On the Use of Thin-client Set-Top Boxes for IPTV Services

Tiago Cruz, Paulo Simões, Pedro Cabaço,
Edmundo Monteiro
DEI-CISUC – University of Coimbra
Coimbra, Portugal

Fernando Bastos,
Alexandre Laranjeira
PT Inovação
Aveiro, Portugal

***Abstract*— together with the developments in terms of broadband connectivity and IP-based service consolidation, the introduction of cloud technologies is opening a whole new window of opportunity for service and telecommunications operators alike, to improve and expand their service portfolio.**

In this line of thought, this paper presents an innovative proposal to improve the delivery of IPTV services for residential users. It departs from the existing model, based on fat Set-Top Box (STB) appliances designed for specific IPTV frameworks, going instead for a solution that turns STBs into stateless devices, with the whole service interface being moved to the provider infrastructure.

This thin-client STB-based service delivery model for IPTV, also includes the necessary management mechanisms to provision and configure the involved components. Overall, this approach is designed to enable providers to improve and simplify their IPTV delivery infrastructure, while maintaining backwards-compatibility with their existing Operations Support Systems.

***Keywords*— Set-Top-Boxes, IPTV, CWMP, Home Networks.**

I. Introduction

Cloud computing is the result of several technology advances, such as virtualization and the pervasive availability of broadband network access, enabling a multitude of new services and support infrastructures. Steadily, this paradigm shift is creating the opportunity to improve and optimize existing services, by consolidating and rationalizing resources. In this sense, cloud and converged technologies are helping to change not only the content and service backend, but also the whole end-to-end support infrastructure, up to the home consumer edge, with benefits ranging from cost reductions to an enhanced end-user experience.

As equipment and resources are getting streamlined and virtualized, a considerable part of the classic software and device ecosystem is also evolving towards a service-based model – this is already happening with *n-play* bundles (such as *triple-play* offers) and cloud services, which are displacing traditional split-medium communication and service delivery models in favour of converged, *everything-over-IP* approaches – IP-based Television (IPTV) is one of such cases.

The possibility of delivering TV content over IP-based streams was one of the main enablers for the *n-play* service convergence trend that is increasingly becoming commonplace everywhere across households. However, the predominating IPTV service delivery model has yet to fully embrace the cloud paradigm and the advantages it has to offer. In fact, conventional IPTV frameworks [1] are still based on a server-consumer paradigm, also requiring some sort of a fat-client appliance with dedicated firmware – a Set-Top Box (STB) - to decode streams and interface with the service. This model is somewhat reminiscent of the decade-old client-server concept, somewhat suggesting that it might exist room for its improvement.

In this line, this paper proposes a solution that leverages a set of different technologies belonging to different domains, such as virtualization, distributed management or thin-client computing, to present an updated vision of what IPTV services could evolve into. The main rationale behind the presented solution is based on simplifying and streamlining the STB, using remote boot capabilities to transform it into a stateless and robust thin-client device with minimal hardware and devoid of any firmware, with the service logic, functionalities and interface being moved to the service provider infrastructure.

For providers, this approach has two main benefits: first, it aims at reducing IPTV infrastructure Capital Expenditure (CAPEX) and Operating Expenditure (OPEX), by targeting the STB as a cost factor; second it enables the possibility of continuously update and improve the IPTV service with a much reduced effort, enabling a short time-to-market (TTM) for the introduction of new functionality and features. These benefits also translate to the end-user, in terms of convenience, service cost and improved user experience.

This solution also incorporates management mechanisms based on Broadband Forum's [2] CPE WAN Management Protocol (CWMP) [3], the *de facto* industry standard for remote management of customer premises devices and services, on residential network environments. This is of particular importance for operators, since it enables them to use their already existing Operations Support Systems (OSS) to manage the proposed solution.

The rest of this paper is structured as such: the rationale for the proposed solution is discussed on Section II. Section III presents the fundamental components of the proposed solution, namely, its management mechanisms, the remote boot support framework and the thin-client STB architecture itself. Concept evaluation including aspects such as the validation methodology and testing environment are discussed in Section IV. Section V analyses related work and Section VI concludes this paper.

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

II. MOTIVATION

The most common and widespread IPTV service delivery model (Figure 1) is based on a complex environment, frequently relying on a single provider for transport, content and service interfaces, often the same entity that provides connectivity services. This is due to circumstances that favoured telecommunications operators for a long time, as they were better positioned than anyone else to control the critical IPTV service delivery path and provide end-to-end Quality of Service (QoS) guarantees for service traffic.

Together with IPTV framework providers, communications operators helped turn the IPTV service ecosystem in a closed environment, where proprietary technologies and standards are used to protect both content and revenue, a situation that is particularly well-illustrated by the specific case of IPTV Set-top Box (STBs) appliances.

