# Automated Performance Testing of End-to-End Streaming Solutions over HbbTV Architecture

Cristinel Gavrilă, Csaba-Zoltán Kertész
Department of Electronics and Computers
Transilvania University
Brașov, Romania
cristinel.gavrila@xu.unitbv.ro, csaba.kertesz@etc.unitbv.ro

***Abstract*—This paper presents an automated test execution environment designed to analyze the accessibility, reliability and streaming performance of an end-to-end Hybrid broadcast broadband TV (HbbTV) solution. Its purpose is to provide the means for the TV providers to test their HbbTV solution by simulating real-world scenarios and online functioning in a local, offline environment, before making it available for the end users. This way, common problems like server overload, poor streaming quality and HbbTV incorrect functionality can be foreseen and corrected.**

***Keywords*—HbbTV; Automated testing; Streaming media; Web services; Network conditions simulation**

## I. Introduction

The evolution of electronic systems technology over the last few decades had a major impact on the television systems. The first significant evolution in the television technology was the transition from analog to digital television (DTV), which opened a new perspective in the matter of services which can be provided. DTV represents the key factor in the transition from common television sets to modern DTV devices which offer new features and a highly improved media experience. Furthermore, this major breakthrough offered the possibility of combining the broadcast and the broadband environments, which were, until now, mutually exclusive. This was the base factor for introducing a new concept in the DTV devices world, the hybrid broadband/broadcast TV concept, and defining a new standard known as HbbTV [1].

The HbbTV specification [2] is based on already developed standards such as Open IPTV Forum (OIPF), Digital Video Broadcasting (DVB), Consumer Electronics Association (CEA) and World Wide Web Consortium (W3C) and has as a main goal: the seamless merge of Television and Internet. HbbTV enables the delivery of multimedia content and metadata in a mixed broadcast/broadband ecosystem, thus providing the consumer both DVB related and IP-based services on the same end-user device. Among the services delivered through HbbTV the most important are video-on-demand, live streaming, catch-up services, enhanced teletext, Electronic Program Guide (EPG), games, voting and advertising.

A generic end-to-end HbbTV solution architecture comprises multiple interconnected stand-alone systems, working together to provide the means for increasing the range of services offered by a TV provider and, at the same time, to enhance the end user interactive experience. Amongst the components of the end-to-end HbbTV solution, the most important are the web server where the HbbTV application resides, the streaming server which offers video-on-demand and live streaming services and the HbbTV compliant device.

The HbbTV specification also defines the necessity of compliance testing for all components of the full HbbTV chain. However, these compliance test cases focus on the individual components only, because usually the different servers, devices and TV provider services are independently developed by different providers [3], [4]. While separate components can pass the compliance test, a full end-to-end testing is also needed, to make sure that these components do work together in all situations. Such testing environments were presented by Lukač et. al. [5] and Köhnen et. al. [6] proving the necessity of further testing of complete HbbTV solutions.

This paper presents a local test environment for end-to-end HbbTV solutions designed not only for HbbTV device standard compliance verification but also for streaming performance analysis over different simulated network scenarios, multiple streaming protocols and using multiple streaming servers. The main purpose of the test system is to simulate the real-world environment in which the HbbTV solution will be deployed, replicate real use case scenarios and retrieve certain parameters from the HbbTV device. In this manner a performance and behavioral analysis can be conducted before making the HbbTV solution available in the marketplace, thus ensuring that it is capable of providing high quality services for the targeted audience. The major advantage compared to other existing implementations is that streaming performance metrics such as start-up time, connection time, initial buffer fill time, rebuffer ratio and average frame rate are also provided. These parameters quantify the quality of the end user experience when accessing a video-on-demand or live streaming service provided by the HbbTV solution and are decisive when selecting the streaming infrastructure in aspects of hardware equipment and software solution.

