A new Approach to IP-based Transmission of Audio and Video Content via DVB-Networks

Heiko Föllscher
Institut für Nachrichtentechnik (Institute for Communications Technology)
Technische Universität Braunschweig (Technical University of Braunschweig)
38106 Braunschweig, Germany
Email: foellscher@ifn.ing.tu-bs.de

***Abstract*— The following article proposes a structure of a network adaption layer (data link layer) for transmitting DVB-compliant audio and video content over an IP-network which itself resides directly on top of the physical layer of a DVB transmission system. Here "directly on top" should mean that no DVB Transport Stream is placed between the physical layer of the DVB system and the IP layer, as it is nowadays done when using Multi-Protocol Encapsulation (MPE). Basic purpose of this article is to summarize the considerations regarding such a data link layer which still has to be placed between DVB physical layer and IP layer. These considerations address the necessary information that has to be transmitted in addition to the IP-based information in order to achieve a similar functionality like in classical DVB. Furthermore, the article shows the approach taken towards DVB-compliant signalling of parameters and services within the IP domain. Finally, the result of a first simulation of the video transmission via such IP-based DVB system is described.**

I. Introduction

Audio and video broadcasting via Digital Video Broadcast (DVB) [1] already found a large acceptance all over the world. By adopting MPEG-2 solutions, among others the MPEG-2 Transport Stream (TS), in the 1990s the DVB Project chose a system which was optimized for the transmission of digital A/V content over disturbed channels.

However, digital transmission created the possibility and also the desire to broadcast not only the classical audio and video content, but also additional data which might be of interest for large groups of users, e.g. text, graphics or information about social or cultural events. Very often this data is also distributed via the internet in parallel to the DVB broadcast. Providers of such information normally process this data within their office network based on the internet protocol (IP) and also prefer to use the IP for distribution to the customers.

For transmission over DVB, IP-based information presently has to be encapsulated into the MPEG-2 TS. Since the TS was originally optimized for A/V transmission, additional data for the handling of IP-based information has to be added. This is achieved by the Multi-Protocol Encapsulation (MPE).

Taking into account all kinds of services which can be transmitted over DVB, the overall structure of the classical DVB protocol stack can be depicted as in *figure 1*. Video and audio information is encoded according to the MPEG-2 Standards [2] [3] and inserted into the MPEG-2 Transport Stream (TS). Housekeeping information, i.e. information about channel frequencies and the content of the transmitted programs etc. is also embedded into the TS by using a mechanism partly defined by the MPEG-2 consortium (encapsulation of Program Specific Information, PSI) and partly by the DVB project (encapsulation of Service Information, SI).

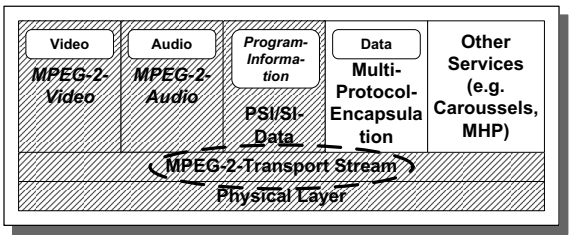

Fig. 1. Protocol Structure of the classical DVB system

So far, only the transmission of A/V content was mentioned for which the MPEG-2 standard was optimized. For data which is carried over other communication protocols like the Internet Protocol (IP), MPE has to be used. This leads to some overhead, caused by the necessary MPE header as will be shown in chapter II.

Other services like Multimedia Home Platform (MHP) applications also use the MPEG-2 TS together with specific encapsulation schemas which also add some overhead. *Figure 2*,

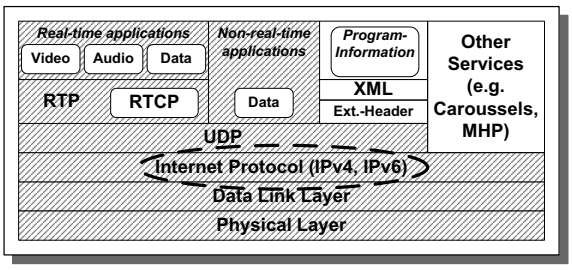

Fig. 2. Idea of IP-only DVB

in contrast, shows a totally IP-based solution. Here all services reside on top of the IP and its associated protocols like the User Datagram Protocol (UDP) or the Real-time Transmission Protocol (RTP). Real-time services like audio and video use UDP and the RTP protocol on top of the IP to ensure a synchronization between different streams by using the RTP time stamps. Program related information and other non-realtime applications which do not require a continuous transmission or special synchronization but rather use data which is downloaded onto the user terminal and then used "offline" simply use the IP/UDP stack. Other services like MHP applications can also be based on the IP protocol in principle, but will not be within the focus of this paper.

