7. ANCILLARY DATA SERVICES

The digital television system affords the opportunity to augment the basic television video and audio service with ancillary digital data services. The flexibility of the MPEG-2 transport layer employed in the system allows new digital services to be easily introduced at any time in a completely backward-compatible manner. Sections 7.1 and 7.2 describe two ancillary services that are specified as a part of the Digital Television Standard from the outset. These basic services provide for the transmission of Program Subtitles (PS), Emergency Messages (EM), and Program Guide information. Section 7.3 describes the required elements for new services that may be added in the future and the method of specifying such future services.

7.1 Text-based ancillary services

The Digital Television Standard provides for two text-based services which augment the basic television service. These are Program Subtitles, which are completely analogous to the Closed Caption service transmitted on line 21 of the analog NTSC signal, and Emergency Messages which are mixed into the baseband video signal in the NTSC signal. The ancillary services envisioned are a mix of associated audio services and textual services. The associated audio services are simply transported as separate audio streams, and are not described here because they follow the specifications of the audio service. Similarly, the text-based services are not described here because they are carried as video user bits and follow the specifications of the video service.

7.2 Program guide

The Network PID, identified as Program 0 in the Program Association Table, is available for use by broadcasters to provide information relating to the programs currently being transmitted and also for programs that will be transmitted at future times. As such, it contains the information necessary to provide a program guide in suitably-equipped receivers. In order to implement a program guide, the syntax and semantics of the stream carrying the program information must be specified and standardized. The ATSC is working to have such a standard developed.

7.3 Specification of additional ancillary services

The MPEG-2 transport layer adopted for use in the Digital Television Standard allows new ancillary services to be added to the basic service at any time in the future. To ensure interoperability of service encoders, broadcast multiplexers and switchers, and consumer receivers, it is important that the signal format and delivery characteristics of any new ancillary services that are added in the future be specified completely and unambiguously.

One way to do this would be to specify precisely the encoder and decoder characteristics. A better way, as it provides more latitude in the design of encoders and decoders while still guaranteeing interoperability, is to specify the syntax and semantics of

the transmitted bit stream and to specify an idealized decoder. This is the method that was used by MPEG and it is the method used in the Digital Television Standard.

Section 5.6.2 of Annex C of the Digital Television Standard requires that the standards for any additional ancillary services be specified in terms of a verification model by defining the characteristics of the transmitted syntax and an idealized decoder.

In MPEG-2 parlance, these ancillary services are referred to as private-data. Private data provides a means to add new ancillary services to the basic digital television service specified in this Standard. Private data is supported in two bit stream locations:

1. Private data can be transmitted within the adaptation header of transport packets (Sections 2.4.3.4 and 2.4.3.5 of ISO/IEC 13818-1).
2. Private data can be transmitted as a separate transport stream with its own PID. The contents can be identified as being ATSC private by using the private_data_indicator_descriptor (Section 2.6.29 of ISO/IEC 13818-1) within the PMT.

In either case, it is necessary that the standards which specify the characteristics of such private_streams be consistent with the Digital Television Standard. Standards for private_streams shall specify precisely the semantics of the transmitted syntax as described in Sections 7.3.1 and 7.3.2.

7.3.1 Verification model

The standard shall be specified in terms of a verification model by defining the characteristics of the transmitted syntax and an idealized decoder. In ISO/IEC 13818-1 and 13818-2, this is accomplished by using the T-STD and VBV models, respectively. The elements required for specification by the Digital Television Standard are described in the following Sections.

7.3.1.1 Syntax and semantics

The syntax and semantics of the transmitted bit stream that implements the ancillary service must be completely and unambiguously specified. The decoding process must also be completely and unambiguously specified.

7.3.1.2 Ancillary service target decoder (ASTD)

An idealized decoder model must be precisely defined for the service. Figure 7.1 introduces a concrete model for pedagogic purposes It is modeled after the T-STD.

The salient features of the model are the size of the transport demultiplexing buffer (TB), the minimum transfer rate out of the transport demultiplex buffer (R_{leak}), the required system buffering (BS_{sys}), and optionally the partitioning of BS_{sys} between the smoothing portion and the decoder portion. The decoding process, represented as the decoding times T_decoder(i), must be completely specified. The behavior of the BS_{sys} buffer must be completely modeled with respect to its input process and its output process. Certain parameters such as bit rate, etc., should also be specified.

