

# Transmission of DVB Service Information via Internet

Artur Lugmayr<sup>1</sup>, Seppo Kalli<sup>1</sup>

<sup>1</sup> Technical University of Tampere, Digital Media Institute, P.O. Box 553,  
FIN-33101 Tampere, Finland  
{lartur, skalli}@cs.tut.fi

**Abstract.** It is a fact, that more and more applications make use of networks. One of those applications will be digital interactive television, relying on a generic protocol solution to provide broadcasting and interaction over network dependent protocols. A digital television transmission, based on a MPEG2 transport stream consists of multiplexed video, audio and Service Information (SI) streams. Service information is important for describing the content of a transport stream and encapsulating various protocols. Because plenty of research work is done on pure MPEG2 audio and video transmissions, this paper deals with service information transmissions, its mechanisms and network requirements if broadcasted over the Internet only. For this reason a DVB-SI broadcast protocol stack will be introduced. UDP and RTP with its packetization schemes will be evaluated and bandwidth consumptions, boundaries and requirements in a LAN/WAN environment evaluated. Finally a software framework for transmission and decoding SI will be presented.

## 1. Introduction

Digital television is going to be realized in the next few years and seems to be the biggest revolution in broadcasting since the introduction of color television [4].

The provided standards are (almost) complete and ready to be implemented. For Europe the key standard will be the *Multimedia Home Platform (MHP)*, published by the *European Broadcasting Union (EBU)*. It is relying on appropriate *Digital Video Broadcasting (DVB)* specifications for digital video broadcast and associated interactive services. The MHP is applicable to all DVB defined transmission media and networks such as satellite, cable, terrestrial and microwave [5].

Several topics of and standards of digital television are covered by the “umbrella-name” DVB: interfaces visible to applications, integrated TV sets, multimedia computers, communication protocols, recommendations, interaction, broadcasting, etc.

Figure (1) presents a generic system reference model for interactive DVB systems, consisting of broadcast service provider, interactive service provider, interaction network, broadcasting network, several interfaces and end user. Both, broadcast and interaction channels - established between user and service provider - are presented.

The broadcast channel represents a unidirectional broadband broadcast channel including video, audio and service information. A bi-directional interaction channel is established between the user and the service provider for interaction purposes. It is formed by a return interaction path and a forward interaction path [6]. The channel specific protocols may be split between network dependent and network independent protocols.

DVB relies on MPEG2 for encoding, decoding and transmission of video, audio and services. This section contains a brief description of MPEG2 basics and focuses on the DVB-SI especially.

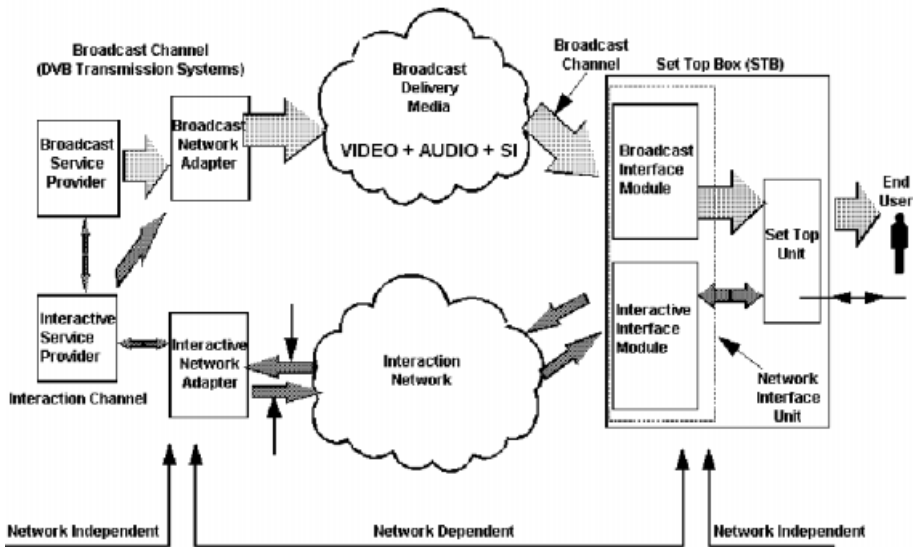


Fig. 1. Generic System Reference Model for DVB [6]

Moving Picture Experts Group (MPEG) is one of the most popular audio/video compression techniques because it is not just a single standard. Instead it is a range of standards suitable for different applications but based on similar principles [7]. Multiple audio, video and service streams, thus program streams, are multiplexed within a Transport Stream (TS). For each program a different compression factor and a dynamic bit rate can be used. This behavior allows dynamical bandwidth handling and is called statistical multiplexing. Besides video and audio a TS contains service information concerning PSI and DVB-SI. Concerning digital video broadcasts service information describes the delivery system, signal content and scheduling/timing of broadcast data streams etc. and includes MPEG2-PSI together with independently defined extension [8].

