Paper

On Implementing IPTV Platform with IPv4 and IPv6 Devices

Jordi Mongay Batalla[a,b] and Piotr Krawiec[b]

[a] *National Institute of Telecommunications, Warsaw, Poland*
[b] *Warsaw University of Technology, Warsaw, Poland*

Abstract—The end of IPv4 addresses is now a reality. Providers not updated to IPv6 will have to hurry up the IPv6 start in its own network. Introduction of IPv6 means not only change of main routers but also change of mentality in operators, applications' programmers besides end users. Even when for the last years the core network is prepared for transferring IPv6 traffic, other built-in parts of the Internet limit the IPv6 start. Examples of these limitations we find in not IPv6-awareness of many applications and services. For instance, voice over IP service, which uses session initiation protocol (SIP) needs to implement IPv6 aware SIP proxies and IPv6 aware AAA (authentication, authorization and accounting) servers as well as adapting application programming interfaces to IPv6. Internet protocol television (IPTV) system includes many different hardware devices, which not always are IPv6 compatible. In this paper, we propose a global solution for integrating all the devices, these one working on IPv4 and these one working on IPv6, under the same IPTV platform. This solution allows end users to receive IPTV stream irrespective of IP protocol used. The proposed solution is particularly relevant for small IPTV systems, which, step by step, are adapting into IPv6.

Keywords—"good practices", IPTV, IPv4/IPv6 interoperability, IPv6, set top box.

1. Introduction

Is IPv6 here already? It seems difficult to answer to this question. For sure IPv6 is arriving for the last 20 years. Mistrust of operators and companies are justified. ICT'2010, one of the most important *Information and Communication Technologies* conferences in Europe organized by the European Commission, presented the current state of implementation of IPv6 in Europe, and the future of IPv6 in Europe does seem overcast; on the one hand the IPv4 addresses are really finishing: deadline is 2012; on the other hand there are several steps missing for total operation of IPv6 all around Europe, and the countries, which will not work with IPv6 risk incomplete operation within a near future.

Building a network fully IPv6-aware comes up as a difficult task because Internet protocol is related with all the systems and devices (horizontal point of view), and at the same time it is the bottleneck of the layer architecture of the Internet (known as hour-glass model of the Internet). The hour-glass model implies that almost all the layers of the network have to do with the Internet protocol (vertical point of view).

From a horizontal point of view, all the systems of the network as, e.g., multimedia systems, distributed databases, even tester platforms [1] should adapt to IPv6. By bringing into operation IPv6 in entire systems as network services the IPv6 traffic increases within the network. For example, Google activated IPv6 in internal Youtube communications, increasing in this way the IPv6 traffic in the Internet up to 3000% [2]. Google needs IPv6 to build all the offered services inside one unique network, which is a requirement of the business plan. Other large systems in the Internet as, e.g., Yahoo and Facebook are actually starting on IPv6. Akamai[1] has just also announced IPv6 start.

Not only larger and universally extended service systems in the Internet must overcome different troubles for IPv6 start, but also the small systems find different difficulties during this process. Notice that small systems have much fewer economic resources for starting IPv6.

In this paper, we analyze the IPv6 start for Internet protocol television (IPTV) centering in the emerged problems when we set in motion IPTV system over devices, some working on IPv4 and some of them working on IPv6. We propose a solution based on sending to the network two parallel multicast streams, each one for one IP protocol (v4 and v6). To double the IPTV stream, we consider two independent networks and locate a server (IPv4/IPv6 server), which transmits between IPv4 and IPv6 "zones". This server is able to receive multicast flows generated by IPv4 devices and resend them in IPv6 multicast transmission to the IPv6 hosts. It can also perform transmission in opposite direction, when the IPTV signal source is located in IPv6 network and the destination is in IPv4 domain.

Let us remark that classical translation mechanisms are not useful in considered IPTV scenario, since multicast addresses may not be simply translated. The shortcoming of the proposed solution may derive from the effectiveness of the IPv4/IPv6 server in case of resending TV streams with high capacity.