In order to place an IP transmission directly on the physical layer of DVB networks without using the TS, a thin additional

layer has to be introduced which adapts the IP packets to the restrictions of the DVB channel. This new layer has to provide some of the functionalities of the MPEG-2 TS which are vital for the transmission over DVB defined networks.
The following chapter describes these essential functions of the TS. Chapter III will then introduce the frame structure of the newly proposed data link layer, after which the parameters for encapsulating the data to be conveyed via the packets of the new data link layer will be examined in chapter IV.

II. MPEG-2 Transport Stream – Essentials

Being optimized for transport of audio and video data, the MPEG-2 TS includes a multitude of information within its header structure which is necessary for this kind of usage [1]. The MPEG-2 TS packet and its header structure are shown in *figure 3*. The main entries to the header which are of special interest here are marked by shaded fields in the figure.

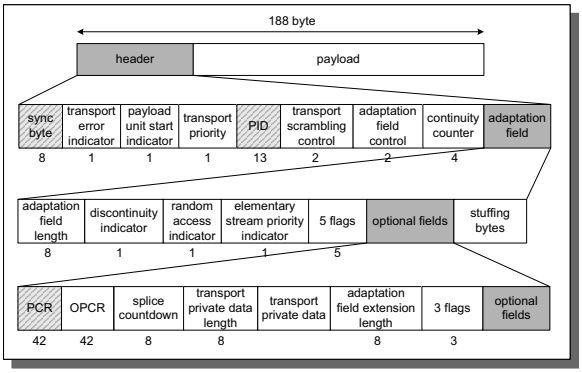

Fig. 3. Structure of the MPEG-2 Header

The *sync_byte* enables the receiver to recognize the beginning of a TS packet and thus the beginning of the header. The *packet identifier* (PID) is a 13 bit long field indicating the type of data encapsulated in the packet payload. By this, the data can be allocated to the program which itself includes several streams, e.g. one video stream, two or more audio streams and also data streams which are related to the program. The *Program Clock Reference* (PCR) is used to convey a separate clock for each program from the transmitter to the receiver. Together with the *Presentation Time Stamp* (PTS) and *Decoding Time Stamp* (DTS) which are transported within the encoded A/V stream, the PCR enables the receiver to synchronize the multiple incoming A/V streams for lip-synchronous display. Besides these entries in the TS header, there are several others, which are supporting the transmission of A/V content via the MPEG-2 TS. However, for further examination these entries are of minor importance, since their functionalities can be covered by higher layer protocols in the IP domain. Thus, for the newly defined data link layer it is crucial to look for a substitute of the "vital" entries mentioned above and to ensure that the higher layers provide the rest of the functionalities needed for proper delivery of the A/V data.
For encapsulation of other data, e.g. IP-based internet information, the already mentioned MPE [4] is used. *Figure 4* shows the structure of the MPE mechanism. The datagrams to be carried – most likely IP datagrams – are provided with a 12 byte long header and a 4 byte long redundancy check sequence and then inserted into the TS packet. This kind of

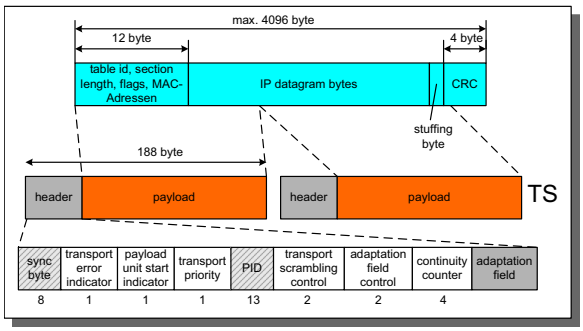

Fig. 4. Multi-Protocol Encapsulation: Insertion of IP datagrams into the MPEG-2 TS

encapsulation causes some overhead, because besides the TS header additional 16 byte have to be transmitted here in order to carry the IP data.
In the following, a structure will be presented which decreases this additional overhead of IP transmission by adopting IP as the transport protocol for all kinds of data and by adjusting the data link layer for usage on top of the DVB physical channels.

