DVB-H link layer

Chapter 8
DVB-H Link Layer

Onno Eerenberg, Arie Koppelaar, and Peter H.N. de With

8.1 Introduction

In the fall of 2004, the European Telecommunications Standards Institute (ETSI) approved the Digital Video Broadcast Handheld (DVB-H) standard [19], which is specifically tailored to battery-powered mobile reception and developed by the International Digital Video Broadcasting (DVB) Project [8]. The DVB-H standard, formerly known as DVB-X [20], is an extension to the DVB-T standard [17] with extra features added to the physical and link layer.

With respect to the DVB-T physical layer, the DVB-H physical layer is extended with Transmission Parameter Signalling (TPS) to, e.g., fasten service discovery, a 4K mode to trade off Doppler sensitivity versus echo sensitivity and an in-dept symbol interleaver to increase the robustness to, e.g., impulsive noise. The DVB-H link layer uses a Time-Division-Multiplex (TDM) broadcast technique, called time-slicing, to transmit a service, enabling power-efficient service reception. On top of this, the DVB-H link layer is foreseen with a second Forward Error Correction (FEC) layer, called MPE-FEC, which protects the received service against various reception impairments, e.g., Carrier-to-Noise Ratio (CNR) ratio, Doppler or impulsive noise influences. The DVB-H additional protection by the MPE-FEC provides a CNR improvement of 4–6 dB advantage to DVB-H transmission with respect to DVB-T and reduces the influence of the speed factor while receiving the signal [7]. According to the DVB-H standard, time-slicing is a mandatory feature, whereas the second FEC layer (MPE-FEC) is an optional feature.

O. Eerenberg (✉) and A. Koppelaar
NXP Semiconductors, Research High Tech Campus 32, 5656 AE Eindhoven, The Netherlands
e-mail: onno.eerenberg@nxp.com, arie.koppelaar@nxp.com

P.H.N. de With
University of Technology Eindhoven, Fac. EE, Postbus 513, 5600 MB Eindhoven, The Netherlands
e-mail: P.H.N.de.With@tue.nl

F.-L. Luo (ed.), *Mobile Multimedia Broadcasting Standards: Technology and Practice*,
DOI: 10.1007/978-0-387-78263-8_8,

Unlike the traditional DVB broadcast members DVB-C, DVB-S, or DVB-T, DVB-H is a datagram-based broadcast transmission standard because of its endurance to buffering, delays, and easy network integration. This allows transmission of data that pertains, e.g., to multimedia services or file-downloading services. The usage of the Internet Protocol (IP) allows the coding to be decoupled from the transport, opening the door to a number of features benefiting handheld mobile terminals including a variety of encoding techniques, which only require low power from a decoder. Therefore, IP is the Open System Interconnection (OSI) [24] Layer-3 protocol used in mobile handheld convergence terminals, creating an IP datagram stream which consists of IPv4 or IPv6 datagrams, each sharing the same IP source and destination address.

Whereas the traditional DVB broadcast members use the Packetized Elementary Stream (PES) container format [25] to encapsulate audiovisual access units, DVB-H uses Multi-Protocol Encapsulation (MPE) sections [18] to carry OSI Layer-3 datagrams or so-called Multi-Protocol Encapsulation Forward Error Correction (MPE-FEC) sections [18] to carry Reed-Solomon (RS) parities. Figure 8.1 visualizes the relation between the documents that describe the DVB-H standard [34]. DVB-H is based on the DVB-T standard. It is for this reason that, besides the new DVB-H system specification standard [19], the system is described by a number of existing documents, which have been amended with the appropriate DVB-H features. The standards depicted in Fig. 8.1 lack the description of the OSI Layers 3-7 protocols, because the definition of the layers above the IP layer is outside the scope of the DVB-H specification. To overcome this shortage, the DVB-H ad hoc group Convergence of Broadcast and Mobile Services (CBMS) developed the Internet Protocol Data Broadcast (IPDC) specification [45]. The purpose of the IPDC specification is to provide both the higher layer protocols for DVB-H that enable the construction of an end-to-end system and the integration

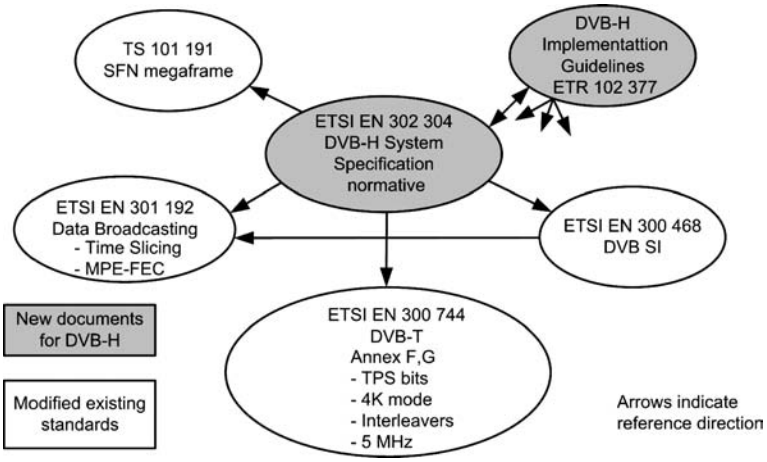

Fig. 8.1 The DVB-H family of standards

of a cellular communication system [12]. The IPDC over DVB-H standard complements the DVB-H standard, by defining OSI Layers 3-7 and influences some of the OSI Layer-2 (link layer) Program Specific Information (PSI) and Service Information (SI).

This chapter describes the aspects of an efficient and robust link layer. This link layer is an interface between the OSI Layer-1 (radio layer) operating in the physical domain and the OSI Layer-3 (network layer). The key words for this interface are *efficiency* and *robustness*. The efficiency aspects of the DVB-H link layer are multifold and are split into two categories. Category one is characterized in the sense that reliable and unreliable received data is both employed to maximize data recovery and reconstruction using erasure FEC decoding. Erasure information is scarcely assigned, leading to a higher flexibility in the usage of error correction. Category two is characterized with respect to the link layer implementation, optimizing on memory footprint, logic area, and cycle consumption. The robustness aspects of a DVB-H link layer are (1) that the link layer subsystem shall not collapse when incorrect data is processed and moreover (2) correctly received data shall always be transferred to the network layer, regardless of the outcome of the FEC decoding.

The sections of this chapter describe the various aspects that are related to the DVB-H link layer. Section 8.2 addresses the positioning of the link layer, a description of the signals processed, starting from the radio signals up to the IP layer, and the DVB-H link-layer features. In Sect. 8.3, the link-layer aspects related to the OSI layers are discussed. Section 8.4 presents the building blocks of an efficient and robust DVB-H link layer. Section 8.5 elaborates on the possible IP de-encapsulation methods required for an efficient link layer. Section 8.6 addresses the verification and validation of an efficient and robust DVB-H link layer. Finally, conclusions are presented in Sect. 8.7.

8.2 Features of the DVB-H Link Layer

This section is divided in to two parts. First, we depict the position of the link layer in a DVB-H terminal and briefly elaborate on the involved information signals and their definitions. Second, we introduce the DVB-H link-layer features, time-slicing aspects, and the MPE-FEC. Finally, the datagram and RS-parity data encapsulation are discussed in detail.

8.2.1 DVB-H Link Layer and Its Information Signals

Figure 8.2 indicates the position of the link layer in a basic DVB-H terminal. At the left-hand side of Fig. 8.2, the antenna signal enters the (silicon) tuner. The channel decoder and demodulator operate at the tuner output signals I and Q. The resulting MPEG-2 Transport Stream (TS) is processed by the link layer. The main

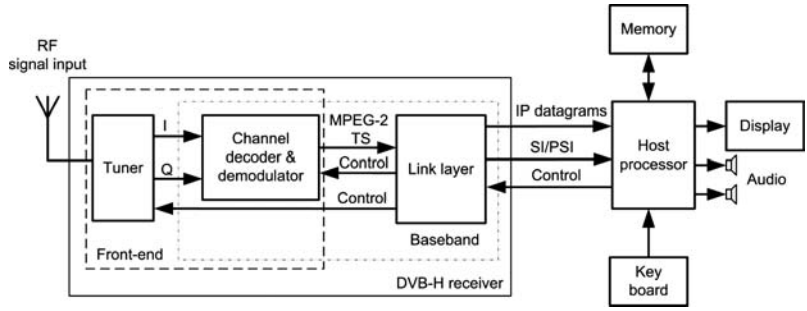

Fig. 8.2 Basic DVB-H terminal setup

responsibilities of the link layer are filtering of IP datagrams from a selected Elementary Stream, filtering of sections from SI and PSI, maintaining the synchronization with the time-sliced service, and front-end control of the receiver.[1] Because the IP datagram information, RS-parity data, and SI/PSI information are all transmitted using sections, a DVB-H link layer only needs to be able to handle section-based information, as this is the only MPEG-2 container format used.

An IP-based DVB-H service is associated to a so-called IP platform. In DVB-H, the SI/PSI information lists the available IP platforms and information to trace them. The SI/PSI information is processed by the middleware, resulting in a database that links an IP address to a particular Elementary Stream and the Transport Stream(s) that carry this Elementary Stream. A service IP address is obtained via the Electronic Service Guide (ESG), which is broadcasted using IP. An IP platform may contain more than one ESG. A special IP/port-number combination is used in every IP platform to transport a service that announces all ESGs to be found in that IP platform. This is the bootstrap ESG Service. The ESG consists of two essential types of information: user attraction and acquisition information. The majority of the ESG information is expressed as eXtended Markup Language (XML) fragments, but a part of the acquisition information are Session Description Protocol (SDP) files that the terminal needs to locate service streams and configure service consumption applications appropriately [13].

The link layer requires configuration, to extract an IP-based service from the received TS. Basically, this means setting the Packet IDentifier (PID) filter, setting up the MPE-FEC decoder (if used), indicating the maximum burst duration and initializing the possible IP filters to source and/or destination addresses. The configuration is done by the middleware.[2] For this, the middleware is invoked by the application requesting for a service with a particular IP address. This IP address is obtained by the application from the ESG. The middleware can be embedded in the DVB-H baseband but may also run on a host processor. Depending on the

[1] Front-end control by the link layer avoids time-slicing knowledge on the host, thereby simplifying the overall system design.

[2] We use the name middleware but for DVB-H this function is also known as Transport Stream Controller (TSC).

middleware system partitioning, the output of the DVB-H link layer contains either SI/PSI sections and IP datagrams as depicted in Fig. 8.2, or IP datagrams and the output of the middleware, e.g., a list of services for a particular IP platform.

8.2.1.1 Relation Between PSI/SI and a DVB Network

Figure 8.3 indicates the relation between DVB networks, Transport Streams, DVB services, and components after [14]. A DVB network is uniquely identified by a *network_id*. A DVB network consists of one or more Transport Streams, each carrying a multiplex and being transmitted by one or more DVB Radio Frequency (RF) signals. Information about a DVB network is available within the Network Information Table (NIT) *sub_table* (identified by *network_id*). The NIT lists all multiplexes and DVB RF-signals available within the DVB network. The NIT is carried within each DVB network. A single multiplex is a set of DVB services multiplexed together and transported by a TS. A multiplex and the corresponding TS are identified by *transport_stream_id* and *original_network_id*. The *Transport_stream_id* parameter is unique within the *original_network_id*. The *Original_network_id* parameter is the *network_id* of the DVB network generating the multiplex. Information about a particular multiplex is available within the Program Association Table (PAT) carried within the multiplex [25]. Each multiplex contains exactly one PAT, listing all DVB services available within the multiplex. A DVB service is a sequence of programme

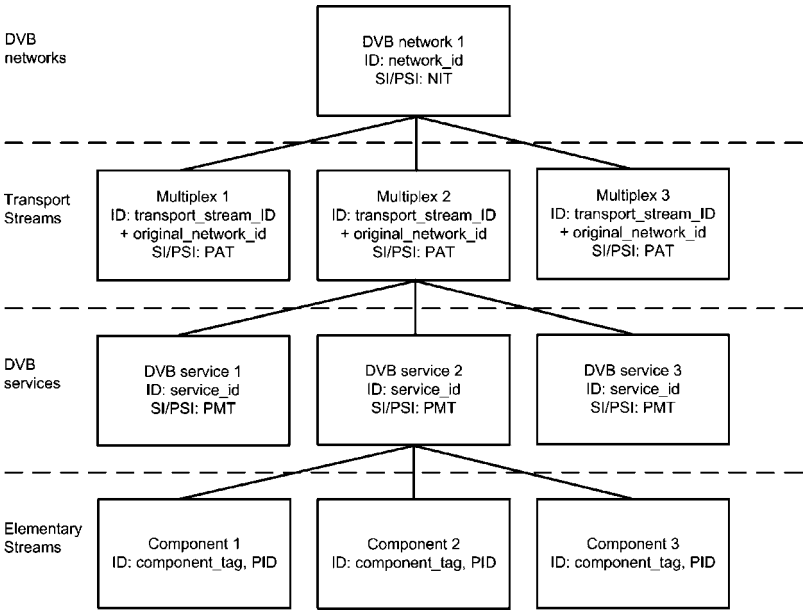

Fig. 8.3 Relation between DVB networks, Transport Streams, DVB services, and components

events, each of which groups together a set of components. Components can either have a seperate Elementary Stream, or share an Elementary Stream.[3] An Elementary Stream is a collection of Transport Stream packets sharing a common *PID* [25]. A DVB service is globally and uniquely identified by the triplet of *service_id*, *transport_stream_id*, and *original_network_id*. The *service_id* is unique at least within a multiplex. All the components of a DVB service are carried within a single multiplex. Information about a DVB service is available within the Program Map Table (PMT) *sub_table* (identified by *service_id*) [25], carried within the same multiplex. A component is identified by the *component_tag*, which is unique within a DVB service. The component is carried within an Elementary Stream, identified by the *PID*. The *PID* is unique within a TS. Mapping between a *component_tag* and the *PID* is signalled in the PMT. It is possible to have one Elementary Stream carrying a component of more than one DVB service.

8.2.1.2 Relation Between IP Platform, IP flow, and IP Stream

An IP platform is a set of IP flows managed by a service provider. The IP platform represents a harmonized IP-address space, that has no address collisions, and represents a set of IP/MAC streams and/or receiver devices. An IP platform may be available on multiple TSs, within one or multiple DVB networks. In such a case, each of the TSs may carry any subset of the IP flows of the IP platform. An IP platform is identified by the *platform_id*, which is carried by the IP/MAC Notification Table (INT) [18]. This table provides a flexible mechanism for carrying information about availability and location of IP/MAC streams within DVB networks. *Platform_id* values are divided into two ranges. One range consists of *platform_id* values that are globally unique. If such a *platform_id* value is signalled in two different DVB networks, it refers to the same IP platform. The second range consists of *platform_id* values, which are unique only within the scope of a DVB network. Data from an IP platform identified by such *platform_id* is not available in any other DVB network. Such an IP platform is globally and uniquely identified by the combination of both *platform_id* and *network_id* only. Each IP platform contains one or more IP flows, each mapped into one or more IP streams. An IP flow is identified within an IP platform by its source and destination addresses, and the involved port number. IP flows are IP platform specific, and two different IP flows on two different IP platforms may use the same source/destination addresses. Each IP flow may be mapped into one or more IP streams. An IP stream is a data stream delivering exactly one instance of an MPE-encoded IP flow. Figure 8.4 depicts the mapping of a single IP datagram on a single MPE section, which is mapped on one or more Transport Stream packets. An IP stream is identified by *transport_stream_id*, *original_network_id*, *service_id, component_tag*, and IP source/destination addresses (all together), which are further described hereafter.

[3] Note that in DVB-H, the definition of an Elementary Stream differs from the definition as used in MPEG-2, where an Elementary Stream refers to the basic information stream prior to TS packetization.

- *Transport_stream_id* and *original_network_id* together identify a single Transport Stream.
- *Service_id* and *component_tag* together identify a single Elementary Stream within a Transport Stream. The Elementary Stream consists of TS packets with the same *PID*.
- IP source/destination addresses identify a single IP stream within an Elementary Stream. This is required to differentiate between multiple IP streams carried within the Elementary Stream.

Figures 8.4 and 8.5 indicate the hierarchical relation and structure of IP platforms, IP flows and IP streams after [14]. Figure 8.4 depicts that a stream of MPE sections sharing the same MAC address, also indicated as MPE-section stream, is delivered within an Elementary Stream. At the top, it is shown that an IP flow consists of a number of IP datagrams. A single IP datagram is encapsulated in a single MPE section, resulting in an MPE-encoded IP flow, which is transported in an Elementary Stream. Figure 8.5 visualizes the hierarchical structure of an IP platform, IP flows, and IP streams. An IP platform consists of one or more IP flows, which can

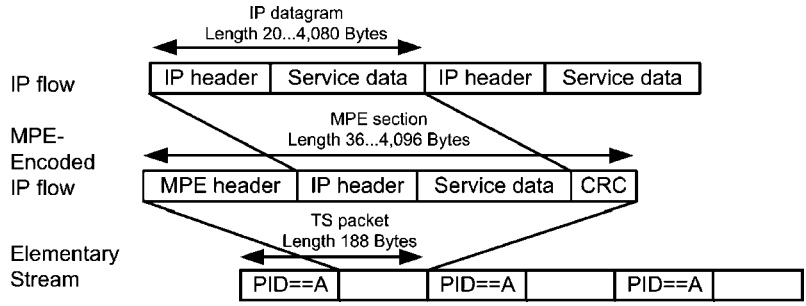

Fig. 8.4 Relation between IP stream, MPE-section stream, and Elementary Stream

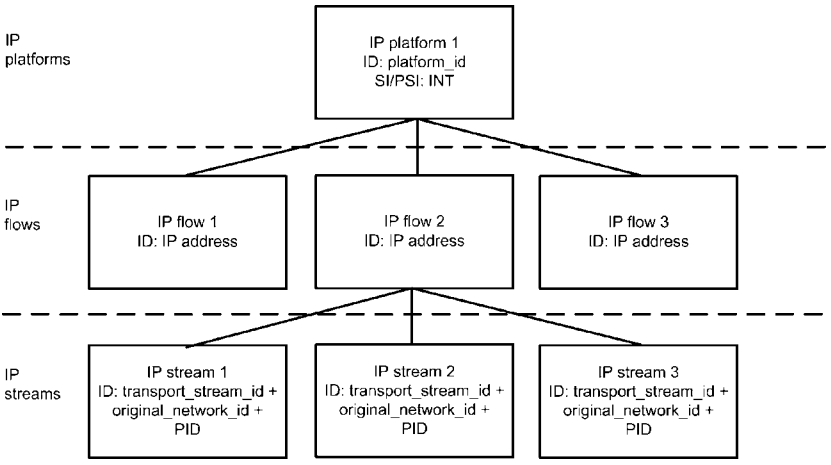

Fig. 8.5 Relation between IP platform, IP flow, and IP stream

be transmitted in one or more IP streams. As a result, the same data is available on one or more Transport Streams in one or more multiplexes and thus on different broadcast frequencies. Two different Elementary Streams, carrying IP streams which contain the same IP flow, may be located in different multiplexes and thus different broadcast frequencies. In such a case, an IPDC DVB-H receiver may accomplish a handover [38, 46] between these multiplexes, while receiving the IP flow. To optimize the bandwidth usage, the INT does not always announce the source address of an IP flow. In this case, IP stream differentiation is based on the destination address only. An IP stream is a single data stream delivering an IP flow. An IP stream is identified by associated parameters indicating its location, *network_id*, *original_network_id*, *transport_stream_id*, *service_id*, and *component_tag*. An IP flow represents the data content of a stream, while an IP stream represents an instantiation of such an IP flow. An IP flow belongs to exactly one IP platform. An IP stream may be announced by multiple INT *sub_tables*, eventually making it part of multiple IP platforms. This enables the transmission of a single service over multiple areas via multiple frequencies.

