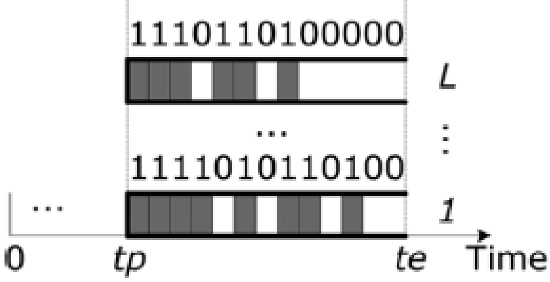


Figure 4 – Layered buffer structure (Liu, 2009, pg. 3)

The figure above illustrates data packets, or layer chunks, spread across layers from 1 to *L*. In a single layered livestreaming system, if one packet were to become lost, it not only would affect the current frame of video, but successive ones a well. This is known as error propagation. Fortunately, layered video aims to reduce error propagation and provide a more graceful degradation of video quality. If a data packet on the highest quality layer (layer *L*) becomes corrupted, then the system will automatically look to a lower layer and use that packet instead.

Lower layers contain lower quality content; however, this is still better than displaying corrupt video content to the screen.

Considering the three aspects of LayerP2P, Electrical and Computer engineers Zhengye Liu, Yanming Shen and Keith W. Ross have constructed a graph containing data the team gathered at PlanetLab as part of their LayerP2P implementation. **Figure 5** is a graph portraying the performance of peers on an overloaded network using both single layer and LayerP2P technology.

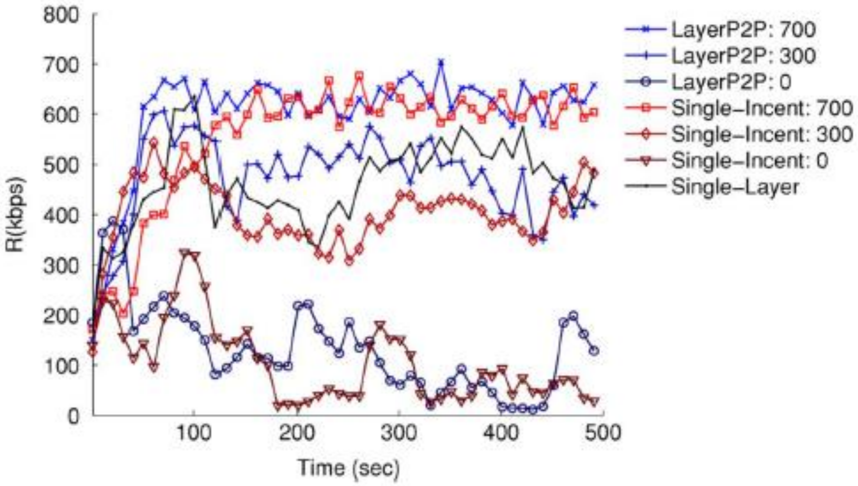


Figure 5 – Behavior of peers in an overloaded scenario (Liu, 2009, pg. 3)

Analyzing the graph above, there are three techniques Liu and his team tested on an overloaded network: Single-Layer, Single-Incent and LayerP2P. Single-Layer is a single-layer video stream with no incentives program. Each peer on the network receives equal bandwidth for consuming livestreams. Single-Incent is a single layered video stream with incentives similar to LayerP2P. LayerP2P includes incentives for peers as well as layered video encoding. The numbers 0, 300 and 700 indicate the source upload rate of the video. In other words, a video source may be outputting at a bit rate of 700kbps but the peers who receive the video over the network will likely see a bit rate less than this. Thus, **Figure 5** is showing how the actual upload rate varies over time with respect to the three different livestreaming methods. Looking at Single-Layer, the upload rate fluctuates within a range of 350kpbs and 600kbps. Interestingly, Single-Incent: 700 and LayerP2P: 700 both appear to reside at approximately the same bit rate, even matching to max 700kbps at some points. Therefore, while LayerP2P provides layered video, **Figure 5** suggests the most important aspect of LayerP2P, and any P2P network for that matter, is the ability to provide incentives to those who contribute. It is not to say layered video is not an important aspect of LayerP2P. One piece of information **Figure 5** does not contain is how smoothly video qualities are transitioned and how single-layered video deals with packet loss. Layered video is still a highly important component of LayerP2P for quality smoothing and flexibility.

