

# IP Data Delivery in HBB-Next Network Architecture

Roman Bronis<sup>(✉)</sup>, Ivan Kotuliak, Tomas Kovacik, Peter Truchly,  
and Andrej Binder

Faculty of Informatics and Information Technologies, Institute of Computer Systems  
and Networks, Slovak University of Technology in Bratislava,  
Ilkovicova 2, 84216 Bratislava, Slovakia  
{roman.bronis,ivan.kotuliak,tomas.kovacik,  
peter.truchly,andrej.binder}@stuba.sk  
<http://fiit.stuba.sk>

**Abstract.** Digital television enables IP data delivery using various protocols. Hybrid television HbbTV enhances digital television with applications delivery. HBB-Next is an architecture which enhances HbbTV with additional features. However it does not specify IP data delivery despite it has access to both broadcast and broadband channel. This paper proposes architecture and protocols for IP data delivery in HBB-Next. To achieve this goal we designed new node (Application Data Handler - ADH) in HBB-Next architecture and new communication protocols (Application Data Handler Control Protocol - ADHCP, and Hybrid Encapsulation Protocol - HEP) for data transmission. We created Stochastic Petri Net (SPN) model of designed protocols and implemented them in ns2 network simulator to verify our solution. Results of SPN model simulation and ns2 network simulation are discussed and HEP protocol is compared to existing encapsulation protocols used in DVB systems.

**Keywords:** IP data encapsulation · HBB-Next · DVB · SPN · ns2 · Application Data Handler (ADH) · Hybrid Encapsulation Protocol (HEP)

## 1 Introduction

Evolution from digital television to hybrid television started with Multimedia Home Platform (MHP) [1]. It was later surpassed by hybrid television standard - Hybrid Broadcast Broadband Television (HbbTV) [2]. HbbTV applications are CE-HTML based and can take advantage of broadband return channel. HbbTV applications can be interactive and mostly serve for TV providers as enhanced EPG (Electronic Program Guide) applications (archive, informations about movies and shows, trailers etc.). HbbTV applications can not only serve as TV information portals, but can be also used in other areas such as in e-learning [3].

To enhance HbbTV application capabilities, HBB-Next platform was designed [4]. It provides additional features as user recognition (by face or voice etc.), content recommendation and user management. As in HbbTV, HBB-Next terminals are connected to broadband Internet which is used for application data or media streams delivery. In HBB-Next, service provider has access to broadcast channel, but it is used only for media streaming.

HBB-Next platform does not specify IP data delivery. Protocols for IP data delivery in digital television could be used, but they were not designed to utilize both broadcast and broadband channel. In this paper we propose enhancement to HBB-Next architecture and we design protocols to deliver IP data from applications to terminals using both broadcast and broadband channel. We compare it to similar solutions and describe its advantages.

This paper is organized as follows: the second section describes protocols used for IP data delivery in DVB systems and current state of next generation of hybrid television. The third section proposes Application Data Handler (ADH) node and ADH-Control Protocol (ADHCP). In fourth section Hybrid Encapsulation Protocol (HEP) and HEP Hash Table (HHT) protocols are described. The fifth section describes Stochastic Petri Net (SPN) model of designed communication, its properties and results of simulations. The sixth section describes implementation of designed protocols in ns2 network simulator and results of simulations. The seventh section concludes this paper.

## 2 IP Data Delivery in DVB and in HBB-Next

In this section we describe current state of IP data delivery in DVB systems and current state of HBB-Next architecture.