III. Framing in IP-only DVB

The main functionality covered by the new data link layer in IP-only DVB is to provide framing to the encapsulated IP packets. IP itself does not support the detection of the beginning of the IP header. For this, it relies on lower layer protocols like the ethernet protocol. In addition, the IP packets can have a variable size of up to 64 kByte.
For the data link layer proposed here it shall be assumed that the basic error correction of the DVB transmission – an outer block code concatenated with an inner convolutional or block code – will be used further. This means that the IP packets of varying length have to be inserted into a fixed length transmission frame of the data link layer. In classical DVB these packets (the TS packets) were chosen to be 188 byte long because the adopted Reed-Solomon(RS) error-correction code is only defined for packet lengths of up to 256 byte. But this restriction defined in 1993, the first phase of DVB, is not in place anymore because error-correction coding evolved since that time. The newly defined DVB-S2 system, an improved DVB satellite system, uses low density parity check (LDPC) codes, which opens the option for much longer packets. In the DVB-S2 case these packets were chosen to be 8100 byte or 2025 byte long. In the following, it shall be assumed that the frame size of the transmission frames in IP-only DVB can also be chosen freely, depending on the overhead caused by this length, and then will be constant over time with the chosen value. These frames will then have to be protected against errors by a suitable error-correction code. However, the Error-correction code that has to be chosen will depend only on the desired error rate and the length of the packet which has to be protected and will be independent from the information (IP based or TS-based data) carried within the packet. For example, a 188 byte long transport frame of the proposed IP-only DVB solution can be protected by the same error-correction code (in this case the RS-code concatenated with a convolutional code) as the classical TS packet. Longer transport frames have to be protected by a code which can cover longer packets than the RS code. However,

the efficiency of the bare transport frames (TF) without the error protection, measured as the ratio of payload and overall frame length (i.e. header plus payload), will not be influenced by this. For this reason, the overhead which is added to the transmitted packets by the error-correction will be ignored in the further examinations. An analysis for varying the length of the transport frames (TF) in order to define the optimized length will be introduced in chapter V.

The structure of the proposed header for the IP-only DVB transport frames is shown in *figure 5*. Within this header, the

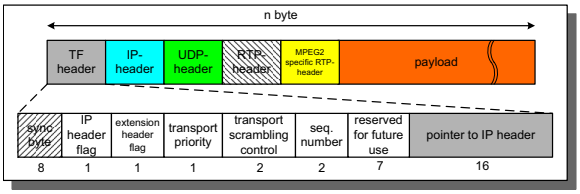

Fig. 5. Proposal of a header for data link layer packets in IP-only DVB

sync_byte serves the same purpose as in the MPEG-2 header. It denotes the start of a transport frame and can be easily distinguished from bytes in the payload having the same bit pattern because the TFs are all of identical length so that the appearance of the sync_byte shows a periodicity.

When encapsulating IP packets into the TFs, the IP packets are very unlikely to fit completely into an integer amount of TFs. There will always be some part which has to be placed into a new TF. But then the rest of the TF has to be filled with meaningless *stuffing bytes* which reduce the efficiency of the transmission. Another possibility is to place the beginning of a new IP packet directly behind the end of the previous packet into the same TF. This decreases overhead, but creates the situation that a new IP header has to be identified somewhere in the payload of the TFs. This is achieved by the *IP header flag* and the *extension header flag* in the TF header. The *IP header flag* signals (when set to 1) that there is an IP header in the following payload. Together with the *extension header flag* set to 0, it denotes that the IP header is directly following the TF header, while an *extension header flag* of 1 indicates that the beginning of the IP header is found at the position described by the 16 bit field *pointer to IP header* field at the end of the TF header. This pointer field will be present only when it is used, i.e. when the *extension header flag* is set to 1. DVB-T knows the concept of hierarchical modulation in which a high priority and a low priority stream are transmitted in parallel. The *transport priority flag* signals data with high priority. Scrambling of the payload for protection against unauthorized access can be denoted by the field *transport scrambling control*. And finally, aside from 7 bits which are reserved for future use (e.g. describing the protocol which is carried, besides IP), the sequence number ranging from 1 to 4 can be used to recognize any loss of packets, which is very unlikely in DVB transmissions.

