An Analysis Framework for Simulated Video Streaming Networks

Patrick Mulumba, Peter Clayton and George Wells

***Abstract*—Distributed media streaming has been driven by the combination of improved media compression techniques and an increase in the availability of bandwidth. This increase has lead to the development of various streaming frameworks, which currently provide the majority of the streaming media available throughout the Internet. This study seeks to analyse a range of existing commercial and open-source streaming media distribution engines, and will attempt to classify them in such a way as to define a Common Analysis Framework. The common framework will be used as the basis for a simulation tool intended to predict the performance impacts of network configuration changes, new features, and general scaling. The intention is to validate the theoretical simulation tool against empirical evidence collected from configuration and engineering extensions to existing systems. The research will be undertaken using an experimental Computer Science approach, in which a system is developed or extended from a hypothesis of possible performance improvements. These improvements are evaluated for performance and functional viability, before making successive design changes or extensions based upon the outcome of the previous iterative design and evaluation steps.**

***Index Terms*—Network simulation, Cache, Video streaming, NS-2, Distributed Multimedia, Evalvid,**

I. Introduction

Modern video streaming consists of distribution engines (or frameworks) that provide streaming services through networks oriented upon the Internet [1]. However, many of these streams are transported through a collection of various network bandwidths and networking structures which can affect the quality of the video transmission. We propose a streaming media framework that provides a sufficient guideline that assists in the prediction of performance impacts of various changes to an existing streaming media network. We provide our motivation in Section II, followed by our objectives in Section III. The design of our framework is provided in Section IV and we detail our progress achieved thus far in Section V. We summarise our future work in Section VI.

II. MOTIVATION

The advancement of network communication has assisted in the growth of modern video streaming media [2]. However, since video streaming media is significantly dependent on the characteristics of the communication medium [3], various streaming optimization strategies and techniques have been applied by existing streaming media frameworks to ensure a successful video stream transmission. Since these strategies and techniques sometimes impose certain limitations on a stream, we propose a Common Analysis Framework for Simulated Streaming-Video Networks (CAFSS-Net). CAFSS-Net is intended to predict the performance impacts of configuration changes, new features, and general scaling of these strategies and techniques. CAFSS-Net will serve as a guideline to aid in the design of video streaming media environments. We also intend to evaluate the performance of CAFSS-Net against existing commercial and open-source streaming networks to validate the accuracy of our solution.

This work was partly sponsored by the Distributed Multimedia Centre of Excellence at Rhodes University, with sponsorship from Telkom SA, Business Connection, Comverse, Amatole Telecommunication, Mars Technologies, Openvoice, Storetech, Tellabs, THRIP and with financial support from National Research Foundation.

III. OBJECTIVES

A simple video media stream consists of three components; The encoder which reproduces the video for streaming, the server which transmits the video and the player which plays the media stream [4]. Our objective is to examine existing commercial and open-source frameworks that encompass these components by first analyzing the standard streaming concepts applied by these frameworks. Our common foundation is implemented in this framework to explore bandwidth performance, the effects of different network topologies and the different caching strategies available within the scope of our framework. We examine the primary complexities of video streaming which are latency, jitter and both dropped and lost packets. We also investigate how different video specifications [5] (such as resolution, motion in the scene, video length) may have an impact on the performance of CAFSS-Net.

IV. DESIGN

CAFSS-Net is in the form of a network simulation tool. Prior to this, we propose an experiment with existing streaming media networks to determine their features and strategies. We aim to incorporate some of these features/strategies into our framework to better represent existing streaming environments. We first implement a simple server-client streaming scenario as an initial test case for our simulation. We expand the client distribution to determine the impact of the overall performance. Secondly, we determine a suitable simulation tool that will encompass a streaming media environment which supports our objectives. We also capture some video streaming network data from both real networks and simulated networks to validate our simulation tool. Once our framework has been successfully calibrated, we will perform various configuration changes and apply features to further enhance our common analysis framework.

V. PROGRESS

Suitable experimentation with existing streaming media networks yielded us with a comfortable foundation for the development of CAFSS-Net. The following streaming environments

were investigated; Apple's Darwin Streaming environment [6], Windows Media streaming service [7], and Real Networks Helix Streaming Service [8]. We determined that each of these systems is capable of providing suitable streaming to all our testing scenarios. We also managed to capture network traffic data [9], [10] from both the server and the client(s). We used this information to validate our simulation tool.

NS-2 [11] is the simulation tool which serves as the foundation of CAFSS-Net. This allowed us to investigate the various streaming concepts stipulated, but no video simulation support was given with its initial distribution. Evalvid is a tool which evaluates the quality of video transmitted over networks [12], both real and simulated. We managed to integrate this tool into our existing NS-2 framework. We verified our simulation tool data against the data obtained from the commercial and open-source streaming networks to validate its accuracy. We found the difference in our initial networking tests between our simulation tool and the existing streaming networks to be within an acceptable margin of three percent.

