Data transfer in HBB-Next architecture

Roman Broniš[*]
Institute of Computer Systems and Networks
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
roman.bronis@stuba.sk

Abstract

Modern television devices are not only connected to broadcasting systems but they are also connected to the Internet. Their purpose is not only to show TV channels but they are also used to provide interactive applications in form of Smart TV applications and HbbTV (Hybrid Broadcast Broadband Television) applications.

HbbTV is an open standard for delivery of interactive applications which was later improved by HBB-Next (Next generation Hybrid Broadcast Broadband) standard. HBB-Next architecture provides additional features and security to HbbTV applications. HBB-Next defines data transfer from application to terminals mainly in form of multimedia.

Delivery of IP data is present in DVB systems in form of MPE (Multiprotocol Encapsulation) and ULE (Unidirectional Lightweight Encapsulation) protocols, however IP data delivery is not specified in HBB-Next.

This dissertation thesis propose changes to HBB-Next architecture and design new protocols for IP data delivery in HBB-Next architecture.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

Keywords

digital television, DVB, hybrid television, HbbTV, HBB-Next architecture, IP data, encapsulation

*Recommended by thesis supervisor: Assoc. Prof. Ivan Kotuliak
To be defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, 2015.

1. Introduction

Digital television is primarily used for television broadcasting, but it can be also used for IP data delivery. To encapsulate IP data into DVB broadcasting, protocols Multiprotocol Encapsulation (MPE) or Unidirectional Lightweight Protocol (ULE) can be used. These are focused on IP data delivery trough broadcast channel but are not utilized to use broadband channel for IP data delivery.

Today's television devices are used not only for watching TV channels, but can be also used to provide interactive applications. This trend has its roots in teletext but current television devices evolved and have more computational power, which can be used to show web pages and run applications. Televisions with these features are referred to as Smart TVs. Every vendor has its own Smart TV platform and their applications are not compatible. To remove this barrier standard HbbTV was created. It presents possibility of CE-HTML (Consumer Electronic HTML) based applications for compatible TV and set-top-boxes. Applications can be connected to broadcasted show and can provide additional information to it. HbbTV applications are currently mainly used for enhanced program information and video-on-demand or TV catch-up services. HBB-Next is an architecture which provides provides user recognition, content recommendation and user management features to HbbTV applications.

HBB-Next architecture requires for terminals to have access to broadband network connection. HBB-Next architecture and terminals have access to both broadcast and broadband channel. These channels could be both used to better IP data delivery for terminals. Aim of this work is to propose changes to HBB-Next architecture to provide more effective IP data delivery from applications to terminals. To achieve this goal new node in HBB-Next architecture was designed and new communication protocols were proposed and simulated.

This paper is organized as follows: state of the art of digital, hybrid, next generation hybrid television is described in Section 2. Section 3 describes IP data encapsulation. Design of changes of HBB-Next architecture and description of new communication protocols are proposed in Section 4. Section 5 summarize results and concludes this paper.

2. Digital television, HbbTV and HBB-Next

In this section we describe current state of digital DVB and hybrid television HbbTV and architecture for next-

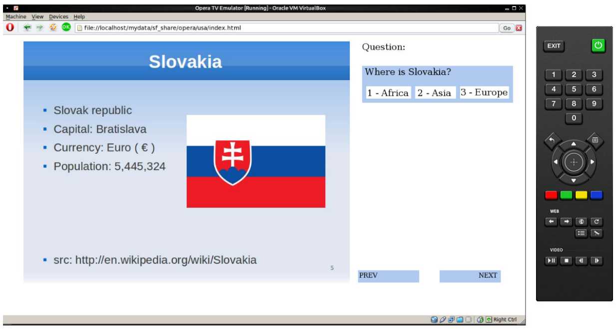

Figure 1: Example of HbbTV application used for e-learning

generation hybrid television HBB-Next.

2.1 Digital television

Digital television is replacing analogue television in many countries. It can provide multiple television channels on one multiplex. Video and audio streams are encoded with MPEG-2, MPEG-4 or H.264 codec [9] and packeted to PES (Packetized Elementary Stream) and into MPEG-TS packets [15]. These audio and video streams are identified by PID (Packet Identifier) numbers in MPEG-TS packet header and according to PSI/SI (Program Specific Information / Service Information) signaling tables mapped to TV channels [6].