The complete protocol stack for the IP-only DVB will then look as depicted in *figure 6*. Here only encapsulation of IPv6 is assumed, because transmission of a multitude of TV programs requires a large amount of different IP addresses, which is better supported by IPv6 than by IPv4. The whole stack is built upon the encapsulation of IP into transport frames

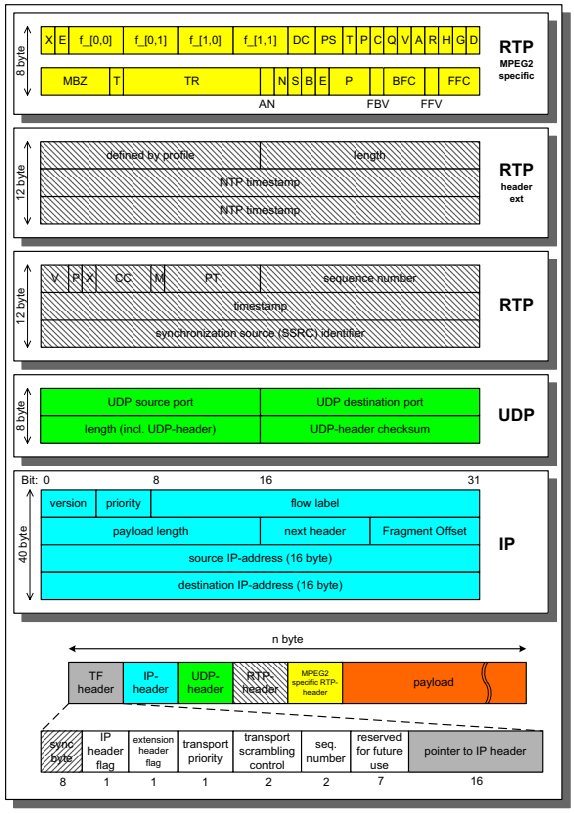

Fig. 6. Proposed protocol stack for IP-only DVB

as described above and then uses the UDP to carry RTP packets. Not yet covered by the transport frame structure is the question how to ensure synchronization mechanisms as they are used in the MPEG-2 TS. For this, the RTP [5] and its extensions can be used. RTP provides RTP time stamps which can be used to substitute the *decoding* and *presentation time stamps* (DTS and PTS). Furthermore, the RTP gives the possibility to use a header extension for individual usage. Within this header extension a system time similar to the *program clock reference* (PCR) can be included, providing a common time reference between encoder and decoder. All other A/V-specific functionalities as mentioned in chapter II will then be covered by the MPEG-2 specific header as defined in [7]. The substitution of the PID can be achieved by using IP addresses and port numbers, which will be described in chapter VI.

IV. Encapsulation of A/V content in IP-only DVB

After definition of a data link layer structure which enables the transmission of IP packets over DVB paths without the usage of the MPEG-2 TS, a closer look will be taken at the different possibilities to implement the encapsulation of IP-based audio and video data into the transport frames of this data link layer.

As shown in *figure 7*, the encoded data from the elementary stream, e.g. one MPEG-2 frame, could simply be taken and plugged into RTP packets by adjusting the RTP payload to the length of this frame and handing it down to the next lower layer, the UDP layer. Here again the payload of the UDP packet could be adjusted to the length of the received RTP packet and the UDP packet could be built from the whole