### 2.1 IP Data Delivery in DVB

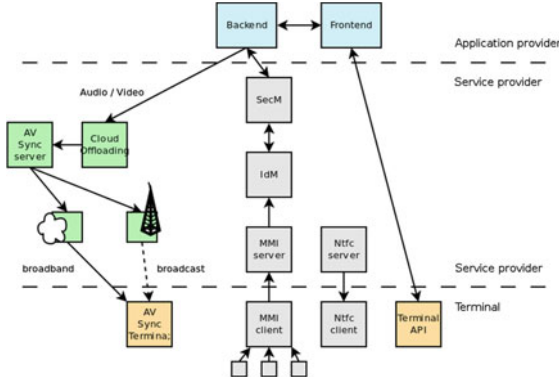
Multi-Protocol Encapsulation (MPE) protocol was designed for IP data delivery in first generation DVB systems (DVB-S/C/T) [5]. It is the most used protocol to receive IP data over broadcast channel in areas without connection or with limited broadband connection. MPE can work in two modes. In padding mode, MPE frame's unused data are filled with invalid data. In packing mode, MPE frame's unused data are filled with next packet. Padding mode is available to all devices with MPE support, however MPE packing mode is optional. Packing mode is more effective but is not supported by all end-devices.

Unidirectional Lightweight Encapsulation (ULE) protocol was designed as lightweight alternative to MPE protocol [6]. It has reduced header (header size: MPE - 16 B, ULE - 4 B) and is using packing mode by default.

For second generation DVB systems (DVB-S2 etc.), new protocol for IP data delivery can be used [7]. Generic Stream Encapsulation (GSE) protocol is the most effective in DVB-S2 systems, but it is not compatible with first generation of DVB systems. Despite GSE's highest efficiency, MPE is still used in many cases. During transition to DVB-S2, providers stayed with MPE protocol because it had wider support in consumers' end-devices and was still used by DVB-S systems.

## 2.2 HBB-Next

HBB-Next is platform of next-generation hybrid television. It was designed to provide additional features to hybrid television. HBB-Next architecture consists of three main layers: application provider, service provider and terminal (Fig. 1).



**Fig. 1.** HBB-Next architecture (high-level view)

Application provider layer represents application, its data and frontend. Applications can be HbbTV compatible and can take advantage of features of Service provider layer.

Service provider is HBB-Next core architecture provider. Service provider layer consist of multiple nodes. It is designed to provide advanced features such as user recognition (Multi-modal Interface)[8], content recommendation, enhanced identity management (IdM) and security management (SecM), and audio and video synchronisation and delivery (CloudOffloading and AV Sync nodes).

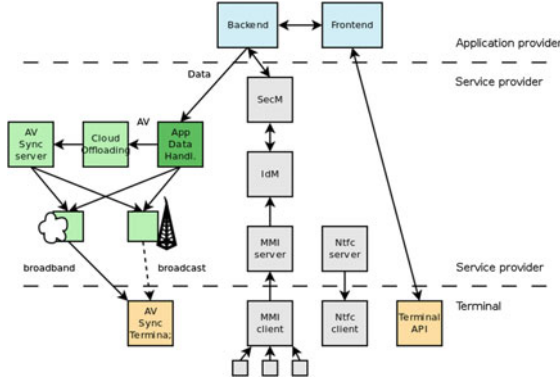
Terminal is end-point device which is able to receive transmission on broadcast channel and is also connected to broadband channel. Broadband channel's bandwidth is not specified but HBB-Next terminal is considered to have enough bandwidth for face recognition data and multimedia streaming reception.

Applications in HBB-Next can send data to terminal by broadband channel. Broadcast channel is solely used for audio and video data delivery. However, this channel could be used for delivery of other application data as well. To provide this feature HBB-Next architecture need to be enhanced.

HBB-Next does not specify IP data delivery in its core. Standard DVB encapsulation protocols (MPE, ULE, GSE) could be used for IP data delivery in HBB-Next only by bypassing its core. HBB-Next platform could take advantage of its core and its connection to broadcast and broadband channels to transfer IP data from applications to terminals.

### 3 Application Data Handler (ADH) and ADH-Control Protocol (ADHCP)

HBB-Next architecture is not suitable for delivery of application data through broadcast channel. Only node which is connected to both broadband and broadcast channel is AV Sync node. For applications to be able to deliver IP data we created a new node - Application Data Handler (ADH) in service layer (Fig. 2).



**Fig. 2.** HBB-Next architecture with ADH node

ADH node is used to receive application data and send them through appropriate broadcast or broadband channel according to applications' needs.