As mentioned above a TS is a multiplex of video, audio and data. The raw output of an encoder, containing only data to approximate the original video and audio, is called *Elementary Stream (ES)*. Note that the continuous elementary stream contains raw data for one audio/video/data stream only and no error recovery is provided. For

practical purposes it is essential to break this continuous elementary stream into packets. Each packet in the *Packetized Elementary Stream (PES)* consists of a header and a payload. The header contains synchronization time stamps and is used for identifying PES packet. The size of each packet varies with the application. Those PES packets are multiplexed within TS packets, which are uniquely identified by its PID.

*DVB Service Information (DVB-SI)* describes the delivery system, content and scheduling/timing of broadcast data streams etc. DVB-SI is multiplexed in the MPEG2 transport stream and is based on *Program Specific Information (PSI)* and several extensions.

## 2. Related Work

The MHP is specified in [5] and is a good starting point to get an overview over digital broadcasting standards and specifications. Furthermore this specification defines the interface visible to applications. The MPEG2 based transport stream multiplex is specified by [9]. An excellent technical description covering MPEG2 compression, streaming and testing can be found in [7]. Network dependent protocols in the DVB context address the interaction channels and are described in [6] and [12]. Network independent protocol specifications can be found in [10], [13], [15], [16], [17] and [19], issuing DVB encapsulation in existing networking protocols. A detailed description of DVB-SI can be found in [8] and implementation issues are covered by [18]. *Basso, Cash and Civanlar* addresses the transmission of MPEG2 streams over non-guaranteed quality of service networks. They discuss issues related to the transport of MPEG2 streams over such network by the *Real-Time Transport Protocol (RTP)* [2].

## 3. DVB Protocol Layers

The DVB protocol stack facilitates the OSI reference model up to the fourth layer. The application layer is not covered by any standards, and applications are left open to competitive market forces [6]. Basically the DVB protocol stack identifies a network dependent (up to the 4th OSI layer) and a network independent protocol (upper layers). The purpose of the different DVB protocol stacks is acting as interface between upper and lower layers and to maintain a logical connection between user and broadcast/interactive server provider, as presented in Figure (1).

It is important to distinguish between different kinds of connections that can be established between broadcast service provider, interactive service provider and end user. The broadcast channel is used as downstream channel for video, audio and data. The interaction network addresses application control data, application communication data and/or data download control [6]. Depending on the transport medium, various protocols for transmission are provided, e.g. [16] addresses the interaction channel through GSM.

Basically two different protocol stacks are specified for combining network independent and network dependent protocol suites: broadcast channel protocol stacks and interaction channel protocol stacks. Figure (2) shows all layers involved in encapsulating a DVB stream within a broadcast medium. The lower four layers are exactly the same as in the ISO-7-layer reference model and consist of physical, data-link, network, transport and application layers. Each layer has its ISO predefined purpose and protocols, depending on the kind of connection established between broadcast service provider, interactive service provider and the end user. A description of ideas, underlying protocols and encapsulation methods of each layer can be found in the following sections.

### 3.1 Broadcast Channel Protocols

The purpose of the broadcast channel protocol stack is to encapsulate the MPEG2-TS to the broadcast medium. The responsibility of the interaction channel protocol stack is to carry information from the user to the provider and vice.

#### 3.1.1 Broadcast Delivery Layers

The responsibility of the broadcast delivery layer is to establish a point-to-multipoint connection for the delivery of a MPEG2 transport stream between the broadcast service provider and the end-user.

There are many possible network configurations covering the currently specified DVB broadcast options including satellite, terrestrial, cable, SMATV and MMDS in conjunction with PSTN, ISDN, cable and other interactive channel options [5]. As this paper focuses on the transmission of DVB-SI over the Internet, only possible Internet protocols are considered and presented.