As with all technology, when a solution is presented, usually a new set of challenges arise. There are two key challenges that face LayerP2P technology consisting of video quality smoothing and data scheduling. Although LayerP2P already provides smoother video quality transitioning, there are two smoothing mechanisms that further enhance the transition. First, *amplitude reduction* is the reduction in the size of jumps between quality levels (Bradai, 2013, pg. 12). Suppose a video source outputs a livestream on five layers with the fifth layer containing the highest quality video. If the current video play-head is on layer five when data packets become lost, amplitude reduction will attempt to drop down to use layer 4 instead of all the way down to layer 1. Essentially, amplitude reduction attempts to make video quality variation subtle as opposed to drastic quality changes. Given the importance of maintaining steady video quality, let us examine amplitude reduction in more detail.

Figure 6 contains a helpful illustration of a layered video timeline. As you can see, the horizontal axis is composed of timing intervals $t_0 \ldots t_{11}$. The vertical axis indicates which layer of video was used during a particular time interval. The smoothing windows are made up of a set number of timing intervals, in this case six, with which to calculate the average quality level $\bar{L}$ for that group of video packets.

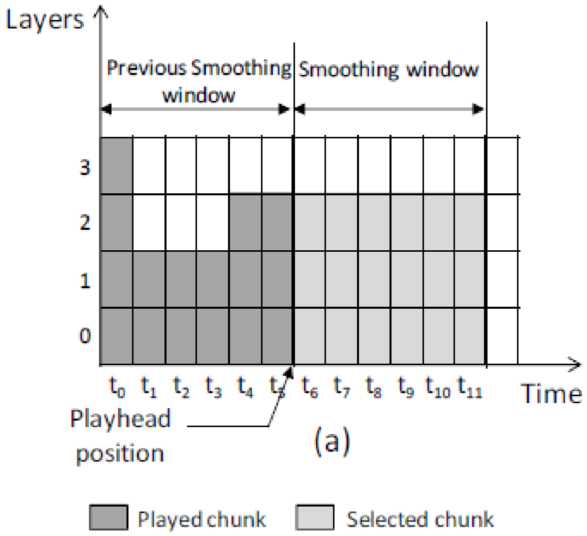


Figure 6 – Layer usage over time (Bradai, 2013, pg. 12).

This equation is used to calculate the average layer achieved during a smoothing window:

$$\bar{L} = \frac{L^{t_0} + L^{t_1} + L^{t_2} + L^{t_3} + \ldots + L^{t_{n-1}}}{n}$$

It is a simple equation to find the mean, or average. L^{t_n} is simply the layer utilized for a particular timing interval. If all the layers used are added up and divided by the number of timing intervals in a smoothing window, it yields the average layer used.