Data are sent from application to ADH node using ADHCP protocol (Fig. 3). This protocol was designed to allow application data encapsulation (data field, <1435 B), link type selection (link type field) and addressing (address type and address fields). Hash field in ADHCP headers is used as checksum for encapsulated data. ADHCP communication is based on request and reply messages and they use two different header formats. Application is requesting data transfer from ADH ((1) in Fig. 3). ADHCP request header consist of link type, address type, address (optional), data (encapsulated data) and hash value (of encapsulated data). ADH node response in case of failure with response ADHCP message ((2) in Fig. 3). ADHCP response header consists of hash value of message which was not delivered correctly and response code. Response code is:

- 00 - refused: in case of ADH refuse to transmit data over selected link,
- 11 - retransmission request: in case any of terminals failed to receive data and they are no longer in ADH cache,
- 01 and 10 - reserved.

In Fig. 3 part A there is ADHCP messages exchange when there is insufficient bandwidth on broadcast channel (ADH refuses to send data). Part B shows

correct data transmission with ADHCP to ADH. In part C ADHCP messages exchange in case of transmission failure on broadcast channel is shown.

HBB-Next's Application layer is considered to be connected with Service provider layer with sufficient bandwidth. ADHCP is encapsulated in TCP/IP to achieve reliable transmission.

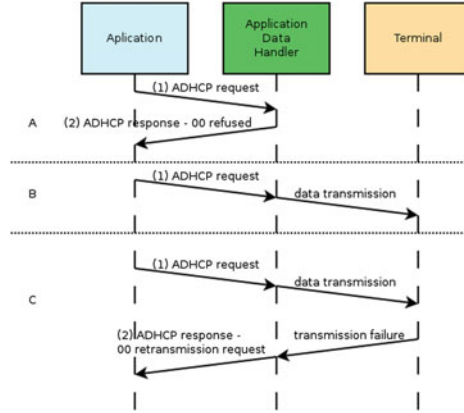


Fig. 3. ADHCP messages flow

#### 4 Hybrid Encapsulation Protocol (HEP) and HEP Hash Table (HHT) Protocol

To transmit data from ADH to terminal, new lightweight protocol was designed. Hybrid Encapsulation Protocol (HEP) can be used either in DVB broadcast channels encapsulated in MPEG-TS or in broadband channel over TCP/IP. Broadcast channels are not reliable medium. To check correct reception of received data, HASH values are sent from ADH to terminal using HEP Hash-Table - HHT ((1) in Fig. 4). HHT consist of list of items. One item consist of fields (size of field in brackets):

- hash value (256 b) - hash value of data received from application,
- link type (4 b) - expected link to receive HEP frame,
- reception time (32 b) - expected time of HEP frame reception, in Unix time format, and
- sequence number (20 b) - position in terminals reception stack.

One or multiple items may be send in one HHT message. HHT is sent to selected terminal by broadband channel using TCP/IP and it consists only from items addressed to selected terminal.

After HHT communication, HEP message is sent to terminal ((2) in Fig. 4). HEP message consists of encapsulated data sent from ADH (originally from

application). For HEP messages received from broadcast channel terminal counts Hash value and check if the message was received correctly. Counted value is compared to one terminal received in HHT. If received correctly, counted Hash value will match to one from terminals HHT list. In case of transmission error, counted Hash value will not match any of items in HHT list. Terminal periodically checks reception time and requests frames ((3) in Fig. 4) which were not delivered within reception time in terminals HHT. Requesting HEP message consist of number of requested frames with their hash values following. After ADH receives HEP request, it can either resend data (if still cached) or request application for retransmission with ADHCP response ((4) in Fig. 4).

HEP encapsulation was designed to be more efficient then previously used protocols. ULE protocol was designed to reduce MPE’s header to achieve higher efficiency [9, 10]. GSE protocol is even more efficient, but it is used only in second generation DVB channels [11]. HEP’s encapsulation over broadcast channel is more efficient as MPE, ULE or GSE because it has no header. It is using HHT protocol instead which is solely transmitted over broadband channel and therefore saves broadcast channel bandwidth.