In terrestrial DVB-T broadcasting, multiplexes are 7 or 8 MHz channels modulated with OFDM or QAM modulation [8]. Other DVB systems use other frequencies and modulations but data structure is same for every first generation DVB systems (DVB-T/S/C).

Digital television standard DVB enables IP data delivery. IP data delivery will be described in next section.

2.2 Hybrid television

Hybrid television HbbTV enhance digital television with added applications. These applications are based on CE-HTML programming language with specific JavaScript HbbTV extensions. These extensions enable controlling of television device functions directly from HbbTV applications (changing channels, volume, resizing picture) [10].

HbbTV applications serve mainly as interactive enhanced program guides, video-on-demand or catch-up television services, but it can also serve as an e-learning platform (Figure 1) [17].

It is possible to deliver HbbTV application in two ways:

- to encapsulate application data into DSM-CC (Digital Storage Media - Command and Control) object carousels and send it by broadcast channel,
- to signal application as a link to remote HTTP server over broadcast channel in AIT (Application Information Table) in PSI/SI tables and download application over broadband channel.

DVB broadcasting systems have a limited bandwidth therefore the most HbbTV applications are broadcasted as a

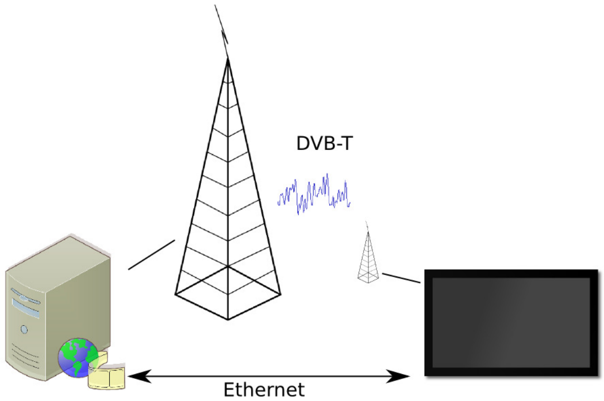

Figure 2: HbbTV application delivery

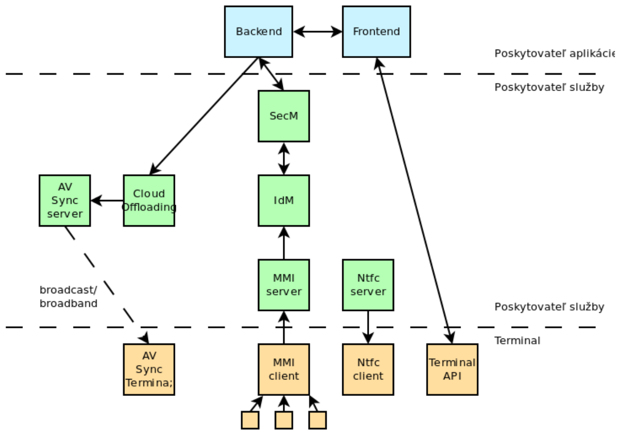

Figure 3: HBB-Next high level architecture

link through broadcast channel, and application and its content is delivered by broadband channel.

2.3 HBB-Next

HBB-Next consortium created architecture for HbbTV applications. This architecture (Figure 3) is designed to provide new features to enhance HbbTV applications [12]. The main features are user recognition by face, voice or iris, content recommendation for users, controlling applications with gestures, user management and security management. HBB-Next architecture consist of three main layers: application layer, service provider layer and terminal layer. Each layer consists of nodes providing additional features (Figure 4).

Application layer represents a HbbTV application which consists of backend and frontend. Backend is core part of application and frontend represents user interface and other interfaces of an application.