## II. Proposed Test Environment Architecture

The test environment is a complex system in which the focus of the testing process is not only on the HbbTV device, but

978-1-5090-1993-9/16/$31.00 ©2016 IEEE

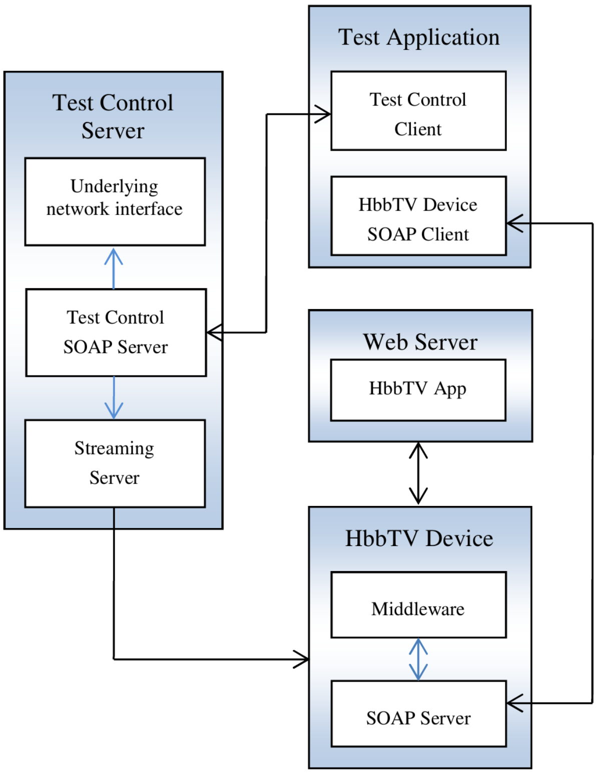

Fig. 1. Automated test environment for end-to-end HbbTV solutions architecture.

also on other entities such as streaming servers and delivery network. Its purpose is to define a viable and cost effective proceeding to perform functional, end-to-end and stress tests by being able to control and monitor all the components of the end-to-end solution chain.

The test environment architecture is presented in Fig. 1. It consists of the end-to-end HbbTV solution and the test system. The test system has a modular design, each of its components being linked to a key point and performing a specific set of tasks. This approach offers an increased flexibility and scalability for the test system. New functionality can be added with no significant effort, and it can be easily adapted to work with different architectures of the end-to-end HbbTV solution. The communication between the components is IP based, thus reducing the dependency on external hardware components such as infrared transmitters or DVB modulators.

The core element of the test system is the Test Application, a central unit from where all the system components are controlled and monitored. Its main tasks are to execute the test suite, manage the test system components and generate reliable reports. In order to simulate real-world conditions, the Test Control and the HbbTV device SOAP (Simple Object Access Protocol) servers are used, each one providing a specific set of features. The network properties and the streaming servers are controlled by the Test Control Server. It receives commands from the Test Application and responds with the command execution status via the SOAP protocol. The HbbTV device SOAP server is used to process requests which simulate key press events, thus enabling IP based remote controlling of the HbbTV device and also to provide information regarding its status and streaming performance metrics.

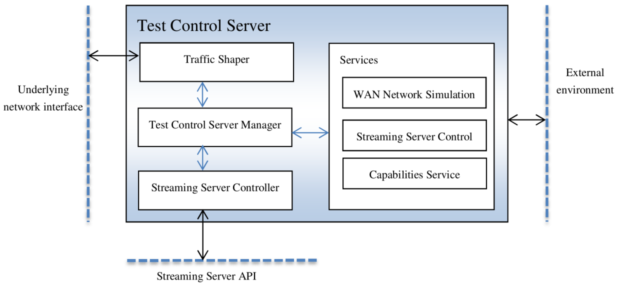

Fig. 2. Test Control Server

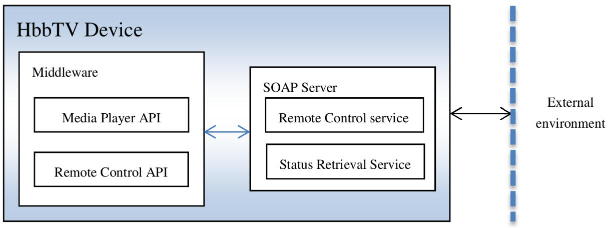

Fig. 3. HbbTV device SOAP server structure overview.