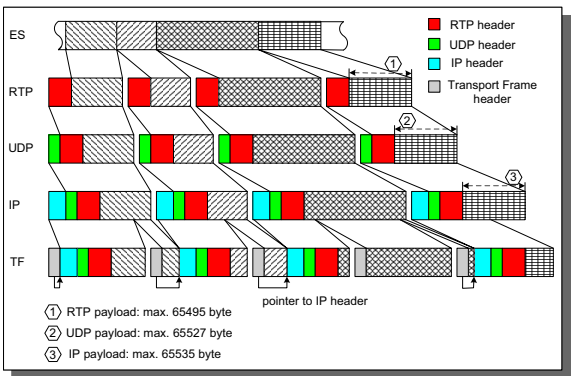

Fig. 7. IP-only DVB: Encapsulation for broadcast

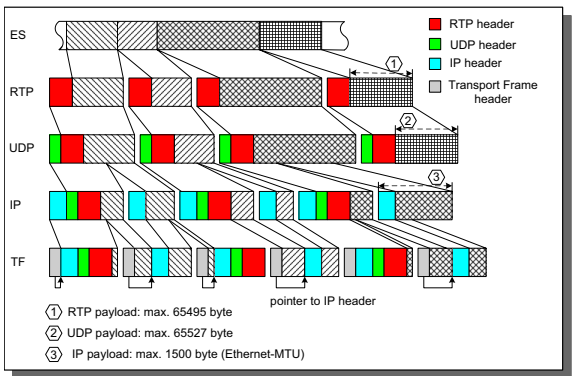

Fig. 8. IP-only DVB: Encapsulation compatible with Ethernet MTU

RTP packet. After handing the newly build UDP packet down to the IP layer, again the whole UDP packet could be inserted into one IP packet. Finally, the IP packet will be chopped into parts which have the length of the Transport Frame payload. These parts then can be protected by the appropriate error correction code and transmitted via DVB.

At first glance, the only stumbling block here could be that exactly one data unit from the encoder (e.g. one frame) has to fit into the payload of one RTP packet. However, since the packet length of RTP is not restricted, and since one RTP packet has to fit exactly into one UDP packet, the length limit of UDP is of importance. UDP packets themselves have to be embedded into IP packets, so the maximum payload of the IP packet is the really limiting factor. In IPv6 this payload length is limited to 65,535 byte (marked with 3 in *figure 7*). Subtracting the 8 byte UDP header from this, we get a maximum UDP payload of 65,527 byte for the UDP payload (2 in *figure 7*). This is concurrently the limit for the RTP packet length, so that the RTP payload (packet length minus header, number 1 in *figure 7*) is limited to 65,495 byte. An encoder has to be set to this limit for the data units it produces. However, 64 Kbyte is quite a large number, so this limitation does not appear too severe.

At second glance, it can be recognized that the usage of packets with maximum length does put some other restrictions onto the reception of the transmission. In the typical internet domain, a variety of different network technologies is normally switched together, each with different transmission capabilities and thus with different maximum transfer units (MTU). The MTU of the ethernet standard for example is 1500 byte, saying that only packets up to this size will be transmitted. So, with the solution of encapsulating IP packets of maximum length into the transport frames, the transmission via DVB means is enabled, but this makes it impossible for the receiver to distribute the IP packets over an in-home network directly.

In order to provide the possibility to send the packets which were received via DVB directly into an ethernet based network, the IP packets have to be limited to a total length of 1500 byte. Such an encapsulation would look as shown in *figure 8*. Here the UDP packets are fragmented into the payload of the 1500 byte long IP packets, causing the UDP fragments to be limited to 1460 byte (IP length of 1500 byte minus IP header of 40 byte). The encapsulation of the RTP packets into the UDP payload is simply done by attaching the UDP header directly to the RTP packet. This encapsulation method takes the MTU of Ethernet into account and simultaneously does not put a limit to the RTP packets besides the maximum UDP length of 64 kbyte. Nevertheless, it forces the UDP into a fragmentation which basically is not wanted.

Fragmentation is possible in each layer, but most likely unwanted within the typical internet layers. Thus, it might be best to already fragment the A/V frames which are produced by the encoders. Future examinations of the different fragmentation possibilities will show the best method. The next chapter will present first results.

V. Simulation Results

The following two graphs (*figures 9* and *10*) show the results from encapsulating two A/V streams into the IP-only DVB protocol stack. For this, two MPEG-2 TSs were taken and their overhead was measured. The first TS, called ”arnold.ts” consisted of an MPEG-2 video stream of 6 Mbit/s and an MPEG-1 encoded audio stream of 256 kbit/s. The overall size of the audio file was 111.880 byte, of the video file 2.884.296 byte. The average size of the video PES packets was determined to be 25.338 byte, of the audio PES packets 1.462 byte. The second TS was called ”daserste.ts”. It contained an MPEG-2 video stream of 2,757 Mbit/s, causing a file size of 2.645.809 byte and an MPEG-1 encoded audio stream of 192 kbit/s, file size 181.440 byte. The average size of the video PES packets in the stream ”daserste.ts” was 3.948 byte, of the audio stream 2.894 byte. Both streams were extracted from a multiplex thus including only the MPEG-2 defined program specific information (PSI), not the DVB specific service information (SI) anymore. First, the existing MPEG-2 transport streams were examined in order to calculate the overhead which was caused by the encapsulation into the TS. That means, the transported A/V sequence was taken out of the packet and the ratio of payload to overall length of the packet was calculated. For ”arnold.ts” an efficiency of 0.96 was calculated, resulting in an overhead of 3.4%. The efficiency of ”daserste.ts” turned out to be 0.942, showing an overhead of 5.8%.