Service provider layer represents a HBB-Next core. It consists of multiple main nodes:

- Multi-modal interface (MMI) - receives information from terminal modalities (microphone, camera etc.) and recognize users,
- Security Manager (SecM) - provides information about current level of authentication of user to applications,

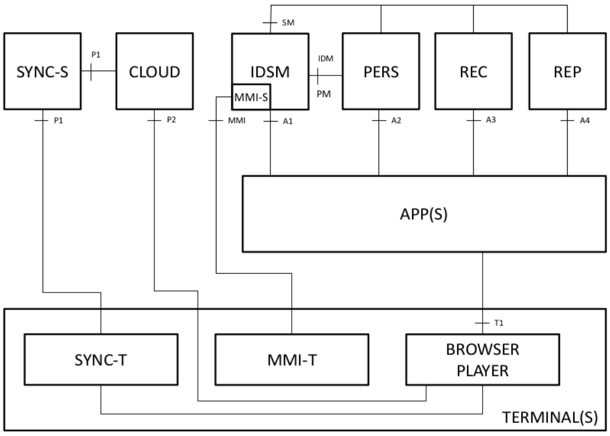

Figure 4: Service provider layer and terminal layer

- Identity Manager (IdM) - provides information about user, receives information about user presence from MMI,
- Recommendation Engine (RE) - provides content recommendation for users,
- CloudOffloading - used for offloading of resource heavy tasks (video encoding) to cloud service,
- Audio/Video Synchronization (AV Sync) - encodes audio and video to contain synchronization data.

Terminal is end-point device which has connected modalities (microphone, camera etc.), can communicate with HBB-Next service layer and is capable of showing HbbTV applications.

HBB-Next is focused on applications. It does not specify IP data delivery despite HBB-Next has access to both broadcast and broadband channel for data transmission, but both channels are available only to AV Sync node. Changing architecture to have an access for applications to both channels would enable more efficient data delivery. With access to both channels, application could encapsulate IP data. Existing DVB protocols for IP data encapsulation could be used in HBB-Next, but they are not designed to utilize both channels. We focus on IP data encapsulation in the next chapter.

3. IP data encapsulation

In this section we describe existing protocols used for IP data delivery in digital television systems.

3.1 Multi-protocol Encapsulation

Multiprotocol Encapsulation (MPE) protocol was designed as a native data encapsulation protocol for DVB systems [7]. It consists of packet header and a data unit (Figure 5). Packet header contains multiple header fields, mainly a MAC address of receiving device. MPE header can be 16 to 44 B long. MPE packets are encapsulated into MPEG-TS packets. There are two options of data encapsulation in MPE - padding and packing mode. In padding mode, MPE's padding data are placed into other MPEG-TS packet and additional meaningless data are added to fill the MPEG-TS packet. In packing mode, MPE's fragment data are placed into other MPEG-TS packet and next MPE's packet follows to fill the MPEG-TS packet.

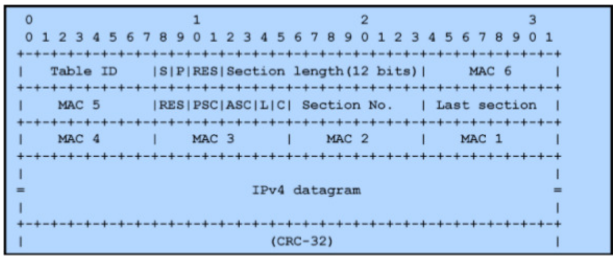

Figure 5: MPE protocol header [19]

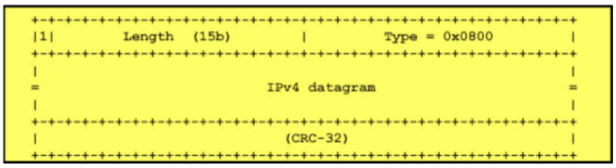

Figure 6: ULE protocol header [19]

This protocol is mostly used for IP data delivery over first generation digital television systems. Second generation systems also implement and use this protocol despite its overhead because of great compatibility of terminals.

3.2 Unidirectional Lightweight Encapsulation

Unidirectional Lightweight Encapsulation protocol (ULE) was designed to reduce MPE header overhead [11]. ULE header can be only 8 B long (half of size of MPE header). ULE consists of packet header and data unit (Figure 6). ULE packet contains multiple fields:

- destination address indicator - 1 b - indicates whether destination address is present in ULE header,
- length - 15 b - length of data unit with checksum field,
- type - 2 B - type of encapsulated data (using standard IANA numbers as in Ethernet 2 frames),
- destination address - 6 B - optional address in XX:XX:XX:XX:XX:XX format,
- data - encapsulated data,
- checksum - 4 B - CRC-32 value counted from beginning of ULE packet to end of data unit.