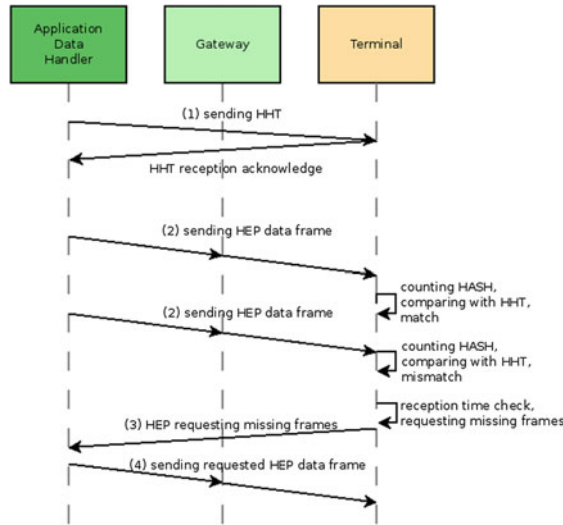
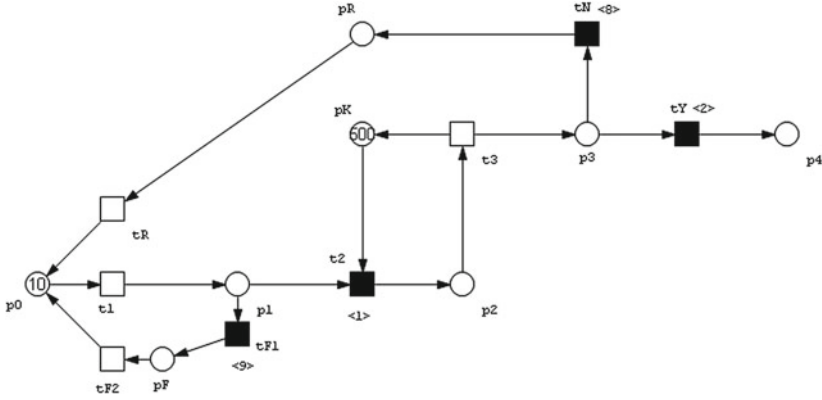


Fig. 4. HHT and HEP communication

## 5 Petri Net Model of Communication

Petri Nets consist of places, transitions, arcs and tokens. Places are connected to transitions by arcs. Firing a transition ( $t_1$  - Fig. 5) can move token between connected places (from  $p_0$ ) in a direction of an arc (to  $p_1$ ). Stochastic Petri Nets enable transitions to be fired with given probability or rate.



**Fig. 5.** Communication model using Petri Nets

To verify properties of designed protocols, we created model of their communication using Stochastic Petri Nets (Fig. 5, Table 1). We verified its selected properties using PIPE tool [12]. Place  $p_K$  was used to simulate broadcast link capacity. Transitions  $t_Y$  and  $t_N$  were stochastic transition set to simulate broadcast channel error rate -  $t_Y$  was executed every 2 times,  $t_N$  was executed every 8 times what represents 20% error rate on a broadcast channel.

**Table 1.** Places and transitions in SPN

$p_0$	ADHCP request sent	$t_1$	send ADHCP request
$p_1$	ADHCP request received	$t_2$	send HHT
$p_2$	HHT received	$t_3$	send HEP frame
$p_3$	HEP received	$t_Y$	data check - correct
$p_4$	correct data	$t_N$	data check - incorrect
$p_F$	full channel	$t_R$	request for retransmission
$p_K$	link capacity	$t_F$	denial of transmissin
$p_R$	incorrect data, retransmission		

Results showed that PN is not safe and can be dead-locked - which means that sent messages can be delivered and halt in terminal's last state  $p_4$  - correct data reception. This is considered as correct protocol behavior, because it represents that data can be delivered to final destination ( $p_4$  - correct data reception). There was no other dead-lock identified. We tested boundedness for places representing link capacity ( $p_K$ ) and results showed that these places are bounded, therefore communication can not overload link capacity. Boundedness for whole model was also tested and results showed that the whole model is bounded.