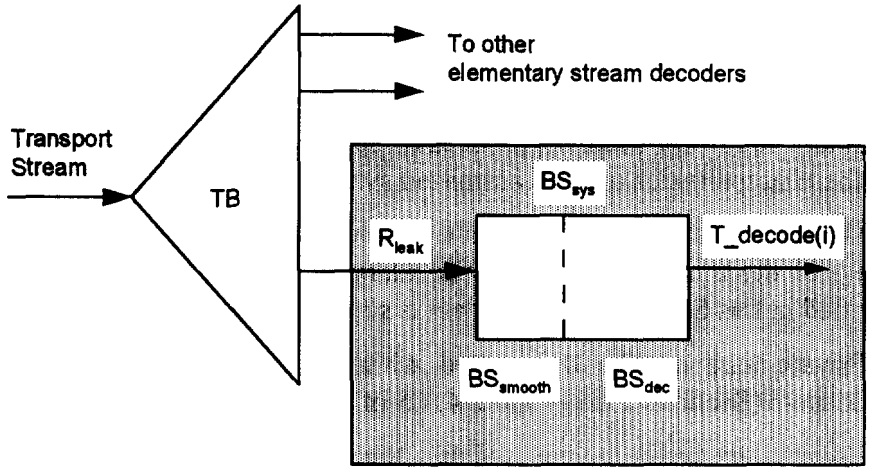

Figure 7.1. Ancillary service target decoder.

7.3.2 Stream type and PMT descriptors

A new ancillary service is described as a program or elementary stream through documented Program Specific Information.

7.3.2.1 Stream type

Several identifiers that are part of the transport section of the Digital Television Standard may be used to identify either the signal or constituent parts thereof; however, the fundamental identifier is the User Private stream type. The stream_type codes are to be unambiguously assigned within the range 0x80 to 0xAF. The AC-3 audio stream is identified by the code 0x81.

7.3.2.2 PMT descriptors

The Ancillary Service specification must include all pertinent descriptors that are found within the Program Map Table. Specifically, it is recommended that either the private_stream_identifier or the registration_descriptor, or both, be included. The value of the 32-bit private_stream_identifier is assigned by the entity specifying the syntax of the Ancillary Service bit stream. Consequently, its value is not guaranteed to be unique. The value of the 32-bit format_identifier in the registration_descriptor is assigned by the ISO-designated registration authority and is guaranteed to be unique. Although inclusion of the private_stream_identifier and/or registration_descriptor is not required for a stream with a unique stream_type code within the Digital Television Standard, it will enhance interoperability in the case where the stream is stored outside this Standard, or transmitted in some other network that has its own set of stream_type codes.

8. SERVICE MULTIPLEX AND TRANSPORT SYSTEMS

8.1 Introduction

This Chapter provides a tutorial description of the functionality and format of the transport system employed in the digital television system. It is intended to aid the reader in understanding and applying the precise specification of the transport layer given in the underlying normative standards documents. The precise transport system standard is based on MPEG-2 and is given in ISO/IEC 13818-1 as constrained and extended for the Digital Television Standard. The constraints and extensions to the MPEG-2 standard that apply are given in Annex C of the Digital Television Standard.

The transport system employs the fixed-length transport stream packetization approach defined by MPEG. This approach to the transport layer is well suited to the needs of terrestrial broadcast and cable television transmission of digital television. The use of moderately long, fixed-length packets matches well with the needs and techniques for error protection in both terrestrial broadcast and cable television distribution environments. At the same time it provides great flexibility to accommodate the initial needs of the service to multiplex video, audio, and data while providing a well-defined path to add additional services in the future in a fully backward-compatible manner. By basing the transport layer on MPEG-2, maximum interoperability with other media and standards is maintained.

8.1.1 Advantages of the fixed-length packetization approach

A transport layer based on a fixed-length packetization approach offers a great deal of flexibility and some significant advantages when attempting to multiplex data related to several applications on a single bit stream. These are described in some detail in this Section.

8.1.1.1 Dynamic capacity allocation

While digital systems are generally described as flexible, the use of fixed-length packets offers complete flexibility to allocate channel capacity among video, audio and auxiliary data services. The use of a packet identifier (or PID) in the packet header as a means of bit stream identification makes it possible to have a mix of video, audio and auxiliary data which is flexible and which need not be specified in advance. The entire channel capacity can be reallocated in bursts for data delivery. This capability could be used to distribute decryption keys to a large audience of receivers during the seconds preceding a popular pay-per-view program, or download program-related, computer software to a "smart receiver."