ULE has smaller packet header and therefore lower packet header overhead in comparison to MPE. By default, it is working in packing mode and is considered more effective protocol for IP data delivery over broadcast channel [20]. This protocol is not common in end-devices. This is the reason why multiple service providers still prefer MPE protocol.

3.3 Generic Stream Encapsulation

Generic Stream Encapsulation protocol (GSE) is protocol for data encapsulation in second generation DVB broadcasting systems (DVB-T2/S2/C2) [5]. It is more effective than MPE or ULE but it is not compatible with first generation DVB broadcasting systems [18].

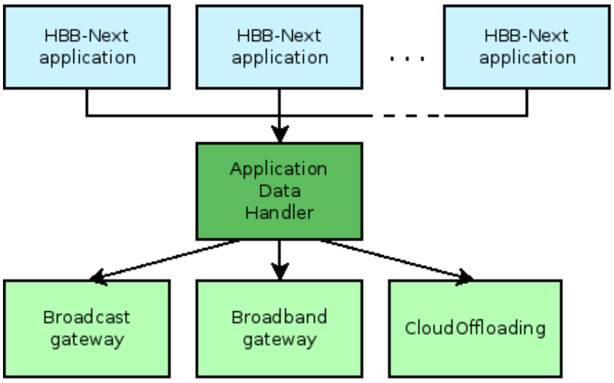

Figure 7: Application Data Handler

4. IP data encapsulation in HBB-Next

In HBB-Next, AV Sync node has only access to both broadcast and broadband channel. This node is used for delivery of audio and video streams. Other application data in HBB-Next has no access to broadcast channel and need to be delivered over broadband channel. If applications could use broadcast channel for data delivery, broadband bandwidth (mainly unicast) could be saved. Using broadcast channel would be also more efficient delivering application data to multiple terminals.

To deliver other than audio-video stream over broadcast channel in HBB-Next architecture, new nodes need to be designed which would receive application data and would have access to both broadcast and broadband channel. Using this channels it could send data to terminals.

Using existing protocols to utilize HBB-Next broadcast channel would be possible, however these protocols are designed for broadcast channels only. New protocol for data encapsulation could also utilize not only broadcast channel but it could be used also for data delivery using broadband channel. Using both channels, terminals could take advantage of parallel data delivery and maximize their download bandwidth.

To deliver IP data over changed HBB-Next architecture, new protocol could be designed to utilize possibility of usage of both broadcast and broadband channels. This new protocol could allow addressing data to one or multiple terminals according to number of terminals or usage of channels.

To enable IP data delivery in HBB-Next we added new Application Data Handler (ADH, Figure 7) node to HBB-Next architecture. This node is directly connected to broadcast and broadband channel. It is able to receive data from applications and sending it to terminal by one or both channels.

Communication with ADH node will be done using two designed protocols. Applications would use ADH Control Protocol (ADHCP) to deliver their data to ADH node. ADH node will process these data and would be able to send them to terminals with Hybrid Encapsulation Protocol, which was designed to allow data delivery over both broadcast and broadband channels.

Encapsulation of proposed Hybrid Encapsulation Protocol (HEP) from ADH to terminal has zero data overhead and minimal processing overhead on terminal side. By reduction of header overhead, better effeciency over MPE or ULE is achieved.

Figure 8: Simple Petri Net

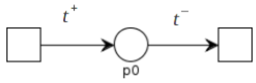

Figure 9: Definition of t^+ and t^-

5. Verification of designed protocols

To verify design of protocols, model of communication using proposed protocols was created. This model was created using Stochastic Petri Nets (SPN) and simulated using various tools. To verify communication closer to environment of real computer networks, proposed protocols were implemented in ns-2 network simulator.

5.1 Petri Nets

Petri Nets can be used to model event-based systems [16]. Petri Net consists of places, transitions and tokens. Places (p) are connected with transitions (t) by arcs. Tokens can be moved from place p_0 to place p_1 by firing transitions t_0 (Figure 8).