8.2.2 Time-Slicing

Although the DVB-T broadcast standard allows mobile reception, it is not optimized to support reception with mobile battery-powered terminals. Therefore, the DVB-H broadcast standard is equipped with a feature called time-slicing, in which services[4] are broadcasted in periodic bursts, see Fig. 8.6 after [18]. This feature is added to the DVB-H link layer to allow service distribution via an existing DVB-T network. Time-slicing enables the receiver front-end to be active for only a fraction of the time, receiving bursts of a requested service [35]. Major power consumers in, e.g., a Personal Digital Assistant (PDA) based terminal are the Liquid Crystal Display

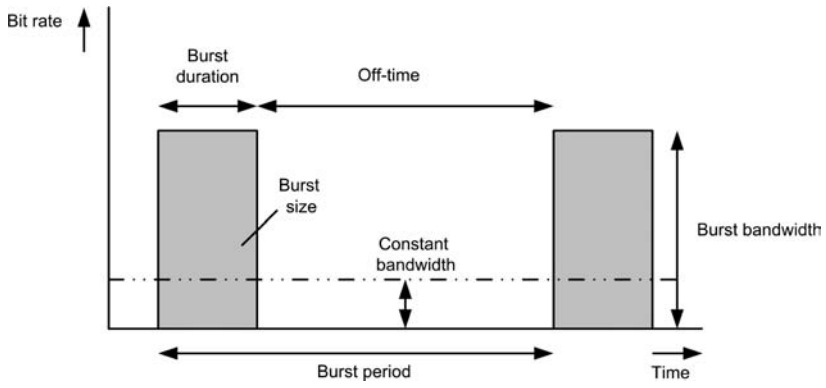

Fig. 8.6 A DVB-H service burst

[4] Note that the transmission of the Transport Stream is not interrupted.

(LCD) backlight and the front-end. In the early phase of DVB-H standardization, the continuous power consumption of such a front-end was estimated at 1 W [34]. As of today, the power consumption has been pushed back to around 150 mW for the silicon tuner and 200 mW for the baseband for continuous reception.[5] With current technology, the consumed DVB-H receiver power in off-time can be brought down to 3 mW. Depending on the values for the burst duration and burst period, power savings up to 95% can be achieved [42]. Although services in DVB-H are broadcasted in time-sliced mode, the SI/PSI information are non-time-sliced broadcasted. A time-sliced service can be sequential or parallel of nature and broadcasted in combination with a continuous broadcasted DVB service, see Fig. 8.7b. For example, in Fig. 8.7a, DVB-H Service 2 and DVB-H Service 3 represent parallel service bursts, whereas DVB-H Service 1 and DVB-H Service 2 form consecutive service bursts. Besides parallel services at Elementary Stream level, as indicated in Fig. 8.7a, parallel services can also share an Elementary Stream and differ in multicast address. Although the parallel services as indicated in Fig. 8.7a share the same burst start and end-time, the standard does not impose any restrictions on this behavior. Parallel services are beneficial for several reasons:

- Fast zapping time
- Simultaneous reception of the main services in combination with low-speed services (ESG, Alarms, ...)
- Local insertion of services
- Better optimization of bandwidth, e.g.
  - Maintain burst length (for impulsive noise / Doppler performance)
  - One service does not utilize the full bandwidth. Inserting two services in parallel results in a better bandwidth utilization

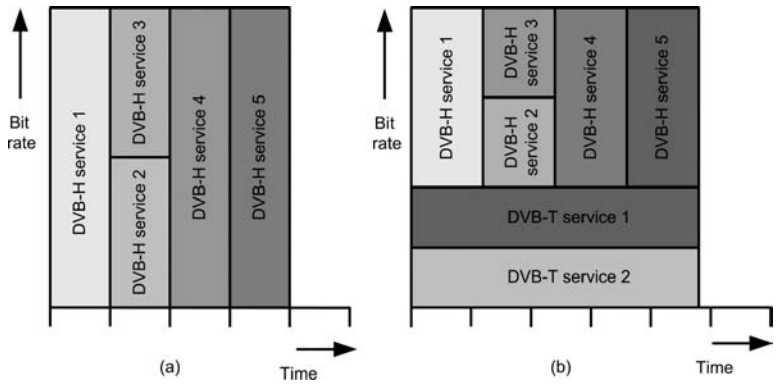

Fig. 8.7 DVB-H service broadcast methods. **(a)** Multiplex snapshot containing only DVB-H services, whereby service 2 and service 3 are broadcasted in parallel. **(b)** Multiplex snapshot containing a mixture of DVB-T and DVB-H services

[5] Note that the LCD backlight was and still is a major power consumer, but with state-of-the-art signal processing, the LCD backlight can be dynamically controlled, thereby reducing the backlight power consumption roughly with a factor 30%.

If parallel services are sharing an Elementary Stream, this may lead to the following complications. For example, one service may require all bandwidth in the Internet Protocol Encapsulator (IPE) leaving no bandwidth for the remaining services. Furthermore, it is difficult to re-encapsulate services in an existing Elementary Stream due to, e.g., sharing the MPE-FEC setting.

Similarly, if parallel services do not share the same Elementary Stream, some beneficial properties occur. First, one service will not influence the available bandwidth of other services. Second, IP data can be encapsulated locally due to the separate MPE-FEC setting. Third, some service types such as, e.g., ESG and alarm services can occur with higher periodicity, due to the disentanglement of the main service to which they were attached. Combining a continuous broadcasted DVB service and time-sliced services, see Fig. 8.7b, decreases the power saving due to a lower available data rate for time-sliced services [36]. When a DVB-H multiplex is combined with a regular statistical multiplexed information signal, the DVB-H burst character is not jeopardized [7]. A time-sliced broadcast technique allows the terminal to switch off the front-end for the off-time period, in which no service data is transmitted. Besides power reduction, time-slicing has also the advantage that the front-end can be reused during the off-times for peeking into neighboring cells to perform service discovery and enabling seamless horizontal handover for a specific service [37].[6]

Time-slicing information is carried by the *real_time_parameters*, which are available in each of MPE and MPE-FEC sections. The off-period starts after the end of a service burst. When due to a transmission error, the *frame_boundary* (see Sect. 8.2.5) is missed, the *max_burst_duration* of the *time_slice_fec_identifier_descriptor* (see Sect. 8.3.2) allows determination of the service burst end. This enables the activation of the power-down mode, provided that at least one MPE or MPE-FEC section has been correctly received. If the time-slicing information for a service burst is missed, the terminal's front-end must be active to detect the Elementary Stream containing the next service burst.

8.2.3 MPE-FEC Feature

To enhance the DVB-H receiver robustness under various reception conditions, an error-protection scheme is added to the DVB-H link layer. This scheme is called MPE-FEC, employing powerful channel coding on top of the DVB-T channel coding, offering extensive time interleaving and allowing service distribution via an existing DVB-T network. The FEC is based on the Reed-Solomon (RS) code [255,191,65] RS. The parity data is stored together with the OSI Layer-3 IP datagrams in a so-called MPE-FEC frame. DVB-H supports four different MPE-FEC frame sizes. The MPE-FEC frames can be constructed using either 256, 512, 768 or 1,024 rows. Figure 8.8 indicates an MPE-FEC frame of k rows, which consists of

[6] Note that there exists also vertical handover, which means service handover between two different standards, e.g., DVB-H and UMTS and is outside the scope of this chapter.

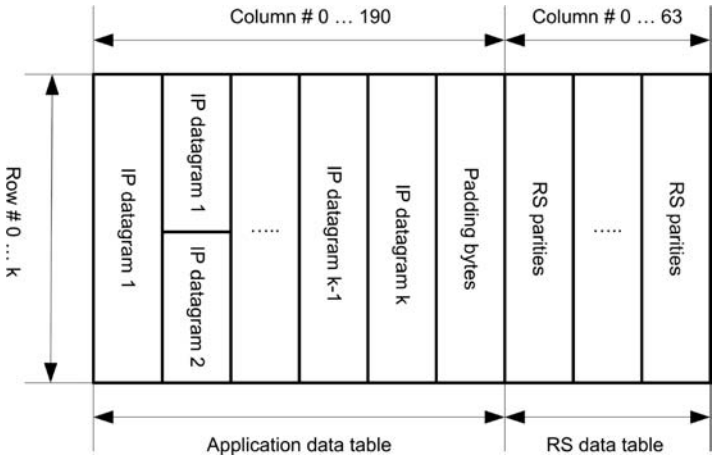

Fig. 8.8 MPE-FEC frame of k rows, one padding column, and 64 parity columns

an application data table and an RS-data table. Although the MPE-FEC frame has a two-dimensional structure, it can be considered of being two one-dimensional arrays that contain the corresponding data. Due to this one-dimensional structure, data of a single IP datagram can be split over two or more application data table columns.[7] Such a situation is depicted in Fig. 8.8, where the length of IP datagram 1 exceeds the number of MPE-FEC frame rows. The number of rows that construct an MPE-FEC frame is signalled in the *time_slice_fec_identifier_descriptor*, see Sect. 8.3.2. For the situation that the transmitted number of IP datagram bytes is less than the number of bytes corresponding to the product of application data table columns and MPE-FEC rows, zero-valued padding bytes fill up the remaining positions of the application data table. The number of fully padded columns is signalled in the MPE-FEC section header, see Sect. 8.2.4 and may vary per MPE-FEC burst. The [255,191,65] RS mother code allows to correct up to e erasures, if the erroneous positions are known, or t unknown errors according to the inequality

$$2t + e < 65. \tag{8.1}$$

Padding columns and subsequently not transmitting these padded columns can also be used to transmit less IP data making the code *stronger*, a process also known as shortening. Alternatively, transmission of less parity columns can also be applied making the code *weaker* than the mother code, a process also known as puncturing. Although MPE-FEC is part of the DVB-H standard, it is not obligatory. As the transmission of the application data table always precedes the transmission of the RS-data table, an MPE-FEC ignorant DVB-H receiver can skip the reception of RS data, reducing the power consumption by maximal 25%. For MPE-FEC capable

[7] Note that the row-wise calculated parity data is column-wise transmitted, using MPE-FEC sections.

DVB-H receivers, skipping the RS data of a service burst is beneficial when all IP datagrams of a particular service burst are received correctly. In general, the usage of MPE-FEC with code rate $\frac{3}{4}$ provides a spectacular improvement of the DVB-H coverage. A laboratory test showed a small variation of the CNR gain (i.e. about 3 dB) when the MPE-FEC coding rate varies from $\frac{7}{8}$ to a robust $\frac{1}{2}$ value. But for MPE-FEC code rate $\frac{7}{8}$, field trials were worse than in a laboratory environment, suggesting that the CNR gain is proportional to the MPE-FEC overhead/coding rate [7]. In the DVB-H standard, a quality measurement indicator MPE-FEC Frame Error Rate (MFER) is introduced, which is defined as the ratio between the number of defect received MPE-FEC frames divided by the total received MPE-FEC frames. Often, this indicator is used to determine the reception conditions that result in an MFER of 5%. Using an MFER of 5%, a Doppler performance of 120 Hz is achieved, corresponding to a speed of 160 km/h (100 mph) in the upper part of band V (800 MHz) or 640 km/h (400 mph) in the lower part of band III (200 MHz) [7].

8.2.4 DVB-H Data Encapsulation

DVB-H a is datagram-based broadcast standard. Unlike the traditional DVB broadcast members, which use the MPEG-2 system standard [25] for encapsulation of audiovisual access units into PES units, which are depicted at TS packets, DVB-H uses MPE for encapsulation of an OSI Layer-3 (Network) datagram (e.g., IP datagram) [18]. If MPE-FEC is used, RS data is encapsulated in MPE-FEC sections [18]. The mapping of the section into MPEG-2 Transport Stream packets is defined in the MPEG-2 Systems standard [25].

8.2.4.1 MPE Section

DVB-H is a broadcast transmission system for datagrams, which may be based on IP or other network layer datagrams [18, 24]. DVB has specified four methods for data broadcasting: data piping, data streaming, Multi-Protocol Encapsulation (MPE), and data carousel, which can all be used for delivering IP data. Data piping and data streaming are used so rarely that they are ignored in this context. Data carousels support delivery of files and other data objects, but are not suited for streaming services. Furthermore, implementing time-slicing on data carousels may be difficult. MPE is well suited for the delivery of streaming services as well as files and other data objects and it supports delivery of other protocols, while implementation of time-slicing on MPE leads to a low-cost solution. MPE sections are compliant to the *DSMCC_section* format for private data [26]. MPE sections provide means to handle either IP datagrams or LLC/SNAP datagrams [22, 23].

Table 8.1 presents the syntax of an MPE section. The *table_id* of an MPE section corresponds to the value “0x3E” [26] after [18]. The value for the *section_syntax_indicator* field is the complement of the *private_indicator* field. If the

Table 8.1 MPE section syntax

Syntax	Number of bits
datagram_section() {	
table_id	8
section_syntax_indicator	1
private_indicator	1
reserved	2
section_length	12
MAC_address_6	8
MAX_address_5	8
reserved	2
payload_scrambling_control	2
address_scrambling_control	2
LLC_SNAP_flag	1
current_next_indicator	1
section_number	8
last_section_number	8
MAC_address_4	8
MAC_address_3	8
MAC_address_2	8
MAC_address_1	8
If (LLC_SNAP_FLAG == '1') {	
LLC_SNAP()	
} else {	
for (j=0; j<N1; j++) {	
IP_datagram_data_byte	8
}	
}	
If (section_number == last_section_number) {	
for (j=0; j<N2; j++) {	
stuffing_byte	8
}	
}	
If (section_syntax_indicator == '0') {	
checksum	32
} else {	
CRC_32	32
}	
}	

section_syntax_indicator field is "1" (*private_indicator* is "0") the section uses a Cyclic Redundancy Checksum (CRC) to detect a transmission error. If the *section_syntax_indicator* field is "0" (*private_indicator* "1") the section uses a checksum to detect a transmission error. The *section_length* field indicates the number of

bytes following this field including the CRC.[8] The maximum MPE section length is 4,096 bytes. With an MPE section overhead of 16 bytes, consisting of 12 bytes MPE-section header plus 4 bytes CRC, the maximum IP-datagram size becomes 4,080 bytes. Within the MPE-section header, a 6-byte field is allocated for the Medium Access Control (MAC) address. The length of the used MAC address is signalled in the *data_broadcast_descriptor*, see Sect. 8.3.2, which is inserted in the Service Description Table (SDT) or Event Information Table (EIT) [16]. The minimum MAC address length is 1 byte, leaving up to 5 bytes for other use. Four of these five MAC bytes, *MAC_address_1, …, MAC_address_4*, are used to deliver the *real_time_parameters*, see Sect. 8.2.5, resulting in a 2-byte MAC field. In case of multicast IP streams, the MAC address is actually redundant data, as the MAC address is a function of the multicast group IP address. For all IP streams, the IP-datagram header following immediately the MPE-section header includes source and destination IP addresses, which uniquely identify the IP stream. A receiver can either ignore the MAC address entirely, filtering IP addresses only, or use the remaining MAC address bytes to differentiate IP streams within the Elementary Stream. Depending on the value of the *MAC_IP_mapping_flag* field, which is carried by the *data_broadcast_descriptor*, IP-to-MAC mapping is applied. As a result, the low-order IP bytes are mapped to *MAC_address_5* and *MAC_address_6*, as described for IPv4 [3] and for IPv6 [2]. The reserved fields are set to "1." The *payload_scrambling_control* field defines the scrambling mode of the section payload. This includes the payload starting after the *MAC_address_1* but excludes the checksum or *CRC_32* field. The applied scrambling method for the payload is user private. The *address_scrambling_control* field defines the scrambling mode of the MAC address. This field enables a dynamic change of the MAC addresses, the applied scrambling method for the MAC address is user private. If the *LLC_SNAP_flag* field is set to "1," the payload carries an LLC/SNAP-encapsulated datagram, following the *MAC_address_1* field. The *LLC_SNAP_flag* indicates the type of datagram conveyed. If this flag is set to "0," the section shall contain an IP datagram without LLC/SNAP information. If this flag is set to "1," the section shall contain an IP datagram preceded by LLC/SNAP fields. The *current_next_indicator* field is set to "1." The *section_number* field is set to "0," due to the fact that one section carries only a single IP datagram. The *last_section_number* shall be set to "0," again because there is only one section carrying a single IP datagram. The *MAC_address_1, ..., MAC_address_4* fields have obtained a new purpose in DVB-H and carry the *real_time_parameters*, see Sect. 8.2.5. The *CRC_32* field contains the calculated CRC according to [25].

8.2.4.2 MPE-FEC Section

The RS data of each MPE-FEC frame shall be delivered in MPE-FEC sections [18], using the syntax as depicted in Table 8.2 after [18]. The MPE-FEC sections are

[8] Note that the section length is always three bytes longer than indicated by the *section_length* field, due to the preceding generic part.