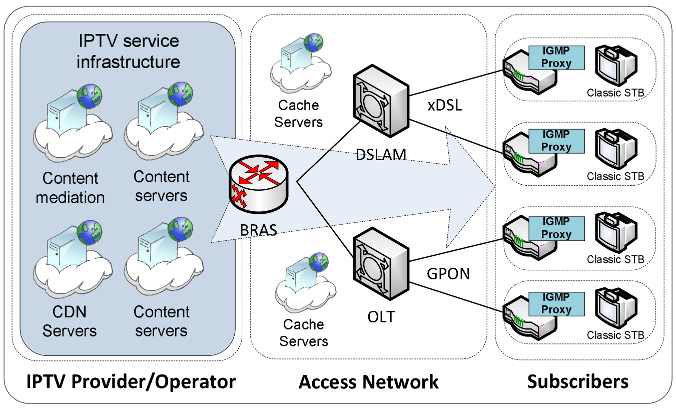

Figure 1: A typical, operator-centric, IPTV service architecture model.

STBs are embedded devices with a considerable degree of complexity that are used to receive TV and media content that is streamed from a service provider, and subsequently decoded and played in a TV set. While some TVs incorporate features that could be used to provide almost the same functionality as standalone STBs (at least theoretically) they are not compatible with existing IPTV frameworks – a situation that will probably remain unchanged for the time being, due to the providers' need for protecting IPTV content and service interfaces. As part of its business model, IPTV and triple-play operators usually bundle their own STBs (whose firmware is based on proprietary technologies and locked to a specific media platform, like Microsoft's Mediaroom [4]) with their service offers, in an attempt to protect content, interfaces and preserve digital rights. This has some drawbacks, namely:

- **Cost.** There are studies that pinpoint STBs as responsible for approximately 70% to the overall CAPEX [5]. Due to licensing policies, this value is influenced (in some cases, almost linearly) by the number of subscribers. By comparison, middleware (20%) and Digital Rights Management (DRM - 7%) account for the second and third highest overheads, respectively.
- **Management overhead.** Managing a large number of STBs is not an easy task. Even considering the fact that modern STBs are designed from scratch to be as reliable as possible, the introduction of new functionality sometimes requires sensitive operations (such as firmware updates) to be mass performed, in order to update existing equipment to the latest software/firmware version.
- **Vendor lock-in.** Most STBs are locked to a particular vendor and IPTV framework. This has the effect of locking the IPTV operator to a specific technology provider, making impossible to switch to another one without replacing existing STBs. While several different IPTV frameworks can coexist on the same infrastructure, it is not a cost-effective solution for the service provider.
- **Flexibility**. The introduction of new features is always dependent on the specific characteristics of both the IPTV framework and the STB equipment. This may have a negative impact in terms of the TTM needed for the introduction of new functionality to service customers, a situation further aggravated by the fact that providers are constantly competing among them to be the first to announce such new features.

In order to address these issues, this paper proposes an alternative, simpler STB architecture that makes use of service and device virtualization techniques to improve the existing IPTV service model. This solution proposes streamlining STBs, by turning them into stateless and robust thin-client devices with minimal hardware and devoid of any firmware or persistent local state. This is possible because, at each reboot, the entire operating environment for each STB is downloaded over the network, directly from the provider infrastructure (eventually using a distributed caching mechanism or even a Content Delivery Network to optimize the process).

Once the STB is turned into a thin-client appliance, the service logic, functionalities and interface can be moved to the service provider infrastructure. The IPTV service interface, which is streamed straight from the IPTV provider infrastructure, is rendered on a full-screen web browser component using HTML and Javascript [6].

The use of modern web technologies provides a standard and feature rich environment for developing interactive and dynamic interfaces that can be rendered on several kinds of devices. Particularly, HTML5 [7] includes new features that were previously only possible using other technologies (proprietary plugins and APIs - Application Programming Interfaces), such as multimedia support, while maintaining consistency with prior versions.

This allows the STB to render the service interface in a transparent way to the end-user, which interacts with the device as it already does with existing STBs. To ensure scalability, multicast video reproduction is supported using an Internet Group Management Protocol (IGMP) [8] proxy component on the Residential Gateway (RGW), for multicast media streams together with a media player component or transparent proxy on the STB (if HTML 5 video capabilities are used). In the latter case, the transparent proxy component is used to overcome two limitations of HTML5 video: lack of support for DRM and multicast video. Nevertheless, the definition of mechanisms for DRM support on HTML5 is already in its early stages, being pushed by companies like

Google, Microsoft and Netflix [9].

For the final customer this change will be completely transparent. For the IPTV operator, this solution brings numerous advantages, in terms of cost, flexibility (customers are always with an updated system). There is also the fact that it makes it easier to integrate the IPTV platform with other applications (e.g., social networks). Moreover, this framework also helps extending the operational life of STB platforms – as long the hardware is able to process and decode media streams, it will be always up to date in terms of service features and interfaces since its operating environment is completely refreshed at each reboot.

This thin-client STB concept can be used by both telecommunications operators and Over-The-Top (OTT) third-party service providers – yet, for the latter, operator involvement will be required to provide infrastructure and management support for both QoS, routing/multicast and service traffic isolation.

III. Proposed Architecture

This section presents and describes the proposed IPTV STB framework, with special relevance to the building blocks that constitute it. Starting with a brief introduction to the CWMP management framework, it goes into describing the implementation of its remote boot capabilities. The final subsection will delve into the design and operation of the thin-client STB.

