Consumer Based Rewind Interface to Play Back Depth Videos Over Network

Gary A Grimm Jr.
Department of Computer Science,
Truman State University
gag706@truman.edu

Zac Crane
Department of information technology,
University of Missouri – Columbia
zdc9pd@mail.missouri.edu

***Abstract*—Users of the MU elder-care program desire the ability to review videos stored in a server at MU over the internet. Currently, some users of the Kinect fall detection live in Kansas City with access to Google Fiber which ensures sufficient bandwidth for fast video playback, but there is also a need for accessing the interface via mobile devices, through wireless internet, or from locations where Google Fiber is not an option. We are trying to find ways to detect the client's bandwidth so that we can compress a video to a sufficient degree that streaming the video will be fast and have minimal negative effect on user experience.**

Index Terms*—*bandwidth, detection, streaming, compression

I. Introduction

Bandwidth detection can allow a website to customize itself in a way that is best suited for the user's connection speed. This can allow faster browsing for the user and more efficient use of resources.

Our project is to find an efficient and accurate way to detect user bandwidth and apply proper compression to videos that user's are viewing. This will allow the user to review the videos more quickly without waiting for the video to load.

II. Related Work

Similar papers are Measuring Bandwidth [2], A new estimation method using RTT for available bandwidth of a bottleneck link [3], And Bandwidth measurements and capacity exploitation in Gigabit Passive Optical Networks [4]. One other work is Method and System of Automatic Bandwidth Detection [1] a patent.

Measuring Bandwidth details several bandwidth detection methods and the pros and cons of each along with some measurement techniques for bandwidth. Measuring throughput allows a measure of the amount of data a transport protocol can transfer over time[2]. Another method used by TCP is to send larger packet after packet until one is dropped[2]. This allows an estimation of the available bandwidth of a network, but is quite slow. Additionally, there is the pair packet method which sends two packets, and measures the time between the receipt of the first packet and the receipt of the second packet.

A new estimation method using RTT for available bandwidth of a bottleneck link mentions Pathload, which uses probe packets and the stream rate of a packet train to determine bandwidth by measuring the delay and its rate of change,[3] (slowing down means the rate is too high for the available bandwidth) and Pathchirp which is similar, but uses packet pairs instead of single packets, [3] as methods that are currently used. However, pathload estimates the bandwidth by overloading the network until failure, and pathChirp is inaccurate when there is other traffic on the network and thus requires several probe packets be sent. [3] Both of these methods can be highly inefficient due to the high volume requirements. The writers suggest a new method using round trip time (RTT) which attempts to incorporate the delay times at the router, in the end-to-end transmission, in propagation, and in queuing. Keeping these variables separate allows one to detect network load by the queuing delays, and bandwidth utilization by the ratio of minimum RTT and total RTT. [3]

Bandwidth measurements and capacity exploitation in Gigabit Passive Optical Networks talks about ways to measure bandwidth along with other aspects of the gigabit passive optical network which will not be covered. The method used finds a ratio between the total bits transmitted of an user datagram protocol (UDP) stream to the client and the bits actually received [4] and uses that information to determine the actual throughput of the network.

The Role of Accessibility in a Universal Web discussed the topic of accessibility. The paper addresses universal design, or creating products that are usable by the most people possible. Some of the considerations for web accessibility include but are not limited to low bandwidth, limited access to the internet, computer skills, hardware and software, and the cost of internet connectivity [5]. The W3C is mentioned as helping to collaborate a way to maximize accessibility to the internet by implementing protocols and work groups. Some things they are working on are HTML 5 standards, as well as CSS. Also, it mentions that the best way to promote web accessibility is by increasing awareness of the benefits of web accessibility for people with disabilities. Accessibility and situational design of websites has a lot of overlap and by understanding accessibility better, it is possible to more efficiently meet both goals [5].

Method and System of Automatic Bandwidth Detection suggests sending a small file of known size and measuring the time taken to send the file [1] in order to give a rough estimation of bandwidth. This is by far the simplest option, but is prone to be incorrect due to fluctuations in the bandwidth because it only checks the bandwidth once and only for a short period of time. This will likely result in a result that is higher or lower than what would be considered the “correct” bandwidth.

III. Problem Formulation and Methods

Patients and researchers in the MU elder-care program desire the ability to review videos stored in a server at MU over the internet. Currently, users live in Kansas City with access to Google Fiber which ensures sufficient bandwidth for full video streaming, but there is also a need for accessing the interface via mobile devices, through wireless internet, or from locations where Google Fiber is not an option. We are trying to find ways to detect the client's bandwidth so that we can send a video that has been compressed to a sufficient degree that streaming the video will have minimal effect on user experience.