In order to enhance the validity of CAFSS-Net, we used videos from an online Controlled Library of Research [13] with each of these videos adhering to specific criteria. These criteria aided us to provide an acceptable foundation for our testing. These specifications include different degrees of video scene motion, different resolutions and a constant number of frames for all videos. We also developed tools that can help us determine the latency, jitter, and both lost and dropped packets for the various transmissions through CAFSS-Net. These tools can provide graphs detailing the statistics of the video transmissions.

Since various caching algorithms and strategies are available for streaming, we propose the development of a simple caching scheme. This scheme enables us to transmit the test videos through different bandwidths imposed in our simulation environment. Our testing thus far, applies to our simple cache scheme to the test videos in order to compare the performance impacts of different network bandwidths within two networking scenarios [14] (a video streaming broadcast network and a video streaming multicasting network).

VI. FUTURE WORK

Additions to our testing still need to provide a holistic implementation of video streaming to CAFSS-Net. We still need to explore the impact of videos with different total frame counts and different video lengths. Since our existing videos contained no audio, we could possibly investigate the performance of a video stream containing audio as well. We also need to experiment with different simulation environment variables [15](file packetization/fragmentation sizes) and the possible implementation of different video streaming media caching strategies.

REFERENCES

[1] Adobe, "A streaming media primer," Adobe Dynamic Media Group, http://www.adobe.com/products/aftereffects/pdfs/ AdobeStr.pdf, 2000. [Online]. Available: http://www.adobe.com/products/aftereffects/pdfs/AdobeStr.pdf

[2] G. Michael, "A powerful trend that represents fundamental change: The internet streaming media boom," *Streaming Media Inc.*, vol. Best Practices: Media, Broadcast & Entertainment, 2006.

[3] P. de Cuetos and K. W. Ross, "Unified framework for optimal video streaming," IEEE INFOCOm, Tech. Rep., 2004.

[4] S. Mack, *Streaming Media Bible*. Hungry Minds, Inc., 2002.

[5] K.-L. Kao, C.-H. Ke, and C.-K. Shieh, "An advanced simulation tool-set for video transmission performance evaluation," in *ACM International Conference Proceeding Series*, P. from the 2006 workshop on ns-2: the IP network simulator, Ed., vol. 202. ACM New york, USA, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1190464

[6] Apple, *Streaming Server Administrator's Guide*, streaming server administrator's guide ed., Apple Computer, Inc., March 2005.

[7] Microsoft, "Windows media services," http://www.microsoft.com/windows/windowsmedia/forpros/server/server.aspx, 2007. [Online]. Available: http://www.microsoft.com/windows/windowsmedia/forpros/server/server.aspx

[8] R. Media, *Helix Server Administration Guide*, helix server version 11 ed., Real Networks, Inc., September 2005.

[9] G. Combs, "Wireshark: Go deep," http://www.wireshark.org/, January 2008. [Online]. Available: http://www.wireshark.org/

[10] sourceforge.net, "Tcpdump/libpcap public repository," http://www.tcpdump.org/, November 2007. [Online]. Available: http://www.tcpdump.org/

[11] K. Fall and K. Varadhan, *The ns Manual (formerly ns Notes and Documentation)*, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC., 10 2005.

[12] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," in *13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*. Technical University of Berlin, Telecommunication Networks Group (TKN), 2003.

[13] A. S. University, A. University, and S. Microsystems, "Video traces research group," http://trace.eas.asu.edu/index.html, 12 2007. [Online]. Available: http://trace.eas.asu.edu/index.html

[14] P. Mulumba, P. Clayton, and G. Wells, "An example of a simple cache system for a video streaming implementation within a network simulation," in *South African Telecommunication Networks and Applications Conference (SATNAC)*. Rhodes University: Rhodes University, May 2008, submitted: Still Under Review.

[15] C.-H. Ke, "How to evaluate mpeg video transmission using the ns2 simulator," http://hpds.ee.ncku.edu.tw/ smallko/ns2/Evalvid/_in/_NS2.htm, 05 2007. [Online]. Available: http://hpds.ee.ncku.edu.tw/~smallko/ns2/Evalvid_in_NS2.htm

P. Mulumba is with Rhodes University currently doing an MSc research in Distributed Multimedia, This author completed both his undergraduate and Honours years at Rhodes University, and his schooling at St Andrews in Grahamstown. The author's research interests are distributed media, networking, and Media compression techniques, (e-mail: g02m2469@campus.ru.ac.za).

P.G Clayton is the Deputy Vice Chancellor: Research and Development at Rhodes University. He was formerly the head of the Telkom CoE (Centre of Excellence) in Distributed Multimedia at Rhodes University. He recently received a National Science & Technology Forum award for Outstanding Contributions to Science, Engineering and Technology, (e-mail: p.clayton@ru.ac.za).

G.C Wells is an associate professor with Rhodes University, Grahamstown. He is the Head of Department of Computer Science at Rhodes University, and teaches programming language-related courses in the Department. He completed his PHD degree in Parallel and Distributed Computing at the university of Bristol. He is also a Sun-certified Java Programmer, (e-mail: g.wells@ru.ac.za).