Table 8.2 MPE-FEC section syntax

Syntax	Number of bits
MPE-FEC_section() {	
table_id	8
section_syntax_indicator	1
private_indicator	1
reserved	2
section_length	12
padding_columns	8
reserved_for_future_use	8
reserved	2
reserved_for_future_use	5
current_next_indicator	1
section_number	8
last_section_number	8
real_time_parameters()	
for (j=0; j<N1; j++) {	
rs_data_byte	8
}	
CRC_32	32
}	

compliant to the *DSMCC_section* type, which are user private [26]. Each MPE-FEC section shall carry exactly one column of the corresponding RS-data table. The number of MPE-FEC sections used to carry RS data of an MPE-FEC frame shall not exceed the number of columns of the RS-data table. However, for the situation that puncturing is applied, the number of MPE-FEC sections indicated by *last_section_number* delivered may be less, indicating that not all RS data is transmitted. In the latter case, the RS decoder shall consider bytes within undelivered columns as unreliable. The number of delivered MPE-FEC sections is notified via the *last_section_number* field. The position of the delivered RS data in the RS-data table is indicated by the *section_number* field and the *real_time_parameters* field *address*. Section 0 carries the first (leftmost) column of the RS-data table, MPE-FEC section 1 carries the second column, and so on. The columns not delivered, e.g., due to puncturing, shall be the rightmost columns of an RS-data table. The *table_id* field equals the value "0x78." An MPE-FEC section only supports a CRC to be used for error detection, resulting in the *section_syntax_indicator* to be set to "1," forcing the *private_indicator field* to be "0" [26]. The two-bit reserved fields of the MPE-FEC section are set to "11." The *section_length* field indicates the number of remaining bytes in the section immediately following this field up to the end of the section, including the *CRC_32*. The number of *rs_data_bytes* carried in the MPE-FEC section shall be equal to the number of rows in the corresponding MPE-FEC frame, as indicated by the *frame_size* field of the *time_slice_fec_identifier_descriptor*, see Sect. 8.3.2. The amount of fully padded-padding columns in the application data

table is denoted by the *padding_columns* field.[9] The *padding_columns* field value shall have a maximum value of 190 and may vary between successive frames.[10] When not used, the 8-bit *reserved_for_future_use* field is set to "0xFF." Similarly, when not used, the 5-bit *reserved_for_future_use* field is set to "0x1F." The *current_next_indicator* field is set to "1." The *section_number* field indicates the number of the section and corresponds to the RS-parity column in the RS-data table. The *section_number* field of the first section carrying RS data of an MPE-FEC frame is therefore set to "0x00." The *section_number* field shall be incremented with unity with each additional section carrying RS data of the concerned MPE-FEC frame. The *last_section_number* field indicates the number of the last section that is used to carry the RS data of the current MPE-FEC frame. The *rs_data_byte* field contains the RS-data bytes delivered. The *CRC_32* field contains the calculated CRC according to [25].

8.2.5 DVB-H Real Time Parameters

Table 8.3 indicates the *real_time_parameters* syntax, which are present in each MPE and MPE-FEC section after [18]. As mentioned in Sect. 8.2.4, the MPE-section syntax fields *MAC_address_1, . . . , MAC_address_4* are replaced by the *real_time_parameters*. The interpretation of the *real_time_parameters* values depends on the broadcast mode, whether time-slicing and/or MPE-FEC are active or nonactive.

8.2.5.1 Real-Time Parameters: *delta_t*

In time-sliced transmission mode and regardless of the presence of MPE-FEC, the *delta_t* field indicates the time to the next time-slice burst within the Elementary

Table 8.3 DVB-H *real_time_parameters* syntax

Syntax	Number of bits
real_time_parameters () {	
delta_t	12
table_boundary	1
frame_boundary	1
address	18
}	

[9] Note that potential padding bytes after the last IP datagram for filling up a column, are not signalled.

[10] An MPE section should not be empty. As a result, the application data table shall contain at least one datagram per service burst, resulting in a maximum of 190 padding columns.

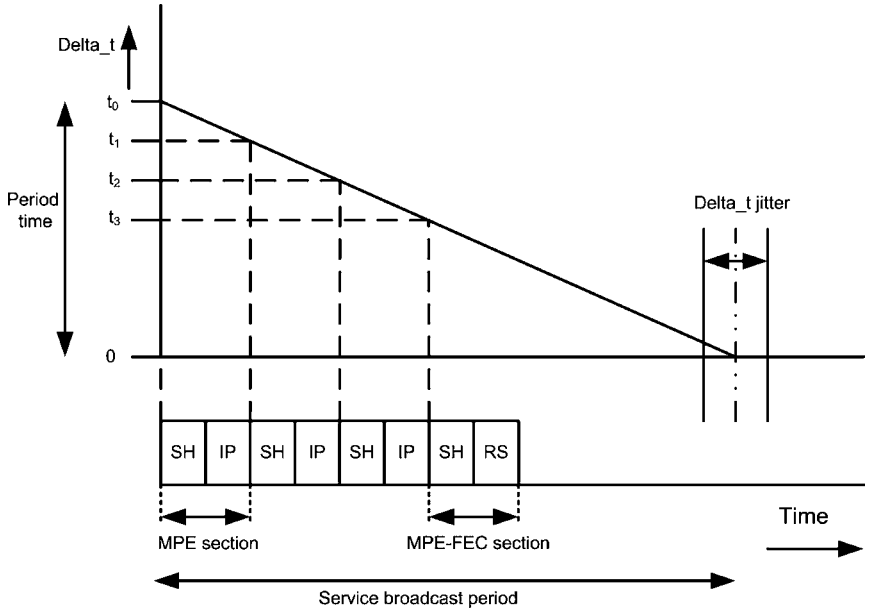

Fig. 8.9 *Delta_t* behavior in an MPE-encoded IP flow. The Section Header (SH) contains the *delta_t* value. The section payload is either an IP datagram (IP) or RS data (RS)

Stream. The time information is in all MPE sections and MPE-FEC sections[11] within a burst and the value may differ section by section, as indicated in Fig. 8.9. The resolution of the *delta_t* value is 10 ms. Value "0x00" is reserved to indicate that no further bursts will be transmitted within the Elementary Stream, i.e. indicating the end of a service. In such a case, all MPE sections and MPE-FEC sections within the burst shall have the same value in this field. The time indicated by *delta_t* shall exceed the end of the maximum burst duration, indicated by the *time_slice_fec_identifier_descriptor* field *max_burst_duration*, of the actual burst. This ensures that a decoder can always reliably distinguish two sequential bursts within an Elementary Stream. The basic objective of the *delta_t* method is to signal the time from the start of the currently received MPE (or MPE-FEC) section to the start of the next burst. To keep the *delta_t* insensitive to any constant delays within the transmission path, *delta_t* timing information is relative. The purpose of delivering *delta_t* in MPE and MPE-FEC section is to remove the need for synchronizing clocks between transmitter and receiver, as is the case for the DVB-T/C/S broadcast standards, which use the Program Clock Reference (PCR) [25]. The receiver has to provide sufficient accuracy for the duration of only one period, as the clock is restarted by each burst. As *delta_t* indicates the relative time rather than absolute one, the method is virtually unaffected by any constant delays within the transmission path. However, jitter on such delays does affect the accuracy of *delta_t*.

[11] Note that MPE-FEC may not be present because it is a nonobligatory feature.

This jitter is later referred to as *delta_t* jitter. If *delta_t* indicates the earliest possible time when the next burst may start, any *delta_t* jitter can be handled by decreasing the *delta_t* and thereby sacrificing the accuracy of the *delta_t*, thereby influencing the achieved power saving. Remultiplexing experiments indicate that the *delta_t* jitter remains in the vicinity of the allowed 10 ms [7].

For the situation that time-slicing is not used and MPE-FEC is applied, the *delta_t* field behaves like a cyclic MPE-FEC frame index within the Elementary Stream. The counter is incremented with unity for each subsequent MPE-FEC frame. If the maximum value "0xFFF" is reached, the counter value wraps back to zero. When large portions of data are lost, this parameter enables the identification to which MPE-FEC frame the actual received section belongs.

8.2.5.2 Real-Time Parameters: *table_boundary*

The *table_boundary* field is set to "1," when the current section is the last section of a table within the current MPE-FEC frame. If the section is an MPE section, this flag indicates the last section of application data table. For each MPE-FEC frame, exactly one MPE section with this flag set shall be transmitted. For each MPE-FEC frame for which any RS data is transmitted, exactly one MPE-FEC section with this flag set shall be transmitted. When MPE-FEC is not used on the Elementary Stream, this flag is reserved for future use, and set to "1."

8.2.5.3 Real-Time Parameters: *frame_boundary*

For the broadcast situation that time-slicing and MPE-FEC are active, the *frame_boundary* field when set to "1" denotes that the current section, which is either an MPE or MPE-FEC section, is the last section within the current burst. When the *frame_boundary* field is set in an MPE section, the burst does not contain MPE-FEC data.

8.2.5.4 Real-Time Parameters: *address*

For the situation that the *time_slice_fec_id* in the *time_slice_fec_identifier_descriptor* associated with the Elementary Stream is set to "0" and MPE-FEC is applied, the *address* field specifies the byte position in the corresponding MPE-FEC frame table that matches the first byte of the payload carried within the section. The first MPE section of a service burst has an *address* value corresponding to the value "0x00000." Consecutive sections carry *address* values in ascending order and can be calculated as follows. Let k be the MPE section index per burst, then the *address* value of section $k+1$ is calculated by adding the length of the IP datagram carried by section k with the *address* value of section k. The first MPE-FEC section after the MPE sections of a service burst starts again with the *address* field, reset to the value

"0x00000." Because an MPE-FEC section carries RS-parity data with a length that corresponds to the number of rows of a particular MPE-FEC frame, the *address* calculation for MPE-FEC section $k+1$ is reduced to adding the number of rows to the *address* value of MPE-FEC section k. If MPE-FEC is not used on the Elementary Stream, the *address* value is set to the fixed value "0x3FFFF."

8.3 OSI Layer Aspects Influencing the DVB-H Link Layer

This section presents the DVB-H broadcast stack and some aspects of the network and transport layer that influences a robust link-layer implementation. Furthermore, to properly de-encapsulate a received Elementary Stream, the link layer needs to be configured by the middleware, based on descriptors that are broadcasted via the SI. The second part of this section elaborates on the descriptors containing the configuration settings of the link layer.

8.3.1 DVB-H Broadcast Protocol Stack

DVB-H is an IP-datagram-based broadcast standard, which is visualized in Fig. 8.10, depicting the first four OSI layers of the DVB-H broadcast protocol stack [13]. From Fig. 8.10 it is clearly visible that the DVB-H link layer output has an IP and an SI/PSI section interface. The IP datagrams are offered to the IP stack on a host processor for further processing, whereas the DVB signalling information in the form of SI/PSI sections are requested by the middleware stack. This stack can be installed on either a host processor or embedded within the receiver baseband subsystem. Let us now highlight two aspects of the DVB-H broadcast stack that are relevant for a robust link-layer implementation. One of the aspects that characterize a robust link layer is the requirement that all correct IP datagrams, either correctly received or corrected by the link layer FEC, will be forwarded to the IP stack. A critical aspect of this requirement is the readout of IP datagrams from the MPE-FEC frame. The IP datagrams in DVB-H can have variable sizes up to 4,080 bytes, the maximum size that fits in an MPE section. It was mentioned in Sect. 8.2.3 that

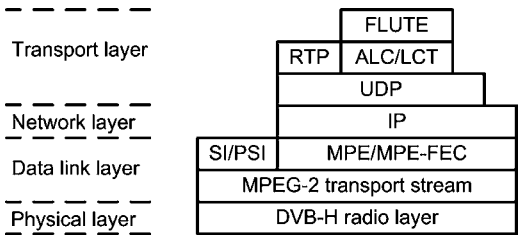

Fig. 8.10 DVB-H broadcast protocol stack

IP datagrams are stored in the application data table in a linear fashion. Due to a possible variance in IP-datagram length, it is necessary to determine the length field for each individual IP datagram in the MPE-FEC frame, which is located in the IP-datagram header. For the MPE-FEC frames that are correctly received or could be fully corrected by the link-layer FEC, IP datagram readout is possible due to the availability of a reliable IP-datagram length field. For the MPE-FEC frames that, even after applying the link layer FEC, still contain errors, the situation is more difficult and may result in the loss of all correct IP datagrams in that MPE-FEC frame. Hence, it is essential to have knowledge on the correctness of the IP-datagram length field. The protocol used by the network layer is based on either IPv4 or IPv6 [4, 39].[12] One of the points in which the IP network-layer protocols IPv4 and IPv6 differ from each other is the absence of an IP-header checksum in IPv6. The absence of an IP-header checksum results in an unreliable IP-datagram parsing in defect MPE-FEC frames, due to the fact that the length field may be corrupted. As a result, all correctly received IP datagrams may be lost, all depending on the location of the defect IP datagrams in the application data table. Figure 8.11 indicates the resulting loss of IP datagrams in the application data table, situated in column interval *l*, caused by a too large number of defect IP datagrams in column interval *i*, exceeding the FEC decoding capabilities. To overcome this protection shortage at the network layer (OSI Layer 3), the checksum of the transport layer (OSI Layer 4) can be used. The User Datagram Protocol (UDP) [40] uses a checksum that is calculated using a pseudo-header, information from the IP header, containing the IP-header source address, destination address, protocol field, and the UDP packet length. Because the IP version used for a service is fixed, the IP-datagram length

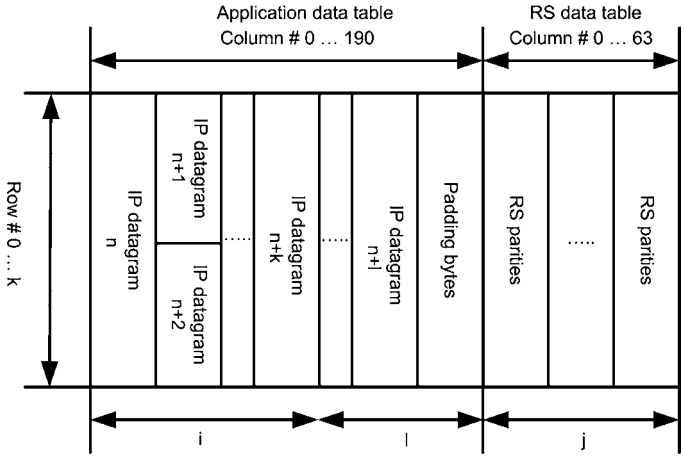

Fig. 8.11 Incorrect MPE-FEC frame, resulting in loss of IP datagrams for column interval *l*, for the situation that interval *i* contains more defect columns than the available parity columns *j*

[12] Note that it is prohibited to mix IPv4 and IPv6 in a single service.

can be calculated using the UDP length field.[13] The calculation of a UDP checksum requires the availability of all UDP data, which under certain conditions may be difficult, e.g., for the situation of a fragmented IP datagram, for example, such a situation occurs when a fragmented IP datagram is split over two time-sliced bursts. In a robust link-layer implementation, the applicability of the OSI Layer 3-4 checksum has to be considered to avoid unnecessary loss of IP datagrams. This aspect will be further discussed in Sect. 8.4.

8.3.2 DVB-H Service Information

Service discovery is performed by the receiver middleware, via interpretation of the received SI and PSI information. In order to receive a requested DVB-H service, the link layer must be configured. This section describes the *time_slice_fec_identifier_descriptor* and *data_broadcast_descriptor*, which contain essential information for configuring the DVB-H link layer.

8.3.2.1 Time_slice_fec_identifier_descriptor

A *time_slice_fec_identifier_descriptor* identifies whether time-slicing and/or MPE-FEC are used on an Elementary Stream, which is available for each time-sliced Elementary Stream describing the DVB-H specific link-layer settings. This descriptor can be transmitted by either the Network Information Table (NIT), the IP/MAC Notification Table (INT), or Conditional Access Table (CAT) [18]. An Elementary Stream can share a *time_slice_fec_iden-tifier_descriptor* with other Elementary Streams and a *time_slice_fec_identifier_descriptor* can override previous settings all depending on which *descriptor_loop* and which table the descriptor appears. Table 8.4 depicts the *time_slice_fec_identifier_descriptor* syntax after [18]. The *descriptor_tag* field is set to "0x77." The *descriptor_length* field specifies the number of bytes of the descriptor immediately following this field. The *time_slicing* field when set to "1" signals whether the referenced Elementary Stream is time-sliced. The value "0" indicates that time-slicing is not used. The *mpe_fec* field is a 2-bit field, which is further explained in Table 8.5. The *frame_size* field is a 3-bit field and provides information that a decoder may use to adapt its buffering usage. The exact interpretation depends on whether time-slicing and/or MPE-FEC are used. In case time-slicing is used (i.e. time-slicing is set to "1"), this 3-bit field indicates the maximum number of bits in section payloads within a time-slice burst for the Elementary Stream. For MPE sections, payload bits are counted over *ip_datagram_data_bytes* or *LLC_SNAP* field (whichever is supported), excluding any possible *stuffing_bytes*. For MPE-FEC sections, payload bits are counted over *rs_data_bytes*. When MPE-FEC is used (i.e. *mpe_fec* is set to "0x1"), this field

[13] The IP version used can be determined from the IP header of a correctly received IP datagram.