The paper is structured as follows. In the next section we present the overview of the IPTV system that we tried to migrate to IPv6. Then, we describe the solution for IPv4/IPv6 environment. After that, in Section 4 we show the results of effectiveness study for proposed solution and conclude the paper in Section 5.

[1] www.akamai.com

2. Overview of the IPTV System

Multimedia communications are crucial for the definitive supremacy of packet networks over other connecting platforms. Practically all the multimedia communications have been or are being placed in the network, one of the most important is the television system carried within the Internet. This system is known as IPTV. IPTV systems have experienced an unexpected success in the network, gaining in popularity compared with other television transmissions. The reason we may find in the fact that consumers, always more, "demand personalized TV experiences that are available anytime, anywhere, on any device"[2]. The capabilities of IP television to fulfill these requirements as well as the fact that the whole complexity of IPTV systems is actually transparent to the consumers give more and more popularity to IPTV systems.

IP television favored changes in business models for the Internet. While before IPTV introduction, users connected more or less occasionally, now with IPTV (classical television or video on demand) the users just do not disconnect the computers from the Internet. The result is that many more consumers are constantly connected and the classical IPv4 addressing is not enough. IPTV demands IPv6 to offer static addressing to all the users. Moreover, another reason for the introduction of IPv6 to carry IPTV streams is the mobility of terminals (UMTS, LTE, etc.). As known, mobility requires enlarged addressing. On the other hand, we may remark another not banal reason for carrying classical television channels by IPv6 network is the enhanced multicast of IPv6 compared to earlier version of Internet protocol. Therefore, IPv6 seems to be crucial for IPTV.

Japan was the first country, which implemented a complete IPTV system working on IPv6. This first system is NTT Plala Hikari TV[3] and its implementation resulted indispensable since Japan developed only-IPv6 network. In any case, Hikari TV resulted very successful and currently has hundreds of thousands of consumers. Toshiba was the first hardware-specialized company commercializing IPTV devices working on IPv6.

The complexity of IPTV systems is due to the high quantity of information carried by television streams. In fact, the IPTV is known as one of the killer applications in the Internet because of the necessity of bandwidth. The demand of higher quality of the images required by the consumers means in practice that the image resolution is always higher and it implies more bandwidth in the IPTV transmissions. Figure 1 shows different image resolution codes (most typical in Europe) standardized or commercialized in the indicated years. As we may observe, the proposed image resolution generally increased as time went by.

To this increasing image resolution, we should add the higher requirements of television 3D, which in its most popular version, consists of uniting two images in one, doubling, in this way, the necessary bandwidth in the network.

[2]www.ericsson.com/campaign/televisionary

[3]www.hikaritv.net

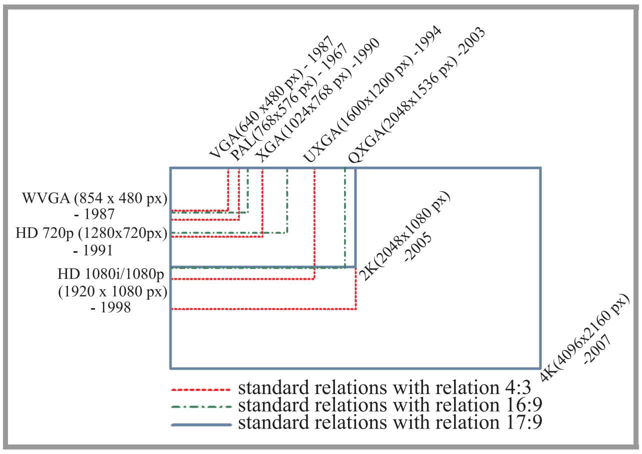

Fig. 1. Standards for image resolution.

Moreover, new applications related to the IP television as, e.g., interactive television, demand new requirements from the network. In the case of interactive television, the requirements are more similar to the interactive games than to the classical television.