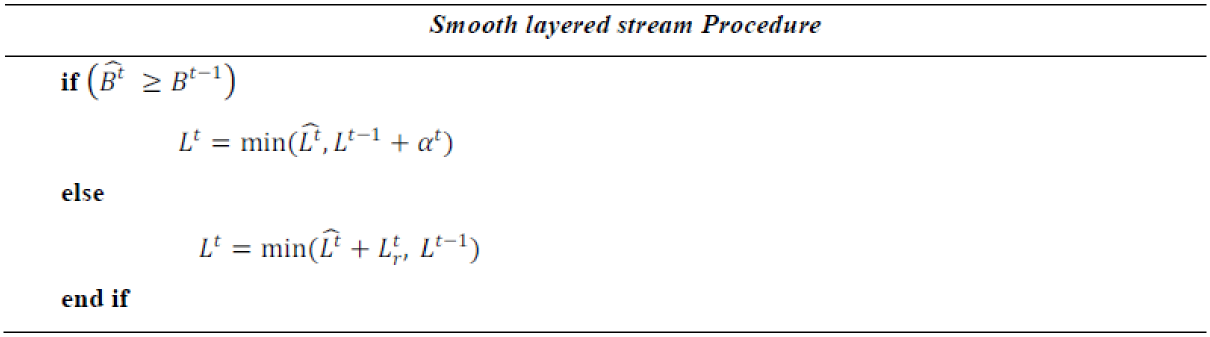
Here is the specific equation used to calculate the average layer used in **Figure 6**.

$$\bar{L} = \frac{L^{t_0} + L^{t_1} + L^{t_2} + L^{t_3} + L^{t_4} + L^{t_5}}{n} = \frac{3 + 1 + 1 + 1 + 2 + 2}{6} = 1.66$$

Thus, the average layer of video used in **Figure 6** was layer 1.66. Ultimately the goal of amplitude reduction is to decide which layer to use in the next timing interval, L^t. This is based on a combination of the average layer selected in the previous smoothing window, $\bar{L}$, and the layer selected prior to the next timing interval, L^{t-1}. The difference between L^{t-1} and $\bar{L}$ is known as the mean deviation. In simpler terms, amplitude reduction measures how much the previously chosen layer deviates from the average of the last *n* layers in the smoothing window. The equation for the mean deviation is shown below:

$$\alpha^t = |L^{t-1} - \bar{L}|$$

Using the information presented thus far, **Figure 7** sketches out pseudocode for how the layer for the next timing interval is chosen.



```
Smooth layered stream Procedure
if $(\widehat{B^t} \geq B^{t-1})$
        $L^t = \min(\widehat{L^t}, L^{t-1} + \alpha^t)$
else
        $L^t = \min(\widehat{L^t} + L^t_{r'}, L^{t-1})$
end if
```

Figure 7 – Algorithm for Amplitude Reduction (Bradai, 2013, pg. 12)

Note, $\widehat{B^t}$ represents the estimated available bandwidth, B^{t-1} is the bandwidth that was available during the previous timing interval, $\widehat{L^t}$ is the maximum sustainable layer allowed at bandwidth

$\widehat{B^t}$ and $\widehat{L_r}^t$ indicates the layer which may be achieved using the remaining bandwidth at the end of the previous timing interval. Using these definitions, the algorithm proceeds in the following manner. If the estimated bandwidth for the current timing interval is less greater or equal to the previous interval's bandwidth, the selected layer will be the minimum value chosen between either the maximum possible layer attained at the estimated bandwidth, or the layer chosen during the previous interval plus the mean deviation. Otherwise, the selected layer will be the minimum value chosen between either the maximum possible layer attained at the estimated bandwidth plus the layer attained using the remaining bandwidth from the previous interval, or simply the layer chosen during the previous frame. The improvements to LayerP2P made by amplitude reduction are evident in **Figure 8**. Clearly, the smooth stream shows a much smoother transition between layers than the raw stream given the available bandwidth.

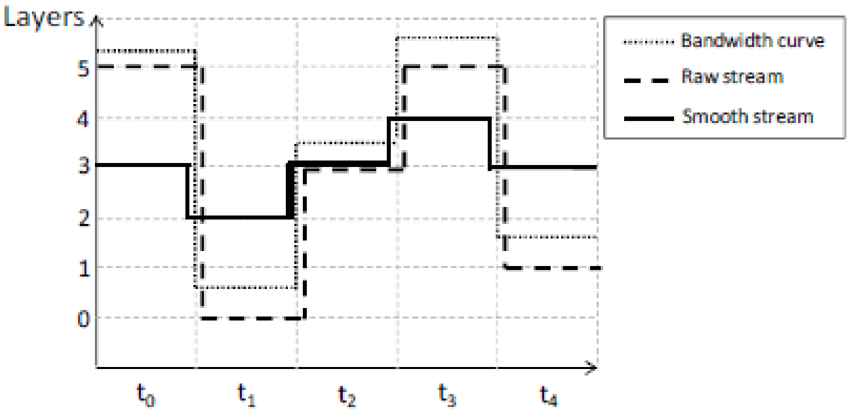


Figure 8 – Smoothing by amplitude reduction (Bradai, 2013, pg. 6)