The content of the stream may be video, audio and/or data. The main application area and profiles addresses *data piping, data streaming, multiprotocol encapsulation, data carousels and object carousels* [12]. Data pipes support asynchronous, end-to-end delivery of data and is carried directly in the payload of a MPEG2 TS packet. Data streaming supports a streaming-oriented, end-to-end delivery of data in an asynchronous, synchronous or synchronized way. This addresses the ability to broadcast streaming media, which contain video and/or audio. Within the TS multiplex such data would be the PES.

Multiprotocol encapsulation is used to map higher-level protocols in the data portion of lower layer protocols. It is essential for carrying transport units of some protocol within another type of protocol. Especially mutliprotocol encapsulation is essential for two reasons: Firstly it allows to broadcast services that require the transmission of IP-datagrams of various communication protocols. Secondly multiprotocol encapsulation guarantees the transmission of a DVB stream over another protocol type, thus broadcasting over the Internet for example.

A point-to-multipoint transmission of a very large amount of data has to be established between the broadcaster and the end-user. The considered protocol has to be able to establish a point-to-multipoint connection. Thus manly the *User Datagram Protocol (UDP)* and the *Real Time Transfer Protocol (RTP)* can be considered.

The UDP provides an unreliable connectionless delivery service using IP to transport messages between machines. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer [1]. UDP utilizes the networking services provided by the underlying IP layer in addition allows multiplexing, checksum services and distinguishing among multiple processes within the operating system by assigning port numbers to each message.

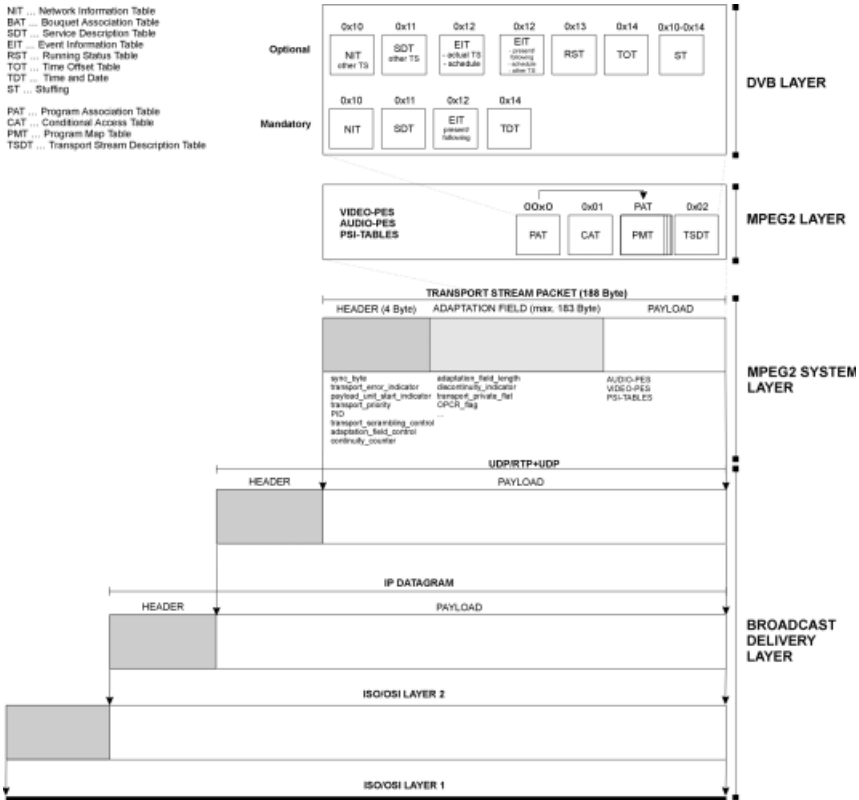


Fig. 2. DVB stream encapsulation by using the Internet as broadcast delivery medium

The *Real Time Transfer Protocol (RTP)* usually relies on UDP and provides payload type identification, sequence numbering, time stamping and delivery monitoring [2]. RTP provides a time stamping service and a synchronization mechanism for jitter measurement. Time stamping is derived from a 90 KHz clock and carried within a 32 bit field. Currently three different types of payload are defined to carry a MPEG2 stream: Firstly the payload based on encapsulated MPEG2 PS and TS as described in [21]. This approach relies on the MPEG2 timing model as based on MPEG2 Program Clock Reference (PCR), Decoding Time Stamps (DTS) and Presentation Time Stamps (PTS) [9], the RTP timestamp mechanism is not used. TS packets are packetized that each RTP packet contains multiple of MPEG2 transport packets. Secondly the payload based on separate packetization of MPEG2 audio and

video elementary streams [21]. This approach reduces packet dependencies, maximizes throughput and uses the RTP timing mechanisms through synchronization provided by the *Network Time Protocol (NTP)* [22].