For the correct management of heavy TV streams served to an increasing demand, the IPTV systems are developing and improving new solutions every day. In this sense, IP television comprises many research areas related to telecommunications. These areas are, among others, storage technologies, video and audio encoding (for example, MPEG-2 codec or more recent MPEG-4 H.264 codec), data encryption, data distribution, transmission by the network (new control and data planes). The complexity of IPTV systems as well as their importance is also proved by the increasing number of projects dedicated to improvement of transmission of television streams by the Internet. Between all the projects within the 7th Framework Program funded by the European Union (EU 7FP) we may highlight the following ones: one of the most successful projects, which is currently finishing is the P2P-Next project[4]. Among other objectives, this project specified and implemented a set top box with an interface for connecting to peer to peer networks which offers to the classical television sets the possibility of gaining access to the contents provided by peer to peer networks. Mobile3DTV[5] researches problems of moving 3D television to mobile environment. As known, mobility has strong limitations of bandwidth availability, which is not according to 3D television bandwidth requirements. Challenges as capture of 3D images, coding, and transmission are investigated in Mobile3DTV. Otherwise, CANTATA[6] is a project proposed inside the information technology for European advancement (ITEA) and develops a subset of functionalities related with interactive TV systems, which defines the requirements for this kind of television. Interactive TV enhances IPTV by offering to the consumer the possibility of interacting with the service provider for, e.g., shopping purposes. Many other

[4]www.p2p-next.org

[5]www.mobile3dtv.eu

[6]www.itea-cantata.org

7FP projects aim at introducing content-awareness within the network, which will undeniably open many new business possibilities to the Internet television. In fact, the new proposed architectures interconnect the four actors delineated in IPTV systems: content providers, IPTV service providers, transport and distribution IP network providers and clients [3]. These projects are grouped together in the future media networks cluster.

Let us remark that IPTV refers not only to classical broadcast television but also to new video on demand (VoD) services. The difference between them lies in broadcast (or multicast) transmission of classical television channels and unicast (or anycast for new content aware network architectures) of VoD transmissions. Anyway, the system studied in this paper refers to classical broadcast (multicast) television.

The concept of our IPTV system is presented in Fig. 2. In this system, the high definition television signal, called digital video broadcasting or briefly (DVB) can be delivered either by satellite (DVB-S) or terrestrial (DVB-T) manner. After receiving by appropriate antenna, television signal is transferred to the dreambox device. The dreambox[7] is a type of set top box and it is responsible for splitting digital DVB signal into IP packets, buffer them and transmit to the network as an integrated IP packet stream. Because unicast communication is not effective for providing IPTV service, often multicast connections are used for transfer packets between dreambox and end devices. Users can watch TV programs directly on their PC computers or laptops, thanks to appropriate IPTV applications. In case when we want to use a display unit such as a TV set, another set top box (STB) is used to transform again the IP packet stream into high definition television signal.

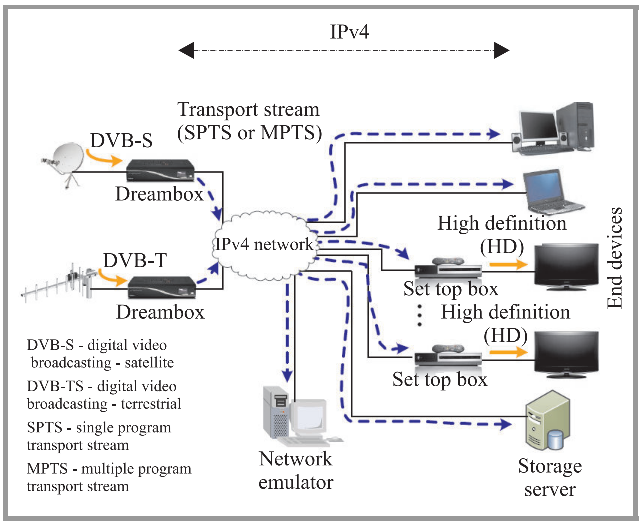

Fig. 2. IPTV system over IPv4.