Table 8.4 Syntax *time_slice_FEC_identifier_descriptor*

Syntax	Number of bits
time_slice_fec_identifier_descriptor () {	
descriptor_tag	8
descriptor_length	8
time_slicing	1
mpe_fec	2
reserved_for_future_use	2
frame_size	3
max_burst_duration	8
max_average_rate	4
time_slice_fec_id	4
for(i=0; i<N; i++) {	
id_selector_byte	8
}	
}	

Table 8.5 MPE-FEC algorithm

Value	MPE-FEC	Algorithm
00	MPE-FEC not used	n/a
01	MPE-FEC used	[255,191,65] RS
01 to 11	Reserved for future use	Reserved for future use

Table 8.6 Size coding

Size	Max burst size	MPE-FEC frame rows
0x00	512 kbits	256
0x01	1024 kbits	512
0x02	1536 kbits	768
0x03	2048 kbits	1,024
0x04 to 0x07	Reserved for future use	Reserved for future use

indicates the exact number of rows in each MPE-FEC frame for the Elementary Stream. If both time-slicing and MPE-FEC are used for an Elementary Stream, both constraints (i.e. the maximum burst size and the number of rows) apply. If *time_slice_fec_id* is set to "0," the coding of the *frame_size* is according to Table 8.6. If *time_slice_fec_id* is set to any other value, coding of the *frame_size* is currently not defined. The *max_burst_duration* field is used to indicate the maximum burst duration in the Elementary Stream. A burst shall not start before $T1$ and shall end not later than at $T2$, where $T1$ is the time indicated by *delta_t* in the previous burst, and $T2$ is $T1 + MBD$, where MBD is the actually computed maximum burst duration in time (Fig. 8.12). If the *time_slice_fec_id* is set to "0," the computed value for MBD shall be between 20 ms and 5.12 s. The MBD parameter is computed according to

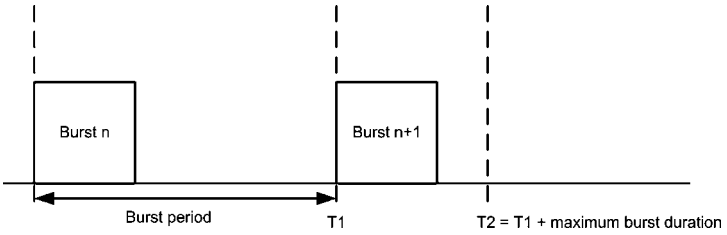

Fig. 8.12 Relation between service burst, *delta_t* and maximum burst duration

Table 8.7 Coding for max_average_rate

Bit rate	Description
0000	16 kbps
0001	32 kbps
0010	64 kbps
0011	128 kbps
0100	256 kbps
0101	512 kbps
0110	1,024 kbps
0111	2,048 kbps
1000 to 1111	Reserved for future use

the following formula: $MBD = (max\_burst\_duration + 1) \times 20ms$, with a resolution of 20 ms. If the *time_slice_fec_id* is set to any other value than "0," the coding of the *max_burst_duration* is currently not defined. When *time_slicing* is set to "0" (i.e. time-slicing not used), this field is reserved for future use and shall be set to "0xFF" when not used. The *max_average_rate* field is used to define the maximum average bit rate in the MPE-section payload measured within one time-slicing cycle or one MPE-FEC cycle, and it is derived by finding the maximum of

$$C_b = \frac{B_s}{T_c}, \tag{8.2}$$

where C_b denotes the actual calculated bit rate, B_s is the size of the current time-slicing burst or MPE-FEC frame in MPE-section payload bits and T_c is the time between the transport packet carrying the first byte of the first MPE section for the current burst (frame) and the transport packet carrying the first byte of the first MPE section for the next burst (frame) within the same Elementary Stream. Note that when MPE-FEC is used, the RS data is not included in B_s. If *time_slice_fec_id* is set to "0," the coding of the *max_average_rate* is according to Table 8.7. If *time_slice_fec_id* is set to any other value, coding of the *max_average_rate* is currently not defined. The *time_slice_fec_id* field identifies the usage of the following *id_selector_byte(s)*. Currently those bytes are not used, and this field shall be set to value "0x0," and *id_selector_byte(s)* shall not be present. Note that this field affects the coding of *frame_size*, *max_burst_duration*, and *max_average_rate*

fields on the actual descriptor, and the *address* field of real-time parameters on the referred Elementary Stream. The definition of the *id_selector_byte(s)* of the *time_slice_fec_identifier_descriptor* will depend on the *time_slice_fec_id*.

Which *time_slice_fec_identifier_descriptor* applies to an Elementary Steam depends on the position within the SI information. When located in the first descriptor loop of the NIT, the descriptor applies to all Elementary Streams within all Transport Streams in the DVB network. If located in the second descriptor loop of NIT, the descriptor applies to all Elementary Streams within the referred Transport Stream, overriding any time-slicing or FEC information from the first descriptor loop. If located in the *platform_descriptor_loop* of an INT, the descriptor applies to all Elementary Streams referenced within the table, overriding any time-slicing or FEC information from the NIT. If located in the *target_descriptor_loop*, the descriptor applies to all Elementary Streams referenced within the current iteration of the *target_descriptor_loop* following the appearance of the descriptor, overriding any time-slicing or FEC information from the *platform_descriptor_loop* and in the NIT.[14] The descriptor may appear more than once, in which case each new occurrence overrides previous occurrence(s) [14].

8.3.2.2 Data_broadcast_descriptor

For each Elementary Stream carrying MPE-encapsulated IP stream(s), *SDT_actual* contains a *data_broadcast_descriptor* [18]. Table 8.8 indicates the *data_broadcast_descriptor* syntax after [16]. The *descriptor_tag* value is set to the value "0x64" [16].

Table 8.8 Syntax *data_broadcast_descriptor*

Syntax	Number of bits
data_broadcast_descriptor () {	
descriptor_tag	8
descriptor_length	8
data_broadcast_id	16
component_tag	8
selector_length	8
for(i=0; i<selector_length; i++) {	
selector_byte	8
}	4
ISO_639_language_code	24
text_length	8
for(i=0; i<text_length; i++) {	
selector_byte	8
}	
}	

[14] Note that the descriptor applies to Elementary Streams with *stream_type* "0x90."

The *descriptor_length* indicates the number of descriptor bytes following this field. The *data_broadcast_id* field is set to the value "0x0005," indicating that the *multiprotocol_encapsulation_info* structure is used. The *component_tag* is employed to label component streams with a unique tag value. The *component_tag* is unique within the DVB service and shall have the same value as a *component_tag* field of a *stream_identifier_descriptor* that may be present in the second descriptor loop of the PMT sub_table. The *selector_length* field is set to the value "0x02." There are two *selector_byte* fields, due to the value of the *selector_field*. The meaning of the *selector_byte* values is defined by the *multiprotocol_encapsulation_info* syntax, as depicted in Table 8.9 after [18]. The *MAC_address_range* field indicates the number of MAC address bytes that the service uses to differentiate the receivers according to Table 8.10. When the *MAC_IP_mapping_flag* field is set to "1," the service applies to the IP-to-MAC mapping as described for IPv4 multicast addresses [3] and for IPv6 multicast addresses [2]. If this flag is set to "0," the mapping of IP addresses to MAC addresses is done outside the standard [18]. The *alignment_indicator* field indicates the alignment that exists between the bytes of the datagram section and the Transport Stream bytes according to Table 8.11. The *alignment_indicator* is set to "1" according to [14]. The reserved field is set to the value "0x07." The

Table 8.9 Syntax for multiprotocol_encapsulation_info structure

Syntax	Number of bits
multiprotocol_encapsulation_info () {	
MAC_address_range	3
MAC_IP_mapping_flag	1
alignment_indicator	1
reserved	3
max_sections_per_datagram	8
}	

Table 8.10 Coding of the MAC_address_range

MAC_address_range	Valid MAC_address bytes
0x00	reserved
0x01	6
0x02	6, 5
0x03	6, 5, 4
0x04	6, 5, 4, 3
0x05	6, 5, 4, 3, 2
0x06	6, 5, 4, 3, 2, 1
0x07	Reserved

Table 8.11 Coding of the *alignment_indicator* field

Value	Alignment in bits
0	(8)-default
1	32

max_sections_per_datagram is set to the value "0x01," indicating that each IP datagram is carried within a single MPE section. The *component_tag* field is set to the value announced within the PMT sub_table of the DVB service for the referred component.

8.4 Efficient and Robust DVB-H Link Layer Implementation

This section will elaborate on the functional blocks together forming an efficient and robust link layer, using the definitions regarding efficiency and robustness as defined in Sect. 8.1.

8.4.1 DVB-H Link Layer Functional Blocks

Figure 8.13 depicts a block diagram of an efficient and robust DVB-H link layer, which is briefly described hereafter. The remaining subsections further elaborate on the individual functional blocks. The MPEG-2 demultiplexer is the first functional block that operates on the received Transport Stream, applying *PID* filters to select the requested Elementary Streams. After *PID* filtering, the Elementary Stream is either forwarded to the SI/PSI section filters or IP de-encapsulation filters. The queue manager forwards the correctly received SI/PSI sections to the host processor, using messages equipped with, e.g., locator information, indicating the queue from which the section originates. The reliable or unreliable de-encapsulated IP datagrams or IP-datagram fragments are temporally stored in the scratch buffer prior to final transferring them to the proper position in the MPE-FEC frame. During

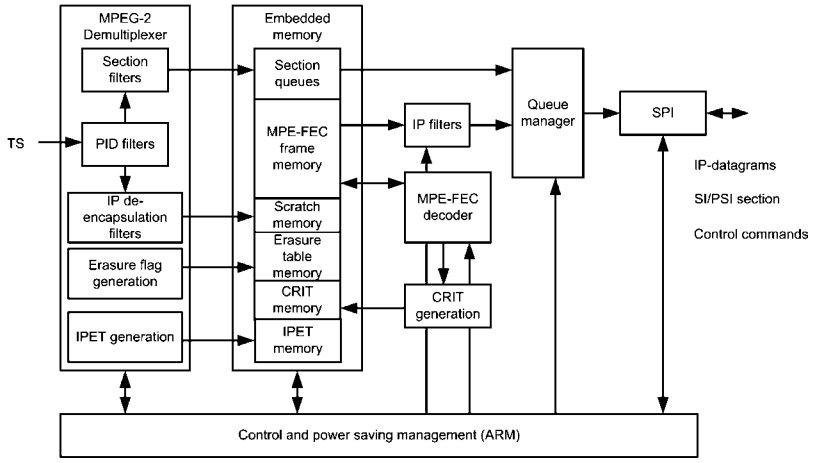

Fig. 8.13 Functional block diagram of an efficient and robust DVB-H link layer

IP de-encapsulation of the reliable and unreliable IP datagrams, locator information indicating the start position in the MPE-FEC frame of correctly received IP datagrams is stored in the Internet Protocol Entry Table (IPET). Furthermore, for each MPE-FEC symbol, corresponding erasure information is preserved in the erasure table. After receiving all data of a particular service burst, the [255,191,65] RS MPE-FEC decoding is applied for the case that not all IP datagrams were correctly received. For each row, the MPE-FEC decoding result is stored in the Corrected Row Index Table (CRIT). For the situation that all IP datagrams are correctly received, or are correctly available after applying MPE-FEC decoding, readout of the individual IP datagram is possible, using the traditional parsing method as discussed in Sect. 8.3.1. For the situation that not all IP datagrams were correctly received and the MPE-FEC decoder was not able to correct all MPE-FEC frame rows, readout of the correct IP datagrams is possible using the IPET and CRIT in combination with the erasure table. The IP datagram readout also applies filtering of IP datagrams on the basis of source and/or destination address(es). All filtered IP datagrams are forwarded to the host processor via a specific messaging technique, allowing the multiplexing of IP datagrams, SI/PSI section information, or system status information. To enable TDM-based broadcast reception, the link layer maintains service-reception synchronization and context control. Time-sliced service broadcasting allows the definition of two or more reception contexts, each of which can be regarded as a virtual receiver. In, e.g., the main reception context, the main service(s) are received. After main service reception, the receiver waits for the next service burst, based on the synchronization information (*delta_t*). During the off-time of the main context, an auxiliary context can be started. In the auxiliary context, different tuning parameters and filter settings for the available resources can be applied. The auxiliary context allows, e.g., to peek into neighboring channels. During such a peek operation, SI/PSI information can be collected as well as monitoring for *delta_t* values of the requested service. Based on the building blocks of Fig. 8.13, a data-flow chart is derived and depicted in Fig. 8.14. This data-flow chart is available for each context. The number of individual filters as indicated in Fig. 8.14, is derived in the corresponding subsections.

8.4.1.1 MPEG-2 Demultiplexer

At the left-hand side of Fig. 8.13, the received MPEG-2 TS enters the MPEG-2 demultiplexer. Elementary Streams, either SI/PSI, or those containing a DVB-H service, that are requested by the application, are filtered on the basis of their *PID* value. After *PID* filtering and processing of the other Transport Stream main header fields, the SI/PSI section information is routed through the SI/PSI section filters, where one or more SI/PSI filters can receive section data from the same *PID* filter. Although MPE encapsulation uses a section to encapsulate a single IP datagram, IP de-encapsulation differs somewhat from the traditional section filtering, due to the presence of the MPE-FEC. The IP de-encapsulation process is also responsible for the generation of erasure flags and the IPET entries. To allow proper erasure-flag

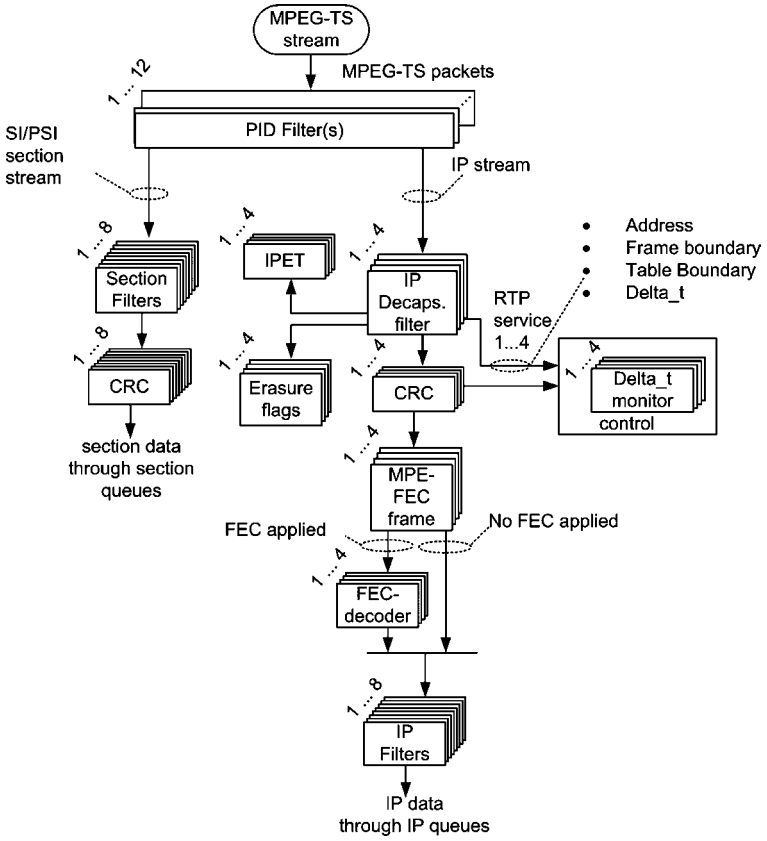

Fig. 8.14 DVB-H link layer data processing flow

generation, the TS main-header filtering (see Sect. 8.5) and the IP de-encapsulation filtering are jointly processed.

The number of available *PID* filters is equal to the sum of the maximum number of section filters and the maximum number of IP de-encapsulation filters.

The number of available section filters depends on the number of simultaneous section filters required by the middleware. From experiments, it can be concluded that the DVB-H middleware shows a satisfactory performance, when eight section filters are available.

The number of available IP de-encapsulation filters depends on the type of application. From an MPE-FEC frame size point-of-view, four IP de-encapsulation filters allow the reception of four parallel services, each with an MPE-FEC frame size of 256 rows. The sum of these four MPE-FEC frame sizes results in 1,024 rows, the maximum MPE-FEC frame size. The availability of four IP de-encapsulation filters allows the reception of parallel services with different MPE-FEC frame sizes, provided that the sum of the individual MPE-FEC frame rows for the various services is less than or equal to 1,024 rows.

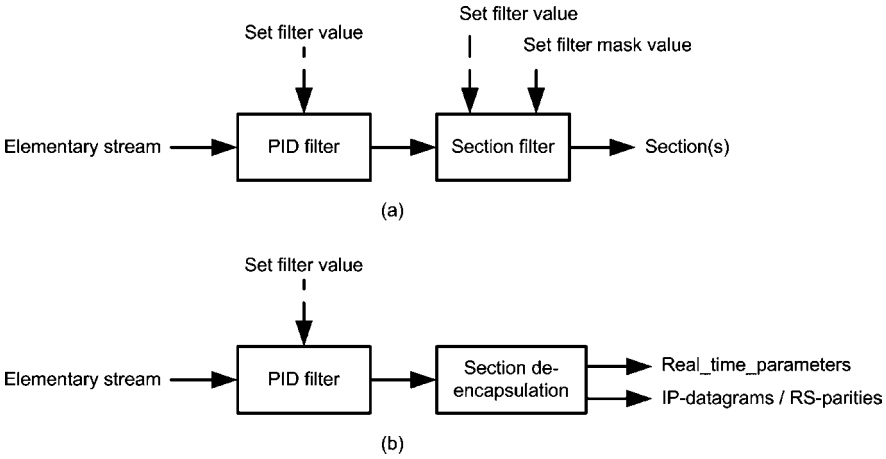

Fig. 8.15 Filter settings for a section-filter pair and IP de-encapsulation filter pair. **(a)** A section-filter pair consisting of a *PID* filter in combination with a section filter. **(b)** An IP-filter pair consisting of a *PID* filter in combination with an IP de-encapsulation filter

Figure 8.15a indicates the programming interface for a section-filter pair, consisting of a *PID* filter and a section filter. Figure 8.15b indicates the programming interface for an IP de-encapsulation filter and a *PID* filter, again forming a filter pair. An IP de-encapsulation filter requires no programming interface. The reason for this is twofold. First, the MPE header does not contain fields that require programmable filter operations. Although the standard allows for filtering on *MAC_address_5* and *MAC_address_6* fields, this filtering can be postponed until the actual IP datagrams are filtered from the MPE-FEC frame, using the IP source and/or destination address(es). On top of this, the distinction between MPE section and MPE-FEC section is achieved via the *table_id* of the corresponding section header, which is a fixed but different value for both MPE and MPE-FEC section type. Second, the payload of an MPE section is byte-aligned, see Sect. 8.3.2, resulting in the absence of padding bytes, avoiding notification of the IP de-encapsulation filter.

8.4.1.2 Section Queues

The total memory footprint for the section queues is the sum of the individual section-queue memory footprints. In the worst-case situation, all eight section filters operate on the same section data, which could be up to 4 kbyte in size, resulting in a total maximum section-queues size of 32 kbyte. Assigning 4-kbyte section-queue space rigid to each section filter will negatively influence the SI/PSI filtering performance. For the situation that a section queue contains a correctly received section, the queue manager needs to forward this data for further processing to a host processor. This is an interrupt-driven process and takes time to be executed. When the section queue contains a section, that is only a fragment of the 4 kbyte it can

maximally store, the remaining space is not accessible by the section filter for further usage. As a result, the section filter is blocked for the duration that a section is forwarded to a host processor. On the average, the received sections will have a size that is much smaller than the maximum size of 4 kbyte. It is therefore advantageous to fragment the 32 kbyte section-queue memory into smaller fragments of, e.g., 256 bytes, allowing a better section-filter performance. The 256-byte fragments are used to construct a section queue that fits best to the received section. This section-queue buffer is allocated with aid of the *section_length* field, which is available in each section header. With some bookkeeping (e.g., section-queue number, memory start-address, and section length), the individual sections are traced in the section-queue memory space. This allows the queue manager to read the individual sections, create the corresponding message, send this to the host processor, and release the allocated memory-queue fragments. As long as there are free section-queue fragments, active section filters are prevented from being blocked, as long as the received sections fit within the available section-queue memory space, thereby increasing the section-filtering performance.

