Geo-based mediaplayer

An interactive interface for geo-based video streaming

Andreas Nordberg
andno793@student.liu.se

Jonathan Sjölund
jonsj507@student.liu.se

ABSTRACT

Being able to interact with a video stream can be both fun and educational; however for it to accomplish that the interaction must be seamless. Creating a media player that can allow user to view a video stream from different geographical position is a challenge that is tackled in this paper. By providing detailed information of the material used and methods implemented this paper will provide the information necessary for the possibilities of that kind of media player, and explain how to achieve a seemingly good and to high extent an enjoyable video streaming experience. A proof-of-concept is shown and provides a result that...(Mention the result here)

Keywords

HTTP Adaptive Streaming, OSMF video player, Interactive video streaming, Geographical based streaming, Seamless playback

1. INTRODUCTION

1.1 Motivation

Streaming has evolved and become more popular over last couple of years. Thousands upon thousands of different streams are being watched and streamed by many people every day, thus the demand for better and more ways to stream are longed for. Stream videos in different ways in order to create a more interesting streaming environment that can provide both better entertainment but also a better way to improve observation in science and other areas more reliably. If a stream can provide the possibility for watching a video from different angles it can give people the option to observe and also enjoy something from different perspectives. This project focuses on accomplishing that, by creating a geo-based video player that can allow people to view a video from different angles and change between them seamlessly without any delay or stuttering. By looking at an existing video streaming player and improve it to accomplish this task we can show and help to prove that it is something worth implementing in already existing video players.

1.2 Narrow Scope

In this project a so called Geo-based Command-and-control video streaming player using geo-tags will be designed and developed. In practice this means a service in which you can choose between a set of recording streams of, for example, the same event but slightly different locations and angles. This would be a useful feature to have in any larger event where you would want to show the same scene from different locations and angles. The interface will be useful for both event-organizers who hires staff to make several different recordings of the same scene for simultaneous viewing, but could also be used by the public who volunteers to record the event.

1.3 Boundaries

The application we provide is only going to be a prove of concept which means that we will only focus on the functionality of the video player. Factors like a pretty interface and usability on broader spectrum will be neglected. We will focus on making the application work for only one user to verify the functionality we want to accomplish. The number of video streams that we will be able to seamlessly switch between will for the purpose of testing be limited to a few, but then expanded upon to support any reasonable number of streams. This is because we firstly want to make sure that it is possible to switch between video streams and not that it can be done over large numbers. The reason for this is that pre-buffering many videos can be hard to accomplish with a large number of video streams. As long as we provide a way to do it the solution can be expanded upon.

2. BACKGROUND

To be able to grasp the concept of how HTTP-adaptive streaming and geo-based streaming works we first need to gather data and theories that we can use in order to further strengthen our methodology and the interpretation of our result. Since we will be using HTTP-adaptive- and geo-based streaming when programming our interface we will study existing theories and articles. We will then use that knowledge to implement a new upgraded video player that is adapted to streaming from different geographical positions.

2.1 Theory

Mobile users streaming media sometimes suffers from playback interruption when faced with a bad wireless connection. HTTP Adaptive streaming, HAS, seeks to resolve this by dynamically changing the bitrate and therefore quality of the stream to make do with the connection that's available to the user. To ensure smooth transitions between these quality swaps HAS also tries to predict the swaps in advance using various methods depending on the HAS framework. There are many algorithms for these predictions, but a brief example would be to use previous logged connectivity history and future connectivity using geo-based methods to make predictions [1]. Most HAS players uses weighted average of past download time/rates in order to estimate download rate of available bandwidth [2, s. 317-326]. With these HAS predictions, a stream quality fitting the user's network quality can be buffered. [1]

With HAS-adaptive streaming it's needed for us to prefetch data from several close-by (if not all, depending on number of streamers) streams and build up a small enough buffer that makes switching between different streams seamless. By looking at how HAS is used when implementing an Interactive Branched video we can say that parallell TCP connections is a must in-order to achieve this with a cost of wasting bandwidth and lower playback quality. This depends mainly on the number of videos that needs to be prefetched. Most HAS video players has a cap on the buffer size in order to avoid wasting bandwidth. As described in the paper Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming they use a customized HAS player that solves the problem of tradeoff between quality and number of chunks downloaded. The playback chunks are stored in the playback buffer while the prefetched chunks are stored in a browser cache thus allowing those chunks to be retrieved quickly. This ensures that no playback interruption occurs for the user. The way they download the chunks are done in a round-robin way to ensure that a buffer workahead is built up enough for seamless playback in parallel TCP downloading. [2, s. 317-326]

In Empowering the Creative User: Personalized HTTP-based

Adaptive Streaming of Multi-path Nonlinear Video they uses this technique of storing prefetched chunks in a playback buffer. Prefetching for different branches to allow seamless switching between videos, using the notion of multi-path nonlinear videos to stitch together videos using a novel buffer management and prefetching policy. This prefetching decreases the time it takes to switch between branches considerably and is something we will take advantage of since the code we use from [2, s. 317-326] are based on a similar policy. Most of these works are mostly focused on branching videos which is simliar but in not entierly to what we will be doing. We will contribute more to the possibility of prefetching several videos parallell then switch to them on demand. [2, s. 317-326] [3]

Figure 1 and figure 2 illustrates an example of a stream consisting of chunks being played, how these chunks are prefetched and stored and a swap between two streams.