The IPTV data are transferred through the network as a transport stream (TS), which is defined in MPEG-2 specification [4]. The TS is a type of container used for multiplexing the audio, video and auxiliary data as, for example, information required for synchronization or error correction. Transport stream is then packetized and encapsulated into the IP packets. MPEG-2 standard distinguishes between two types of TS: single program transport stream (SPTS) and multiple program transport stream (MPTS). SPTS correspond to transmission of a single TV channel, whereas MPTS allows transfer of more TV channels together within the same TS. The part of MPTS stream is program associated table (PAT), which contains the list of all transmitted TV channels. From the network point of view, the most important difference between the SPTS and MPTS is the necessary bandwidth for transmission. As we will see further below, this difference results crucial for the efficiency of the proposed IPTV solution for IPv4/IPv6 environment.

The stated IPTV system additionally contains a server to storage transmitted video files for further use, as well as a network emulator to perform diverse measurements in the IPTV system, such as measurements of QoS metrics experienced by IPTV flows for different (e.g., high load) network conditions.

The IPTV system described above was originally built to work on IPv4 only. Our aim was to migrate it on IPv6 protocol. The first difficulties that we met during this process were related with used IPTV application, which does not cooperate with IPv6.

Problems with applications may hinder the widespread use of IPv6 protocol. Although many applications nowadays are already IPv6-enabled (especially those associated with Linux system[8], the process of adapting some of them to support IPv6 is still pending. For example, up to year 2009, the MySQL application, a very popular open source database, makes possible the communication over IPv6 protocol between MySQL main programs (mysqld), called MySQL servers, as well as between the MySQL server and the MySQL cluster management server program (ndb_mgmd). Nonetheless, for now the communication between ndb_mgmd program and database repositories (the MySQL cluster data node daemon – ndbd program) is still IPv4-only aware [5].

In our IPTV system, we replaced the existing IPv4 commercial application by the open-source VideoLAN Client (VLC) media player [6]. VLC can handle most of the media codecs and video formats, as well as various streaming protocols. It permits also to send and receive data using both IPv4 and IPv6 protocols. Observe that using IPv6-aware application is obligatory at least in these networks, which are natively IPv6-only. VLC cooperates with video LAN manager (VLMa), which is able to manage broadcasts of TV channels from DVB-S or DVB-T sources and streaming audio and video files. Furthermore, VLMa can be used to stream a received unicast stream in multicast way.

The main problem we found during IPv4/IPv6 migration was that the set top box (STB) devices, used to convert

[7] www.dream-multimedia-tv.de/en

[8] See e.g. www.deepspace6.net/docs/ipv6_status_page_apps.htm

IP packet stream into television signal, could not operate with IPv6 protocol. This issue does not affect dreambox devices, which work on Linux-based operating system Enigma2. Enigma2, as a large majority of Linux variants, supports IPv6. Moreover, thanks to open source concept of Linux, dreambox software can be easily upgraded by users, if need be. Unfortunately, we were not able to modify software in other STB devices. Taking into account that we had many such STB devices, it was not viable to replace all of them in IPv6-compatible equipment. To solve this, we proposed to divide the network into two subdomains, isolating the devices, which may work on IPv6 and these ones, which may work on IPv4 only.

3. Transmission of IPTV Streams on IPv4/IPv6 Environment

Creating two networks, which separate the IPv4 and IPv6 equipment, effects on the MPEG-2 transport stream transferred through the network between dreamboxes and end devices. Now we should send the transport stream twice:

- in IPv4 subdomain, transport stream is encapsulated into IPv4 packets, what is done by dreamboxes,
- in IPv6 subdomain the same transport stream is encapsulated into IPv6 packets.