### A. *Test Control Server structure overview*

The Test Control Server resides on the same machine where the streaming servers operate and acts as a network interface manipulator and streaming server controller. An overview of its structure is presented in Fig. 2. The services provided by this server are: WAN network simulation, streaming server control and capabilities service.

The component which handles the WAN network simulation is the Traffic Shaper. This is accomplished by altering the underlying network interface properties such as bandwidth, network delay, packet corruption, packet loss and jitter. The streaming servers are controlled by the Streaming Server Controller component which accesses the streaming servers control API, thus being able to perform operations such as start, stop, retrieve available video-on-demand files or check a live stream health. Those components are controlled by the Test Control Server Manager which is also responsible for initializing and starting the SOAP and the HTTP server.

### B. *HbbTV device SOAP server*

The means for controlling and retrieving parameters from the HbbTV device under test (DUT) are provided by the SOAP Server integrated in the device software. An overview of this component is presented in Fig. 3.

The services provided are the remote control service and the status retrieval service.

The IP based remote controlling is preferred because it is less prone to external interference and also eliminates the

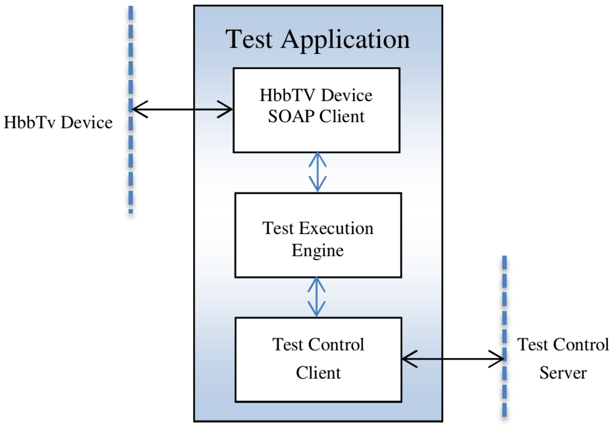

Fig. 4. Test application structure overview.

dependency on external IR devices. The status retrieval service has a major role in the test system and offers information regarding the overall device state and streaming performance metrics by accessing the device middleware API. The advantages of this approach are that it is noninvasive, easy to integrate and the fact that an external component for analyzing the audio and video output of the DUT is not needed.

The retrieval of streaming performance metrics is highly dependent on the middleware implementation. In general, the middleware API provides device status retrieval functionality such as player state or web browser state, but support for retrieving specific streaming performance parameters such as player start-up time, connection time or initial buffer time is not always available. This obstacle can be overcome by monitoring the elapsed time between player state changes.

### C. Test Application

The test application is the central unit of the test system. Its structure is presented in Fig. 4. It consists of:

- a test execution engine which selects, loads and executes the test suite;
- HbbTV Device SOAP Client which ensures the communication with the HbbTV device;
- Test Control Client which ensures the link with the Test Control Server

The Test Application receives as input parameter an XML file in which the test suite is defined. Defining the test suite is the only required user input and after this step is completed, the Test Application is ready to initialize and perform the testing process.

## III. Running the Test Suite

Test suite definition is XML based and can contain one or multiple test cases. The outcome of each step in the test is compared with the predicted outcome and the result (failed or passed) is registered in the test log. A failed step in the test leads to a test case failure and information regarding the failure is registered in the test log. Each test case can be configured to use a certain streaming server, a specific network scenario and desired number of streaming server simultaneous connections. This feature confers flexibility in test suite definition, thus increasing the test diversity. After the definition of the test suite, the testing process can be started. The final outcome consists of a test report specifying the result and the streaming performance metrics for each test step.