A. CWMP-based Management

Broadband Forums' CWMP protocol suite is the established standard for secure device and service management on broadband environments, being designed to enable secure auto-configuration, dynamic service provisioning, diagnostics, software/firmware management and status/performance monitoring of devices and associated services. It provides a management API of Remote Procedure Calls supported by a set of extensible data models defined by related standards such as TR-106 [10] or TR-157 [11]. The standard data model for a CWMP-capable device follows a common set of requirements for which the detailed structure, hierarchically organized like a directory tree, depends on the nature of the device. Data model information is structured using objects and parameters - each object is a container for other objects and parameters, the latter storing the configuration properties of the managed device.

Specifically, CWMP already has provisions for an STB data model, in the form of TR-135 [12], which was implemented for the proposed STB management agent. For this purpose, a previously developed modular agent integration framework [13] was used. This agent is included on the downloaded payload.

The adoption of CWMP for the proposed IPTV framework enables operators to use their already existing OSS infrastructure to handle service and device management for the home LAN. Also, one of the characteristics that make CWMP particularly fit for management of devices and services inside the residential LAN has to do with its operation model: a managed device always initiates management sessions, either directly or by request of the management server (designated by Auto-Configuration Server or ACS). This has the benefit of better coping with firewalls and other mediation mechanisms.

B. Managed cloud boot

As an alternative to the use of stateful firmware (as it happens with conventional STBs) the stateless STB is a thin-client device that is able to download its entire operating environment from the network, at boot. This is possible due to a previously developed technology that enables remote OS boot capabilities on broadband access network environments [14], using the Preboot eXecution Environment (PXE) protocol [15].

PXE is the *de facto* standard for network boot firmware, allowing a device to download and execute an agent or a complete operating system image – the Network Bootstrap Program (NBP) – over a LAN at boot time, for deployment, diagnostic or bare metal recovery. PXE can also be used to support completely stateless thin-clients [16] whose operating environment is downloaded from the network when powered up, instead of using local firmware (see Figure 2).

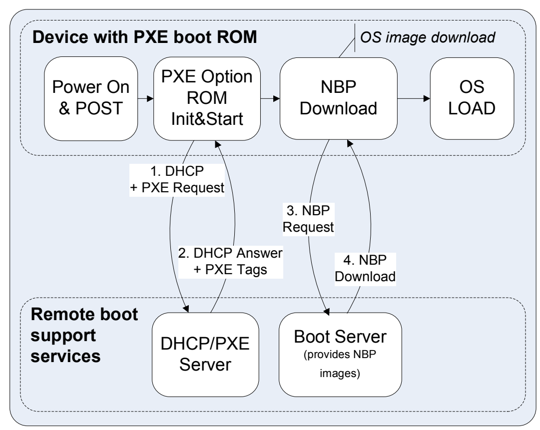

Figure 2: PXE-based network boot process

PXE operates in close integration with the Dynamic Host Configuration Protocol (DHCP) [17], using specific embedded option tags to configure the IP stack of the Boot ROM and to locate the location from which the NBP must be downloaded. For this reason, the DHCP server of the RGW must be adequately configured to provide the PXE firmware with the information it needs. In this scope, CWMP provides the remote configuration mechanisms for such purpose.

In this scenario (see Figure 3), CWMP is used by the ISP ACS to configure all PXE-related parameters, mapped on the CPE/RGW data model. The CWMP agent of the RGW uses this data to configure the embedded DHCP server, so that it can provide the correct option tags to the PXE boot ROM. Once the STB boots, its Boot ROM component will use DHCP-embedded information (provided by the RGW DHCP server) to locate an external boot server, from which it will download its NBP using an HTTP(S) [18] connection.

To enhance PXE operation, an ISP may also use CWMP to configure a private virtual circuit pipe in order to offer QoS guarantees to PXE, related management traffic and remote desktop services. This makes it possible to establish SLA agreements between ISPs and third-party providers of IPTV services to allow end-to-end differentiation of service traffic, therefore enabling effective OTT service delivery to STBs.

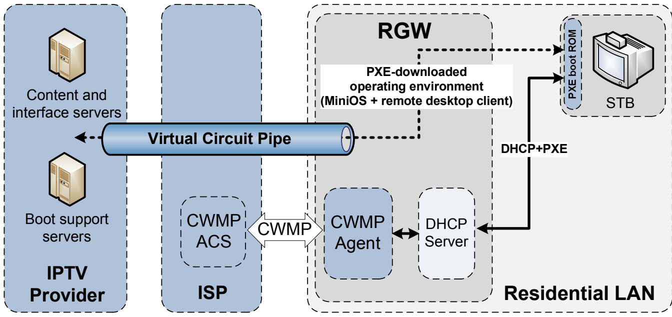

Figure 3: PXE-based STB remote network boot.

C. Stateless STB architecture

While HTML5 does provide support for embedding video content on web pages, it doesn't provide support for two features considered critical in this context: multicast/RTP streaming support and DRM. To overcome these specific limitations, we propose two different approaches: the first embeds a transparent proxy component on the STB or RGW to provide multicast RTP-to-HTTP stream decryption, buffering and translation (Figure 4), while the second (Figure 5) replaces the native HTML5 video capabilities with a plugin that enables embedding a media player (such as VideoLAN Client - VLC [19]) inside webpages.