To perform the latter, we propose to use special tool, called the IPv4/IPv6 server. This server is placed at the border between IPv4 and IPv6 networks and has two network cards. One of them receives multicast IPv4 stream generated by dreambox, while the second one is responsible for resending the same stream after encapsulating it in multicast IPv6 packets. Figure 3 presents the concept of resulting network. Summarizing, the IPv4/IPv6 server works as a gateway between the networks.

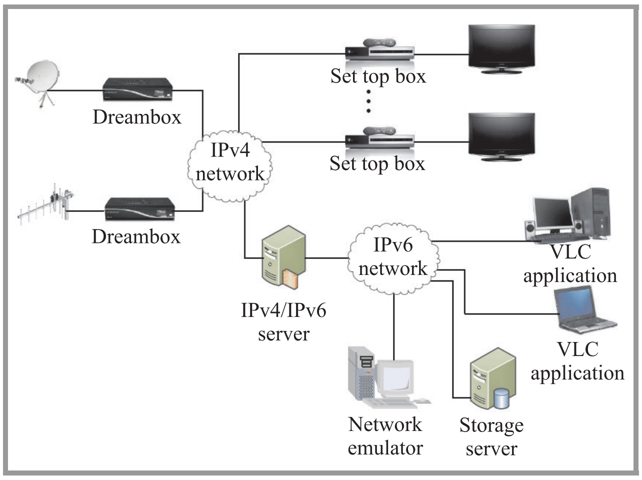

Fig. 3. IPTV system in IPv4/IPv6 environment.

In the IPv4 domain, the IPv4/IPv6 server acts as an ordinary multicast client, which subscribes to the IPv4 multicast stream in a standard way, using IGMP protocol [7]. On the other hand, in the IPv6 domain, the IPv4/IPv6 server operates as a shared root of distribution tree for an IPv6 multicast group. We assume that in the IPv6 network the protocol independent multicast – sparse-mode (PIM-SM) [8] is implemented, which is the most widely used multicast routing protocol because of its independency from underlying unicast routing protocols and overcoming the scalability problems [9]. In our case the IPv4/IPv6 server plays role of a so-called PIM-SM rendezvous point (RP) for the entire IPv6 domain. An RP can be considered as the meeting place for sources and receivers of multicast data. Setting up the IPv4/IPv6 server as a RP is crucial if there are more routers in the path between the IPv4/IPv6 server and the end IPv6 multicast clients.

RFC 3956 [10] defines an address allocation policy (called embedded-RP) in which the address of the RP is encoded in an IPv6 multicast group address. The document specifies a subrange of unicast prefix-based IPv6 multicast addresses [11], which starts with FF70::/12 prefix, by setting one of previously undefined bit from flags field to 1. Furthermore, it prescribes a method for embedding the RP address, which serves given multicast group, to IPv6 multicast address of this group. Thanks to it, there is no requirement for any multicast pre-configuration of the other routers belonging to multicast tree, if they are not operating as an RP, because routers can automatically obtain information about the RP from IPv6 multicast group address.

According to RFC under consideration, we enforce the multicast group address to be

FF77:0xxx:aaaa:aaaa:aaaa:aaaa:gggg:gggg,

where all the bits “x” together with “a” bits represent the rendezvous point address, whereas “g” bits represent the identifier of the multicast group. For implementation purpose, we notice that our IPv6 multicast group address should be mapped into Ethernet multicast address on the following form: 33:33:gg:gg:gg:gg [12].

Now we illustrate the procedure of establishing multicast connection by one IPv6 host, which wants to receive the IPTV stream generated by the IPv4 dreambox. Let us suppose, for the sake of argument, that:

- dreambox has the IPv4 address 210.165.23.7,
- the IPv4/IPv6 server has the IPv6 address
  FF77:0130:1111:1111:1111:1111::,
  which enclose the embedded rendezvous point address 1111:1111:1111:1111::1.