Required time to download a 15 min video with different compression rate and on different networks

Network	Available Bandwidth							
			8	16	32	64	128	Video BitRate (Kbps)
	Original	Normalized Scale	7200	14400	28800	57600	115200	Size of 15 min Video
2g	236 Kbps	236 Kbps	30.5	61.0	122.0	244.1	488.1	Time required to download the video(Seconds)
3g	14 Mbps	14336 Kbps	0.5	1.0	2.0	4.0	8.0	
802.11a	54 Mbps	55296 Kbps	0.1	0.3	0.5	1.0	2.1	
4g	100 Mbps	102400 Kbps	0.1	0.1	0.3	0.6	1.1	
802.11n	600 Mbps	614400 Kbps	0.0	0.0	0.0	0.1	0.2	
Google Fiber	1000 Mbps	1024000 Kbps	0.0	0.0	0.0	0.1	0.1	

Figure 1. Download Speeds

Figure 1 shows the download speeds of some lower quality videos on different connections. The primary issue at this level of quality is mobile data. 2G and 3G connections don't have the bandwidth to download 64 and 128Kbps videos without a slight delay. This problem can be exacerbated by HD video quality. 1080P quality can be around 8Mbps which, even at 100Mbps, would take several seconds to download.

Once a suitable method for bandwidth is decided on, we will ask participants in the program to review the web page. We will then ask them to assess its quality 1) in regards to the speed at which the video loads considering their bandwidth 2) how easy this automatic setting is to over ride manually along with 3) the simplicity of the web page.

IV. Implementation

The implementation of our web page is almost complete and only needs small details added like positioning of the video player and date selection that will be covered at a later time. Original time spent on the web page was used to learn HTML and JavaScript, and some PHP, CSS, AJAX, and MySQL. Learning the basics of these languages took up our first week as we had little or no experience in the majority of them. We created simple web pages to practice using the new languages. Some short time was also spent downloading and getting resources to work that we would need later like MySQL Workbench and XAMPP (Apache, MySQL, PHP, and Perl)

Simultaneously, we spent some time researching ways that web sites can connect to databases. We found that this was something in common practice and that we could simply use PHP to run a MySQL query through a pre-made connection, also created with PHP.

We also spent time researching some mobile/web programming frameworks like the Intel framework, which had a good library, used HTML5 and CSS3 and included a drag and drop interface for simplicity, the Sencha API which also used HTML5, but if mass produced, would require payment, jQuery which is free, has excellent examples for use, allows 3rd party plug-ins and has a large community and finally, Kendo API which is built on jQuery so it allows 3rd party plug-ins, but does not support device API's and also has a cost.

We began to look at graphing solutions for patients to visualize their data in the third week. We looked at Google Visualization, JQPlot, Highcharts, and Flot charts. We determined that Google Visualization was easier to use and should work well with building graphs with dynamic data.

After picking the API we created graphs from static data to practice using Visualization. Later we tried to incorporate data from a database and had issues with the page not being ready and figuring out how to send the data we needed to the graph. Eventually, with Moein's help we were able to get this working. At present this feature in the web page has been rescinded to focus on the research. Given time, we can re-add a graph meant to visualize where motion occurs and identify any falls in the video if they occur.

We also learned how to use HTML forms with GET and POST (along with PHP) and we created sample pages that access a database and either search for the given input, or insert new data. Initially we had some trouble communicating between the web page and the database, but resolved those issues after researching how to make the connection and feed it to functions.

At our presentation for week 2 we were asked to find more specific OKR's or goals so we decided we needed to:

- Create a WebApp that connects to the database and makes charts that will be based on data from the database and have:
  - selectable start/end dates
  - markers in the video player to indicate falls
  - graphs to represent motion or the lack thereof
  - selectable list of upcoming videos
  - next, previous, skip to, play/pause buttons
  - database authentication
  - Find a video player that uses MP4

Unfortunately, because of the many changes to our project over time, most of these functions are no longer relevant and have been removed.

We also read several of the Eldertech publications to get a better idea of where the whole project was headed and what had been done so far. After this, we wrote up descriptions of our understanding of the project and after meeting with Moein, we found out that our understanding of the project was incorrect and that we were meant to create a web interface meant for consumer use.

In week four, we decided to use VideoJS as our video player because it provided functionality with MP4 and had many available plug-ins already created.

We also added more goals for the week which included:

- add list of static videos to the web page
- Add AJAX functionality
- Add Authentication via database
- move website to online server

At the Tuesday meeting we were asked what our hypothesis was for our research. At this point we had not been told to create a hypothesis so later, we sat down with Dr. Skubic and Moein to decide on a hypothesis for our research. We ended up deciding that a question to drive our research should be sufficient. We discussed making a responsive web site that could detect bandwidth available and change available features based on bandwidth available and decided this would be a more interesting project and our question would be:

"Can we create a web page that detects bandwidth and provides service through a lower bandwidth network that patients find as satisfactory (within 15%) as a fully capable network (Google Fiber)."