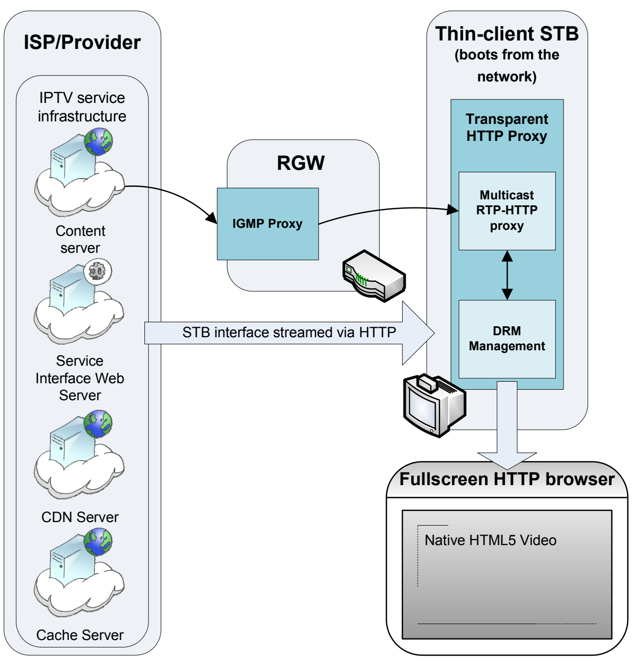

Figure 4: Thin-client STB proposal for native HTML5 video support

The prototype implementation (discussed on Section IV) was built using the second variant of the proposed architecture (Figure 5). On the ISP/Operator infrastructure side (considered the same entity for simplification purposes) there are servers for content (belonging to the provider or to a CDN) and the web-servers that provide the IPTV service interfaces, in the form of web pages, which are rendered on the STBs.

In this variant, content such as IPTV channels is provided as multicast RTP streams, which are subscribed by the Proxy IGMP component of the customer RGW, on behalf of all the consumer devices inside the residential LAN (similar to most IPTV frameworks). On the STB side, a full screen web browser provides access to the service interface, with content being rendered using an embedded or overlayed media player, accordingly with interaction needs.

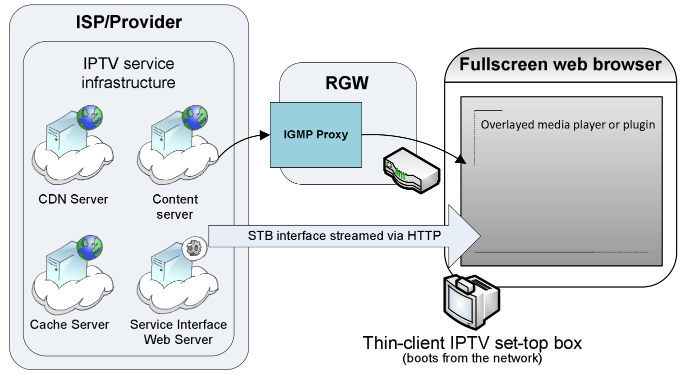

Figure 5: Prototype thin-client STB operation

In both variants, the entire operating environment is downloaded at boot, in the form of a NBP. In this scenario, the persistent device firmware footprint is reduced to the Power On Self-Test (POST) routines and boot ROM firmware. Moreover, the proposed solution might also be used to implement a hybrid approach, using NBPs for recovery or incremental firmware component management, instead of downloading the entire firmware image at each boot.

This architecture has the benefit of being compatible with most mechanisms being used by current IPTV frameworks. While it may not be compatible with specific encoding or DRM characteristics of proprietary solutions (making it unable to directly access protected content specifically provided for conventional STBs), it is able to make use of most of the already existing service support infrastructure (such as existing transport and aggregation topologies or multicast IP mechanisms) to provide an improved service.

IV. Validation

This section discusses a proof-of-concept prototype implementation of the proposed solution, its experimental validation process and obtained performance results. It also includes a detailed description of the testbed built for emulation of the end-to-end IPTV delivery scenario which supported the validation process.

A. STB Prototype, testbed and test metodology

As previously mentioned, a proof-of-concept prototype was built, using the second variant of the STB architecture. For this purpose, a streamlined NBP image with an OS and a payload complete enough to provide the desired functionality was assembled. For this purpose, a small custom Debian-based Linux distribution [20] was put together, incorporating graphics support, the Firefox browser [21] (a custom Webkit-based [22] browser is currently being developed for this purpose), the VLC [19] media player (together with its browser integration plugin), a CWMP management agent and the LIRC library (Linux Infrared Remote Control) [23] for support of infrared remote controls.

After some fine-tuning, the NBP image was reduced to 373 MB, which can be considered acceptable for our proof-of-concept prototype. However, even without any considerable reduction in terms of NBP size, there are adequate mechanisms that could be adopted to ensure adequate performance and scalability in large-scale scenarios. This would be the case for multicast-based file streaming protocols, together with distributed caching or CDN techniques – a subject out of scope for this paper.

The proof of concept STB prototype was then integrated on a purpose-built testbed (see Figure 6), designed to emulate the conditions of commercial broadband access networks based on Gigabit Passive Optical Networks (GPON) or Digital Subscriber Line (xDSL) technologies – including the operator (OSS, services) and customer premises domains. To mimic the conditions of access network links, a transparent *Dummynet* bridge [24] interconnects the customer with the ISP – in previous validation testing, this bridge proved to be capable of a sustained forwarding throughput of 800Mb/s. A PC with *Wireshark* [25] captures and measures network traffic, using a mirroring port on the Ethernet switch. A Linux system hosts the CWMP ACS and the related profile management services.