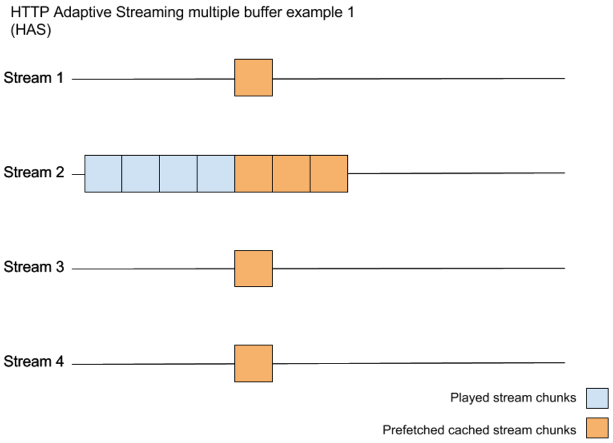
HTTP Adaptive Streaming multiple buffer example 1 (HAS)
Stream 1
Stream 2
Stream 3
Stream 4
Played stream chunks
Prefetched cached stream chunks

Figure 1: HAS Parallell Stream Buffer 1

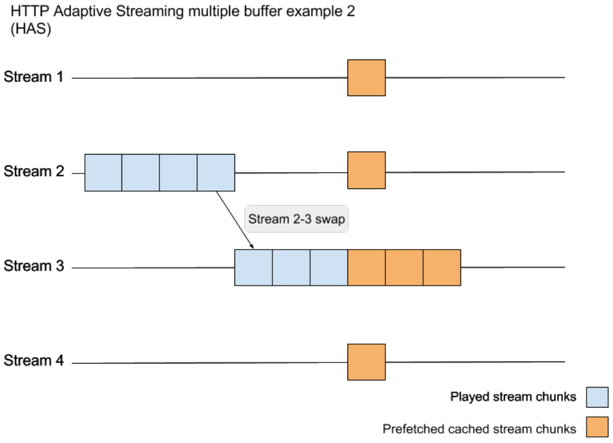
HTTP Adaptive Streaming multiple buffer example 2 (HAS)
Stream 1
Stream 2
Stream 2-3 swap
Stream 3
Stream 4
Played stream chunks
Prefetched cached stream chunks

Figure 2: HAS Parallell Stream Buffer 2

There can occur several problems in HAS players [2, s. 317-326]. The paper "Confused, timid, and unstable: Picking a video streaming rate is hard they" show that when a competing TCP flow starts a so called "downward spiral effect" occurs and the downgrade in throughput and playback rate becomes severe [4]. This is caused by a timeout in the TCP congestion window, high packet loss in competing flows and when a client has a lower throughput the playback rate is lower due to smaller buffer segments which makes a video flow more susceptible to perceiving lower throughput and thus creating a spiral. A possible to solution is to have larger segment size by having an algorithm which is less conservative, meaning that requesting a video at lower rate than is perceived. [4] This is something that we can keep in mind since quality can decrease drastically when having several videos buffering in parallel, though we will not have to buffer a full video at the same time but only bit sized ones while the main stream is being watched.

To display the stream in our application we will be using a media player called Strobe Media Playback (SMP), created with the Open Source Media Framework (OSMF) by Adobe Systems. The OSMF itself is build upon Adobe Flash Player, while becoming more outdated by day and discontinued by some, it is widely used for media and other graphic applications and suffices to use for the proof-of-concept of our application. In practice this means that the video player is created using the tools that OSMF provides, compiled into a runnable flash file bytecode and run by Adobe Flash Player. OSMF supports a number of important features that will be used within our interface. Most importantly it enables the use of HAS with its HTTP streaming support and progressive downloading. It also enables the player to seamlessly switch between several media elements by using a composition of "nested serial elements", which will be prominently used within our application. [3]

2.2 Method and materiel

To advance in this project we will mainly be programming, designing and developing the app. The programming language of choice will be Java Action Script and the IDE Flash Builder, which is very similar to the IDE Eclipse. The functionality of the interface to be developed is multiple. We want the interface to accept incoming video streams tagged with a location and cardinal direction from expected sources. The video streams will have to be tagged with these geographical datas somehow, which is not a common included feature with most video recording softwares. Developing a seperate recording app to create these kind of video streams, for the sake of this project, might be outside of our goal limitations for this project. If it is, we will prove the functionality of our interface with fabricated video geo-tags. These streams will then be made to work with a custom OSMF player. We will create a user interface with these streams as inputs so multiple video streams are shown to the user for the choosing. When one of these streams are being played the user should then be able to swap to another stream using the applications interface. Under-the-hood features will include HAS to ensure a smooth playback of the streams, both for a buffering a single stream but also for prefetching and buffering a fraction of the other streams to ensure uninterrupted playback during stream swaps. To help us focus on the main problem of developing this interface, we are being provided with some existing code by our supervisors. This includes a working SMP player created with a modified version of OSMF. We will also use code for an existing HAS-interface using prefetching. [2, s. 317-326] During the project we will receive constructive feedback about our work to help us with our process and provide us with other point of views. For any obstacles, expected and unexpected, we will read up on relevant theories that might help us.