The third approach uses one transmission and one reception port per bundled A/V and services, as audio/video and relating service information are inseparable and mostly used together. This approach provides an implicit synchronization of audio/video and service information, reduces the header overhead and reduces overall receiver buffer size allowing a better bandwidth control [2]. This should reduce any overheads through redundant information in various layers and provides easier error recovery. According to [2] the overhead can be up to 2-3% of the overall bandwidth.

### 3.1.2 The MPEG System Layer

Each transport unit within a MPEG2 transport stream is called a *transport stream packet*. Conceptually a transport stream packet consists of three parts: 4-byte header, adaptation field and payload area. The whole length of a transport stream packet is exactly 188 bytes. The length of the payload depends on the length of the adaptation field, the length of which is limited to 183 bytes. Figure (2) shows the header fields of a TS packet. The adaptation field is optional and need not be used at all. It is used if more header information is essential and the header is larger than usually. To maintain the packet length of 188 bytes, the payload is reduced. The extended header information includes priority and clock reference.

### 3.1.3 MPEG and DVB Layer

The MPEG layer contains demultiplexed audio and video streams as *Packetized Elementary Streams (PES)*. Service information is represented in PSI tables. This layer describes and holds information of only one transport stream. The description of more transport streams can be found in the DVB layer. The DVB layer contains not only SI tables of one TS, but it contains various other tables containing information of a whole program bouquet, networks, satellite transmission, program events and service descriptions. The table and service information mechanism is described in further detail in the next sections.

The raw output of a MPEG encoder is an ES and contains any information needed to approximate the original video/audio. No fault tolerance and no error recovery are provided. The continuous ES needs to be broken in packets out of practical purposes. This stream is called PES.

## 3.2 Interaction Channel Protocols

For the interaction channel the whole bunch of Internet protocols can be used. The MHP standard allows access to IP, TCP, UDP and HTTP. But there is no limitation in using other protocols that can be encapsulated within IP. MPH also provides a registry mechanism for new broadcast protocols for data broadcast services.

## 4. DVB Service Information (DVB-SI)

Various tables, defined by ISO/IEC or ETSI, describe DVB-SI. Each DVB-SI carried in transport stream packets has unique PIDs, whether standardized or assigned by the *Program Association Table (PAT)* or *Conditional Access Table (CAT)* described in this section. Each packet carrying DVB-SI must be sent periodically in every TS. ISO/IEC 13818 [9] specifies service information, which is referred to as PSI. The PSI data provides information to enable automatic configuration of the receiver to demultiplex and decode the various streams of programs within the multiplex. For identifying services and events for the user, additional information is needed. This includes information on services and events carried by other transport streams or networks. The coding of this data is defined by ETSI in ETS 300 468 [8].

In the following the most important tables are mentioned:

- *Program Association Table (PAT)*: The PAT provides the correspondence between a program number and the PID value of the TS packets that carry the program definition. The program number is the numeric label associated with a program [9]. It specifies the PIDs of all Program Map Tables (PMT) packets.
- *Conditional Access Table (CAT)*: The CAT table provides the association between one or more CA systems, their streams and any special parameters associated with them [9]. The CAT always has a PID of 1 and must be multiplexed in every TS.
- *Program Map Table (PMT)*: The PMT provides the mappings between program numbers and the program elements that compromise them. Thus, the PMT is the complete collection of all program definitions for a simple TS [9].
- *Network Information Table (NIT)*: The NIT conveys information relating to the physical organization of the multiplexes/TSs carried via a given network and the characteristics of the network itself. Through the combination of two fields in the NIT section (original\_network\_id and transport\_stream\_id) each TS can be uniquely identified [8].
- *Service Description Table (SDT)*: Each sub-table of the SDT describes services that are contained within a particular TS. The services may be part of the actual TS or part of other TSs, these being identified by means of the table\_id [8].
- *Event Information Table (EIT)*: The EIT provides information in chronological order regarding the events contained within each service [8]. The EIT contains information about the current TS or other TSs. It covers present/following event information or event schedule information.

