COMP4801 Final Year Project

Project Plan

Peer-to-Peer Live Streaming
Web Application

Prepared by: Ho Kwan Hei (3035500841)
Supervised by: Dr. Anthony T.C. Tam
Table of Contents

Introduction..................................................................................................................................... 3

Background.................................................................................................................................. 3

Motivation................................................................................................................................... 3

The Project .................................................................................................................................. 3

   Peer-to-peer (P2P) application................................................................................................. 3
   Web Application....................................................................................................................... 3

Existing Products ......................................................................................................................... 3

Objective.......................................................................................................................................... 3

Methodology................................................................................................................................... 4

Project Timeline............................................................................................................................... 5

References....................................................................................................................................... 6
Introduction
Background
Live Streaming refers to live media content delivered to computers on the Internet. With popular live streaming platforms such as YouTube and Twitch, streamers can broadcast live video and interact with their audience simultaneously [1]. These platforms often utilize Content Delivery Network (CDN) to ensure the platform can service a large number of audience, providing global users smooth and low latency (delay) video playback [2]. However, CDN also carries a high setup cost and a high maintenance cost [2].

Motivation
While platforms such as YouTube and Twitch provide live streaming service free of charge, videos hosted on their platforms will be subject to their respective content guidelines. Live stream can be terminated if copyrighted materials appear in the video [3]. Streamers can be banned from the platform if they do not adhere to the community guidelines [4]. For these reasons, this project aims to create a live streaming application where streamers can exercise more control over their video contents.

The Project

Peer-to-peer (P2P) application
P2P network allows users to exchange files amongst themselves, without the need of a centralized server. The proposed live streaming application will construct a P2P network for content delivery. This approach provides a low entry barrier for live streamers since it is not necessary to rent expensive servers to handle the large amount of traffic generated by viewers. In addition, it provides scalability since each new peer will introduce extra bandwidth into the P2P network.

Web Application
The application will be developed to run on browsers, which will provide users a convenient way to access the live streaming service. Developing a web application also offer platform independence, which avoids the need of writing separate programming codes for difference operating systems such as Windows and macOS.

Existing Products
There are existing commercial products [5] [6] which offer live streaming over hybrid P2P network, where centralized servers are employed in addition to the P2P network to ensure reliability for content delivery. There are also P2P Internet Protocol Television (IPTV) which streams live television content over the Internet [7].

Objective
In currently available free live streaming platforms, streamers have little control over their video content. In response, this project hopes to make P2P live streaming accessible with minimal cost required.

Initially, the application will support two major functions:
• Allow streamer to broadcast video obtained from web cam
• Viewers can discover current live streams and join a live stream

After the core components of the application have been completed, several optional functionalities can be further implemented:

• Allow streamer to use Web Cam for broadcasting or use Screen Sharing
• Chat room inside a live stream

Methodology

JavaScript and Node.js will be used to build the web application.

Traditionally, two browsers cannot establish direct communication without a middle server relaying messages in between. However, it is now possible with the advent of WebRTC [8]. This browser built-in technology requires a signaling server to assist in connection setup between two browsers, and a direct browser-to-browser connection can be established afterwards. For this application, WebRTC provides API for data transmission and obtaining video stream from user computer’s web cam. The video can then be split into several-second video chunks using the browser’s Media Source Extension API. The video chunks are now ready to be shared in the P2P network. The (fixed) time length of each video chunk is known to each peer when they join the live stream.

An additional central server will be implemented to handle peer entry in the P2P network and allow viewers to discover current live stream.

For peer discovery, Kademia [9] Distributed Hash Table (DHT) will be used. In this P2P network, a video chunk is identified by a key (hash value of the video chunk’s timestamp). Each peer will receive a unique peer ID assigned by central server on entry. Peer ID and content key will share the same key space, and the XOR of these two numbers reveals the “closeness” of a peer to the content. Each peer also maintains a list of peers it has connected to. If a peer wishes to search for a specific video chunk, XOR distance metric [9] is applied to both the key and the peer IDs of connected peers to obtain several “closest neighbors” (connected peers). Then content requests are sent in parallel to these neighbors. If one of these peers holds the requested content, the content data is sent back. If not, the neighbor will send back several of its closest neighbors, and the query will repeat in the same fashion. If one iteration of querying closest neighbors does not reduce the XOR distance to the request content, new peers will be queried. Alternatively, storing content in the DHT network works in the similar manner, by selecting peers with closest XOR distance to the content key. For the implementation of Kademia DHT, the JavaScript library libp2p will be used. The video chunks stored in user computer’s memory will carry a expiration time, since it is not necessary to store the entire video for the purpose of live streaming and computers with low memory can potentially run out of system resources.
### Project Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Sep, 2020</td>
<td>First meeting with supervisor</td>
</tr>
<tr>
<td>4 Oct, 2020</td>
<td>Complete project research. Delivered of Phase 1:</td>
</tr>
<tr>
<td></td>
<td>• Detailed project plan</td>
</tr>
<tr>
<td></td>
<td>• Project web page</td>
</tr>
<tr>
<td>31 Oct, 2020</td>
<td>Build prototype application without P2P component. Video/Audio Stream can be correctly separated into chunks and fed to the video player.</td>
</tr>
<tr>
<td>30 Nov, 2020</td>
<td>Working implementation of DHT network</td>
</tr>
<tr>
<td>11-15 Jan, 2021</td>
<td>First presentation</td>
</tr>
<tr>
<td>24 Jan, 2021</td>
<td>Delivered of Phase 2</td>
</tr>
<tr>
<td></td>
<td>• Preliminary implementation</td>
</tr>
<tr>
<td></td>
<td>• Detailed interim report</td>
</tr>
<tr>
<td>7 Feb, 2021</td>
<td>Improve and complete the core functionalities of the application. (Live streaming over P2P network).</td>
</tr>
<tr>
<td>7 Mar, 2021</td>
<td>Allow live stream broadcaster choose to use web cam or screen sharing as video source.</td>
</tr>
<tr>
<td>18 Apr, 2021</td>
<td>Chat Room inside live stream. Delivered of Phase 3:</td>
</tr>
<tr>
<td></td>
<td>• Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>• Final report</td>
</tr>
<tr>
<td>19-23 Apr, 2021</td>
<td>Final presentation</td>
</tr>
<tr>
<td>4 May, 2021</td>
<td>Project exhibition</td>
</tr>
</tbody>
</table>
References