Marking describes position of tokens in places in selected state. Initial marking is referred to as m_0 marking.

Petri Net models have various properties. Every Petri Net can be rewritten into incidence matrix, where rows represent places and columns represent Δt, where Δt is difference between value of transitions entering place (t^+) and leaving place (t^-) as is showed in Figure 9.

Table 1: Basic symbols of Petri Nets (PN)

p	places in PN
t	transitions in PN
p_0	selected place in PN
t_0	selected transition in PN
t^+	transition giving tokens to selected place
t^-	transition taking tokens from selected place
Δt	$\Delta t = t^+$ - t^-
m	marking of PN
m_0	initial marking

Petri Nets have various properties, which can be verified. During our design of protocols we will focus on selected properties:

- boundedness - represent upper limits of tokens in selected place, it is examined for selected places but it can be also examined for whole Petri Net,
- reversibility - represent if Petri Net transitions can be fired repetitively from initial marking m_0 to another occurrence of initial marking m_0,
- reachability - represent if chosen marking can be reached from initial marking,

- liveness - represents firebility of transitions of Petri Net, show whether firing selected transitions doesn't lead to dead-lock.

These properties can be verified by application of algorithms on incidence matrix of Petri Net. Translation of call flow to Petri Nets leads to complex incidence matric. To help with complex Petri Nets and verifiction of their properties, there exists multiple Petri Net tools - PIPE [4] and Snoopy [13] etc.

5.2 Stochastic Petri Nets

Stochastic Petri Nets are Petri Nets with added probability of firing on selected transitions. This kind of Petri Nets is more suitable for modeling of communication flows of proposed protocols. It allows timings of execution of selected transition with given probability. This method is more comparable to real life scenarios, where communication processing is time related and probability can simulate various errors in communication.

Properties of SPN are inherited from Petri Nets. Created model can be therefor examined checking multiple Petri Net properties. Boundedness of Petri Net can show overflows in model of communication. Detecting sources and sinks (liveness) can show proper communication flows. Reversibility can show loops or point to reuse of created model. Reachability can show wheter final state (correct reception of application data) can be reached from initial marking (application sends data).

To create a communication model using SPN we transformed call flows of designed protocols to places and transitions of Petri Net. To model errors on broadcast channel we used stochastic transitions. Probability of firing of strochastic transition represented error on broadcast channel.

We simulated and verified properties of created SPN communication model using Snoopy and PIPE tools. Results showed expected properties of modeled communication.

5.3 Simulation in ns-2

There are existing results for comparison of MPE and ULE efficiency. These were done by simulating their implementation in ns-2 network simulator [14].

To verify design of proposed protocols, we created their implementation in ns-2. Using ns-2 simulator, designed protocols could be implemented on different protocol layers.

We implemented protocol ADHCP as an *Agent* to simulate data link layer connection between HBB-Next Application and ADH node (Figure 10). This approach more appropriately simulated connections in providers backbone network. Protocol HEP was implemented as data link layer protocol using broadcast channel and as application protocol of TCP using broadband channel (Figure 11). HHT table data transmission was implemented as data structure in payload of protocol TCP.

We created multiple network topologies to test our designed protocols. We started with node to node (Application and ADH, ADH and Terminal) communication using ADHCP and HEP protocols. Later we created simple