Sections defined by the broadcaster can be sent via private sections. They can occur in transport streams, that are exclusively assigned for sending private sections, or in various other streams, whose PID is defined in the PMT.

### 4.1 Decoding of DVB-SI

The first issue in demultiplexing a transport stream is to look for PIDs 0 and 1 in the multiplex. Packets with the PID 0 contain the PAT, whereas the PID 1 is carrying the

CAT table. Upon reading the PAT, the PIDs of the NIT and each PMT can be found. By decoding the PMT, the decoder is able to obtain each PID of each elementary stream. If programmes are encrypted, the CAT has to be accessed; therefore the CAT section at PID 1 has to be read. For the whole DVB-SI more tables have to be decoded and read. The previous four tables describe only information to demultiplex various streams of programs within the multiplex and therefore called program specific information. Figure (2) shows the process of decoding PSI data from a transport stream. More detailed information for identifying services and events for the user can be found in other DVB-SI tables.

Each DVB defined service information table might consist of a descriptor loop. The descriptors identify the content of the current table and are defined in various ETSI standards. Each descriptor consists of a `descriptor_tag` for identifying the descriptor, the `descriptor_length` and the specific descriptor data – for example the title of the current movie or the classification information for a current event is classified by its `descriptor_loop`.<sup>1</sup>

## 4.2 Mapping of Sections in Transport Stream Packets

It is essential to distinguish between tables and sections. A section is a syntactic structure used for mapping all predefined service information into transport stream packets [8]. The length of one section is variable, but does not extend 1024 bytes, except the EIT, whose maximum length limit is 4096. Whereas a table consists of one section minimum and is uniquely identified by the `table_id`. The content of the table depends on its use and its definition. A table provides information about a certain matter, addressing various needs of a DVB transmission. Each service information section is uniquely identified by `table_id` and `table_id_extension`. All sections are directly mapped into transport stream packet, without any mapping into PES. The start of a section in the payload of a transport stream packet is pointed to by the only one `pointer_field` or by default at the beginning of the payload. One pointer field is enough, as the length of each section can be estimated by the length field of each section and as one restriction of the MPEG standard issues, that no gaps between sections within a transport stream packet are allowed. Each section has to be finished, before the next one can start in packets with single PIDs, as otherwise it is not possible to identify to which section the data belongs.

In case a section does not completely fill up one transport stream packet and it is not advised to start a new section, stuffing has to be used to fill up the free space. There are two ways to perform stuffing: Firstly through the adaptation field mechanism, secondly by filling remaining bytes with the value 0xFF. The decoder can discard the bytes with the value 0xFF. Therefore the `table_id` 0xFF is prohibited.

---

<sup>1</sup> Example: The EIT consists of a `content_descriptor` to provide classification information for an event. If the current stream carries a romance, this is classified by its `content_descriptor`.



## 5. Requirements in Broadcasting a SI Stream

Network requirements in broadcasting a DVB stream are related to signal acquisition time, bandwidth utilization, table repetition rate and protocol performance. These concepts are presented more detailed in the following:

*Signal Acquisition Time:* Signal acquisition time is the time needed to decode a particular table as part of the SI. It is assumed in [9], that the worst-case contribution to signal acquisition time for retrieving PSI is approx. 80ms based on a 25Hz PSI frequency. Signal acquisition time depends on protocol performance, implementation, hardware configuration and error recovery.

*Repetition Rate:* The repetition rate addresses the frequency at which each service information table is sent. MPEG2 defined PSI tables should be sent with a frequency of 25Hz. For DVB rates only minimum repetition rates are defined. The NIT, BAT, SDT for other TSs, EIT for other TSs and the EIT for the first 8 days ahead shall be transmitted every 10 seconds. Whereas SDT and EIT for the actual TS should be transmitted at least every 2 seconds. TDT, TOT and EIT for further than 8 days need only to be transmitted every 30 seconds.