Using Snoopy [13] we simulated SPN model sending 10 and 1000 messages (Fig. 6). Simulation results show correct reception of all messages - purple line

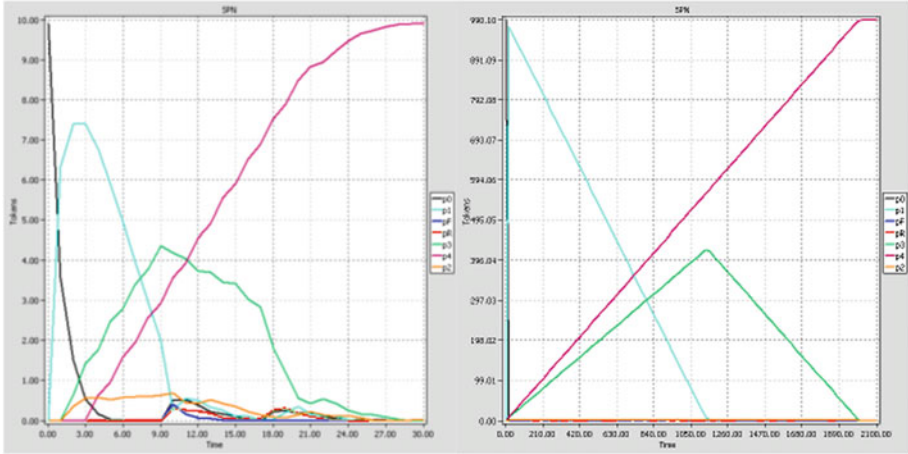


Fig. 6. SPN simulation - 10 messages, 1000 simulations

of  $p_4$  place leads to sent messages count, and correct retransmission request in case of an error on broadcast channel - highlighted red line of  $p_R$  place.

### 6 Simulation in ns2

In order to simulate our protocols, we implemented their version using ns2 network simulator. Application (APP) was sending its messages using ADH node broadcasting to 100 terminals through DVB gateway (DVB-GW). We set terminals' error reception rate on broadcast channel to various percentages (0–99%). Terminal Term\_(0) had 0% error rate reception, terminal Term\_(1) had 1% error rate reception, and so forth. Simulation scenario was set to request every erroneous message again over broadband channel through IP gateway - IP\_GW (Fig. 7).

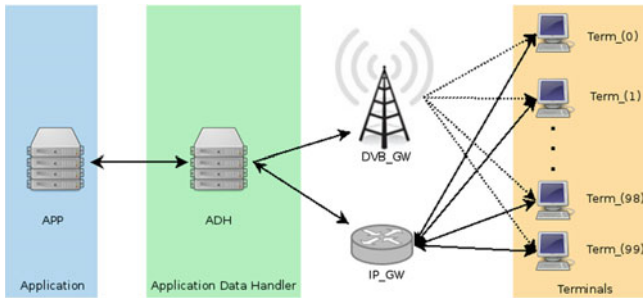


Fig. 7. Simulated topology



Results (Fig. 8) show dependence of received frames on broadcast channel error. Red line represent percentage of messages received through broadcast channel and green line represents percentage of messages received through broadband channel. Independently on broadcast channel error rate, summary of both line (channels) gives 100 % reception of messages on terminal. Results verified correct transmission and retransmission behaviour of designed protocols in network simulator.

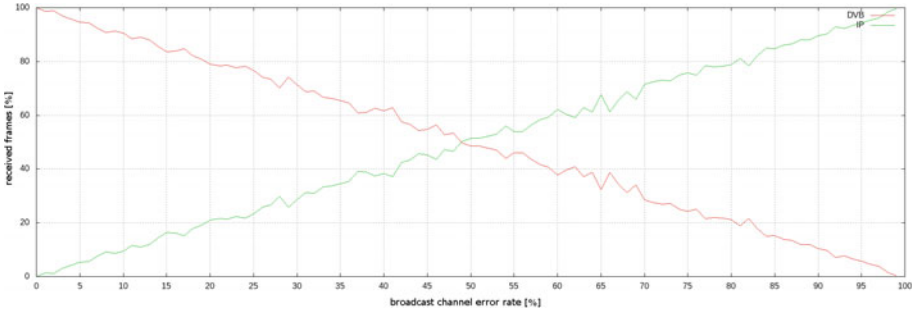


Fig. 8. Results of ns2 simulation

## 7 Conclusion

In this paper we proposed architecture and protocols for IP data delivery in DVB broadcast channels in next generation hybrid television - HBB-Next. We identified missing specification for IP data delivery in HBB-Next, which could take advantage of broadband channel.