Therefore, the embedded-RP multicast prefix is

FF77:0130:1111:1111:1111:1111::/96.

To start receiving the dreambox IPTV stream, the IPv6 host should send a multicast listener report message of multicast listener discovery protocol (MLD) [13] to the destination multicast group address FF77:0130:1111:1111:1111:1111:210.165.23.7. When the IPv4/IPv6 server receives this message, it joins the new host to given IPv6 multicast group. Next, if there was no transmission of multicast data so far (since there was no any IPv6 multicast listener), the IPv4/IPv6 server starts resending the IPTV stream to the joined IPv6 host.

Because the IPv4/IPv6 server operates in IPv6 domain as a source of IPTV streams, the IPTV packets will arrive to the IPv6 host with source address of the IPv4/IPv6 server. It means that IPv6 multicast transmission is performed with destination multicast group address FF77:0130:1111:1111:1111:1111:210.165.23.7 and source address FF77:0130:1111:1111:1111:1111::. In this way, different multicast streams from more than one dreambox are allowed if they have different IPv4 addresses. However, resending more IPTV streams by the IPv4/IPv6 server could cause incorrect work because of hardware limitations. The effectiveness of the IPv4/IPv6 server we study in the next section.

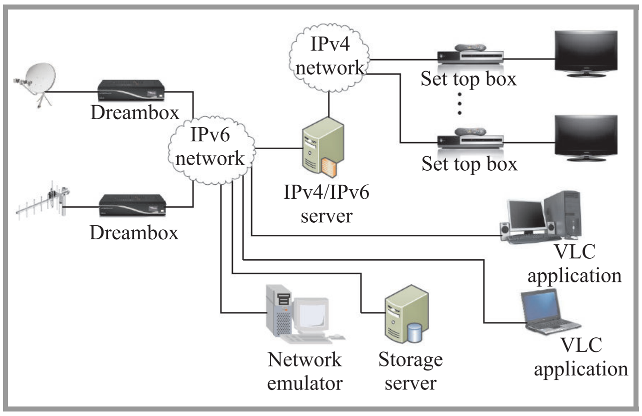

Fig. 4. IPTV system in IPv4/IPv6 environment (second approach).

The second investigated approach is when dreambox sends IPv6 stream and the server is the responsible to translate the stream into IPv4 as shown in Fig. 4.

In this case the server uses the MLD protocol to join to IPv6 multicast tree in the IPv6 network. On the other hand, in the IPv4 domain, the IPv4/IPv6 server operates as a shared root of distribution tree for an IPv4 multicast group.

4. Effectiveness Study of the Proposed Solutions

In this section we aim at investigating the effectiveness of the IPv4/IPv6 server in both the proposed solutions, i.e., when the server translates IPv4 stream into IPv6 and in the opposite way.

In the first approach, we assume that the dreambox at the IPv4 domain sends an IPTV packet stream at rate, which increases from one trial to the next. For this purpose the dreambox works in MPTS mode. The MPTS service allows to group together many TV channels, which may be encoded with standard definition (SD) or high definition (HD) resolution. During the tests, dreambox generates one MPTS flow with different number of TV channels, and then the total bandwidth of IPTV stream can be easily obtained as multiplication of bandwidth of the SPTS flow (9.47 Mbit/s). Although we could increase IPTV data rate by simply growing the number of SPTS multicast flows, we believe that the chosen approach imitates better a real IPTV scenario, where one IPTV service provider offers different number of TV channels. Then we monitor whether the IPv4/IPv6 server is able to transfer received IPTV packets to the IPv6 network.

The test run as follows: firstly the multicast tree was created in both IPv4 (using IGMP protocol) and IPv6 domains (using MLD protocol). Next dreambox streamed the DVB-T signal as a unique SPTS in IPv4 multicast mode. The IPv4/IPv6 server captured the IPTV stream as IPv4 multicast listener, and resent it to the IPv6 end devices in an IPv6 multicast connection. We calculated the rate of packet flow received by the IPv4 network card of the IPv4/IPv6 server (incoming stream) and the rate of packet flow sent by the IPv6 network card (outgoing stream). The obtained results are presented in Fig. 5.