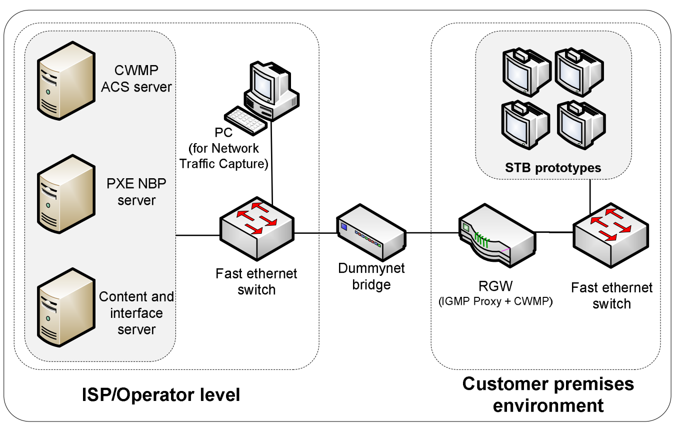

Figure 6: Testbed environment

To measure the impact of the access network conditions on the performance of the proposed solution, the *Dummynet* bridge was configured to enforce bandwidth and traffic conditions representing typical commercial offers (see Table I). The rationale for the specific configurations applied to *Dummynet* is discussed at [14]. Also, a 100Mb/s LAN scenario was established, for reference purposes.

TABLE I. BROADBAND TEST REFERENCE SCENARIOS

	Nominal bandwidth (b/s) (Down/Up)		Effective bandwidth (b/s) (Down/Up)	
LAN	100M	100M	100M	100M
GPON	100M	10M	93M	9.3M
	30M	3M	27,9M	2.79M
ADSL	24M	1M	20,04M	835K

On the operator side, there is a PXE boot server, the CWMP ACS and the web/content servers. On the customer premises side, the domestic LAN is equipped with a prototype home gateway. This is a Linux system with two network interfaces, a CWMP agent, and an IGMP proxy. It is based on a platform with modest computing capabilities (using cheap, off-the-shelf components and a single-core Atom CPU clocked at 1.6GHz, paired with 512MB of RAM) to mimic, as much as possible, the hardware constraints of typical commercial routers. The STB prototype was built using a Pentium M system (clocked at 1.6GHz), with 2GB RAM, an Infrared receiver, and a network interface with gPXE [26] firmware support.

This testbed was used to host a series of measurements, following a test methodology devised to validate the performance of three different aspects of the proposed architecture: time-to-boot, interactivity and management interfaces. For each batch, experiments were repeated 10 times. All equipment in this scenario had their clocks synchronized by an NTP (Network Time Protocol) [27] server – while there are some alternatives with higher accuracy, such as PTP (Precision Time Protocol) [28], its precision was considered enough for test purposes.

B. *Test Results for boot performance*

The first batch of tests attempted to measure boot performance in terms of latency and network traffic generated by the STB boot process. All time values were obtained from service log files, while network traffic analysis and capture was performed using *Wireshark* [25], which was also used to validate log timestamps.

The tests considered the emulated access network scenarios from Table I, in order to analyze the influence of the access network technology and its specific characteristics in STB boot times. Figure 7 shows obtained results.

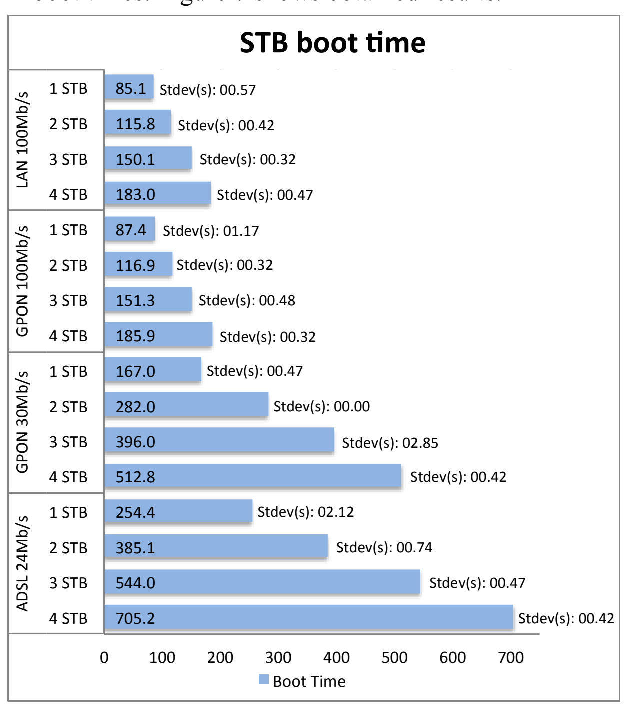

Figure 7: STB boot time (seconds)

Since the NBP payload for all scenarios is the same, tests for generated network traffic were only performed for the 100Mb/s LAN reference scenario (see Figure 8). It should however be mentioned that, in circumstances where packet losses may occur due to problems with the access network (link instability, congestion) it is expected that the transport overhead may increase, due to retransmission of lost information.