8.4.1.3 Erasure-Table Memory

The erasure table stores 2-bit erasure flags, see Sect. 8.5, generated by the IP de-encapsulation filter, for each MPE-FEC symbol that constructs the application data table. If each symbol of the application data table would have its own erasure flag, the memory footprint requires $2 \times 255 \times 1{,}024$ bits. Fortunately, the symbols of the application data table are transmitted via TS packets, which means that more than one symbol will have the same 2-bit erasure value. Let us assume that an IP datagram can be 32 bytes in size and transmitted in a single TS packet. An MPE-FEC frame of size 1,024 rows can hold 32 IP datagrams per column, resulting in 32 fragments of 32 bytes per MPE-FEC frame of size 1,024. We also assume that the RS data can be transmitted in a similar way as the IP datagrams. As a result, the erasure table will have $32 \times 255 = 8{,}160$ entries. Although the erasure-flag information requires a 2-bit storage, the erasure-transition coding requires 3-bit coding, see Table 8.12. Another 5 bits are required to indicate the position where a transition occurs within the 32 byte fragment. The total memory involved per erasure

Table 8.12 Coding of erasure type transition

Code	First stage	Second stage
000	OK	False
001	OK	Unknown
010	False	OK
011	False	Unknown
100	Unknown	OK
101	Unknown	False

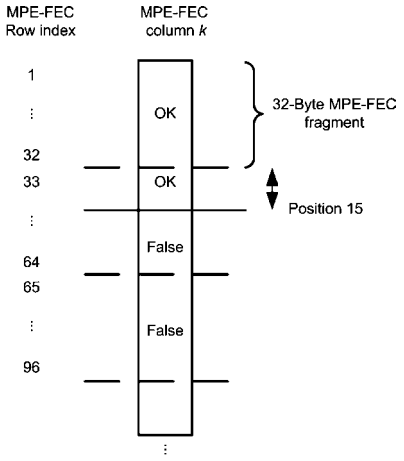

Fig. 8.16 Storage example of an erasure transition in the erasure memory

Table 8.13 Mapping of transition-type bits and transition-position bits forming an erasure entry

Bits	Name
7 : 5	Transition type
4 : 0	Transition position

entry is 1 byte, resulting in a total erasure table footprint of 8,160 bytes. Figure 8.16 indicates an example transition. The erasure-flag information in column *k* is OK for the first 46 MPE-FEC frame rows. When the erasure-type transition bits form the MSB bits of the erasure entry and the 5-bit transition position form the lower bits, see Table 8.13, then the first erasure entry for column *k* is "0x1F." The length value "0x1F" indicates that the transition lies outside this fragment. The second erasure entry will have an OK to False transition at position 15, resulting in an entry value of "0x0F."

8.4.1.4 Scratch Memory

This memory is used during the IP de-encapsulation and will be elaborated in Sect. 8.5. The size is determined by the maximum size of an IP datagram for MPE encapsulation, which is 4,080 bytes.

8.4.1.5 MPE-FEC Frame Memory

In principle, the MPE-FEC frame memory is equal to the product of the maximum MPE-FEC frame size, which is 1,024 rows and the sum of the application data table and RS-data table columns, which is 255 columns, resulting in total MPE-FEC

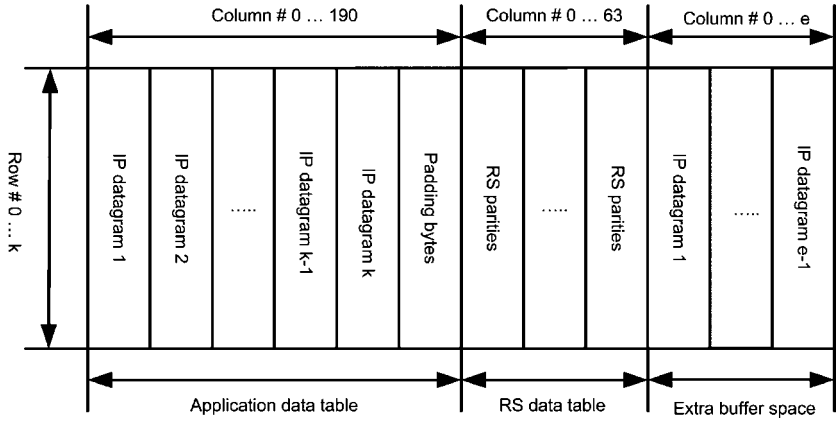

Fig. 8.17 MPE-FEC frame extended with extra columns allowing continuous reception of consecutive services

frame size of 255 kbyte. With this memory pool, various reception situations are supported as long as the sum of the individual MPE-FEC frame sizes are less than or equal to 1,024 rows. The MPE-FEC calculation time and IP datagram transfer time are neglected in the calculation, so is the data rate at which the service enters the link layer. These aspects have their influence on the reception of consecutive services. To allow the reception of two consecutive services with, e.g., an MPE-FEC frame size of 1,024 rows and MPE-FEC, extra memory is required to buffer the new incoming IP data, while the previous burst is still in process. Figure 8.17 indicates an MPE-FEC frame with an extra memory extension. Let us assume a situation, as depicted in Fig. 8.7, in which consecutive Services 4 and 5 both equipped with MPE-FEC are requested by the application. Let us consider the case that the application data table and RS data table are filled with data from Service 4. After finishing the Service-4 IP de-encapsulation process, the MPE-FEC is applied, provided that Service 4 was subject to errors. While the FEC is calculated, the Service-5 Elementary Stream simultaneously enters the MPEG-2 demultiplexer. To avoid Service-5 IP datagrams loss, extra MPE-FEC frame memory, columns 0...*e*, are added to the MPE-FEC frame. This extra memory space provides buffering to cover the time that the MPE-FEC calculation is performed. After finishing the MPE-FEC decoding of Service 4, the RS-data-table memory space is available for storing IP datagram or RS-parity data of Service 5. The columns that formed the application data table for Service 4 will become available somewhere during the reception of Service 5, but after transferring the Service-4 IP datagrams to the host. In this example, only two consecutive services are received. When the amount of consecutive services that is received is increased, MPE-FEC memory fragmentation occurs during the reception of those services. This fragmentation stops after reception of the last consecutive service burst.

8.4.1.6 IP Filter

The received Elementary Stream may contain more IP flows than actually required for the requested service. Such a situation occurs when services are multiplexed at IP level, as indicated in Sect. 8.2.2. An IP filter operates on IP source and/or destination address(es), allowing various filter operations. For the case that an Elementary Stream contains multiple IP flows, IP filtering lowers the amount of IP data that needs to be transferred to the host processor, reducing the transmission time of the IP data, lowering the power consumption. Because a service can be based on IPv4 or IPv6, two filter versions, one for each standard, would be required to perform the required source and/or destination address(es) filtering. Such a situation is avoided when the filtering is achieved by a filter which is IP-version agnostic. Such a filter consists of a filter value, filter-mask value, and an offset value, indicating the start position from where in the IP datagram, the filter value, and filter-mask value are to be applied. The offset is relative to the start of an IP datagram, as indicated in Fig. 8.18. In this way, only the application needs to be aware of the used IP version, calculating the proper filter value, filter-mask values, and offset value. The lengths of the filter value and filter-mask value are fixed to 40 bytes, allowing to operate on IPv4 and IPv6 datagram headers. With aid of the filter-mask value, a filter can be created that filters on only the source address, destination address, or both source and destination address(es) for both either IPv4 or IPv6. The number of IP filters is based on the number of available IP de-encapsulation filters and the number of IP flows per service. An audiovisual service results in two IP flows, one that contains the audio, while the other contains the video information. With a maximum of four IP de-encapsulation filters, minimally eight IP flows need to be handled, resulting in eight IP filters, which can be switched off for creating a bypass mode, resulting in all IP data to be delivered at the host processor.

8.4.1.7 Queue Manager

The queue manager is responsible for avoiding a queue from overflowing, by creating messages that are send-toward the host processor. Since the messages are sequentially transmitted over the external interface, a message header containing vital information regarding, e.g., data type, original queue number, is required to

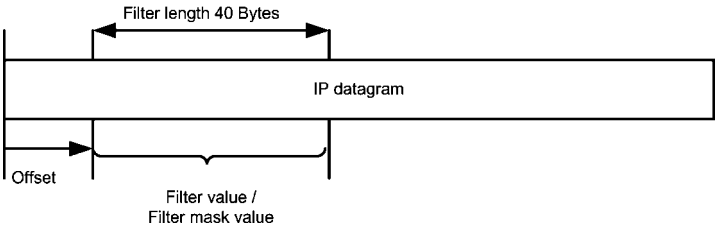

Fig. 8.18 An IP version agnostic IP filter method

allow proper routing of the message data at the host, e.g., invocation of the correct callback function for the middleware. A message can contain receiver-status information, or either one or more IP datagrams, or SI/PSI sections. Combining two or more IP datagrams, or SI/PSI sections into a single message, increases the transfer performance, avoiding unnecessary polling or even interrupt generation.

8.4.1.8 SPI Interface

A popular interface for connecting devices in, e.g., a mobile phone is the four-wire Serial Peripheral Interface (SPI). SPI requires a master to be present in the system providing the SPI clock. The value for the SPI clock depends on various aspect such as:

- Maximum link-layer input bit rate
- IP-datagram transfer time
- Receiver-software download time

The maximum link-layer input bit rate is determined by the transmission modulation settings, which is for an 8-MHz channel bandwidth equal to 31.6 Mb/s. Such an input bit rate results in an SPI clock of 32 MHz, if the amount of buffering is to be minimized. The modulation setting resulting in a channel throughput of 31.6 Mb/s will most probably not be used, due to poor mobile reception performance. Modulation settings that result in good or even excellent reception performance have a bit rate around the 14 Mb/s. As a result, the SPI clock will be at least 14 MHz, if the amount of buffering is to be minimized. The upper bound may be far above the 32 MHz, especially when the SPI bus is shared with other slaves or simply to reduce the receiver power consumption. A receiver can go to sleep mode,[15] consuming only a few milliWatts, see Sect. 8.2.2, as soon as all requested data has been fetched by the host processor. Finally, the SPI clock speed must be such that it allows to satisfy the requirements regarding the receiver-software download time. A guideline value for the software-image download time is in the range of 200 ms.

8.4.1.9 Link-Layer Control

The link-layer control has to deal with the following aspects:

- Front-end control
- Power-mode control
- Booting of the downloaded software image
- Command interpretation and control
- Context management
- Maintain service synchronization

[15] Note that the receiver front-end can be switched off, right after receiving the last data of a burst, or after the max burst duration if the service burst end is missed.

- Collect and calculate receiver condition
- Provide handover control

To avoid knowledge on the host concerning time-slicing, the link layer controls also the receiver front-end. Power consumption is reduced to a minimum, when the receiver is switched off completely. As a result, the receiver control will enable receiver booting, when the receiver module is activated. After booting, which may take around 10 ms, the link-layer control will notify the host that the receiver system is booted and ready for software download. Software download by the host avoids permanent storage inside the receiver module. Once the receiver is booted, the software image is downloaded and installed. An interrupt notifies the host that the receiver is ready to receive commands. Possible commands initiated by the host are, e.g., reset_link_layer, tune_to, setup_queue, start_queue, etc. The commands are equipped with a context parameter, indicating for which context, e.g., the filter settings apply. A context can be compared to a task in an Operating System (OS), where a task is executed based on its priority. A context in DVB-H has a priority but can be blocked by a context with an even higher priority. The context with the highest priority is the "main" context, delivering the service with the highest priority requested by the application. Other contexts are, e.g., peek context and handover context. The peek context allows to peek into neighboring channels and collect relevant information such as, e.g., SI/SPI, whereas the handover context prepares the receiver to switch to a new receiver setting. The link-layer control maintains service-reception synchronization for the service associated to the main context, using the transmitted *delta_t* value, and offers the receiver resources to the other contexts, e.g., the peek context, during the main-context off-time. At various stages in the receiver chain, signal-quality parameters, also known as Quality-of-Service (QoS) parameters, are available or can be calculated. Figure 8.19 indicates the high-level receiver building blocks and the associated signal-quality parameters. Such parameters are:

- Received Signal-Strength Indicator (RSSI), indicating the strength of the selected DVB-H signal, necessary for the handover algorithm
- Bit Error Count before Viterbi decoding (VBER), indicating the number of erroneous bits during the estimation period before Viterbi decoding
- Bit Error Count after Viterbi decoding (BER), indicating the number of erroneous bits during the estimation period after Viterbi decoding

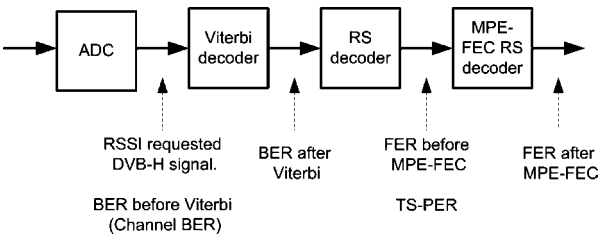

Fig. 8.19 Monitored signal-quality parameters

- Transport Stream Packet Error Count (TS-PER), indicating the number of erroneous TS-packets during the estimation period
- Frame Error Count before MPE-FEC decoding (FER)
- Frame Error Count after MPE-FEC decoding (MFER)

Horizontal handover is issued by the application, based on deterioration of the received QoS for the main context. The application configures a handover context and request for a handover. The actual handover is in the hand of the link-layer control, again avoiding the host processor from being aware of any form of time-slicing. For the situation that more than one service is received, the handover is performed for the service with the highest priority. This priority is set by the application and is, e.g., part of the setup_queue command.

8.4.2 IP Datagram Recovery in the DVB-H Link Layer

The DVB-H block diagram as depicted in Fig. 8.13 provides means to guarantee that correctly received IP datagrams will be forwarded to the host under all circumstances, realizing the required robustness as defined in Sect. 8.1. Furthermore, there are means to determine the location of corrected MPE-FEC frame rows, for MPE-FEC frames that remain defect after FEC decoding, which enable the location of corrected IP datagram bytes. The robustness is obtained via the usage of two tables. The first table is the Internet Protocol Entry Table (IPET) [5], which stores the start address of a correctly received IP datagram in the MPE-FEC frame. The second table is the Corrected Row Index Table (CRIT), which stores for each MPE-FEC frame row the result of the FEC decoder. Both tables are elaborated in the remaining part of this subsection.

Figure 8.20 depicts the results of the IPET and CRIT concept. Due to the IP de-encapsulation concepts as presented in Sect. 8.5, the receiver can correct up to a TS-packet error rate of 10%. For higher TS-packet error rates, the MPE-FEC frame becomes uncorrectable. With the aid of IPET and CRIT, considerable amount of IP datagrams can be retrieved from the defect MPE-FEC frame. Although the audiovisual information is corrupted, smart error-concealment techniques can camouflage this loss up to a certain extent. The IP datagram recovery due to the CRIT highly depends on the IP datagram size and on the MPE-FEC frame size. For IP datagram sizes around 128 bytes and MPE-FEC frame size of 1,024 rows, 4% of the total amount of IP datagrams contained by a defect MPE-FEC frame can be recovered. For IP datagram sizes larger than 512 bytes and MPE-FEC frame size of 1,024 rows, the recovery due the CRIT is below 1%.