8.1.1.2 Scalability

The transport format is scalable in the sense that the availability of a larger bandwidth channel may also be exploited by adding more elementary bit streams at the input of the multiplexer, or even multiplexing these elementary bit streams at the second

multiplexing stage with the original bit stream. This is a valuable feature for network distribution, and also serves interoperability with a cable plant's capability to deliver a higher data rate within a 6 MHz channel.

8.1.1.3 Extensibility

Because there will be possibilities for future services that we cannot anticipate today, it is extremely important that the transport architecture provide open-ended extensibility of services. New elementary bit streams could be handled at the transport layer without hardware modification by assigning new packet IDs at the transmitter and filtering on these new PIDs in the bit stream at the receiver. Backward compatibility is assured when new bit streams are introduced into the transport system as existing decoders will automatically ignore new PIDs. This capability could possibly be used to compatibly introduce "1000-line progressive formats" or "3D-HDTV" by sending augmentation data along with the basic signal.

8.1.1.4 Robustness

Another fundamental advantage of the fixed-length packetization approach is that the fixed-length packet can form the basis for handling errors that occur during transmission. Error correction and detection processing (which precedes packet demultiplexing in the receiver subsystem) may be synchronized to the packet structure so that one deals at the decoder with units of packets when handling data loss due to transmission impairments. Essentially, after detecting errors during transmission, one recovers the data bit stream from the first good packet. Recovery of synchronization within each application is also aided by the transport packet header information. Without this approach, recovery of synchronization in the bit streams would have been completely dependent on the properties of each elementary bit stream.

8.1.1.5 Cost effective receiver implementations

A transport system based on fixed-length packets enables simple decoder bit stream demultiplex architectures, suitable for high speed implementations. The decoder does not need detailed knowledge of the multiplexing strategy or the source bit-rate characteristics to extract individual elementary bit streams at the demultiplexer. All the receiver needs to know is the identity of the packet, which is transmitted in each packet header at fixed and known locations in the bit stream. The only important timing information is for bit-level and packet-level synchronization.

8.1.1.6 MPEG-2 compatibility

The transport system is based on the MPEG-2 System specification. While the MPEG-2 System layer has been designed to support many different transmission and storage scenarios, care has been taken to limit the burden of protocol inefficiencies caused by this generality in definition.

An additional advantage of MPEG-2 compatibility is interoperability with other MPEG-2 applications. The MPEG-2 format is likely to be used for a number of other

applications, including storage of compressed bit streams, computer networking, and non-HDTV delivery systems. MPEG-2 transport system compatibility implies that digital television transport bit streams may be handled directly in these other applications (ignoring for the moment the issue of bandwidth and processing speed).

While the transport format conforms to the MPEG-2 Systems standard, it will not exercise all the capabilities defined in the MPEG-2 standard. Therefore, a digital television decoder need not be fully MPEG-2 Systems compliant, in that it will not need to decode any arbitrary MPEG-2 Systems bit stream. However, all MPEG-2 decoders should be able to decode the digital television bit stream syntax at the transport system level. Documents defining the extent to which the MPEG capabilities are supported in this digital television transport specification have been submitted to the MPEG committee and have contributed to the standard. MPEG-2 features not supported in ATSC Standard A/53 are constrained[10] when deemed not applicable to broadcast/cable delivery of digital television.

8.1.2 Overview of the transport subsystem

Figure 8.1 illustrates the organization of a digital television transmitter-receiver pair and the location of the transport subsystem in the overall system. The transport resides between the application (e.g., audio or video) encoding/decoding function and the transmission subsystem. At its lowest layer, the encoder transport subsystem is responsible for formatting the encoded bits and multiplexing the different components of the program for transmission. At the receiver, it is responsible for recovering the bit streams for the individual application decoders and for the corresponding error signaling. (At a higher layer, multiplexing and demultiplexing of multiple programs within a single bit stream can be achieved with an additional system level multiplexing or demultiplexing stage before the modem in the transmitter and after the modem in the receiver.) The transport subsystem also incorporates other higher level functionality related to identification of applications and, as illustrated, synchronization of the receiver. This document will describe these functions in greater detail.