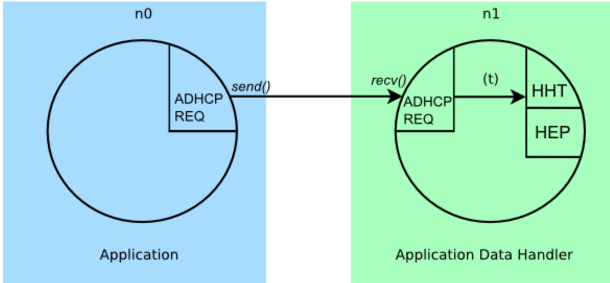

Figure 10: ADHCP implementation in ns2

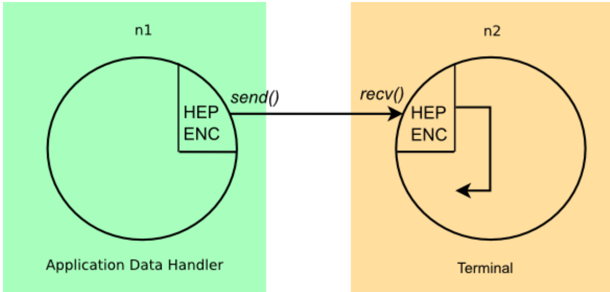

Figure 11: HEP and HHT implementation in ns2

topology with application, ADH node and two terminals. To test more complex scenarios we created simulation with application, ADH node and 100 terminals. Data were send from application, trough ADH, to terminals both using broadcast and broadband channels. Results showed correct delivery of application data to terminals. In case of error on broadcast channel, data were correctly requested and retransmitted using broadband channel.

5.4 SPN and ns-2 simulation comparison

SPN simulations are driven by firing of transitions. Simulations in ns-2 are time driven (data transfer on connections, data handling). Despite both simulation being in different domains, we were able to compare their results. Comparison showed only small difference (<5 % at maximum).

6. Implementation on real devices

To verify proposed protocols, implementation on a real terminal devices can be done. To access lowest data level possible on both channels, Linux operating system (OS) with a DVB driver and its network stack is preferred.

DVB interfaces can be easily managed in Linux using DVB driver and DVB API [3]. This driver creates devices (*frontend0, demux0, dvr0*) in */dev/dvb/adapter0*, where *adapter0* is first recognized DVB adapter. Modulation can be set using *frontend0* device. Device *demux0* can be than opened for reading using DVB API (written in C). Data can be filtered by MPEG-TS PID. Its HEP payload can be accessed, decapsulated and processed.

Broadband channel can be accessed using standard Linux network interface [1]. For data processing of proposed protocols PCAP library can be used to access L2 data. For better integration with Linux and better performance, data processing of proposed protocols can be also done by using socket structure *sk_buff*. Using this structure data from proposed protocols can be also accessed in L2.

Data from both channels can be parsed, merged and redirected into virtual network interface. This behavior is similar to MPE decapsulation using open-source OpenCaster tool [2], which process and redirects data from DVB interface into virtual network interface. Data from virtual network interface can be accessed by other application using network IP data (web browser etc.).

7. Conclusions

Current digital television is improved by HbbTV hybrid television, which brings interactive applications to television devices. HBB-Next architecture enhance this applications with additional features. HBB-Next architecture does not specify IP data delivery. We improved HBB-Next architecture creating ADH node which has access to both broadcast and broadband channel. This node is used to deliver IP data from applications to terminals. For this we propose new protocols ADHCP and HEP. We designed these protocols to be more effective than existing MPE or ULE protocols and to be able to use both broadcast and broadband channel for data delivery. We created model of communication of proposed protocols using Petri Nets and Stochastic Petri Nets (SPN). We verified its properties and simulated its behavior. Results verified design and properties of proposed protocols. Later we implemented proposed protocols in ns-2 network simulator. Results verified correct behavior of designed protocols. Results of simulation of SPN model and simulation in ns-2 were compared providing only a small difference. This is a good result taking into account simulations in two different domains. Results verify correct design of proposed protocols.

Future work includes an implementation on a real devices, preferably Linux devices. Proposed protocols can be easily added to both broadcast channel (dvb interface) and broadband channel (network interface). Using Linux compatible DVB tuner, even MPEG-TS data can be accessed directly what can be used to directly filter HEP data (direct payload of MPEG-TS packet).

Acknowledgements. This work was done in addition to FP7 HBB-Next project and proposed changes were partly presented in Deliverable 1.2.8 document.

References

[1] C. Benvenuti. *Understanding Linux Network Internals*. O'Reilly Media, 2005.

[2] A. Berger, L. Pallara, and R. Schatz. An open source software framework for dvb-* transmission. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 1093–1096, New York, NY, USA, 2008. ACM.

[3] Chehab, C. M. et al. Linux DVB API, 2012.

[4] N. J. Dingle, W. J. Knottenbelt, and T. Suto. Pipe2: A tool for the performance evaluation of generalised stochastic petri nets. *SIGMETRICS Perform. Eval. Rev.*, 36(4):34–39, Mar. 2009.