Frequency reduction is another method of smoothing to enhance video quality on LayerP2P networks. Whereas amplitude reduction seeks to minimize the severity of layer fluctuations as bandwidth changes, the goal of frequency reduction is to reduce the frequency at which the video quality fluctuates. **Figure 9** provides an overview of frequency reduction similar to that of amplitude reduction.

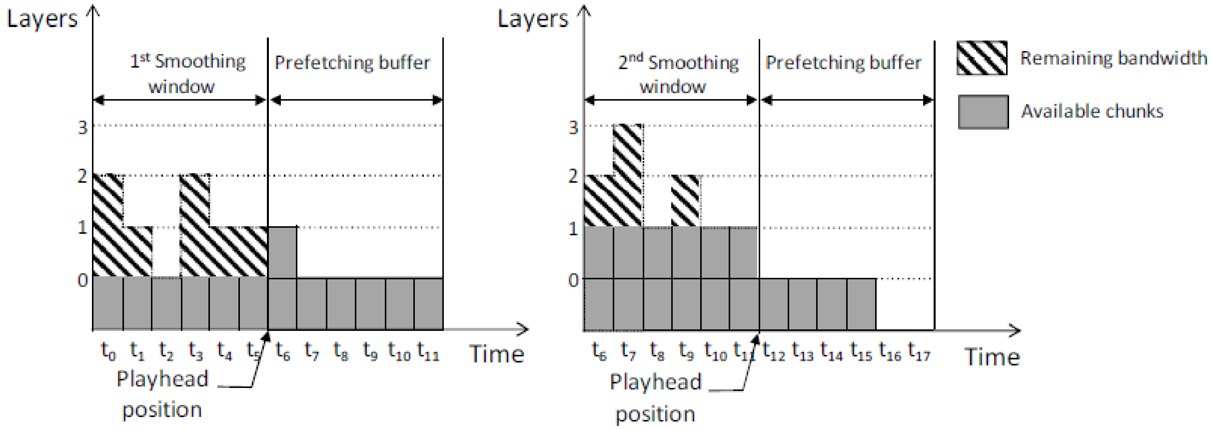


Figure 9 – Frequency reduction mechanism (Bradai, 2013, pg. 14)

Frequency reduction uses smoothing windows as well as prefetching buffers. The prefetching buffer utilizes any remaining bandwidth from the smoothing window to acquire packets ahead of time for the next smoothing window. The first smoothing window begins at the lowest layer, layer 0. At the end of the first smoothing window, the highest layer selected in the prefetching buffer is used as long as there remains enough bandwidth to support it. In this case the layer is upgraded to layer 1 in the second smoothing window.

Figure 10 shows the effects of frequency reduction on LayerP2P networks.