As described earlier, the data transport mechanism is based on the use of fixed-length packets that are identified by headers. Each header identifies a particular application bit stream (also called an *elementary bit stream*) which forms the payload of the packet. Applications supported include video, audio, data, program and system control information, etc. The elementary bit streams for video and audio are themselves wrapped in a variable-length packet structure called the packetized elementary stream (PES) before transport processing. The PES layer provides functionality for identification, and synchronization of decoding and presentation of the individual application. The format and functionality of a PES packet is described in Section 8.5.[11]

---

[10] The constraint takes the form of a limitation of functionality. In this instance, certain flags will be permanently configured, and some fields will not appear in the bit stream. This allows a simpler decoder, as it will not be necessary to handle elements that are not used.

[11] Note that the PES layer is not required for all applications. Its use is required for both the video and audio in the Digital Television Standard.

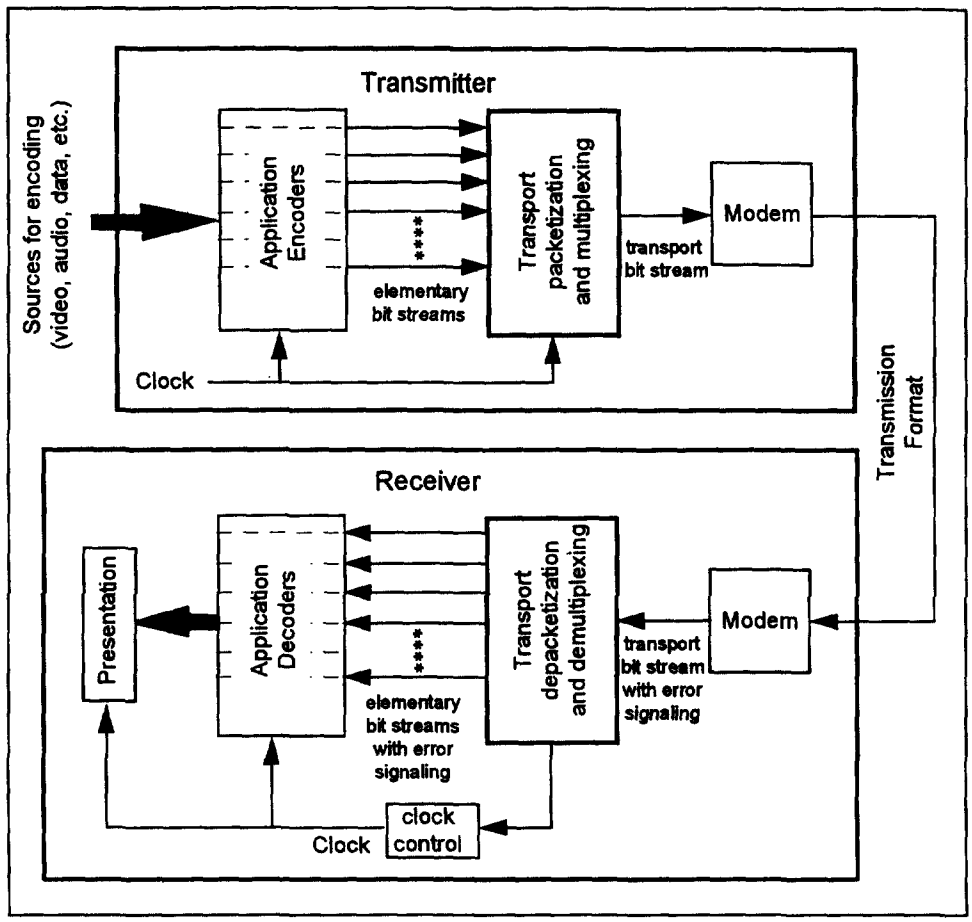

Figure 8.1. Sample organization of functionality in a transmitter-receiver pair for a single digital television program.

Moving up one level in the description of the general organization of the bit streams, elementary bit streams sharing a common time base are multiplexed, along with a control data stream, into *programs*. Note that a program in the digital television system is analogous to a channel in the NTSC system in that it contains all of the video, audio, and other information required to make up a complete television program. These programs and an overall system control data stream are then asynchronously multiplexed to form a multiplexed *system*. The organization is described in detail in Section 8.3.