	1			10			128		
	PSI	DVB	Sum	PSI	DVB	Sum	PSI	DVB	Sum
1	3008	155063	158071	4512	234692	239204	33088	18305184	18338272
*	75200	73397	148597	112800	101104	213904	827200	9132741	9959941

**Table 1.** Comparing bandwidth usage (in bps) of DVB SI with frequency table sent and number of programs per TS. “\*” frequency marks the minimum frequency all tables must be sent according the standards

*Bandwidth Utilization:* Bandwidth utilization varies in number of programs, MPEG bandwidth for audio/video, frequency of sending service information tables, size of tables and number of encapsulated IP datagrams and private sections. Previous experiments already evaluated the performance issues of different MPEG2 packetization schemes for video/audio [2]. The following bandwidth calculations assume that the transmission of minimum SI without any protocol overhead, conditional access, scrambling and stuffing tables. The following are based on following assumptions: Average descriptor length of 20 bytes, 10 descriptors per descriptor loop and minimum SI information sent. This means minimum additional information in various table sections; furthermore no information about other TSs or networks, no user defined tables and encapsulated IP datagrams are included. Each section is fitted into a single TS packet, which is filled up with essential stuffing bytes. The number of TS packets needed to transmit one-table results in the bandwidth usage (bps). Only one section can be fitted into each packet.

Table (1) shows the MPEG and DVB related bandwidth usage. Generally the SDT and the EIT are the largest tables in the multiplex. Assuming ten 8Mbps programs (MPEG video/audio at high quality), the total SI bandwidth would be 213,904bps. This leads to an SI overhead of about 0.27%.

*Protocol performance* can mostly be evaluated on practical tests as it highly depends on the implementation of the protocol stack, packetization scheme and

network load. One way of measurement is to set up criterion addressing the whole bandwidth of disturbance that might occur in transferring service information. This includes packet loss, counting the number of transmission disturbances at the client side and packet acquisition time delay in a LAN and WAN environment.

## 6. Implementation Issues

The implementation platform is a Intel Pentium II with Windows98 as operating System. For retrieving a real MPEG2 TS a COCOM CC1016 DVB-S data receiver connected to a satellite dish was used.

### 6.1 Software APIs

The whole implementation is based on Java with native interfaces to the SDK to the DVB-S receiver card. It can be divided in four packages: *COCOM Drivers/SDK*, *CC1016 API*, *MHP SI Access API* and *Application API*.

The COCOM Drivers/SDK API provides a wide range of functionality: selection channel parameters and frequencies, open/close a TS packet stream, add/remove PIDs to each stream, stop/start data reception, read data, get statistics and various system status calls. This API represents the lowest layer of the implementation and allows access to the data received by the COCOM DVB-S receiver card.

The objective of the CC1016 API is to provide access to data in MPEG sections. It should allow a convenient application-level access to data transported in MPEG sections and filtering mechanisms. Furthermore it provides mechanisms to retrieve pure PES packets, TS packets and channel navigation possibilities. Through the implementation is purely in Java and to provide fully functionality for upper software layers several access methods of the COCOM SDK are included by the use of Java native interfaces.

The MHP SI Access API is completely MHP (Revision 16/Section M) compliant. It makes use of the CC1016 API as well as the COCOM SDK. Its objective is to provide an accurate SI access. This includes parsing of SI tables, managing and monitoring SI, handling SI requests and initializing the whole filtering process.

The Application API is multifunctional. Its responsibility is to provide an accurate user interface, controlling the initialization and configuration process, managing network connections, accessing lower layer APIs and retrieving SI out of a file or from the DVB-S adapter. Currently there are three major possibilities in retrieving SI: from a file, server or directly from the DVB-S adapter. SI can be visualized and accessed by sending requests to the MHP SI Access API - independent by which way SI is obtained - by a very simple user-interface.

The generation of SI service out of a file bases on minimum repetition rates and table sizes as defined in the file. This included the sending of PAT, PMT, NIT, SDT and EIT per TS. The length of the tables was restricted to their minimum lengths<sup>2</sup>.

---

<sup>2</sup> 2 byte for program associations; a PMT size of 26 bytes; an EIT size of 4048 bytes, assuming average descriptor length; NIT, TDT and SDT of their minimum size;

The applied encapsulation schemes have been described in previous sections. UDP and the three different packetization schemes of RTP have been used. In one UDP packet one TS packet got encapsulated.