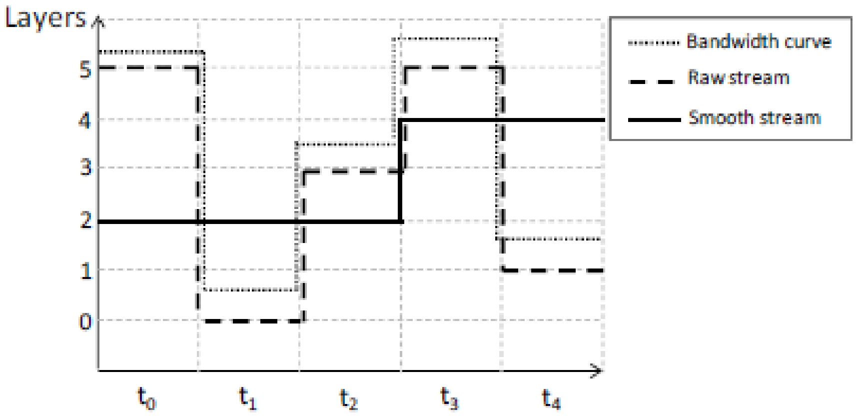


Figure 10 – Smoothing by frequency reduction (Bradai, 2013, pg. 6)

As the bandwidth varies on the network, the raw stream appears to change layers four times whereas the smooth stream changes layers only once. Applied to a much larger time frame, this difference becomes substantial in providing the most consistent video quality in LayerP2P livestreaming.

Another challenge presented by LayerP2P is data scheduling. Naturally, it is easier to encode and send video data to peers in a single-layered video system. With the addition of multiple layers of video, scheduling becomes a challenge. Computer scientist Xin Xiao and his team proposes a 3-stage model for data scheduling how video packets are sent to peers. **Figure 11** on the next page shows a summary of this 3-stage model.

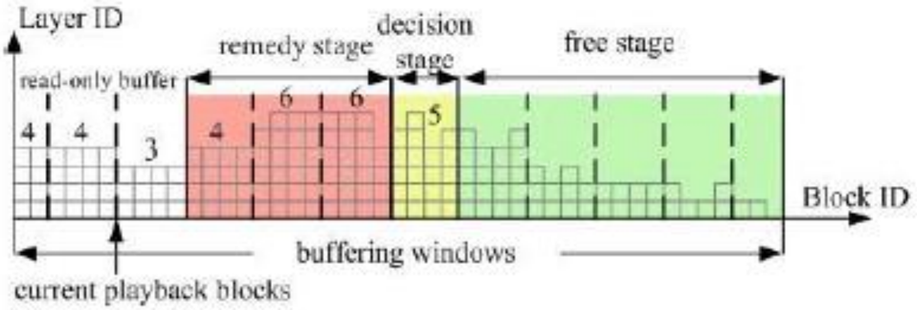


Figure 11 – 3-stage data scheduling model (Xiao, 2009, pg. 4)

The first stage is known as the *Remedy Stage*. This stage has priority over the other two as it contains data blocks that are closer to the playback time. Data blocks are the same as packets. The remedy mechanism is of *k* windows in length and the number of blocks that enter the remedy stage from the decision stage is *n*. Thus, *n* blocks must be delivered from neighboring peers before a window leaves the remedy stage. The choice of value for *k* is critical in this case. If *k* is chosen to be too small, it is likely there will not be enough time to request missing blocks. If *k* is too large, then there will be much larger waiting times to download new incoming blocks.

The *Decision Stage* consists of one window and its scheduling goal is to determine the proper number of layers to subscribe to for the contained window. Essentially, this stage determines the value *L*, the number of possible layers a window may choose from. To determine how many layers to subscribe to the decision layer uses the function shown below (Xiao, 2009, pg. 4).

$$SP(l) = \begin{cases} 1, if DR_l \geq ratio_{std} \\ DR_l / ratio_{std}, otherwise \end{cases}$$

This is a function that determines to probability a window will subscribe to a particular layer *l*. The probability will result in a 1 if the delivery ratio of layer *l* is greater or equal to a constant threshold. Otherwise, the probability of subscribing to layer *l* is the delivery ratio divided by a constant threshold.

The objective of the *Free Stage* is to freely request blocks that have not yet entered the decision stage. Bandwidth utilized by this stage consists of whatever bandwidth remains after being allocated to the Decision and Remedy stages. This stage is the furthest from the playback time thus, it acts similarly to the frequency reduction prefetch buffer found in **Figure 9.** The free stage uses an algorithm called the Min-Cost Flow Problem to schedule the collection of data blocks for this stage.