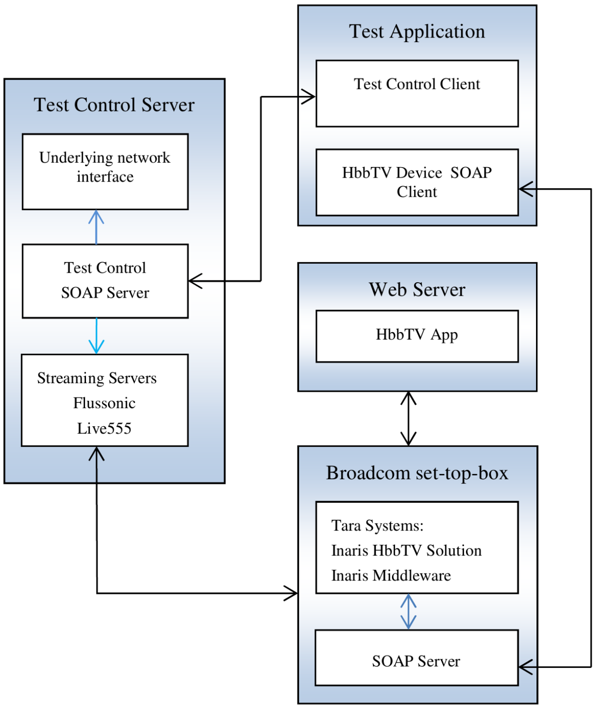

Fig. 5. Test environment for an experimental end-to-end HbbTV solution.

## IV. Experimental Results

The architecture, behavior and performance of an end-to-end HbbTV solution has a major impact on the user experience. Assuring a seamless viewing experience and correct functionality requires intensive testing.

To demonstrate the capabilities of the proposed testing solution, the test environment for an end-to-end HbbTV solution has been setup in a local host medium with the architecture presented in Fig. 5.

The hardware equipment used for setting up the test environment consists of: a Linux machine containing the streaming servers, the Test Control Server and an Apache HTTP server, an HbbTV compliant set-top box and a Windows machine on which the HbbTV Test Application resides. For the HbbTV device a Broadcom based set-top box is used, running the Inaris DVB/IPTV Middleware provided by TARA Systems GmbH. Video streaming services are provided by Flussonic Streaming Server made by Erlyvideo company and Live555, an open source solution. The test suite was defined to replicate a real-world scenario, where the end user starts the HbbTV application, navigate through its menu and accesses live streaming and video on demand services. For each test case the desired streaming server is selected and different impairments