8.4.2.1 Internet Protocol Entry Table

During reception of a service burst, the IP de-encapsulated IP datagrams are stored at predetermined positions in the MPE-FEC frame. The IP datagram start-position

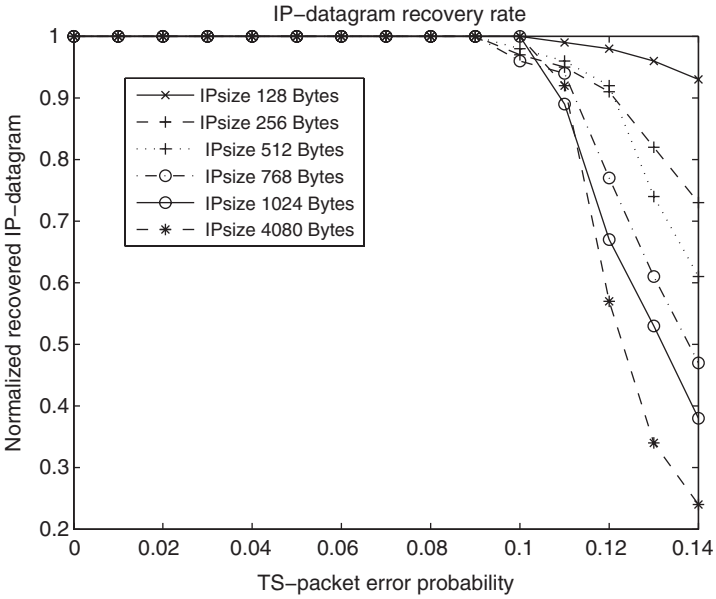

Fig. 8.20 Normalized IP datagram recovery in defect MPE-FEC frames for various IP datagram sizes and varying TS-packet error rate

in the MPE-FEC frame is for each IP datagram indicated by the broadcasted *real_time_parameters* field *address*, which is part of the MPE section header. When after IP de-encapsulation, the MPE section is signalled reliably by means of the CRC code, the *address* field value points to a correctly received IP datagram. The DVB-H standard does not provide means to store this *address* field value, resulting in the loss of this locator information after receiving the service burst. When the *address* field value is stored as locator information in the IPET, each correctly received IP datagram can be retrieved from the MPE-FEC frame. Discontinuities in the IPET-stored locator information indicate the positions of erroneously received IP datagram(s). The IPET locator information solves the intrinsic problem of locating the start positions of IP datagrams in the MPE-FEC frames, which is caused by linked-list way the IP datagrams need to be retrieved from the MPE-FEC frame. The received IP datagrams are stored in a linear fashion in the application data table, as mentioned in Sect. 8.3.1. IP datagram retrieval requires inspection of each IP header to determine its length, in order to retrieve the IP datagram and automatically locating the start position of the next IP datagram. This parsing method collapses as soon as an IP header length field is corrupted, or an IP datagram is missing. The IPET guarantees retrieval of correctly received IP datagrams, regardless of the outcome of the link-layer FEC. Besides information on the correct IP datagrams, IPET also provides information about unreliable or missing IP datagrams. Figure 8.21 visualizes two data application tables. The data application table of Fig. 8.21a contains only correctly received IP datagrams, whereas Fig. 8.21b holds three correct and

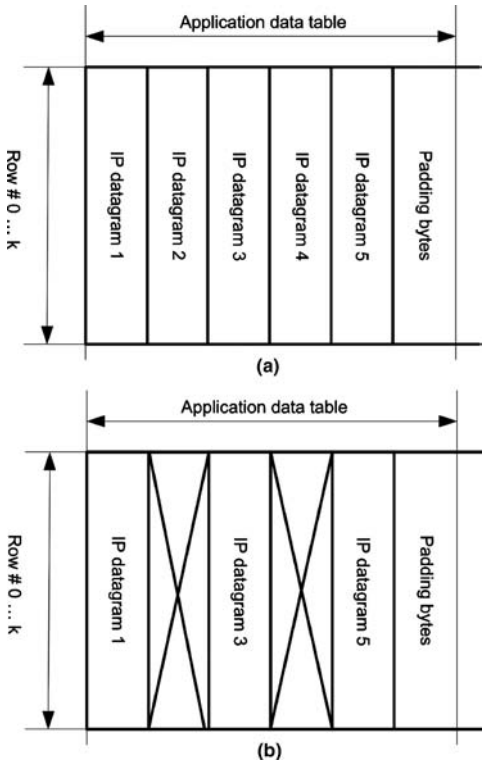

Fig. 8.21 Data application table of an MPE-FEC frame with size $k = 1,024$ rows, containing five IP datagrams. **(a)** Data application table contains only reliable IP datagrams. **(b)** Data application table contains reliable and unreliable IP datagrams

two defect received IP datagrams. Let the IPET locator information corresponding to Fig. 8.21a be $\{0;1,024;2,048;3,072;4,096\}$ and the IPET locator information corresponding to Fig. 8.21b be equal to $\{0;2,048;4,096\}$. The IPET of an application data table that is constructed of only correctly received IP datagrams shall not contain discontinuities. A discontinuity in the IPET occurs when the stored locator information incremented with the IP datagram length of the IP datagram indicated by the locator information does not correspond to the next IPET entry. The IPET corresponding to Fig. 8.21a does not contain discontinuities, because each IPET entry incremented with the length of the IP datagram to which it refers, which is in this example 1,024 bytes, corresponds to the next IPET entry. The IPET associated to Fig. 8.21b contains two discontinuities, indicating that two or more IP datagrams have been received incorrectly.[16] The IPET memory footprint depends on the IP

[16] In this particular example, the IP datagrams have a fixed length of 1,024 bytes. In practice, IP datagrams will have a variable length, resulting in an unknown number of IP datagrams per IPET discontinuity.

datagram size and the locator size. The locator size is directly derived from the *address* field size, which requires 18 bits. The IP datagram size depends on various aspects. On the one hand, there is a dependency on the Maximum Transmission Unit (MTU) size for Ethernet, resulting in IP datagrams with a maximum size of 1,500 bytes. On the other hand, it is influenced by the MTU for the MPE, which is 4,080 bytes. Finally, the third aspect is the MPE-FEC performance, which depends strongly on the IP datagram size [44]. According to [44], optimal IP datagram sizes have a range between 1,024 bytes and 2,048 bytes. The IPET memory footprint for an MPE-FEC frame size of 1,024 rows results in roughly 1,024 bytes. This memory footprint size is based on an IP datagram size of 1,024 bytes, using 3 bytes per IPET locator, although 573 bytes would be sufficient. In practice, H.264 encoded video will be transmitted using IP datagrams that are in the range of 1,024–1,500 bytes and AAC+ audio will use IP datagrams that are in the range of 512–1,024 bytes. Taking into account that video requires more bandwidth, the small-sized IP datagrams occur less often. As a result, the remaining IPET address space is sufficient for handling typical broadcast situation, with the mix of audio and video packets. Depending on the result of the FEC decoding, all erroneous positions in the MPE-FEC frame are repaired or some of the erroneous positions are still defective. The possible erroneous positions correspond to the regions covered by the IPET discontinuity positions. Data reliability can be determined using the OSI Layer 3-4 checksum, as discussed in Sect. 8.3, but this introduces some drawbacks. These drawbacks are avoided when using the CRIT, which is elaborated in the next subsection.

8.4.2.2 Corrected Row Index Table

The CRIT is a table giving the FEC decoding result for each MPE-FEC frame row. After FEC decoding there are two situations. In the first situation, all MPE-FEC frame rows are corrected. As a result, the CRIT entries all indicate a successful FEC decoding operation. In the second situation, not all MPE-FEC frame rows are corrected, so that not all CRIT entries indicate a successful FEC decoding operation. The CRIT is a table storing a 1-bit flag to signal the FEC decoding result, leading to a memory footprint of 128 bytes for an MPE-FEC frame size of 1,024 rows. For the case that an MPE-FEC frame still contains errors even after FEC decoding, the combination CRIT, IPET, and erasure-flag information allows to determine whether an erroneously received IP datagram is corrected. The advantage of this concept is that there is no need to process the data using higher OSI layer protocols, resulting in a uniform recovery approach, regardless of the used network and transmission protocol. Figure 8.22 visualizes the CRIT and erasure-flag processing, for a fragment of column k, indicated by the IPET as a discontinuity region. In Fig. 8.22, the CRIT positions indicated as "0" correspond to MPE-FEC frame rows that are correct, whereas the positions indicated by a "1" mean that the MPE-FEC frame row could not be corrected. The positions in the erasure table indicate the reliability of the symbol position in the MPE-FEC frame (for more details, see Sect. 8.5). The 2-bit erasure flags allow to differentiate between four reliability situations.

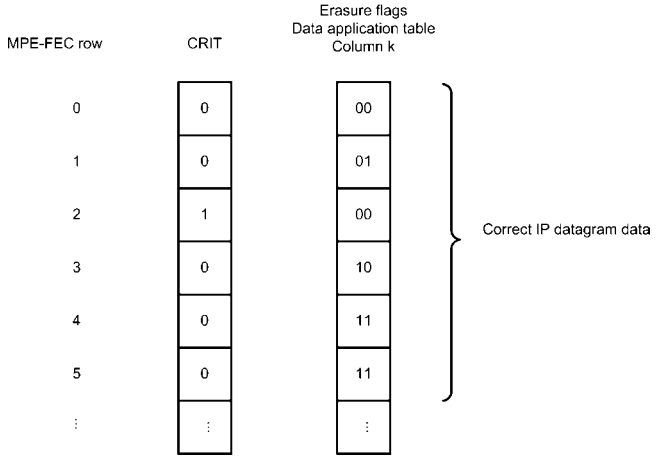

Fig. 8.22 IP datagram recovery using CRIT, IPET, and erasure table

The situation "00" indicates that the symbol position is reliable, whereas all other situations indicate that the symbol position is unreliable. For the situation that the FEC decoder could not correct an MPE-FEC row, the CRIT signals this by means of a "1." Although a particular MPE-FEC row could not be corrected, this does not mean that all symbols of that row are incorrect. By combining the IPET, CRIT, and erasure-flag information, an incorrectly received IP datagram may be recovered from the MPE-FEC frame. The mechanism works as follows. A discontinuity in IPET indicates a region that is unreliable, but this region can contain symbols with a corresponding erasure flag set to "00," signalling that the symbol is correct. The discontinuity region consists of a number of rows, which are either correct or could be corrected by the FEC decoder, or remain defective after FEC. When the CRIT and the erasure-flag information are combined for each defect row of a discontinuity region, each symbol of that region is correct when the CRIT is "1," and the erasure flag is "00," allowing to retrieve a correct IP datagram from a defective MPE-FEC frame. Moreover, the symbols that have an erasure flag value equal to "01" or "10" could also be correct, see Sect. 8.5. For the situation that an incorrectly received IP datagram has symbols constructing the IP header that are correct, but the payload has erasure flags "01" or "10" and the corresponding CRIT has value "1," this IP datagram can be reliably retrieved from the MPE-FEC frame, but it may still be defect. Retrieval of such an IP datagram is advantageous, because the erasure flags may not be correct, leading to another recovered IP datagram. Moreover, if the discontinuity region is larger than the length of this IP datagram, this means that there is at least another incorrectly received IP datagram. This *hidden* IP datagram would otherwise not be visible and definitely be lost. Transmitting possible incorrect IP datagrams will be detected by the IP-stack checksum calculation, preventing them from ending in the application.

8.5 IP De-encapsulation and MPE-FEC Decoding for DVB-H Link Layer

The DVB-H link layer IP de-encapsulation processing has the objective to retrieve in an efficient way one or more IP flows from an Elementary Stream. These efficiency aspects are visible as follows. First, the link layer IP de-encapsulation techniques use received data in such a way that under deteriorating channel conditions, the QoS can be maximized. Second, another efficiency aspect is that link layer processing results in a power and memory friendly way, hence the implementation is also efficient.

Let us briefly explain the IP de-encapsulation process once more. One or more Elementary Streams form a service, see Sect. 8.2.1 for a more elaborate description about the relation between IP streams, IP flows, and Transport Streams. Deriving the requested IP flows from the received Elementary Stream(s) requires a filter-chain as depicted in Fig. 8.14. The filter chain settings are derived from the received SI/PSI information, which is processed by the receiver middleware. The middleware configures the receiver such that the requested IP flows are extracted from the received Elementary Stream(s). For the link layer, the configuration is such that the requested Elementary Stream is *PID* filtered, resulting in MPE and MPE-FEC sections. These sections are forwarded to the IP de-encapsulation filters, which are elaborated extensively thereafter. Furthermore, MPE-FEC decoding is discussed, emphasizing the erasure-flag generation and the usage of the erasure flags in the MPE-FEC decoding. Within a Transport Stream (TS), an Elementary Stream can be uniquely identified by the *PID*, which is a 13-bit field in the TS packet header, for TS packet main header syntax details see Table 8.14. The *PID* filter, as shown in Fig. 8.15b, is a programmable filter that blocks TS packets, which *PID* of which the *PID* values do not match the filter criteria. Every output of the *PID* filter is a sequence of TS packets that have an identical *PID* value, i.e., these packets form an Elementary Stream, see Fig. 8.4. The payload of TS packets belonging to a DVB-H compliant Elementary Stream contains two types of sections: MPE sections and MPE-FEC sections. Each MPE section contains a single IP datagram and an MPE-FEC section carries a single column, having parity data for the MPE-FEC decoding. The *section_length* field and *address* field, indicating the start position of the section payload in the MPE-FEC frame, are present in the header of the MPE sections, see Tables 8.1 and 8.3. The MPE-FEC frame *address* is required if FEC is applied on the received IP data. For this situation, the IP datagrams have to be stored column-wise in the MPE-FEC frame. For checking the integrity of the received data (IP datagram or Reed-Solomon (RS) parity data), sections are provided with a 32-bit Cyclic Reduncancy Check (CRC). The DVB-H Implementation Guidelines [10] suggest to use CRC failures for discarding received data. Accordingly, the corresponding positions in the MPE-FEC frame should be declared as erasures in order to facilitate simpler RS decoding. As indicated earlier, the MPE-FEC sections contain the RS parity data for the extra layer of FEC, which is the FEC Reed-Solomon decoder. Each MPE-FEC section contains one MPE-FEC frame column with parity data. In the MPE-FEC section header, the *section_number* can be used for determining in

Table 8.14 Syntax MPEG-2 Transport Stream main header

Syntax	Number of bits
transport_packet() {	
sync_byte	8
transport_error_indicator	1
payload_unit_start_indicator	1
transport_priority	1
PID	13
transport_scrambling_control	2
adaptation_field_control	2
continuity_counter	4
if (adaptation_field_control == '10' || adaptation_field_control == '11') {	
adaptation_field()	
}	
if(adaptation_field_control == '01' || adaptation_field_control == '11') {	
for (i = 0; i < N; i++) {	
data_byte	8
}	
}	
}	

which column of the MPE-FEC frame the parity data should be placed. For storing the parity data in the MPE-FEC frame, one could also use the *address* field of the *real_time_parameters*, because both fields indicate the start position in the MPE-FEC frame. After receiving the IP datagrams and the RS-parity data, RS decoding is applied when erroneous IP datagrams were received.

8.5.1 Reconstructing the MPE-FEC Frame Under Erroneous Conditions

The IP de-encapsulation method as described in the Implementation Guidelines [10] operates at section level. This means that after calculation of the section CRC, the section payload is either accepted or discarded and erased. In the sequel, the term *Section level IP de-encapsulation* refers to the de-encapsulation method as described in the DVB-H Implementation Guidelines. The discarding of whole sections leads to large erased fragments in the MPE-FEC frame and complicates the reconstruction of the MPE-FEC frame. Therefore, it is advantageous to operate at TS packet level instead of the section level. In literature, several publications [21,30,31,33] support the concept of TS level de-encapsulation. The following facts explain why TS-level IP de-encapsulation has to be preferred over section-level IP de-encapsulation.

A first fact is that not all fragments of a corrupt section have to be erroneous. Discrimination between correct and incorrect fragments is achieved by observing the TS packet field *transport_error_indicator* (TEI), for syntax details see Table 8.14. The TEI flag is a 1-bit flag in the TS packet header that signals whether the RS decoder in the physical layer ([204,188,17] RS) was able to correct the TS packet.

A second fact is that a corrupt TS packet (TEI=1) contains at least 9 byte errors per 204-byte code word. Hence, in the best case, only 9 byte errors occur, and quite some data of the 184-byte payload is useful and actually employed for restoring the original MPE-FEC frame. Both simulations and measurements with an experimental receiver [43] show that under typical channel conditions, often less than 50-Byte errors occur in a corrupted TS packet. In Fig. 8.23 relative distributions

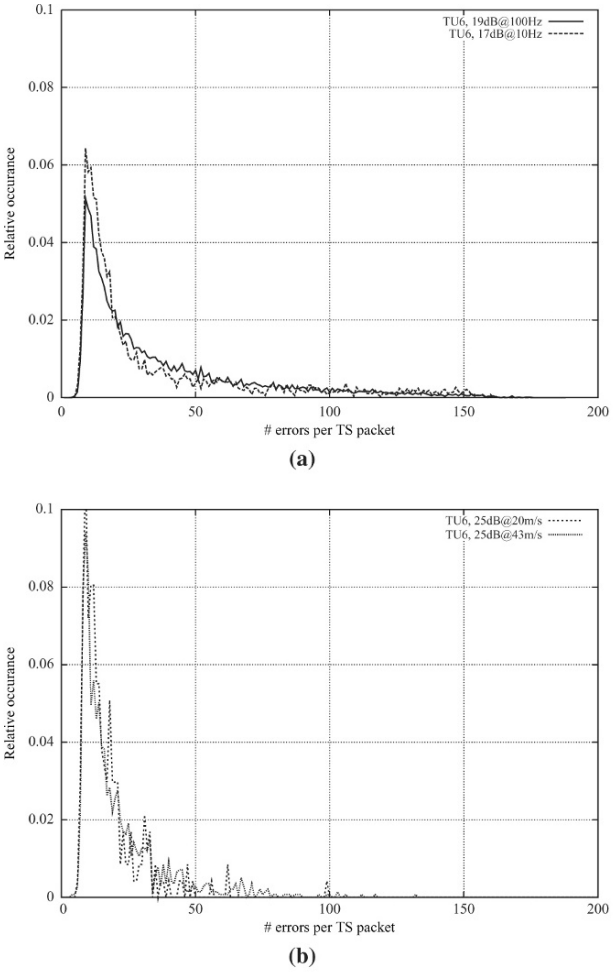

Fig. 8.23 Average number of defect bytes in an erroneous TS packet. (a) Simulated number of byte errors per TS packet. (b) Measured number of byte errors per TS packet.

of byte errors per corrupt TS packet are shown. Fig. 8.23a indicates distributions obtained with computer simulations and Fig. 8.23b depicts distributions obtained with measurements on an engineering sample of a DVB-H receiver. Although the distributions show a high similarity, the measured distributions are somewhat more concentrated to a lower number of errors per corrupt TS packet, an effect that is due to differences in the channel demodulator algorithms, but also from the different channel conditions.

A third fact is that a consistency check on the TS-packet-header *PID* value and *continuity_counter* value, see Table 8.14, is applied to detect whether the payload of a corrupt TS packet is useful. The consistency check involves a comparison of the received *continuity_counter* value of the current TS packet and the expected *continuity_counter* value. The expected *continuity_counter* value is equal to the previous *continuity_counter* value incremented with unity, provided that the *PID* value of the current TS packet is identical to the *PID* value of the previous TS packet. Since not all payload is corrupted, the stored fragment is equipped with an erasure flag of type "unknown," see Table‘8.12 for possible erasure types. Because not all data is corrupted, the previously mentioned erasure flag is handled as a medium-priority erasure flag.

A fourth fact is that the MPE-FEC decoder can correct up to 64 erasures per row. Differentiating in erasure priority can prevent the decoder from being overloaded with erasures. For example, high-priority erasures should always be used by the RS decoder, and medium- and low-priority erasures only when there is sufficient correction capacity available within the boundary of the $2t + e < 65$ correction capacity of the FEC decoder. With 2-bit erasure flags, we can distinguish between missed or discarded fragments (high priority), fragments with some errors (low and medium priority) and fragments that have probably no errors (no priority). The use of multi-level erasure information is also reported in literature [21, 31].

Summarizing, TS packet-level IP de-encapsulation comprises:

- If (TEI=0), place the payload of the TS packet at the appropriate position in the MPE-FEC frame and assign "no priority" erasure information to the corresponding positions.
- If (TEI=1) and the TS packet header is consistent, put the TS-packet payload at the appropriate position in the MPE-FEC frame and assign "medium priority" erasure information to the corresponding positions.
- If (TEI=1) and the TS packet header is not consistent, discard payload and assign "high priority" erasure information to the corresponding positions in the MPE-FEC frame.
- Perform error and erasure decoding of the MPE-FEC frame with ordered granting of erasure information.