After that, we changed SPTS for MPTS flow and repeated the tests for increasing number of TV channels encoded in the stream. Logically, when MPTS contains more channels, larger bandwidth is necessary to transfer it. In the same way as previously, we calculated the data rate of the incoming stream (to the IPv4/IPv6 server from IPv4 network) and the outgoing stream (from the IPv4/IPv6 server to the IPv6 network). Figure 5 presents these values.

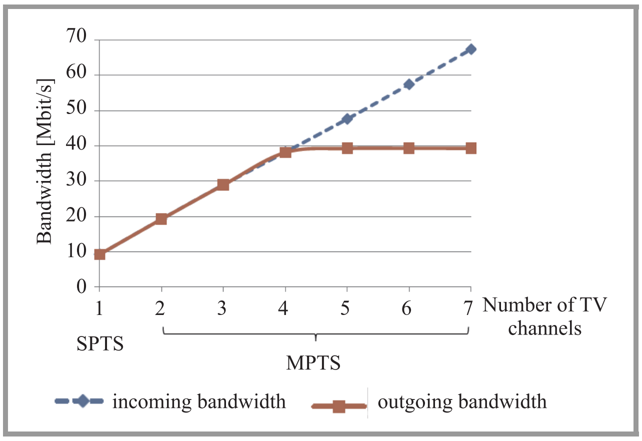

Fig. 5. Results of IPv4/IPv6 server's effectiveness (SPTS – single program transport stream, MPTS – multiple program transport stream).

As one can observe, for low rates the IPv4/IPv6 server does not affect resent IPTV stream. The limit value corresponds to four SPTS flows' bandwidth. Higher rates of IPTV traffic results in packet losses within the IPv4/IPv6 server. We may indicate that the hardware limitations of the server cause this effect. The IPv4/IPv6 server was implemented on PC with processor Intel® Core^{TM}2 duo desktop processor E8500 3.16 GHz and Linux operating system with kernel version 2.2.17. Anyway, presented studies show that the proposed solution has limitations. Certainly, the IPv4/IPv6 server may be used for providing to user a single TV channel (SPTS) as, e.g., a football match in a pay-per-view video service, but the hardware limitations cause that it is not suitable for serving, e.g., the public television, which transmits many TV channels.

In the second approach, we assume that IPTV packet stream is sending by the dreambox, which is in this case located in the IPv6 domain. As in the previous test, the dreambox generated IPv6 packets with increasing rate by working in MPTS mode and emitting the same number of channels as described above. The hardware used to implement the IPv4/IPv6 server was the same one.

The test run similarly to the preceding one, i.e.: firstly, the multicast tree was created in both IPv6 (using MLD protocol) and IPv4 domains (using IGMP protocol). Next dreambox streamed the DVB-T signal as SPTS or MPTS (in consecutive trials) in IPv6 multicast mode. The IPv4/IPv6 server captured the IPTV stream as IPv6 multicast listener, and resent it to the IPv4 set top boxes in an IPv4 multicast connection. We calculated the rate of packet flow received by the IPv6 network card of the IPv4/IPv6 server (incoming stream) and the rate of packet flow sent by the IPv4 network card (outgoing stream). The obtained results are presented in Fig. 6.