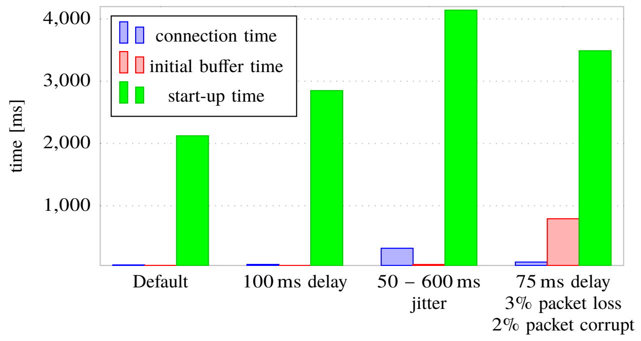

Fig. 6. Performance of MPEG-TS live streaming over HTTP

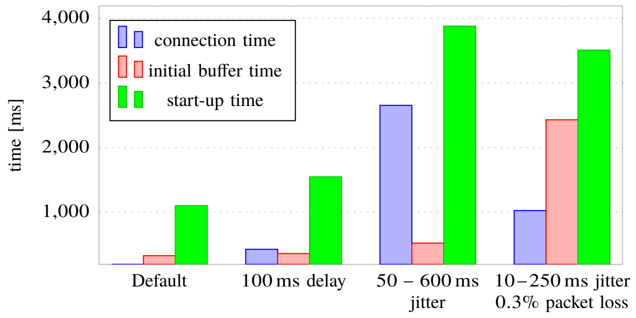

Fig. 7. Performance of video-on-demand streaming over HTTP

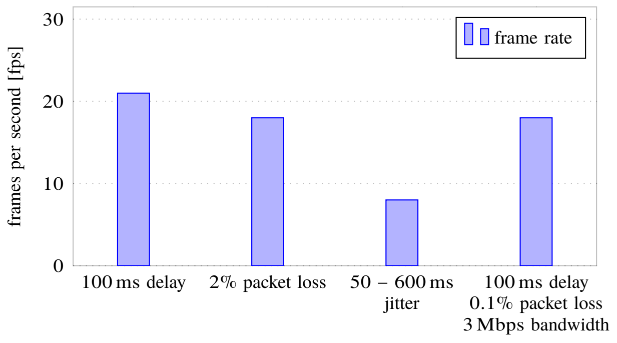

Fig. 8. Average framerate of video-on-demand streaming over RTSP

TABLE I
Generated Test Report

| Streaming Performance Test | | | |
|---|---|---|---|
| ***Streaming performance metrics*** | ***Value*** | ***Streaming Server*** | ***Result*** |
| Total play time [ms] | 120000 | Flussonic | PASSED |
| Start-up time [ms] | 2416 | | |
| Connection time [ms] | 156 | | |
| Initial buffer time [ms] | 2173 | | |
| Rebuffer ratio [%] | 0 | | |
| Average framerate [fps] | 21 | | |

or combinations of impairments are injected in the IP traffic flow in order to simulate the conditions that occur as traffic traverses large IP networks.

Table I presents the results of a test run where Flussonic is used as streaming server, a network scenario with 0.1% packet loss, 50 ms delay and 8 Mbps bandwidth limitation is established and the video-on-demand service is accessd. Some example statistics regarding different timings obtained from the experimental setup simulating live streaming and on demand streaming are presented in Fig. 6 and Fig. 7 respectively. Here, a default network scenario, with no impairments, is used as reference, separate traffic impairments (certain delay and jitter) are used to analyze the effect of each impairment on the streaming performance and also, to simulate a real-world network scenario, a combination of impairments (delay, jitter, packet loss, packet corruption) is used. Another statistic regarding frame rate variation in different network conditions is presented in Fig. 8. Such statistics can be used to check if the end-to-end solution meets the QoS (Quality of Sevice) demands.

## V. Conclusion

This paper presents an automated performance testing system, highly flexible and easy to integrate, able to provide useful and reliable information in aspects of behavior and performance analysis of end-to-end streaming solutions over HbbTV architecture. Its strong point is that it can simulate the real-world environment in which the HbbTV solution will be deployed, making it possible for TV providers to test their HbbTV solution before making it available in the marketplace. This provides a major advantage not only in designing a proper cost effective and QoS compliant solution targeting a defined audience, but also in ensuring a high quality end user interactive experience.

## Acknowledgment

The authors would like to thank TARA Systems GmbH for providing the HbbTV compliant device and integration support, Erlyvideo for providing the streaming server and Mindware Solutions for support and additional hardware equipment.

## References

[1] R. Malhotra, “Hybrid Broadcast Broadband TV: The way forward for connected TVs,” *IEEE Consumer Electronics Magazine*, vol. 2, no. 3, pp. 10–16, July 2013.

[2] *Hybrid Broadcast Broadband TV*, ETSI TS 102 796 V1.2.1, 2012.

[3] (2016, Jan) MIT-xperts HBBTV testsuite. [Online]. Available: http://itv.mit-xperts.com/hbbtvtest/

[4] Y. Liu, B. Du, S. Wang, H. Yang, and X. Wang, “Design and implementation of performance testing utility for RTSP streaming media server,” in *Proc. of First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA’10)*, Harbin, China, Sept 2010, pp. 193–196.

[5] Ž. Lukač, V. Zlokolica, B. Mlikota, M. Radonjić, and I. Velikić, “A testing methodology and system for functional verification of general HbbTV device,” in *Proc. of 2012 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, Nevada, USA, Jan 2012, pp. 325–326.

[6] C. Köhnen, C. Köbel, and N. Hellhund, “A DVB/IP streaming testbed for hybrid digital media content synchronization,” in *Proc. of 2012 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, Germany, Sept 2012, pp. 136–140.