The first three techniques are related to extracting IP datagrams, fragments from a TS packet and placement of these fragments at the proper location in the MPE-FEC frame, combined with the erasure assignment, i.e., so-called IP de-encapsulation. In the next paragraph, TS-packet-level IP de-encapsulation is described.

8.5.1.1 TS-Packet-Level IP De-encapsulation

TS-packet-level IP de-encapsulation is based upon a concept in which fragments of IP datagrams are first collected in a kind of scratch memory, prior to a possible fragment placement of the partially reconstructed IP datagram in the MPE-FEC frame. In Fig. 8.24, the concept for TS-packet-level IP de-encapsulation is depicted. The purpose of the following concept is to linearly fill a temporary buffer (scratch memory) with both reliable and unreliable payload of the received TS packets. This filling process can fail, leading to data gaps in the linear address space which hinders proper placement of scratch data in the MPE-FEC frame. As a result, the MPE-FEC frame filling procedure starts at the beginning or at the end of the scratch memory, to maximize the capturing of useful data. The maximum IP datagram size in DVB-H context is 4,080 bytes. A TS packet can contain at most 184 bytes of payload. A fragment is defined as the part of an IP datagram that is contained in one TS packet. The scratch memory containing the fragments of an IP datagram is called fragment memory (scratch memory). Assuming that the fragments are efficiently encapsulated in a TS packet, a maximum-sized IP datagram is distributed over $N = \lceil 4,080/184 \rceil = 23$ fragments. From a received TS packet, the IP fragment is placed in the fragment memory. Using the *continuity_counter* in the TS packet header, one can determine (to a certain degree) the position of the fragment (i.e., the fragment pointer) in the fragment memory. The *continuity_counter* is also used for detecting missed fragments. Note that the *continuity_counter* is a 4-bit counter, so its range is actually too small for a unique identification of fragments of a maximum-sized IP datagram. However, according to [44], the optimal IP datagram size is between

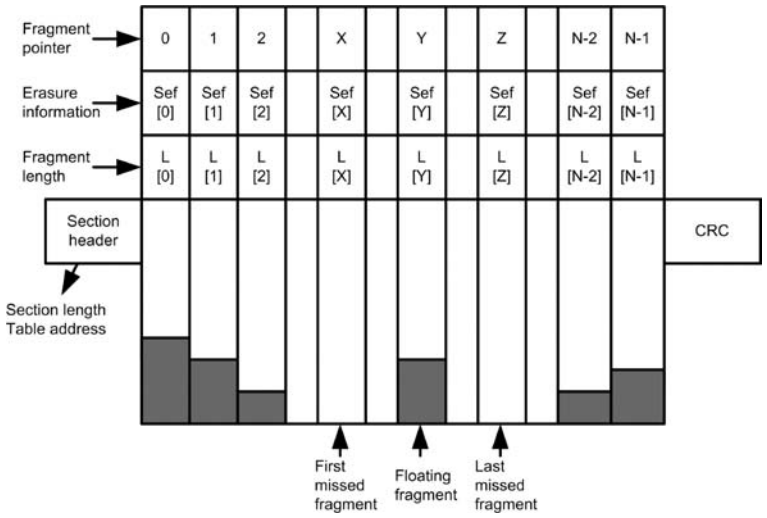

Fig. 8.24 TS level IP de-encapsulation using a fragment memory

the 1,024 bytes and 2,048 bytes. The number of TS packets required to transmit IP datagrams lying in this range fits within the *continuity_counter* range, whose maximum value is 15.

Fragments can vary in length, due to a multiplexing technique called stuffing. Stuffing is a technique for properly aligning data entities like sections, in several TS packets by introducing stuffing bytes (bytes with value "0xFF") in the payload of a TS packet. In case of private sections, two mechanisms can be used for stuffing:

- The first mechanism is based upon adaptation fields. In case of adaptation field stuffing, the stuffing is preceding the actual payload. If *adaptation_field* is used for stuffing, this is signalled in the TS header by the *adaptation_field_control* parameter.
- Another mechanism for stuffing occurs at section level, and thus also applies to MPE- and MPE-FEC sections. In this case, stuffing takes place between the last byte of a section and a new section starting in the next TS packet with a *pointer_field* having value zero. At the decoder, this kind of stuffing can be detected by comparing the section length with the number of extracted bytes. Hence, if the number of already extracted bytes is equal to the section length and the *payload_unit_start_indicator* does not signal the start of a new section, then the remaining bytes of the TS packet payload should be stuffing bytes, thus "0xFF."

As shown in Fig. 8.24, several types of fragments can be distinguished. All these fragment types have to be properly placed in the MPE-FEC frame. For this purpose, the MPE-FEC table address for every individual fragment is determined. The fragments that are received right after the section header can be placed in the MPE-FEC frame, starting from the table *address* that is signalled in the section header. The addresses of succeeding fragments can be determined, either from extrapolating the *address* in the section header together with the section length, or from backward extrapolation, based on the *address* signalled in the next section header. Floating fragments are fragments that are preceded and succeeded by one or more missed fragments. If all fragments (except for the first and maybe the last) have an equal length, one can use address interpolation for the floating fragments.

Since the fragments of IP datagrams originate from different TS packets, they can have different erasure priority information. Fragments that originate from TS packets with TEI = 0 are error-free and will obtain erasure information with the lowest priority level. Usable fragments from TS packets with TEI = 1 will obtain erasure information with medium-priority level. Missed fragments are definitely erroneous, resulting in high-priority level erasure information. Finally, it may happen that all fragments seem to be error-free (TEI = 0 for all TS packets), but that due to mis-correction in the channel decoder, one or more TS packets are mis-corrected. This is detected with the section CRC. The corresponding IP datagram should be assigned with low-priority level erasure information. In Table 8.15, the 4-level erasure assignment is summarized.

Table 8.15 Assignment of 4-level priority erasure's

level	TEI	Other conditions	Symbol state	scope
00	0	CRC=0	Error free	Section
01	0	CRC=1	Mis-corrected TS packet(s)	Section
10	1	'Readable' TS packet	>8 Errors in TS packet	TS packet
11	1	Missed fragment	False	TS packet

8.5.1.2 MPE-FEC Decoding

After the best-effort filling of the MPE-FEC frame, as described in the previous section, MPE-FEC decoding can be started. As stated earlier, error and erasure decoding is applied. The MPE-FEC code is a [255,191,65] Reed-Solomon code, with which one can correct up to t errors and e erasures, provided that $2t + e < d$, where $d = 65$. The assignment of 4-level erasure information is incompatible with a conventional error and erasure RS decoder, in which only two levels of erasure information (reliable versus unreliable) can be handled. Moreover, by assigning erasure information to large data fragments, the risk exists that the number of erasures exceeds the erasure-correction capacity of the RS decoder. For this purpose, the erasure information is rearranged to 2-level erasure information in descending order of priority, a process called "granting of erasure information." Figure 8.25 show the flow-chart for granting erasure information (transforming 4-level into 2-level erasure information). The variables N_i stand for the number of erasures of level i, where a high level corresponds to a high-priority erasure and a low level to a low-priority erasure. The granting procedure shown in Fig. 8.25 implements a method in which erasures of lower priority class are granted only if the number of total erasures does not exceed the erasure correction capacity $d - 1$ or a preselected maximum $p_{\max}$. For the situation that the correcting capacity is not exceeded, MPE-FEC decoding can be applied. The MPE-FEC decoding result is notified to the CRIT in the case that an MPE-FEC frame row could not be corrected, see Sect. 8.4.2. In [31], a similar algorithm for transforming 3-level erasure information to 2-level erasure information is described.

8.5.1.3 Performance of TS-Level IP De-encapsulation

In a simulation, the performance of TS-level IP de-encapsulation and decoding techniques is validated on a Mobile Channel (TU6) [29]. The modulation parameters are: 8K FFT, Guard Interval $\frac{1}{4}$, 16-QAM nonhierarchical and Convolutional Code with R=2/3. The simulated receiver does not use advanced Doppler compensation techniques in the channel demodulator.

In Fig. 8.26, the required average CNR for achieving an MFER of 5% is shown as function of the Doppler frequency on the TU6 channel. The gain in CNR of the TS-level IP de-encapsulation method compared to the section-level IP de-encapsulation

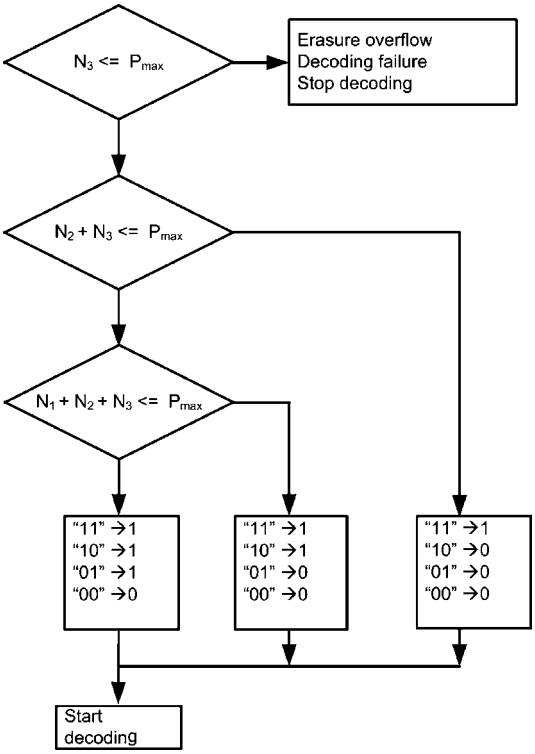

Fig. 8.25 Transformation of multilevel erasure information to bi-level erasure information

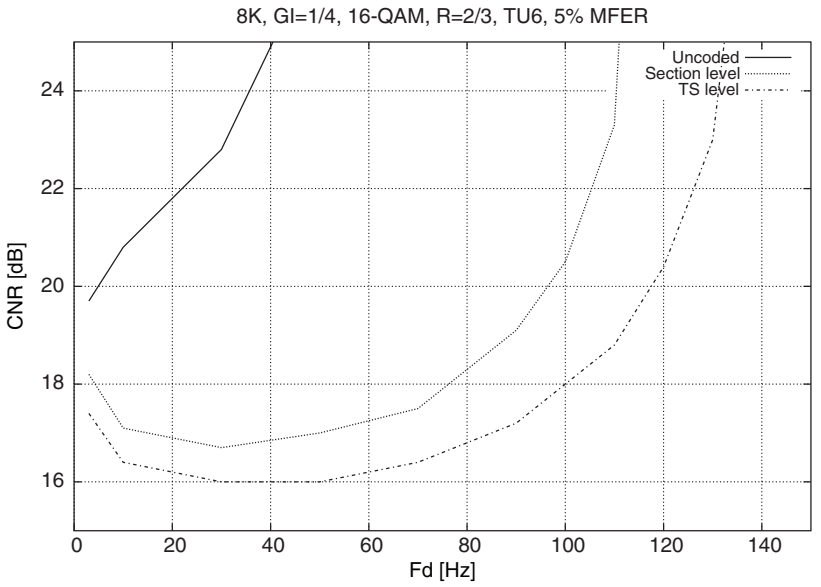

Fig. 8.26 Required CNR for achieving MFER 5%

Table 8.16 Performance figures at TU6 channel for (M)FER 5%

	$CNR_{\min}$ [dB]	$f_{d,3\text{dB}}$ [Hz]	$f_{d,\max}$ [Hz]
Uncoded	20.8	35	–
Section level	17.1	97	125
TS level	16.4	114	141

method ranges from less than 1 dB at 10 Hz to more than 1 dB at 70 Hz. For higher Doppler frequencies, the CNR gain is even higher. The TS-level method also results in a larger Doppler robustness. At an average CNR of 20 dB, we find a difference of approximately 15 Hz between the two curves. The simulation results for $f_d \leq 40$ Hz show that with increasing Doppler frequency, the required average CNR decreases. This phenomenon can be explained by the fact that due to increasing Doppler frequency, more time-interleaving is realized. Time-interleaving is beneficial in the case of rapid changes in channel conditions (e.g., due to motion), so that the duration of bad and good reception conditions is mixed. The advantage of the time-interleaving can be traded off with the limitations of channel-estimation algorithms. The performance criteria $CNR_{\min}$ and $f_{d,3\text{dB}}$ are often used as key performance figures. $CNR_{\min}$ is the required average CNR for achieving the desired Frame Error Rate (FER) (e.g., 5%) at TU6 with 10 Hz Doppler. The parameter $f_{d,3\text{dB}}$ is the maximum Doppler that is allowed for achieving the desired (M)FER with a CNR that is equal to $CNR_{\min}$ + 3 dB. In Table 8.16, these key performance figures are listed. Another performance criterion is $f_{d,\max}$, which is the maximum Doppler that is allowed for achieving the desired FER with a large CNR approaching infinity.

8.5.2 Techniques in the Link Layer for Extra Power Saving

In the DVB-H Implementation Guidelines [10] it is mentioned that in case the application data table is received without errors, MPE-FEC decoding is not applied. Hence, the reception of the RS data table is not necessary anymore. As a consequence, the receiver can be put into sleep mode and wait for the reception of the next burst. At good reception conditions (e.g., error-free reception), a power-saving ratio of up to $64/255 \approx 25\%$ can be realized, in case an [255,191,65] RS code is used. The number 64 in the previous fraction corresponds to the number of columns in the RS data table and 255 in the fraction is the total number of columns in an MPE-FEC frame. However, the requirement of error-free reception of the application table is a requirement that is difficult to satisfy. In practice significant power-savings are realized only if the channel conditions are rather good.

A more promising power-saving method (see also [1, 32]) is based upon the observation that by using an $[n,k,d]$ RS code and applying erasure decoding, one can correct $d-1$ erasures. This means that if k correct symbols are received, decoding of the corresponding word and reconstructing the code word are possible. This

observation may be applied to a DVB-H receiver, where MPE-FEC decoding is then applied after reception of k correct columns, while simultaneously switching off the receiver front-end (tuner and channel demodulator). When k columns are received correctly, the RS decoder can reconstruct the whole MPE-FEC frame by erasing the columns that still had to be received (this is a kind of virtual puncturing). For example, when in the application data table one column is corrupt, only a single correct RS data column is required for reconstruction of an MPE-FEC frame. The state of received columns can be determined by monitoring the section CRC and TS packet headers (*transport_error_indicator* and *continuity_counter*). The decision to switch off the receiver front-end can be further refined. The requirement of receiving at least k correct columns can be transformed into a more relaxed requirement of having (at least) k correct symbols per row. This demands somewhat more bookkeeping, but gives possibly an earlier switch-off time. In order to anticipate on some undetected errors, it is required to have $k+m$ correct columns or at least $k+m$ correct symbols per row. For example, with $m=2$, a single undetected error can be corrected, in addition to the k erasures per row. Summarizing, two criteria for early switch-off of the receiver front-end can be distinguished [32]:

- **Column criterion.** Switch off front-end when $k+m$ correct columns $(m \geq 0)$ are received.
- **Row criterion.** Switch off front-end when all rows have $k+m$ or more correct symbols.

The performance of the extra power-saving methods has been simulated. The channel is modelled by having a TS packet error probability P_{TS}. It is assumed that in a corrupt TS packet all bytes are erroneous such that the payload cannot be used. Furthermore, TS packet errors occur independently from each other. In Sect. 8.5.1.3, it is shown that TS-level IP de-encapsulation has a positive effect on the MFER after MPE-FEC decoding, see Fig. 8.26, as it operates with the lowest CNR. Therefore, it can be expected that the power-saving performance also will improve.

In the simulation, 1,000 MPE-FEC frames are generated and TS packets are corrupted randomly (i.i.d.) using an error probability P_{TS}. From these 1,000 MPE-FEC frames, two histograms are made registering the number of required columns fulfilling each of the two criteria. The histogram is used for calculating the expected average power-saving, which is depicted in Fig. 8.27. The proposed extra power-saving method outperforms the "error-free" method significantly. In case of a TS-packet error probability of $P_{\mathrm{TS}} = 0.01$, no power-saving is realized with the "error-free" method, while the proposed methods achieve a power-saving 15% or more. As expected, the method based upon the row criterion outperforms the method based upon the column criterion. Moreover, applying the TS-level IP de-encapsulation method implicitly improves the power-saving performance. Furthermore, this power-saving improvement from TS-level IP de-encapsulation is larger for the row-based method than for the column-based method. The TS-level IP de-encapsulation method combined with the row-based power-saving method realizes a power-saving ratio of more than 10% over a wide range of TS-packet error probabilities.

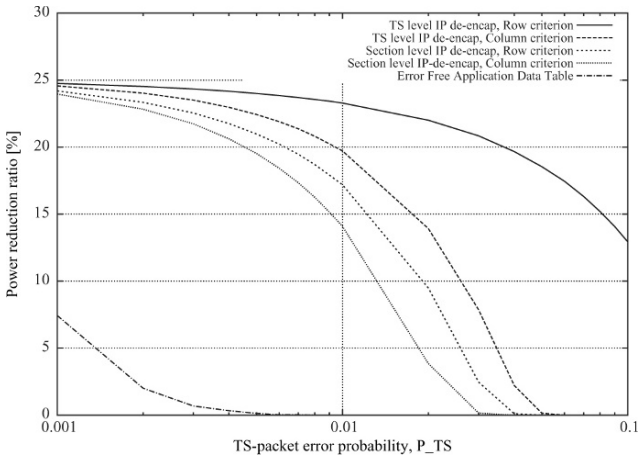

Fig. 8.27 Simulated power-saving

8.6 Verification/Validation of a Robust DVB-H Link Layer

One of the key functions in a DVB-H receiver is the link layer. In order to differentiate on performance, manufacturers can choose between basic implementation, e.g., according to the Implementation Guidelines [10], or advanced link-layer implementation concepts as discussed in Sect. 8.5, resulting in modest or excellent performance. This section presents the DVB-H generator and analyzer concept required for validation and verification of a *robust* DVB-H link layer, as depicted in Fig. 8.13, capable of generating and analyzing extreme signal conditions [6]. The generator and analyzer system aspects are based on the link-layer architecture to be validated. Five functional blocks, MPEG-2 demultiplexer, MPE-FEC decoding, IP-filters, queue management, and link-layer control, are distinguished in the robust DVB-H link-layer architecture, see Fig. 8.13. Four of the five blocks are directly influenced by the incoming TS, whereas the MPE-FEC decoder can only be indirectly influenced. The link-layer functional blocks impose the following system requirements on the DVB-H TS generator:

- **MPEG-2 demultiplexer:** The generator must be able to generate error-free and error-prone TSs, thereby exploring the syntax variations according to [19, 25].
- **MPE-FEC decoding:** This block operates on the MPE-FEC frame, using erasure information derived by the MPEG-2 demultiplexer block.
- **IP filter:** The generator must be able to create error-free or error-prone IPv4 or IPv6 traffic, with uniform or nonuniform destination addresses.
- **Queue management:** The generator must be able to generate error-free or error-prone SI/PSI traffic for average signal conditions up to worst-case timing conditions, maximum section sizes and parallel or consecutive IP-based services based on different MPE-FEC dimensions.