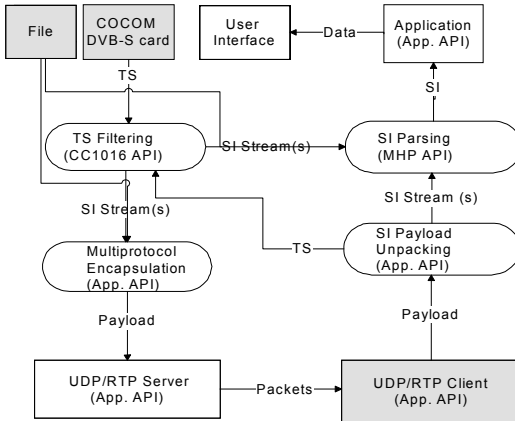
## 6.2 Process Flow

Figure (3) shows the data flow diagram of the whole system. Grey boxes illustrate SI retrieving elements, which represent source nodes for SI information. Setting initialization and configuration options start the whole process. This information defines which TS is obtained by defining its frequency and channel parameters. These parameters are static during the whole process. For the file version and the client version these information does not have to be set at all, as the server or the content of the file specifies it. The same applies for navigation information, which is also static during the whole process. Navigation information includes mainly filter options.

Filter options are *pure PID filtering*, *PID filtering with a positive filter* and *PID filtering with a negative and positive filter*. The first method filters only objects, which match a specific PID. It also allows getting raw stream of TS packets with this certain PID, if no table identifier is specified. The other two methods allow section filtering by specifying filter patterns. Positive filtering means, that a certain filter gets only triggered if the defined values match the specified bytes in the section. The third filter does completely the opposite. It gets triggered if the defined pattern does not match. This can be used to retrieve a section when its header changes (e.g. *version\_number*).

The filtered SI stream can be treated in several ways: A UDP/RTP server can forward encapsulated SI by using mutliprotocol encapsulation or it can be passed directly to the MHP SI Access API. Encapsulating raw stream of TS packets is the simplest method: One UDP/RTP packet contains one or more TS packets. In the current application one TS packet can be sent in one UDP packet. If section filtering is applied, one specific section has to be encapsulated in one UDP/RTP packet. The maximum payload length is limited to 4048 bytes, as the EIT section has this maximum length. The client side retrieves the encapsulated sections or TS packets and has to unpack the payload of RTP and UDP packets. If the payload contains whole sections they can be passed easily to the MHP SI Access API. Otherwise, if the payload contains TS packets, they have to be parsed to obtain whole sections, which can be passed to the MHP SI Access API.

The MHP SI Access API represents a lightweight database, which stores the whole SI. Applications have the possibility to request information by several criteria (e.g. *table\_id*, *transport\_stream\_id*, *PID*, *description tags*).



**Fig. 3.** Data Flow Diagram of the whole System

## 7. Results

Various test constellations have been applied. The first was based on a generation of SI by a file and forwarding them to LAN/WAN clients. The second test consisted of a forwarding of a whole feed of SI from a dedicated satellite card. Each test was carried out under consideration of UDP and the different packetization schemes of RTP. Preliminary assumptions were that no CAT access, no transmission of private sections and encapsulated IP datagrams were included. The criterion to analyze protocol performance was to measure the number of disturbances at the client side. This was done by setting up the client in a LAN and WAN environment under usual network load. Bandwidth measurements were performed under typical internet/intranet network load by using a network sniffer tool. The repetition rate depended on the kind of experiment: The generation of a service information stream out of a file assumed minimum rates, whereas the forwarding of service information from the COCOM card relied on the transmission rate broadcasted by the satellite. As estimated, UDP encapsulation showed the best performance values, through its less header overhead. But RTP offered a more robust behavior concerning reliability and synchronization possibilities with payload-based packetization and joint packetization. The whole service information could be retrieved in a satisfying amount of time, even through some packet losses and re-readings of information tables. The main problem that occurred was in network environments with high packet losses, where lost packets lead to a high amount of table re-readings. But as the DVB environment is based on a not reliable transmission media packet losses are always expected. Packet losses lead to longer acquisition times of the tables and therefore the frequency in sending service information is assumed to be high enough. The encapsulation of one TS packet per

UDP/RTP packet caused a big header overhead and is avoidable. As suggested in [24] more TS packets could be encapsulated in one UDP/RTP packet. Very surprisingly Java was fast enough to generate service information in appropriate time and also the forwarding time delay caused by the server was negligible. The reason for this is the native interface to C++ source code in an implementation.

## 8. Conclusions and Future Work

This paper considered the bandwidth utilization of a DVB service information stream in a LAN and WAN environment. The utilization highly depends on the number of programs and information that is sent to the end-user. It showed that not only the video/audio stream needs its bandwidth - also the service information transmissions demand its part. Future work will focus on the demand of multiprotocol encapsulated IP over a broadcast network and its demand in bandwidth, signal acquisition time and reliability of transmission of different protocol types. Especially multiprotocol encapsulation issues this matter. Reliability is essential because a broadcast media does not allow "conventional" acknowledgement schemes. Therefore a reliable transmission has to be established on the basis of other mechanisms. Their exploration and practical studies is another major subject<sup>3</sup>.