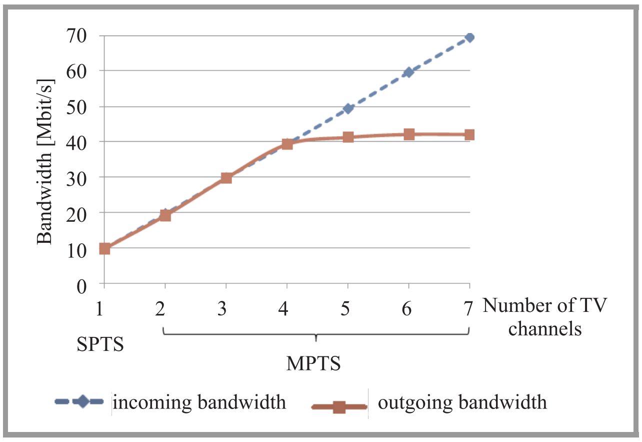

Fig. 6. Results of IPv4/IPv6 server's effectiveness (SPTS – single program transport stream, MPTS – multiple program transport stream) – second approach.

As we may observe in Fig. 5 and Fig. 6, the effectiveness is very similar in both of the approaches. The minimal differences (rather imperceptible in the figures) in favor of the second option could be provoked by the more complexity in sending multicast IPv6 packets than multicast IPv4 packets.

5. Conclusions

To support smooth transition between IPv4 and IPv6 protocols, a set of *good practices* in this direction should be proposed. In this paper we present a solution for deploying the IPTV system in an scenario which involves presence of two kinds of devices: IPv4-only and IPv6-only. The proposal exploits special server for transferring IPTV multicast traffic among IPv4 and IPv6 domains. The proposed solution may be framed as one of these *good practices* because it allows a simple step towards widespread introduction of IPv6.

From the performed experiments we could demonstrate that our IPTV system properly works on IPv4/IPv6 environment. As a consequence, we may conclude that the presented implementation issues are correct. We implemented two solutions, the first one when the multicast IPv4 stream is translated into multicast IPv6 stream and the second one in the opposite direction. Both the solutions properly worked and showed that they may be valid solutions for the case when IPv4-only and IPv6-only receivers are in the IPTV system.

On the other hand, the obtained results of effectiveness let us to realize that, in case of large bandwidths of IPTV streams, the proposed IPv4/IPv6 server does not properly run and is not capable to transfer the whole incoming IPTV traffic. We deliberated that this issue depends on the used hardware but it should be an advice note when one considers using the proposed solution in systems, which demand high bandwidth as classical IPTV does. The effectiveness of the two proposed solutions is similar and it is not possible to conclude which of them behaves better in wide IPTV systems.

References

[1] "Mu Dynamics introduces Mu Test Suite for IPv6 to significantly accelerate and ensure successful IPv6 migration", Mu Dynamics Inc. 2010 [Online]. Available: http://www.tmcnet.com/usubmit/2010/04/26/4749761.htm

[2] C. D. Marsan, "YouTube turns on IPv6 Support, net traffic spikes", *PCWorld Mag.*, Feb. 2010.

[3] D. H. Ramirez, *IPTV Security: Protecting High-Value Digital Contents*. Wiley, 2008.

[4] "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems", ITU-T Recommendation H.222.0, Inst. Telecommun. Union, Geneva, Switzerland, 2000.

[5] *MySQL 5.1 Reference Manual*, paragraph 17.7.2.22. Oracle USA, January 2009.

[6] "VLC media player: The cross-platform open-source multimedia framework, player and server" [Online]. Available: http://www.videolan.org/vlc/

[7] B. Cain, S. Deering, I. Kouvelas, B. Fenner and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, IETF, October 2002.

[8] B. Fenner , M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, IETF, August 2006.

[9] E.-M. Lee, Y.-T. Han, H.-S. Park, "Rendezvous point relocation for IPTV services with PIM-SM", in *Proc. 14th Asia-Pacific Conf. Commun. APCC 2008*, Tokyo, Japan, 2008.

[10] P. Savola and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, IETF, November 2004.

[11] B. Haberman and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 4601, IETF, August 2002.

[12] M. Crawford, "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, IETF, December 1998.

[13] R. Vida and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, IETF, June 2004.

Jordi Mongay Batalla – for biography, see this issue, p. 11.

Piotr Krawiec – for biography, see this issue, p. 13.