For comparison purposes, the same payload as carried in the examined TSs was then encapsulated into the IP-only DVB protocol stack. For this, the length of the TFs was varied from 15 byte to 1020 byte and for each step the efficiency was calculated. *Figure 9* shows the encapsulation of the stream ”arnold.ts”, while *Figure 10* shows the encapsulation of the stream ”daserste.ts”

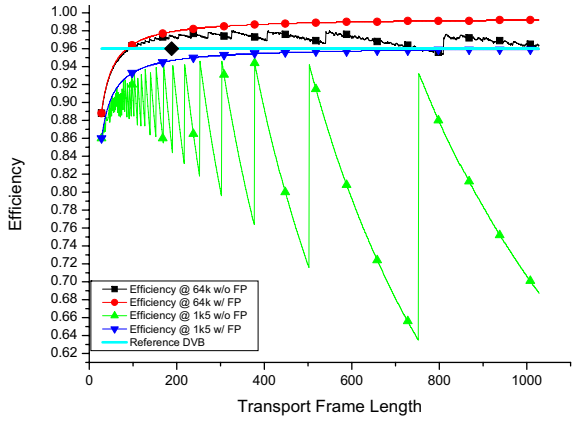

Fig. 9. IP-only DVB: Encapsulation of file ”arnold.ts”

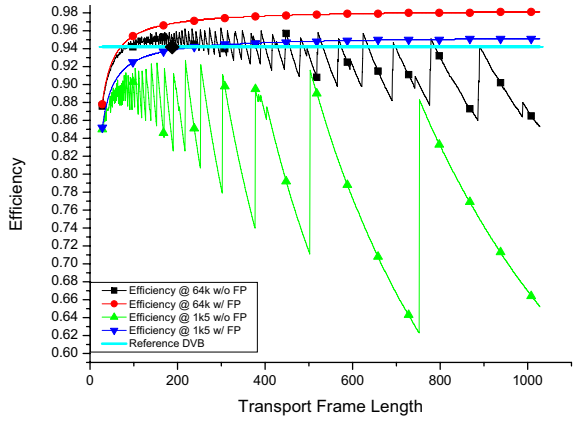

Fig. 10. IP-only DVB: Encapsulation of file ”daserste.ts”

The simulations include the line ”Reference DVB”. This line shows the efficiency of the 188 byte long MPEG-2 TS packets without error-correction in DVB for the chosen video sequence. As described in chapter III, this line describes a threshold for the efficiency of the transport frames in the proposed IP-only DVB solution. Here a direct comparison between the 188 byte long TS packets and TFs of the same length can be done. In each figure four different encapsulation schemes are presented, The curves marked by squares and dots denote the encapsulation of the A/V streams into transport frames according to the broadcast case depicted in *figure 7*. The zigzag curve with squares shows the encapsulation without frame packing (FP). This means that the transport frame is filled with stuffing bytes whenever an IP packet has ended. These stuffing bytes add to the overhead, causing the efficiency to decrease. The smooth curve marked by dots results from encapsulation with frame packing, so that a new IP packet begins within a TF as soon as the previous ends.
The curves with triangular marks describe the encapsulation shown in *figure 8*, enabling the receiver to further distribute the packets into a network using the ethernet technology. Here the zigzag curve with upright triangles shows the result without frame packing, while the smooth curve with upside-down triangles includes frame packing.
The light gray line parallel to the x-axis symbolizes the efficiency of the transport via MPEG-2 TS, which constitutes a threshold for the comparison. As soon as the efficiency of the IP-only DVB solution exceeds this threshold, it proves to be better than the classical TS method. Since this threshold is only valid for 188 byte long TS packets, the efficiency for a TS packet length of 188 byte is depicted by a large black diamond. *Figure 9* shows that the IP-only DVB encapsulation scheme for broadcast with frame packing reaches the threshold already with a frame length of approx. 100 byte and is always more efficient for transport frames longer than 100 byte. For the more realistic case of Ethernet-compatible encapsulation with frame packing the IP-only encapsulation is slightly worse than the MPEG-2 TS. For the stream ”arnold”, the IP-only encapsulation is 2.3 percentage points (efficiency 93.7% instead of 96%) worse than TS encapsulation, while for the stream ”daserste” IP-only is only 0.3 percentage points (93.9% instead of 94.2%) less efficient. The different efficiency values between the two streams are caused by the different lengths of the PES packets.