## References

1. Comer, E.D.: Internetworking with TCP/IP; Vol. I: Principles, Protocols, and Architecture. Prentice-Hall Inc. 1995.
2. Basso A.; Cash G.L.; Civanlar M.R.: Transmission of MPEG-2 Streams over Non-Guaranteed Quality of Service Networks. Proceedings of Picture Coding Symposium: Berlin Germany 1997.
3. Krupczak, B.; Calvert, K.; Ammar, M.: Protocol Portability through Module Encapsulation. Network Protocols, IEEE Proceedings: pp. 56-63, 1996.
4. Fox, B.: Digital TV comes down to earth. IEEE Spectrum: October, 1998.
5. TAM232r16: DVB: Multimedia Home Platform; Revision 16. European Broadcasting Union: February 2000".
6. ETS 300 802: DVB: Network-Independent Protocols for DVB Interactive Systems. European Telecommunications Standards Institute: November 1997.
7. Tektronix: A Guide to MPEG Fundamentals and Protocol Analysis. Tektronix Inc.: 1997. <http://www.tektronix.com>.
8. ETS 300 468: DVB: Specification for Service Information (SI) in DVB systems. European Telecommunications Standards Institute: January 1997.

---

<sup>3</sup> *Acknowledgements:* The FutureTV project is funded by the National Technology Agency of Finland together with major Finish television, telecommunications and digital media companies, especially YLE. We would like to thank especially Teemu Lukkarinen for all his help and the whole FurtureTV tream: Stephane Palomba, Florina Tico, Jukka Rakkola, Olli Savia, Olli Stroem and Chengyuan Peng.

9. ISO/IEC 13818-1: Generic Coding of Moving Pictures and Associated Audio: Systems. International Organization for Standardization: November 1994. Recommendation H.222.0.
10. ETS 300 800: DVB: Interaction Channel for Cable TV Distribution Systems. European Telecommunications Standards Institute: January 1997.
11. Veeraraghavan, Malathi; Karol, Mark: Internetworking Connectionless and Connection-Oriented Networks. IEEE Communications Magazine: Vol. 37, Issue 12; pp. 130-138; December 1999.
12. EN 301 192: DVB: Specification for Data Broadcasting. European Telecommunications Standards Institute: December 1997.
13. ETS 300 801: DVB: Interaction Channel through Public Switched Telecommunications Network (PSTN)/Integrated Services Digital Networks (ISDN). European Telecommunications Standards Institute: August 1997.
14. ETS 300 468: DVB: Specification for Service Information (SI) in DVB systems. European Telecommunications Standards Institute: January 1997.
15. EN 301 193: DVB: Interaction Channel through DECT. European Telecommunications Standards Institute.
16. EN 301 195: DVB: DVB Interaction Channel through GSM. European Telecommunications Standards Institute.
17. EN 301 199: DVB: DVB Interaction Channel for LMS Distribution Systems. European Telecommunications Standards Institute.
18. ES201218: DVB: DVB Interaction Channel for LMS Distribution Systems. European Telecommunications Standards Institute.
19. TR 101 201: DVB: DVB Interaction Channel for SMATV Systems. European Telecommunications Standards Institute.
20. Civanlar, M. Reha; Glenn, L.; Cash; Barry, G.: RTP Payload Format for Bundled MPEG. draft-civanlar-bmpeg-01, Internet draft. February, 1997.
21. Hoffman, D.; Fernando, G.; Goyal, V.; Civanlar, M.R.: RTP Payload Format for MPEG1/MPEG2 Video. draft-ietf-avt-mpeg-new-01, Internet draft. June, 1997.
22. RFC1305: Network Time Protocol (Version 3); Specification, Implementation, Analysis. David L. Mills: March, 1992.
23. Sarginson, P. A.: MPEG-2: A Tutorial Introduction to the Systems Layer. The Institution of Electrical Engineers: pp. 4/1-4/13; 1995.
24. Chengyuan Peng; Petri Vuorimaa.: A Digital Television Navigator. ACM Multimedia 2000: March 20th, 2000.