- **Link-layer control:** The generator must be able to generate time-sliced and dispersed elementary streams, either with or without FEC.
- **Reference data:** The generator must deliver the reference data for the analyzer to perform SI/PSI and IP-data comparisons.

The functional blocks of the link layer, see Fig. 8.13, impose the following system requirements on the DVB-H analyzer:

- **Erasure information:** The analyzer must have the knowledge about erasure-location information and its usage by the FEC decoder to determine the FEC decoder result.
- **MPE-FEC frame:** The analyzer must have knowledge about the transmitted MPE-FEC frame.
- **IP-filter settings:** The analyzer must have knowledge of the IP-filter settings to determine which IP datagrams are expected.
- **SI/PSI sections:** The analyzer must have knowledge on all SI/PSI sections, their position in time, the SI/PSI-filter settings, and the DVB-H receiver reception mode.

The DVB-H TS-generator system aspects result in a DVB-H TS-generator concept, in which the error-prone TS generation aspects dominate the DVB-H generator architecture. Verification of the robust link layer as depicted in Fig. 8.13 is to a large extent determined by the MPEG-2 demultiplexer, because of its ability to handle both reliable and unreliable TS packets. The MPEG-2 demultiplexer requires syntax variation to verify robustness, reliable and unreliable data to fill the erasure table, MPE-FEC frame, and IPET. Besides the MPE-FEC decoder, the remaining functional link-layer blocks operate on reliable data, which simplifies the verification.

Let us now discuss the DVB-H generator system aspects. The first system aspect is the syntax variation. For DVB-H, syntax variation is limited to the presence or absence of the *adaptation_field* and the occurrence of a section header split. Basically, a TS header can be succeeded by an *adaptation_field*, see Table 8.14, where the usage of an *adaptation_field* is selected at the IP encapsulator. In DVB-H, the presence of the *adaptation_field* allows stuffing at TS level, instead of section level. A section-header split can occur when the first bytes of a section, which by default are part of the section header, are placed in the payload of Transport Stream packet N and the remaining section header bytes are placed in the payload of Transport Stream packet $N + 1$. Due to bandwidth optimization, all TS payload bytes are used for the transmission of section data, resulting in a possibility of a section header split.

To avoid the development of a link-layer reference model that produces the data reference set, the DVB-H generator is defined such that it provides data from which the reference set can be derived in a low-cost way. This allows the generation of error-free and error-prone streams. As a result, the DVB-H generator will apply a concept called error back-annotation, enabling the injection of errors anywhere in the stream-generation process. Error back-annotation indicates which MPE-FEC frame data is reliable and which data is unreliable. The error back-annotation takes

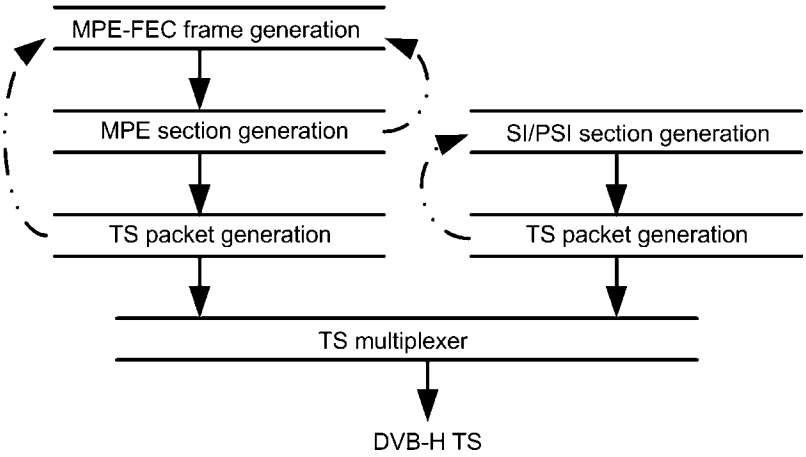

Fig. 8.28 DVB-H generator concept with error back-annotation

into account the IP de-encapsulation scheme of the link layer under test. Error back-annotation is invoked when errors are injected at MPE-section or TS-packet level, or when error expansion occurs. When error back-annotation is applied on SI/PSI data, erroneous sections will not be part of the SI/PSI reference set. Figure 8.28 indicates the stream generation process, where the dashed arrows indicate the error back-annotation.

8.6.1 DVB-H Stream Generator

Let us now discuss an error-prone stream generator allowing an analyzer to verify the link-layer result in an elegant way. At the upper-left side of Fig. 8.28, an MPE-FEC frame is generated. This MPE-FEC frame can be equipped with or without errors. The byte values at erroneous positions in the MPE-FEC frame are not modified, but equipped with a label that is used during the MPE encapsulation and TS packatization process to modify the byte value and finally cause the TS packet-header TEI flag to be set to "1." Error back-annotation will occur for the following situations:

- Injection of errors in the MPE-FEC frame
- Injection of an error at MPE-section level
- Injection of an error at TS-packet level

For the situation that a single error is injected in the MPE-FEC frame, error expansion can occur. Error expansion is caused by the fact that when the erroneous MPE-FEC frame byte is MPE encapsulated and TS packetized, that particular byte will most probably be part of a TS packet containing more than one byte payload. The TS packet that carries the erroneous payload byte(s) shall have the TEI flag set to "1,"

indicating that the TS packet is unreliable.[17] For the situation that the unreliable TS packet contains more than one byte payload, all other payload bytes are signalled as unreliable via the error back-annotation concept. As a result, a single injected error in the MPE-FEC frame causes error expansion in column direction, possibly affecting preceding or succeeding column data. Note that although the other payload bytes are signalled unreliable, they may still be correct, all depending on what the type of error causing the error back-annotation.

For the case that an error is injected at MPE-section level or TS-packet level, error back-annotation is applied taking into account the IP de-encapsulation process of the link layer that is to be tested. For example, if a link layer is implemented using the de-encapsulation method as described in the DVB-H Implementation Guidelines [10], a whole section is discarded when erroneous. Using the error back-annotation concept, all data of the IP datagram encapsulated in that particular MPE section is hard erased. The advantage of using the error back-annotation is that with little effort, various IP de-encapsulation schemes can be validated, without the need to modify the IP de-encapsulation analyzer. Furthermore, the DVB-H generator and associated analyzer allow simulation of various IP de-encapsulation concepts, without the need for an actual implementation.

The back-annotated MPE-FEC frame is the result of error-prone IP encapsulation. The SI/SPI-generation process can be implemented in a simpler way, due to the lack of an additional FEC layer. At the right-hand side of Fig. 8.28, the SI/PSI-generation process is depicted. So when the TS packet containing SI/PSI-section data is corrupted, this is notified to the SI/SPI generator. As a result, the SI/PSI section that is corrupted is not available in the reference set. Each section that is transmitted correctly is equipped with an absolute time-stamp and stored in the reference set. The inserted sections meet the requirements as indicated in [14], involving the time distance between succeeding sections and their corresponding bit rates.

The SI/PSI sections also allow to characterize the receiver wake-up time and power-down time, because of the attached absolute time-stamp. Such a measurement requires a test stream in which sections with a size smaller or equal to the TS-packet payload are inserted in each TS packet. For the situation that the receiver has started too early, sections outside the service burst are received and thereby appear at the output. Similar for the power-down mode, if sections are available in the link-layer output that are transmitted after the end of a service burst, the receiver is active for an unnecessary long period, thereby negatively influencing the power consumption. For this test method, the constraints implied in [14] are not applicable.

8.6.2 DVB-H Stream Analyzer

The analyzer verifies the link-layer-forwarded IP datagrams and SI/PSI sections. This requires the following parameters and data:

[17] Note that in a normal receiver situation, the TEI flag is set by the channel decoder if more than 8 byte errors occur.

- Error back-annotated MPE-FEC frame
- IP filter settings
- Rules for erasure granting
- SI/PSI reference set
- Reception mode, continuous or time-sliced
- Time-slice duty-cycle, and service start-time

The error back-annotated MPE-FEC frame contains all transmitted IP datagrams in reliable format and the corresponding labels of the fragments that will be unreliably received. For the situation that no IP filters are set, all transmitted IP datagrams will be expected, provided that the number of erasures per MPE-FEC row is less than 65. Simply counting the number of erasures (medium and high priority) per row in the error back-annotated MPE-FEC frame indicates which rows can be corrected and thus which IP datagrams can be expected.

When IP postfiltering is applied, the analyzer needs to have the filter values and filter-mask values to reduce the number of expected IP datagrams from the error back-annotated MPE-FEC frame.

For the situation that the number involved erasures exceeds the erasure correction capacity of the RS decoder, erasure granting can be applied. If applied, the DVB-H analyzer needs to be aware of this, and have knowledge of the applied rules.

The section reference set contains the time-stamped correctly transmitted sections. The analyzer uses this reference set to determine which section is to be delivered by the link layer under test. It therefore uses the link-layer reception mode and if operated in time-sliced mode, also the time-slice duty-cycle and the service start-time. Furthermore, it employs an off-set to determine the location of the first service burst in the TS, to extract the sections from the reference set using the absolute time-stamp value.

8.7 Conclusions

When designed for robustness and performance, the DVH-H link layer becomes a quite complex subsystem. The features time-slicing and MPE-FEC added to the DVB-H link layer result in a performance gain when compared to DVB-T. On the one hand, time-slicing reduces the power consumption, while on the other hand, it allows the creation of a virtual receiver, which positively influences the overall system cost. Such a virtual receiver allows, e.g., peeking into neighboring channels during the off-time of the received service, monitoring for possible candidate services, thereby avoiding the need for a second front-end. However, a virtual front-end is essential, due to the cell-based character of a DVB-H network, where for nomadic usage of a DVB-H receiver, horizontal handover is frequently applied without discontinuities in the reception.

The significant reception improvement gained by MPE-FEC can only be achieved when the correct and incorrect received service data is handled properly and the corresponding erasure flags are carefully assigned. At the expense

of 12% extra memory on top of the 255 kbytes required for the MPE-FEC frame of maximum size, a robust and efficient DVB-H link layer is achieved. Such a link layer allows the reception of consecutive service reception for services with maximum MPE-FEC frame size and provides graceful degradation of the received audiovisual information under varying channel conditions. Furthermore, applying intelligent erasure monitoring either on a column or a row basis allows an extra 10% power reduction due to early front-end switch-off.

Fields of future research may be based using the higher OSI layers to enable improved IP de-encapsulation. This is based on the recognition that the OSI layers contain a considerable amount of redundancy. Another field that will provide valuable understanding is the transport-packet burst length. This burst length will strongly depend on the modulation settings and the channel characteristics and will directly influence the IP de-encapsulation performance, perhaps making certain techniques obsolete.

Acknowledgments We wish to acknowledge the NXP Research management for offering us the time and means to write this chapter. In addition, we would like to thank A. Stuivenwold (NXP BL-PES) for putting us on the DVB-H development team. Finally, we would like to thank all the colleagues from various NXP departments involved in the DVB-H project from the cities of Caen, Dresden, Eindhoven, and Nijmegen for their contribution.

References

1. Diego Balaguer de E, Fitzek FHP, Olsen O, Gade M (2005) Performance evaluation of power saving strategies for DVB-H services using adaptive MPE-FEC decoding. IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 4, pp 2221–2226
2. Crawford M (1998) Transmission of IPv6 packets over ethernet networks RFC 2464
3. Deering S (1989) Host extensions for IP multicating RFC 1112
4. Deering S, Hinden R (1998) Internet protocol version 6 (IPv6) specification RFC 2460
5. Eerenberg O, Koppelaar A, Stuivenwold AM, With de PHN (2006) IP recovery in the DVB-H link layer for TV on mobile. International Conference on Consumer Electronics, pp 411–412
6. Eerenberg O, Wendrich P, Orsouw EHW, With de PHN (2007) Efficient validation/verification of a robust DVB-H link layer. International Conference on Consumer Electronics, pp 1–2
7. DVB-H validation task force: final report (2005)
8. Digital Video Broadcasting Project, www.dvb.org (2005)
9. European Telecommunications Standards Institute (ETSI) (2001) Digital video broadcasting (DVB); Allocation of Service Information (SI) Codes for DVB systems European Standard TR 101 162 v1.2.1
10. European Telecommunications Standards Institute (ETSI) (2007) Digital video broadcasting (DVB); DVB-H Implementation Guidelines European Standard draft TR 102 377 v1.3.1
11. European Telecommunications Standards Institute (ETSI) (2005) Digital video broadcasting (DVB); Transmission to Handheld Terminals (DVB-H); Validation Task Force Report European Standard TR 102 401 v1.1.1
12. European Telecommunications Standards Institute (ETSI) (2005) Digital video broadcasting (DVB); IP Datacast over DVB-H: Set of Specifications for Phase 1 European Standard draft TR 102 468 v1.1.1

13. European Telecommunications Standards Institute (ETSI) (2006) Digital video broadcasting (DVB); IP Datacast over DVB-H: Architecture European Standard TR 102 469 v1.1.1
14. European Telecommunications Standards Institute (ETSI) (2006) Digital video broadcasting (DVB); IP Datacast over DVB-H: Program Specific Information (PSI)/Service Information (SI)European Standard TS 102 470 v1.1.1
15. European Telecommunications Standards Institute (ETSI) (2006) Digital video broadcasting (DVB); IP Datacast over DVB-H: Electronic Service Guide (ESG) European Standard TS 102 471 v1.2.1
16. European Telecommunications Standards Institute (ETSI) (2006) Digital video broadcasting (DVB); Specification for Service Information (SI) in DVB systems European Standard EN 300 468 v1.7.1
17. European Telecommunications Standards Institute (ETSI) (2004) Digital video broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television European Standard EN 300 744 v1.5.1
18. European Telecommunications Standards Institute (ETSI) (2004) Digital video broadcasting (DVB); DVB specification for data broadcasting European Standard EN 301 192 v1.4.1
19. European Telecommunications Standards Institute (ETSI) (2004) Digital video broadcasting (DVB); Transmission system for handheld terminals European Standard EN 302 304 v1.1.1
20. Henriksson J (2003) "DXB-X" DVB-Scene magazine, p 7
21. Himmanen H, Hazmi A, Paavola j (2006) Comparison of DVB-H link layer FEC decoding strategies in a mobile fading channel. IEEE 17th International Symposium on Personal, p 1–5
22. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (2001) Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 1: Overview of Local Area Network Standards International Standard 8802-1
23. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (1998) Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control International Standard 8802-2
24. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (1996) Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model International Standard 7498-1
25. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (2000) Information technology – Generic coding of moving pictures and associated audio information: Systems International Standard 13818-1
26. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (1998) Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC International Standard 13818-6
27. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (2001) Information technology – Coding of audio-visual objects – Part 3: Audio International Standard 14496-3
28. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) (2003) Information technology – Coding of audio-visual objects – Part 10: Advanced video coding International Standard 14496-10
29. International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC)(2002) Mobile and portable DVB-T/H Radio access – Part 1: Interface Specification International Standard ISO/IEC62002-1
30. Jokela T, Paavola J, Himmanen H, Ipatov V (2006) Performance analysis of different Reed-Solomon erasure decoding strategies at the DVB-H link layer. IEEE 17th International Symposium on Personal, pp 1–5
31. Joki H, Paavola J (2006) A novel algorithm for decapsulation and decoding of DVB-H link layer forward error correction. IEEE International Conference on Communications, Vol. 11, pp 5283–5288
32. Koppelaar AGC, Eerenberg O (2006) Power-saving by adaptive RS decoding in a DVB-H receiver. Proceedings of the 27th Symposium on Information Theory in the Benelux, pp 9–16

33. Koppelaar AGC, Eerenberg O, Tolhuizen LMGM, Aue V (2006) Restoration of IP-datagrams in the DVB-H link-layer for TV on mobile. International Conference on Consumer Electronics, pp 409–410
34. Kornfeld M , May G (2007) DVB-H and IP datacastBroadcast to handheld devices. IEEE Transactions on Broadcasting, Vol. 53 No. 1, pp 161–170
35. Loncaric S, Grgic S, Zovko-Cihlar B (2006) Minimizing power consumption of the DVB-H receiver. 48th International Symposium ELMAR-focused on Multimedia Signal Processing and Communications, pp 309–313
36. May G (2004) The IP datacast system overview and mobility sspects. IEEE International Symposium on Consumer Electronics, pp 509–514
37. May G (2005) Loss-free handover for IP datacast over DVB-H networks. Proceedings of the Ninth International Symposium on Consumer Electronics, pp 203–208
38. Milosavljevic ZD (2007) A varactor-tuned DVB-H antenna. International Workshop on Antenna Technology: Small and Smart Antennas Metamaterials and Applications, pp 124–127
39. Postel J (1981) Internet protocol – DARPA internet program protocol specification RFC 791. USC/Information Sciences Institute
40. Postel J (1980) User datagram protocol RFC 768. USC/Information Sciences Institute
41. Schulzrinne H, Fokus G.M.D, Casner S, Frederick R, Jacobson V (1996) RTP: A Transport Protocol for Real-Time Applications RFC1889
42. Television on a Handheld Receiver, www.digitag.org (2005) Ver. 1.1
43. Trauschke E (2007) Untersuchungen von Algorithmen zum Reed-Solomon Erasure-Decoding von MPE-FEC Daten nach dem DVB-H Standard unter Berücksichtigung der Eigenschaften des Mobilfunkkanals. Studienarbeit TU Dresden
44. Vadakital VKM, Hannuksela MM, Razaei M, Gabbouj M (2006) Optimal IP packet size for efficient data transmission in DVB-H. Proceedings of the 7th Nordic Signal Processing Symposium, pp 82–85
45. www.ipdc-forum.org (2005)
46. Yang X, Vre J, Owens TJ (2006) A survey of handover algorithms in DVB-H. IEEE Communications Survey 4th Quarter 2006, Vol. 8, No. 4, pp 16–29