When we implement our geo-based media player we need the gps coordinates and compass direction of the streamer. In Figure 3 the layout for how our new operation of our geo-based streaming interface will look like. The layout will display a view of every stream and the possible “Point of interest”, like a concert or the center of any larger event and this will have it’s own geographical position. The view will also display the north, west, east and south cardinal directions to know the direction of the every stream relative to those. The angle (theta) in Figure 3 can be calculated by taking the magnitude heading from a recording client. This will give us the direction relative to the north cardinal direction.

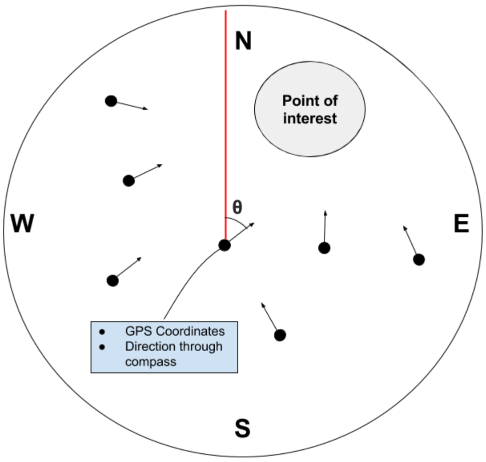

Figure 3: Concept interface of GPS & Direction selection map

The default SMP player that we’ll use and modify can be seen in Figure 4. We will be adding a button that allows us to open a “map” that is similar to Figure 3. There we will be able to choose which stream we want to watch by clicking on it. The stream will immediately play thanks to HAS functionality and if we want to switch to another the process can be repeated and another stream can be chosen. This player is by default set to play a stream of video located at a server supporting HTTP-streaming. For testing purposes at the least, we have created a WAMP server to enable video streaming playback with the used video player. Since we will be using a similiar OSMF player that were used in Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video the quality of our prefetched chuncks will be adaptive to the available bandwidth [5, s. 591-596].

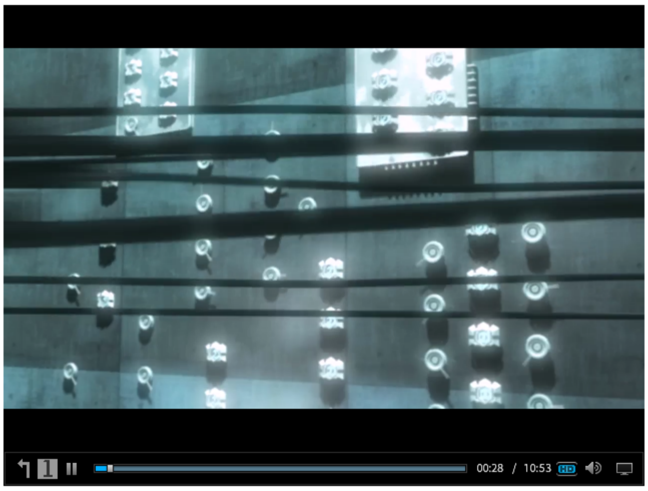

Figure 4: Strobe Media Player

3. DISCUSSION

3.1 Result

4. CONCLUSION

5. REFERENCES

[1] Jia Hao, Roger Zimmermann, Haiyang Ma, *GTube: Geo-Predictive Video Streaming over HTTP in Mobile Environments.*
Published: 2014. Fetched: 2016-03-09
URL: http://goo.gl/6DiQiW

[2] Vengatanathan Krishnamoorthi, Niklas Carlsson, Derek Eager, Anirban Mahanti, and Nahid Shahmehri, *Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming.*
Proc. ACM International Conference on Multimedia (ACM Multimedia) (Orlando, FL, Nov. 2014)

[3] Greg Hamer, *Open Source Media Framework: Introduction and overview.*
Published: 2010-03-15. Fetched: 2016-03-09
URL: http://www.wi-fiplanet.com/tutorials/article.php/3433451/Implementing-Wi-Fi-Multicast-Solutions.htm

[4] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, *Confused, timid, and unstable: Picking a video streaming rate is hard..*
In Proc. ACM IMC (November 2012)

[5] Vengatanathan Krishnamoorthi, Patrik Bergström, Niklas Carlsson, Derek Eager, Anirban Mahanti, and Nahid Shahmehri, *Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video.*
Proc. ACM Proc. ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking (FhMN)) (Hong Kong, Aug. 2013)