We designed new node in HBB-Next architecture - Application Data Handler (ADH). ADH receives all application data and transmits them to terminals using either broadcast or broadband channels. In order to communicate with ADH we designed ADH-Control Protocol (ADHCP). For data delivery to terminals through different channels we designed Hybrid Encapsulation Protocol (HEP) and HEP Hash-Table (HHT) protocols.

We created Stochastic Petri Net (SPN) model of communication with designed protocols. We analysed properties of SPN model and simulated its behaviour. Results showed desired properties and simulations verified correct transmission and retransmission of messages. Later we implemented our protocols in network simulator and simulated communication with multiple terminals with different error rate on broadcast channel. Results showed correct data transmission over broadcast channel and retransmission over broadband channel.

Our work enables IP data delivery in HBB-Next from applications to terminals over HBB-Next service provider layer. Applications can not only serve as multimedia provider and HbbTV content provider, but with our changes they can also behave as IP data providers. Our HEP encapsulation has also reduced

frames' header overhead on DVB broadcast channels. Instead it is using broadband channel to deliver HHT (header-like data).

Future work includes further comparison of SPN simulations with network simulations, testing of parallel transmission in various complex scenarios and implementation and testing on real hardware.

**Acknowledgments.** This work is a result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005 and for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF and was supported by the Slovak national research project VEGA 1/0708/13, KEGA 047STU-4/2013 and Slovak Research and Development Agency project APVV-0258-12.

## References

1. ETSI TS 102 812: Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.3, ETSI, v1.3.1, May 2012
2. ETSI TS 102 796: Hybrid Broadcast Broadband TV, ETSI, v1.2.1, November 2012
3. Kovacik, T, Bronis, R., Kotuliak, I.: HBB platform for e-learning improvement. In: IEEE 10th International Conference on Emerging eLearning Technologies and Applications, ICETA 2012, November 2012, Stara Lesna, The High Tatras, Slovakia, pp. 207–211 (2012). ISBN: 978-1-4673-5122-5
4. Podhradsky, P.: Evolution trends in hybrid broadcast broadband TV. In: Proceedings of 55th International Symposium ELMAR-2013, September 2013, pp. 7–10 (2013). ISBN: 978-953-7044-14-5, ISSN: 1334-2630
5. ETSI EN 301 192: Digital Video Broadcasting (DVB); DVB specification for data broadcasting, ETSI, v1.4.2, April 2008
6. IETF RFC 4326: Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS), IETF, December 2005
7. ETSI TS 102 606: Digital Video Broadcasting (DVB) - Generic Stream Encapsulation (GSE) Protocol, ETSI, v1.1.1, October 2007
8. Kovacik, T., Bronis, R., Kotuliak, I.: Towards novel HBB application platform: experimental testbed. In: 2012 5th Joint IFIP Wireless and Mobile Networking Conference, Bratislava, Slovakia, 19–21 September 2012, pp. 133–137 (2012). ISBN: 978-1-4673-2994-1
9. Fairhurst, G., Matthews, A.: A comparison of IP transmission using MPE and a new lightweight encapsulation. In: IEEE Seminar on IP over Satellite - The Next Generation: MPLS, VPN and DRM Delivered Services, pp. 106–120 (2003)
10. Xilouris, G., et al.: Unidirectional lightweight encapsulation: performance evaluation and application perspectives. *IEEE Trans. Broadcast.* **52**(3), 374–380 (2006). ISSN: 0018-9316
11. Mayer, A., et al.: Analytical and experimental IP encapsulation efficiency comparison of GSE, MPE, and ULE over DVB-S2. In: International Workshop on Satellite and Space Communications, IWSSC 2007, pp. 114–118. IEEE, September 2007. ISBN: 978-1-4244-0938-9
12. PIPE: tool for Petri Nets. <http://sourceforge.net/projects/pipe2/>
13. Snoopy2: tool for Petri Nets. <http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy>