STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 3, 2010

INTRODUCING VAST: A VIDEO-AUDIO STREAMING TESTER

ADRIAN STERCA AND CLAUDIU COBARZAN

Abstract. We present a testing package aimed at video and audio streaming across best-effort networks like the Internet. VAST is intended to be a testing framework for protocols transporting audio-video streams across IP networks. It offers the simplicity and predictability of deterministic simulators like ns-2 combined with the testing power of real-world experiments.

1. Introduction

The proliferation of multimedia data, especially the video and audio components, across the Internet has been exponential in recent years. As a result, the stress on the delivery infrastructure is increasing, as do the efforts to find and test audio-video streaming solutions in either simulated or real test environments. One of the most popular tools used for testing network transport protocols is the ns-2 simulator [8]. Although, simulation packages are in general easy to use and the obtained results are simple to assess due to their deterministic context, they lack the ability to capture the true randomness from real-world experiments, and thus they do not have the testing power of real-world experiments. On the other hand, because of all the randomness involved in real-world experiments and because of many noise sources, the results from those experiments are usually difficult to assess, as they reflect so many interconnections within the components of the experiment.

Our aim is a testing framework which does not rely on simulations, but borrows some of its deterministic approach and simplicity in assessing the test

Received by the editors: August 15, 2010.

2010 *Mathematics Subject Classification.* 90B18, 68M20.

1998 *CR Categories and Descriptors.* C.4 [**Computer Systems Organization**]: Performance of Systems – *Measurement techniques*; I.6.3 [**Computing Methodologies**]: Simulation and Modeling – *Applications*.

Key words and phrases. video-audio streaming, testing, rate control, congestion control, FFMPEG.

results and it also tries to capture true randomness of real-world experiments by using the available networking infrastructure as test environment, not a simulated network one. The name of this framework is **VAST**, which is an acronym for **Video-Audio Streaming Tester**.

The paper is structured as follows. In Section 2 we present other testing tools available on the Internet and related to VAST. Section 3 underlines the goals of VAST while the next section introduces the architecture of the proposed package. Section 5 deals with the pluggable structure of a VAST experiment. Section 6 outlines VASTs strengths and weaknesses compared to the testing tools presented in Section 2 while Section 7 presents our conclusions and intended future work.

2. Related Work

Ns-2 [8] is a well known network simulation package which is focused on congestion control. Although it is very powerful and has implementations for a wide set of transport protocols, it does not capture the possible benefic influence of real-world audio-video data on the network transport control algorithms and, conversely, it does not reflect the effect of network transport control algorithms on the audio-video stream perceived by the end user.

Another testing package for audio-video streaming is ViTooKi [10] which is an open source tool for developing real world audio-video streaming applications. One of the disadvantages of ViTooKi is that it has very low support for media-friendly and TCP-friendly congestion control and congestion control in general.

VLC [11] is yet another open source streaming project similar to ViTooKi which also implements limited congestion control (besides the one implemented by TCP in the OS kernel).

3. Package Goals

By using the VAST package, we want to be able to test in real network conditions (not in simulated environments) the performance of various congestion control as well as stream rate control algorithms for audio-video streaming in best-effort networks. Their performance will be measured by objective metrics (e.g. number of lost packets, throughput, buffer fill level etc.) but also by pseudo-subjective metrics (artifacts on the rendered video etc.).

Another goal of VAST is testing the performance of caching strategies for multimedia (audio-video) objects in audio-video proxy-caches [1].

Finally, we want to have the possibility to test in real-world conditions, algorithms for adapting multimedia streams at application level (e.g. temporal adaptation, grayscale reduction, spatial reduction, transcoding etc.).

The main advantages of the proposed package over other network simulators (e.g. ns-2) are the following:

- the data source is a real audio-video stream and is not random data from memory; because of this the distribution of the audio-video data can influence the semantics of the rate control algorithms;
- the performance of the transmission rate control and stream rate control algorithms can be visually assessed/quantified.

Because it uses real-world conditions in experiments, VAST has also some drawbacks compared to network simulators and the most important is the difficulty to assess the test results due to many noise sources generated by randomness in the experiments conditions (which is a characteristic of all real-word experiment packages). This difficulty is counteracted by the pluggable architecture of VAST which is described in the next section.

4. The Vast Architecture

Generally speaking, there are two approaches in network testing: *the deterministic approach – simulations* and *real-world experiments.*

In network simulations, it is easier to measure, predict and interpret results and no costly network infrastructure is needed for deployment. The main weakness of network simulations is the lack of the ability to capture true randomness and transient events. The exponent of network simulators is represented by ns-2 [8].

In real-world experiments, it is harder to measure and interpret results (i.e. difficult to isolate causes) and an elaborate network infrastructure is required for deployment. The advantage of real-world experiments is the possibility to capture the randomness in the real-world and transient events. Examples of tools useful for audio-video streaming testing are ViTooKi (The Video Toolkit) [10] and VideoLan VLC[11].

VAST tries to combine the advantages of both approaches mentioned above. On one hand, it lies in the real-world experiments category because it relies on real-world network infrastructure and network conditions, thus having a testing power larger than a simulator. On the other hand, through its flexible pluggable architecture, VAST provides an easy way of assessing test results, as do simulators. This is because in each experiment, many components of VAST can be *plugged-in* or *plugged-out* from the experiment, thus adding or removing a noise/randomness source from the experiment. By subsequently eliminating a noise source from the experiment, the test results can be better separated and more easily assessed.

The intended VAST testing architecture is shown in Figure 1. The VAST package can be deployed on a LAN which is linked by several routers. VAST has different data flow sources which can be monitored when competing for network resources: TCP flow source, fixed (parametrized) transmission rate UDP source and audio-video streaming sources/servers. Each VAST source type has a corresponding receiver. Also an audio-video proxy/cache module can be deployed inside the VAST network.

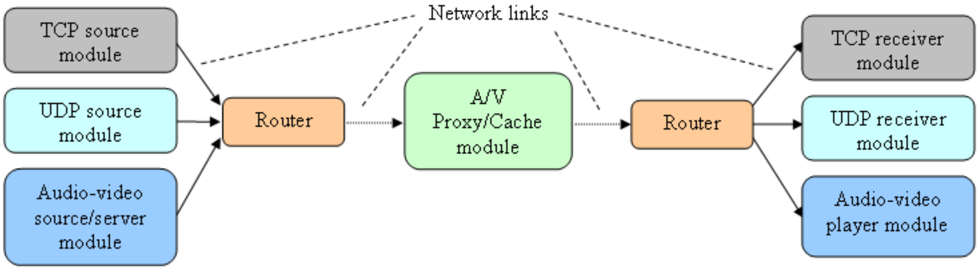

Figure 1. The VAST testing architecture

The VAST A/V player architecture is depicted in Figure 2 and has the following major components: *IO System* for interfacing with the network, *Session Management* for controlling the streaming session using standard streaming protocols and also our experimental streaming protocol (SMSP – Simple Multimedia Streaming Protocol), a *Codec Module* based on FFMPEG [7] for decoding audio-video data and a *Rendering Module* based on the SDL library [9].

The *IO System* is the flow control component. On the client side (i.e. A/V player) the *IO System* consists of feedback producers for various flow control protocols. On the server side, the *IO System* contains flow control components for computing/updating the transmission rate and enforcing the computed transmission rate. The A/V streaming session can use several flow control protocols: DCCP [4], TCP and newer flow control algorithms build on top of UDP: NoCC (No Congestion Control - an algorithm for sending the

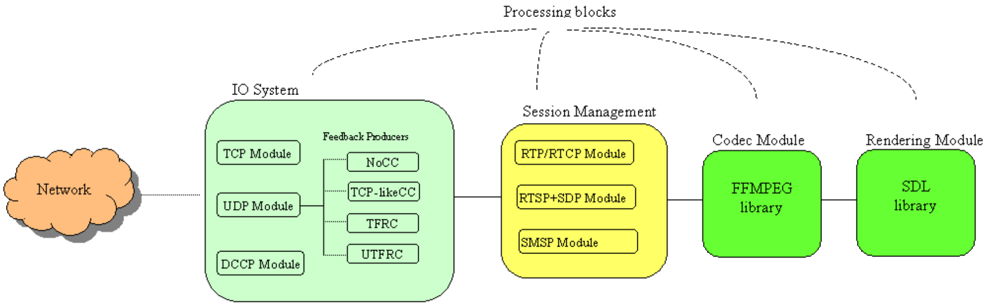

FIGURE 2. The VAST Audio/Video Player architecture

audio-video data at constant rate), TCP-like congestion control, TFRC [3], UTFRC [5, 6].

The *Session Management* component provides implementations for standardized streaming protocols (RTP, RTCP) and streaming control protocols (RTSP, SDP) and also our experimental SMSP – Simple Multimedia Streaming Protocol, whose header consists only of a sequence number and a timestamp.

In Figure 3 the architecture of the A/V streaming server is depicted. It has similar components as the A/V player.

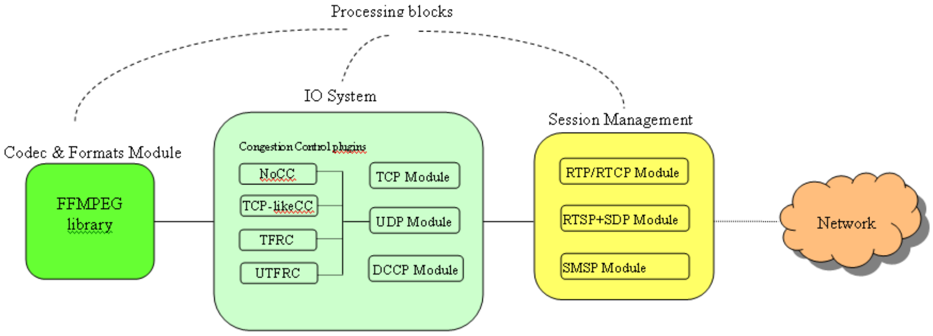

FIGURE 3. The VAST Audio/Video Server architecture

5. THE PLUGGABLE ARCHITECTURE OF A VAST EXPERIMENT

It is important to note that almost each component of the server or client can be disabled in a VAST experiment setup which offers flexibility in isolating causes (noise sources). This is very useful in experiments in which the obtained results can not be explained due to the interconnections of the experiments components and other noise sources. In such a case each component can be independently and subsequently disabled (actually, it is not fully disabled, but replaced with a dummy one which does not generate too much noise because of its simplicity – it leaves data unchanged), until each noise source is identified. For example, if we do not want to consume CPU for the actual rendering of the video or we want to reduce the concurrency randomness at the client, we can disable the SDL-based *Render Module* and the *Codec Module* from the Player and no decoding or display will happen at the Player. Another example is that full fledged session management protocols like RTP/RTC, RTSP and SDP can be disabled and a simple custom header (SMSP) can be used instead, thus reducing the degree of concurrency and complexity of the data flow. Also, the congestion control module can be disabled at the server and replaced with a NoCC, one which transmits multimedia data at a constant rate, thus eliminating variations (randomness) in the transmission rate. To reduce the computing load at the server, the *Codec Module* can be disabled and random data from the memory can be sent instead to the client.

In essence, the architecture of a VAST streaming server or player is a highly configurable and a pluggable one.

6. VASTs advantages and disadvantages

Because of its pluggable and flexible architecture, the VAST package has several advantages over existing audio-video streaming simulators or streaming solutions:

- the ability to test new media-friendly congestion control algorithms which are influenced by the data distribution in the audio-video stream; this is because an audio-video source uses a real audio-video stream instead of random data from memory;
- the performance of the congestion control and stream rate control algorithms can be visually assessed (in the quality of the stream perceived by the client);
- testing specific proxy-cache operators can be done in real-word network setups/conditions;
- the possibility to isolate noise sources in the experiment results by enabling/disabling various components in the streaming server or client by taking advantage of the pluggable architecture of VAST.

Due to the fact that it is still in the development state, VAST has the disadvantage that is not as stable as a classic simulator like ns-2 which has been used for a number of years. Also because it is a real-world experiment tool, it does not have the deterministic properties of a network simulator, thus the results can not be as easily assessed as in the case of a simulator. And because real network infrastructures are used in experiments, the setup for the experiment can take longer than in the case of simulators.

7. Conclusions and Future work

We presented in this paper VAST, a video-audio streaming tester package based on FFMPEG. The strength of VAST comes from its highly pluggable and configurable architecture.

The current state of the implementation is:

- functional local video player;
- simple video streaming server;
- partial congestion control implementation (TFRC/UTFRC);
- naive TCP/UDP sources/receivers implemented.

As future work, we intend to complete the implementation of VASTs modules and compare the VAST package with the previously developed *vpcSim* [2] tool.

References

[1] Claudiu Cobarzan, *Distributed Video Proxy-Caching in High-Bandwidth Networks*, PhD thesis, 2008

[2] Claudiu Cobarzan, Doriana Dorutiu, *vpcSim: A Video Proxy-Cache Simulator*, In Proceedings of the Symposium Colocviul Academic Clujean de Informatica, pp. 135-140, 2005

[3] Sally Floyd, Mark Handley, Jitendra Padhye, Joerg Widmer, *TCP Friendly Rate Control*, RFC 3448, January 2003.

[4] Eddie Kohler, Mark Handley, Sally Floyd, *Datagram Congestion Control Protocol (DCCP)*, RFC4340, 2006

[5] Adrian Sterca, *Congestion Control in Streaming Protocols*, PhD thesis, 2008

[6] Adrian Sterca, *UTFRC - Utility-driven TCP-Friendly Rate Control for Multimedia Streams*, in Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, IEEE Computer Society, Germany, February 2009.

[7] *** FFMPEG library, `http://ffmpeg.org/`

[8] *** The ns-2 simulator, `http://www.isi.edu/nsnam/ns/`

[9] *** SDL library – Simple DirectMedia Layer, `http://www.libsdl.org`

[10] *** ViTooKi – The Video Tool Kit, `http://vitooki.sourceforge.net/`

[11] *** VLC, `http://www.videolan.org/vlc/`

Babes-Bolyai University, Faculty of Mathematics and Computer Science, 1
M. Kogălniceanu St., 400084-Cluj-Napoca, Romania

E-mail address: {`forest,claudiu`}`@cs.ubbcluj.ro`