Chapter 5: Conclusion

LayerP2P is a peer-to-peer networking model tailored towards livestreaming with several key components that improve upon foundational P2P concepts. The first and most important aspect of LayerP2P is that it incentivizes users who contribute more of their personal system resources such as upload bandwidth and storage. The policy states that users will receive video quality commensurate with the amount of resources they provide. The incentives program thus discourages the behavior of free-riding, where users consume livestreams but contribute nothing to the network. Another feature LayerP2P offers is layered video encoding, where video packets are sent with multiple layers of video, each layer containing a higher quality than the previous. As a result, this allows for a more graceful degradation in video quality if packets become lost. Further enhancements to LayerP2P include video smoothing through amplitude reduction and frequency reduction as well as the 3-stage data scheduling model.

After a thorough investigation of LayerP2P, this technology has the potential to provide higher quality and more stable livestreams than ever before. The question is, how will this technology impact society? It is without question, entertainment and media are more accessible and enticing than ever before. YouTube, TikTok, Twitch and a slew of other media platforms continue to lead the way in the media landscape. However, none of these platforms are P2P. Rather, they rely on client-server networking. With that said, real-time data rates are soaring, expected to be 51 zettabytes in 2025 and it’s only a matter of time before our current networking infrastructure becomes overloaded and incapable of keeping up with the big data demand (O’Dea, 2021). It is at this time when P2P networks will thrive due to the number of personal devices far outnumbering the number of servers on the planet. Another possible factor leading to the rise of P2P technology is the increasing number of online communities appearing on media platforms. The ability to communicate and broadcast yourself from anywhere has created digital communities pertaining to just about anything. This idea of a community is very important to P2P networking due to its nature. Recall that the more users that are active on a P2P network, the more powerful it becomes as each peer contributes a portion of its system resources. Given the popularity and psychological effects of being a part of a community of like-minded individuals, the world may see P2P networks evolve into niche networks, where each network contains peers all of whom support a similar motive. In fact, this case is already beginning to emerge as Bitcoin and BitTorrent are P2P networks geared specifically towards cryptocurrency and file sharing respectively. As for now, P2P livestreaming services pioneering the way are CoolStreaming, PPLive, PPStream and UUSee. Although none of these services use LayerP2P, they are steppingstones in the process and will eventually give rise to greater streaming platforms to come.

Works Cited

Abbas Bradai, Ubaid Abbasi, Raul Landa, and Toufik Ahmed, “An Efficient Playout Smoothing Mechanism for Layered Streaming in P2P Networks” 2013 Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom pp. 1-27

Liu Zhengye, et al. “LayerP2P: Using Layered Video Chunks in P2P Live Streaming.” *IEEE Transactions on Multimedia*, vol. 11, no. 7, 2009, pp. 1340–1352., doi:10.1109/tmm.2009.2030656.

O'Dea S, “Global Datasphere: Real Time Data Volume 2010-2025.” *Statista*, 5 Feb. 2021, www.statista.com/statistics/949144/worldwide-global-datasphere-real-time-data-annual-size/.

Tankovska, H. “Number of e-Mail Users Worldwide 2025.” Statista, 19 Mar. 2021, www.statista.com/statistics/255080/number-of-e-mail-users-worldwide/.

Thampi Sabu M., “A Review on P2P Video Streaming” Indian Institute of Information Technology and Management – Kerala (IIITM-K), India IEEE 2013 pp.2-48.

“What Is Client-Server? Definition and FAQs.” *OmniSci*, www.omnisci.com/technical-glossary/client-server.

Xin Xiao, Yuanchun SHI, Qian ZHANG, Jianhua SHEN, and Yuan GAO, “Toward Systematical Data Scheduling for Layered Streaming in Peer-to-Peer Networks: Can We Go Farther” IEEE 2009, Tsinghua University Library pp.1-13.