Result of the comparison is, that in case of pure broadcast the encapsulation of A/V content into IP packets and their transport via the proposed transport frames is more efficient than the usage of the MPEG-2 TS. A transport frame length of approximately 100 byte already shows a better efficiency than the 188 byte long TS packets. Thus, the length of the transport frames can be chosen freely. When frame packing is enabled, this length can be anything above 100 byte, since the curves for this case do not show any explicit peaks. The choice then is mainly influenced by the maximum frame length that can be handled by the system in respect to available memory and required time for processing. When frame packing is disabled, a frame length which causes a high efficiency will be chosen. Furthermore, the comparison shows that the usage of 1500 byte long IP packets in IP-only DVB, combined with frame packing, causes the transmission to be only slightly less efficient than the transmission via the classical transport stream. However, the 1500 byte long packets open up the possibility to distribute the received IP packets directly over an ethernet network without further conversion into other formats. Here a trade-off between slightly less efficiency of the transmission and easy retransmission has to be achieved. Nevertheless, it has to be noted that the shown comparison only takes into account the transmission of V/A data for which the TS itself is optimized. Transmission of already IP based data from the internet will shift the efficiency of both encapsulation schemas so that the IP-only DVB encapsulation will very likely be much more efficient for this usage than the TS. Further simulations will have to show how high this gain in efficiency will be.

VI. Service Discovery/Selection in IP-only DVB

The only entry in the MPEG-2 TS header which was not yet substituted is the *packet identifier* (PID). Since the transport frames of the IP-only DVB stack simply provide a container for the IP packets, such an identifier does not have to be included in the TF header. Identification of the payload and the destination of the packet can be denoted by the IP address within the IP header. Thus, the PID can be substituted by a combination of IP address and UDP port number.
In the area of A/V broadcast this implies that a broadcast IP address which is well-known to the receiver has to be used to transport the service information (i.e. the information about the used frequencies and the included services of the

transmitted stream). Starting with this information, the receiver can discover the services which are included and select the desired service by its dedicated IP address. The components of this service (e.g. video, audio or other related data streams) can then be addressed by using different UDP ports.
Transmitting the service information constantly on a well-known broadcast IP address is mandatory, because the DVB broadcast network does not necessarily provide a return channel. Thus it is not possible for the receiver to send a request for the service information to the transmitter. All data which is necessary for discovery and selection of broadcast services has to be present in the stream and accessible for the receiver without any interaction with the transmitter.

VII. Transmission via DVB-H

Recently, a new derivative of the DVB-T system was introduced, the so called DVB-H [6], [8]. It is aimed at handheld devices and was developed to decrease power consumption in battery powered receivers. DVB-H resides on top of the classical DVB-T transmission channels and introduces *time slicing* of the channel resources. While DVB-T transmits the data continuously, thus forcing the receiver to stay tuned all the time, DVB-H conveys the data in bursts, the so called time slices. *Figure 11* shows the principle of time slicing in DVB-H. In parallel to several conventional TV programs the DVB-T