We planned on researching bandwidth detection, responsive web design, and video compression for streaming.

We also met with Moein to learn about SSH and how to get it to work with Eclipse. There were some issues so we came back the next day when Brad could meet with us all and explain the process of tunneling. We downloaded WinSCP and Putty so that we could access Robby through the tunnels in Kronos.

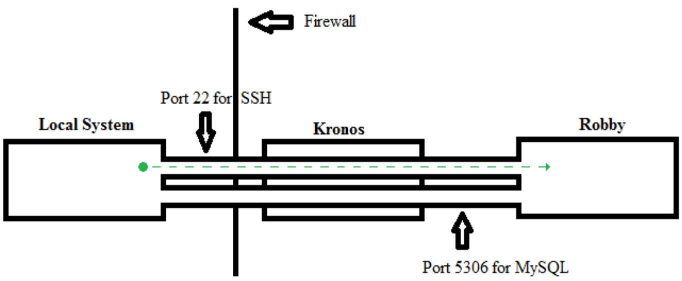

Figure 2. Tunneling

Figure 2 represents the tunneling process through Kronos into Robby using the SSH ports and MySQL ports. Tunneling is necessary because of the firewall that makes Robby inaccessible from an outside connection. This process allows us to add our web page files to the web server where the database is also located. This simplifies the connection and allows the web page access to the database.

Moein helped us to get the video player working with plug-ins and we added a next button that is able to update the video player's source and reload it along with a play/pause button. In addition, we implemented a "skip to" button that, when given the list of times in the video when there is movement, will skip to that section and play it.

Previously we were unable to load the web page onto the server due to inadequate permission but we met with Brad and Moein and were given the permissions to add files to the servers. Once the web page was put on the server we were able to hook it up to the database. This took some time due to the format of the video filepath and putting the results of the MySQL query into the proper format.

At this point (week 5) Moein suggested we focus more on the literature survey so we began researching papers about video compression, bandwidth detection, responsive web design, and web site usability and performance metrics. Most we found turned out to be irrelevant after we had read them and talked to Moein about the uses. He told us to focus on video streaming and bandwidth detection.

He also asked us to look into a way to limit bandwidth so that we could simulate a crippled network. We found a program called Mininet that was able to simulate a network from the server side but Brad was unavailable to put it on the server so we looked for another program. We found Netbalancer, which allows you to apply rules to certain applications that limit the bandwidth they use. We will likely use something similar to simulate our slower networks.

At the presentation on Tuesday (week 6) it was decided that we would no longer be concerned with the video interface or the video streaming aspects of the project and many of the papers we were reading became irrelevant so we found two new papers about bandwidth detection methods.

At present we are looking into a process by which we can detect the bandwidth of the user to a reasonable degree. One option we are considering is the "speedof.me" API which uses HTML5 and downloads small files of increasing size until the file takes more than 8 seconds to download. At which point, it will indicate your download speed based on that file. This is the most appealing option, as it has its own API and uses bandwidth detection that balances out many of the problems with other methods of detection because it uses several files over a span of time so that fluctuations in the network don't interfere as much.

V. Experimental Results

We want our video interface to provide a superior experience to the users by automatically selecting the best quality video for their available bandwidth. We expect that most users will be willing to sacrifice some quality in their videos in order to have fast loading time. We also want the interface to be easy to use so as to increase the Quality of Experience and satisfaction for the user. We expect that users will be as satisfied or nearly as content with the lower quality videos as long as they quickly.

References

[1] Florschuetz, Alan. "Method and system of automatic bandwidth detection."

[2] Lai, Kevin, and Mary Baker. "Measuring bandwidth." INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol. 1. IEEE, 1999.

[3] Imai, Masaharu, Yoshio Sugizaki, and Koichi Asatani. "A new estimation method using RTT for available bandwidth of a bottleneck link." *Information Networking (ICOIN), 2013 International Conference on*. IEEE, 2013.

[4] Rufini, Arianna, et al. "Bandwidth measurements and capacity exploitation in Gigabit Passive Optical Networks." *Photonics Technologies, 2014 Fotonica AEIT Italian Conference on*. IEEE, 2014.

[5] Henry, Shawn, Shadi Abou-Zahra, and Judy Brewer. "The Role of Accessibility in a Universal Web." *Proceeding of the 11th Web for All Conference* 17: n. pag. *Association for Computing Machinery*. Web. 23 June 2014.