[5] DVB. Factsheet: Generic stream encapsulation - enabling the carriage of ip directly over dvb networks, May 2011.

[6] DVB. Document A38: Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems, Oct. 2013.

[7] ETSI. EN 301 192: Digital Video Broadcasting (DVB); DVB specification for data broadcasting, Apr. 2008.

[8] ETSI. EN 300 744: Framing structure, channel coding and modulation for digital terrestrial television, Jan. 2009.

[9] ETSI. TS 101 154: Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream, Nov. 2012.

[10] ETSI. TS 102 796: Hybrid Broadcast Broadband TV, Nov. 2012.

[11] G. Fairhurst and B. Collini-Nocker. Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS). RFC 4326, RFC Editor, Dec 2005.

[12] HBB-Next Consortium. Initial Version of the HBB-NEXT System Architecture. Deliverable 6.1.1, June 2012.

[13] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. Snoopy - a unifying petri net tool. In S. Haddad and L. Pomello, editors, *Application and Theory of Petri Nets*, volume 7347 of *Lecture Notes in Computer Science*, pages 398–407. Springer Berlin Heidelberg, 2012.

[14] T. C. Hong, W. T. Chee, and R. Budiarto. A comparison of ip datagrams transmission using mpe and ule over mpeg-2/dvb networks. In *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, pages 1173–1177, 2005.

[15] ISO/IEC. 13818-1 Information technology: Generic coding of moving pictures and associated audio information: Systems, Dec. 2000.

[16] G. Juhás. Modelovacie formalizmy udalostných systémov, May 2011.

[17] T. Kovacik, R. Bronis, and I. Kotuliak. Hbb platform for e-learning improvement. In *Emerging eLearning Technologies Applications (ICETA), 2012 IEEE 10th International Conference on*, pages 207–211, Nov 2012.

[18] A. Mayer, B. Collini-Nocker, F. Vieira, J. Lei, and M. Castro. Analytical and experimental ip encapsulation efficiency comparison of gse, mpe, and ule over dvb-s2. In *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pages 114–118, Sept 2007.

[19] B. C. Nocker and G. Fairhurst. ULE versus MPE as an IP over DVB Encapsulation. In *Proceedings of the 2nd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs-04)*, 2004.

[20] G. Xilouris, G. Gardikis, H. Koumaras, and A. Kourtis. Unidirectional lightweight encapsulation: performance evaluation and application perspectives. *Broadcasting, IEEE Transactions on*, 52(3):374–380, Sept 2006.

Selected Papers by the Author

T. Kováčik, R. Broniš, I. Kotuliak, P. Trúchly. Innovative Broadcasting Architecture for E-learning Improvement. In *ED-MEDIA 2012. World Conference on Educational Media & Technology, June 26-29, 2012 Denver, Colorado. Chesapeake*. AACE, 2012, s.1589-1594. ISBN 1-880094-95-9.

T. Kováčik, R. Broniš, I. Kotuliak, R. Martinický, M. Hreha. Wireless Sensor Networks for Next Generation Networks and HBB-Next Architecture. In *ICETA 2013. 11th IEEE International Conference on Emerging eLearning Technologies and Applications. October 24-25, 2013 Stary Smokovec, Slovakia.* IEEE, 2013, s. 229-233. ISBN 978-1-4799-2161-4.

T. Kováčik, R. Broniš, I. Kotuliak. Towards novel HBB application platform: experimental testbed. In *Proceedings of 2012 5th Joint IFIP Wireless and Mobile Networking Conference : Bratislava, Slovakia, September 19-20, 2012.* Piscataway : IEEE, 2012, s.133-137. ISBN 978-1-4673-2994-1.

T. Kováčik, R. Broniš, I. Kotuliak. HBB Platform for E-learning Improvement. In *ICETA 2012. IEEE 10th International Conference on Emerging eLearning Technologies and Applications, November 8-9, 2012 StarÃą LesnÃą, The High Tatras, Slovakia.* Piscataway : IEEE, 2012, s.207-211. ISBN 978-1-4673-5122-5.