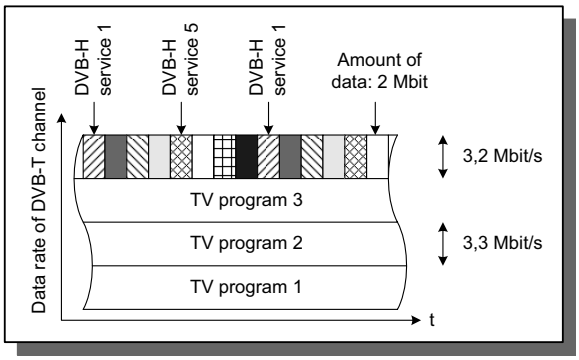

Fig. 11. Time Slicing in DVB-H

TS also carries a number of DVB-H services which share a specific fraction of the data rate of the channel. This fraction is separated into time slices and each service periodically uses the full fraction of the data rate reserved for the DVB-H services. *Figure 11* shows 8 DVB-H services in the 3,2 Mbit/s fraction of the DVB-T channel. Service 1 uses the DVB-H data rate for e.g. 625 ms, thus transmitting 2 Mbit during this time. The next cyclic occurrence of a time slice carrying data from service 1 will then be after 6 slices. During the transmission time of the slices carrying other services the receivers which are tuned to service 1 can go to an idle mode, thus saving energy. In addition to the time slicing, a supplementary error correction for the transmitted IP datagrams was invented. The parameters of the time slicing and the error correction are signalled within the header of the MPE mentioned above. Here 32 bits are used to convey the periodicity of the time slices and the parameters of the error correction code.
In the proposed IP-only DVB system, compatibility to DVB-H transmission can easily be achieved by inserting the 32 bit necessary for signalling of the DVB-H parameters directly into

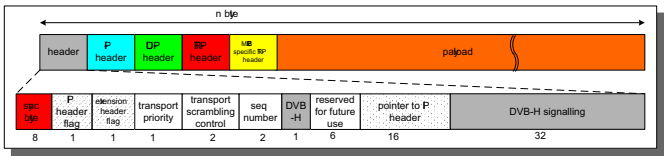

Fig. 12. Time Slicing in DVB-H

the TF header as shown in *Figure 12*. By this, the receiver can learn the time distances between the slices which belong to the desired DVB-H service and cease decoding of the DVB-T TS in the intermediate time intervals. Within the DVB Project, currently the IP Datacast (IPDC) [9] specification is developed, specifying the higher protocol layers for DVB-H, which are completely based on IP. Thus, an IP-only DVB system would facilitate the transmission. In the present DVB-H implementation, the MPEG-2 TS and MPE are only used in order to ensure downward compatibility to DVB-T. Utilization of the IP-based system would avoid the additional overhead of these protocols and increase the efficiency of transmission.

VIII. Conclusion

In this article, a new data link layer for direct transmission of IP-based data on top of the physical layer of DVB broadcast networks without the usage of the MPEG-2 transport stream was introduced. The structure of the data link layer frames was derived from the vital functionalities originally provided by the MPEG-2 transport stream and a header structure was defined providing these vital functionalities for broadcast. Functionalities provided by the internet protocol and its affiliated protocols were described. Different possibilities for encapsulation of the IP packets carrying A/V content into the newly defined data link layer were shown. Finally, a comparison between A/V transmission via the new encapsulation mechanism and via the conventional MPEG-2 TS was made. For this comparison two television streams were transmitted, using each of the described mechanisms. The result proved that the IP-only DVB encapsulation was more efficient than the MPEG-2 TS in a typical broadcast usecase. Furthermore, it was shown that the proposed IP-based transmission opens the possibility for seamless transition into ethernet networks. This option is only slightly less efficient than the TS encapsulation, but gives the benefit of much easier IP handling. Another advantage is the easy adoption of DVB-H in combination with the IP-only DVB system and the decreased overhead compared to TS based DVB-H usage.

References

[1] U. Reimers, *DVB The Family of International Standards for Digital Video Broadcasting*, 2nd ed., Springer 2005
[2] ISO/IEC 13818-2 Information Technology - Generic coding of moving pictures and associated audio information: Video
[3] ISO/IEC 13818-1 Information Technology - Generic coding of moving pictures and associated audio information: Audio
[4] ETSI EN 301 192 Digital Video Broadcasting (DVB); DVB specification for data broadcasting
[5] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications
[6] ETSI EN 302 304 Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)
[7] RFC 2250 - RTP Payload Format for MPEG1/MPEG2 Video
[8] Kornfeld/Daoud, *DVB-H: Digitaler Rundfunk für Smartphone, PDA & Co.*, FKT 1-2/2005
[9] G. May, *The IP datacast system - overview and mobility aspects*, IEEE Conference ISCE 2004