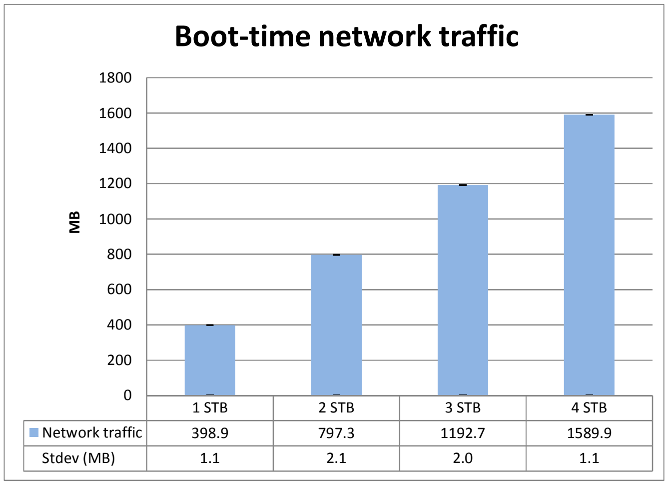

Figure 8: Network traffic generated at Boot (in MB)

Considering the initial conditions and taking into account the size of the boot image, both boot latency and generated network traffic for the STB are within acceptable values, with a low standard deviation.

As expected, boot latency is affected both by the access network technology characteristics and the number of active STBs. These tests have also shown the traffic overhead for downloading the NBP (that is, the STB operating environment image) to be small, when compared with the payload size. Nevertheless, these numbers can be improved by replacing the unicast protocol used for NBP download by a more suitable multicast-based file distribution protocol, such as FCAST [29]. FCAST makes use of cyclic transmission ("carousel") combined with forward error correction techniques, the latter being instrumental to deal with packet loss scenarios, while avoiding the side effects of other protocols that may have a potentially negative impact on scalability, such as (N)ACK packet storms,.

Also, the introduction of a deep sleep mode for the device (to reduce the number of power cycles) is a complementary technique that could also be used to improve boot performance, especially when using a unicast protocol for NBP download or in single-STB scenarios.

´In order to complete this validation section, an empirical study was performed, in order to compare the boot latency of the STB prototype with STBs from leading IPTV service providers in Portugal (Figure 9). It should be noted that these values correspond to test scenarios with a single STB, in GPON 100 scenarios. The identification of the providers was deliberately omitted, for legal reasons, as many operators do not allow for publication of comparative test results without prior consent.

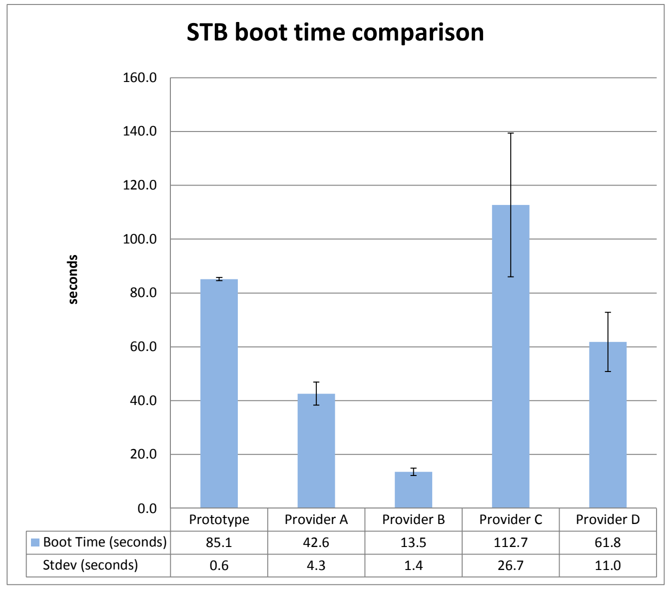

Figure 9: Boot time comparison for different STBs (in seconds)

C. *Interactive performance*

For the interactive performance evaluation of the STB prototype, we chose to analyze the latency for switching between SDTV channel streams encoded with the h.264 [30] codec (H264 High Profile Level 3 720x576 25fps). Average results are shown on Figure 10, for a 100Mb/s GPON scenario, with a single STB. Once again, results were compared with STBs used by leading IPTV services available in Portugal.

Results show the channel change latency time to remain mostly unaffected by the network access technology being used. A high standard deviation and lack of linearity shows that switching channels takes 1 to 8 seconds, on average.

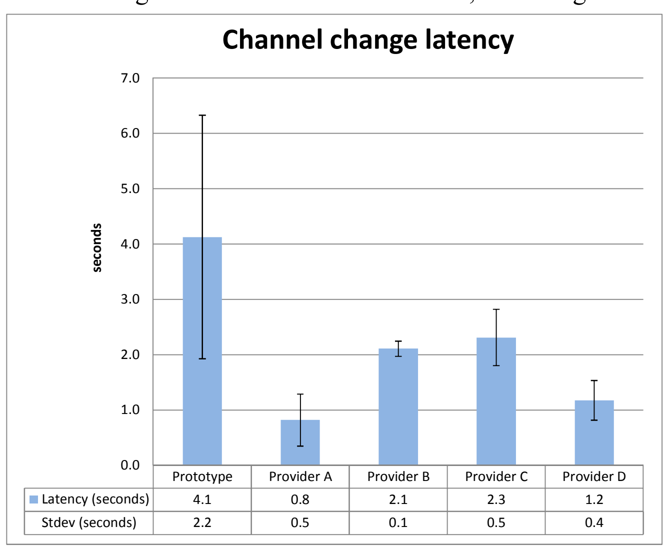

Figure 10: Channel change latency comparison for different STBs (in seconds)

While the values are not referential, they are acceptable, especially considering this is a proof-of-concept prototype. These results can be mostly justified by a lack of optimization for the media player being used, something that can be improved using cache and CDN techniques (akin to Microsoft Mediaroom's *Instant Channel Change* technology [31], which reduces channel shifting latency by using dispersed cache servers geographically closer to the customer) and specific codec parameter fine-tuning.

D. Test Results for management interface performance

The STB CWMP management interface was also evaluated, both from a functional and a performance point of view. Functional tests demonstrated the STB to behave like a conventional TR-135, CWMP-compliant STB. To the ACS, this was completely transparent, as it was possible to manage the device as any other CWMP-enabled device.

As for performance evaluation of the CWMP management interfaces, a set of common operations was chosen, namely:

- End-to end latency of the device configuration operation for changing the set of available channels (channel bundle), from the initial operation to the point where the execution status of the operation is returned.
- End-to end latency of the device configuration operation for placing the STB into maintenance mode, from the initial operation to the point where the execution status of the operation is returned.

Results have shown the average operation latency of about 1 second, from invocation to the effective execution of the operation, regardless of the access network technology in use. Considering the nature of the CWMP protocol, this means that those operations are not significantly affected by access network latency and capacity characteristics (as long they are above the acceptable minimum thresholds) as they involve few protocol interactions. In fact, that measured network traffic between the ACS and the CPE, for the two operations, was only around 45KB, on average.

V. Related Work

While the use of virtualization in IPTV scenarios has already been the subject of discussion [32], there are still few developments in this area, with probable exception of Comcast's project X1 [33] (formerly known as "Xcalibur") .

Accordingly with press releases X1 [34], uses "cloud servers on Comcast's network that allow Comcast to integrate interactive, customized apps and social media features with its traditional video services to create an entirely new television experience". While this is an important move towards the consolidation of the IPTV infrastructure, it lacks some of the features proposed on this paper, while still also using fat client STBs to support service delivery.

Another approach [35], whose fundamental concept is, to a certain extent, within the scope of the proposed solution (even if it targets desktop computing environments), proposes to offload media processing and transcoding to the cloud. While fundamentally different of what is being proposed here (media processing is still done in the STB device), it is nonetheless compatible, and even complementary, to the thin-STB paradigm hereby described.

Moreover, at research level, most STB-related developments still target the classic server-consumer fat-client STB paradigm, being centered on aspects such as energy efficiency [36], interactivity [37] or user profile customization [38]. To our knowledge, this is the first proposition of a network-boot capable, thin-client STB, specifically designed for IPTV service delivery.

VI. Conclusions and Future Work

This paper presented a new STB concept that goes in hand with a new model for IPTV service delivery. It shows how virtualization and consolidation, together with integrated device and service management can be used to improve the delivery of cloud-sourced services to the home environment.

This proposal attempted to overcome the limitations of the currently used IPTV service delivery model, by offering a device-neutral solution that offers increased flexibility and cost reduction benefits for operators and customers alike. In this sense, this solution opens a way for the introduction of new features and functionalities to the STB, namely:

- Lower CAPEX and OPEX, since the support infrastructure for the IPTV service model hereby proposed allows for a more centralized and efficient resource allocation and management. The thin-client STB greatly contributes to this since it is cheaper to operate (more reliable, less components) and manage. The absence of a local state also contributes to this, by turning the replacement of malfunctioning units into a simple device swap operation.
- Shorter TTM to introduce interface changes, new services and functionality. Neutrality of thin-client STBs provides independence from specific firmware, software components or hardware characteristics, also eliminating any kind of persistent local software components. By making use of industry standards, this concept also eases integration of new services and applications into the IPTV service bundle, such as social networks, weather forecast, traffic information and other third-party services.
- Improved end-user experience. For example, customization changes can simultaneously affect several STBs at once or only a specific device instance, accordingly with the user profile and its desires. Moreover, since all IPTV service content and interface is streamed from the provider infrastructure, users can easily access their content and information from any STB.

Moreover, the proposed IPTV model also hints for a post-OTT business model, based on a collaborative paradigm involving operators and third party service and content providers to ensure end-to-end service delivery performance. For such scenarios, we are evaluating the possibility of developing multi-tenancy mechanisms compatible with CWMP, fostering the creation of shared management interfaces for both operators and third-party service providers.

Future developments will also include optimization (of the support infrastructure and components) and fine-tuning of media decoding mechanisms in order to improve channel change latency and boot times.

ACKNOWLEDGEMENTS

This work was partially funded by PT Inovação (Project Virtuoso) and by Project QREN ICIS (Intelligent Computing in the Internet of Services – CENTRO-07-0224-FEDER-002003).

REFERENCES

[1] M. Ketcham, T. Mirlacher, and M. Monpetit, IPTV: An End to End Perspective, Academy Publishing's Journal of Communications, 5(5): 358-373, May 2010 (DOI: 10.4304/jcm.5.5.358-373).

[2] Broadband Forum, http://www.broadband-forum.org.

[3] Broadband Forum,"TR-069 - CPE WAN Management Protocol specification v1.2, Amendment 3", November 2010.

[4] Microsoft Corporation, "Microsoft Mediaroom Presentation Framework - Building Differentiated Experiences 2.0", February 2010.

[5] S. Husain, P. Siebert, "Business Case and Technical Requirements for IPTV STBs, in Proc. of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Bilbao, Spain, 13-15 May 2009.

[6] ECMA International, "ECMAScript Language Specification, 5th edition", Standard ECMA-262, December 2009.

[7] W3C Consortium, "HTML 5.1 A vocabulary and associated APIs for HTML and XHTML" , W3C Working Draft, April 2013.

[8] B. Cain, et al., Internet Group Management Protocol, Version 3, IETF RFC 3376, October 2002.

[9] D. Dorwin, A. Bateman, M. Watson, "Encrypted Media Extensions", W3C Editor Draft, February 2013.

[10] Broadband Forum, "Data Model Template for TR-069 Enabled Device, TR-106 Amendment 4", February 2010.

[11] Broadband Forum," Component Objects for CWMP, TR-157 Amendment 3", November 2010.

[12] Broadband Forum, "TR-135 Amendment 1 – Data Model for a TR-069 Enable STB", November 2010.

[13] T. Cruz, P. Simões, et al., "CWMP Extensions for Enhanced Management of Domestic Network Services", in Proc. of LCN'2010 (The 35th IEEE Conf. on Local Computer Networks), Denver, USA, September 2010.

[14] T. Cruz, P. Simões, et al., "Integration of PXE-based Desktop Solutions into Broadband Access Networks", in Proc. of CNSM'2010 (The 6th IEEE/IFIP Conf. on Network and Services Management), Niagara Falls, Canada, October 2010.

[15] Intel Corporation, "Preboot Execution Environment (PXE) specification version 2.1", September 1999.

[16] T. Cruz, P. Simões, "Enabling PreOS Desktop Management", in Proc. of the IM'2003 (IFIP/IEEE Int. Symposium on Integrated Network Management), Colorado Springs, May 2003.

[17] R. Droms, "Dynamic Host Configuration Protocol", IETF RFC 2131, March 1997.

[18] R. Fielding et al., "Hypertext Transfer Protocol – HTTP/1.1", IETF RFC 2616, June 1999.

[19] VideoLAN Client, project page at: http://www.videolan.org.

[20] Debian – the universal OS, project page at: http://www.videolan.org.

[21] Firefox HTTP browser, project page at: http://www.mozilla.org.

[22] The Webkit Open-Source web browser engine, project page at: http://www.webkit.org.

[23] Linux Infrared Remote Control Library, project page at: http://www.lirc.org.

[24] M. Carbone, L. Rizzo, "Dummynet revisited", SIGCOMM CCR, Vol. 40, No. 2, November 2009.

[25] CACE Technologies, Wireshark Network Protocol Analyzer, http://www.wireshark.org.

[26] H. P. Anvin and M. Connor, "x86 Network Booting: Integrating gPXE and PXELINUX", 2008 Ottawa Linux Symposium.

[27] D. Mills, U. Delaware, J. Martin, J. Burbank, W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", IETF RFC 5905, June 2010.

[28] IEEE, "IEEE Std1588TM – 2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Instrumentation and Measurement Society", July 2008.

[29] V. Roca, B. Adamson, "FCAST: Object Delivery for the ALC and NORM Protocols", IETF Reliable Multicast Transport (RMT) Draft 08, March 2013.

[30] T. Wiegand, G. Sullivan, G. Bjøntegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard", in IEEE Transactions on Circuits and Systems for Video Technology vol. 13, no 7., July 2003.

[31] J. Williams, "Microsoft TV Test", JDS Uniphase Corporation (JDSU) Application Note, May 2011.

[32] I. Pardillos, "In Brazil! Ready for the Broadcast and Cable 2011 Show", Video Stream Networks - Tapeless Community, http://www.vsn-tv.com/blog/view/50-in-brazil-ready-for-the-broadcast-and-cable-2011-show, 22 August 2011.

[33] Comcast Inc, "Introducing the X1 platform from XFinity", available at: http://www.comcast.com/x1.

[34] Comcast Inc., "Comcast Begins National Launch of X1: Next-Generation Cloud Enabled Television Platform and Introduces The X1 Remote Control App", Press Release available at: http://corporate.comcast.com.

[35] S. Islam and J. Grégoire, "Giving Users an Edge: A Flexible Cloud Model and its Application for Multimedia", Future Generation Computer Systems, Vol. 28, No. 6, 2012, pp. 823-832.

[36] S. Lee, J. Yun, "Design of Energy-Efficient Set-Top Box", IEEE 15th International Symposium on Consumer Electronics, 14-17 June 2011.

[37] B. Hu, G. Wu, L. Pan, H. Ni, M. Zhu, "An Implementation of Interactive Set-Top-Box and Its Applications", 8th International Conference on Signal Processing, 16-20 November 2006

[38] M. Bjelica, B. Mrazovac, I. Oaoo, N. Teslic, "Set-Top Box-Based Communication Client with the Automatic Operation Profile Selection", IEEE Transactions on Cosumer Electronics, August 2011.