At this level, the transport is also quite flexible in two aspects:

1. It permits programs to be defined as any combination of elementary bit streams; specifically, the same elementary bit stream can be present in more than one program (e.g., two different video bit streams with the same audio bit stream); a program can be formed by combining a basic elementary bit stream and a supplementary elementary bit stream (e.g., bit streams for scaleable decoders); programs can be tailored for specific needs (e.g., regional selection of language for broadcast of secondary audio); etc.

2. Flexibility at the systems layer allows different programs to be multiplexed into the system as desired, and allows the system to be reconfigured easily when

required. The procedure for extraction of separate programs from within a system is also simple and well defined.

The transport system provides other features that are useful for both normal decoder operation and for the special features required in broadcast and cable applications. These include:

1. Decoder synchronization
2. Conditional access
3. Local program insertion

The elements of these features that are relevant to the Digital Television Standard will be discussed in detail.

The transport bit stream definition directly addresses issues relating to the storage and playback of programs. Although this is not directly related to the transmission of digital television programs, it is a fundamental requirement for creating programs in advance, storing them, and playing them back at the desired time. The programs are stored in the same format in which they are transmitted, i.e., as transport bit streams. The bit stream format also contains the hooks needed to support the design of consumer digital products based on recording and playback of these bit streams, including the use of the "trick modes" that one is familiar with for current analog VCRs. It should be noted that the issues related to storage and playback of digitally compressed video bit streams are quite different from those that apply to analog systems such as NTSC.

8.1.3 General bit stream interoperability issues

Bit stream interoperability at the transport level is an important feature of the digital television system. There are two aspects of interoperability to consider. One is whether the transport bit stream can be carried on other communication systems, and the other is the ability of the system to carry bit streams generated by other communication systems.

In general, there is nothing that prevents the transmission of a bit stream as the payload of a different transmission system. It may be simpler to achieve this functionality in certain systems, e.g., cable television, DBS, ATM, etc., than in others, e.g., computer networks based on protocols such as FDDI, IEEE 802.6, etc., but it is always possible. Because ATM is expected to form the basis of future broadband communications networks, the issue of bit stream interoperability with ATM networks is especially important. ATM interoperability has been specifically addressed in the design of the protocol. This is discussed in more detail in Section 8.5.4.2.

The other aspect of interoperability is the transmission of other, non-television, bit streams within the digital television system. This makes more sense for bit streams linked to television broadcast applications, e.g., cable television, DBS, etc., but is also possible for other "private" bit streams. This function is achieved by transmitting these other bit streams as the payload of identifiable transport packets. The only requirement is to have the general nature of these bit streams recognized within the system context. Note that

there is also a certain minimum system-level processing requirement defined by the Digital Television Standard that needs to be implemented to extract all (even private) bit streams. The details are made clearer in the Sections that follow. It is also important to remember that the system is essentially a broadcast system and hence any private transmissions that are based on a two-way communications protocol will not be directly supported, unless this functionality is provided external to the system definition.

8.2 The packetization approach and functionality

The transport bit stream consists of fixed-length packets with a fixed and a variable component to the header field as illustrated in Figure 8.2.

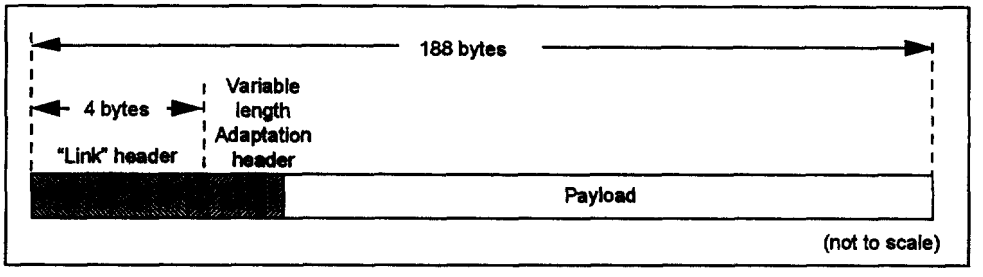

Figure 8.2. Transport packet format.

Each packet consists of 188 bytes and is constructed in accordance with the MPEG-2 transport syntax and semantics. The choice of this packet size was motivated by a few factors. The packets need to be large enough so that the overhead due to the transport headers do not become a significant portion of the total data carried. They should not be so large that the probability of packet error becomes significant under standard operating conditions (due to inefficient error correction). It is also desirable to have packet lengths consistent with the block sizes of typical, block oriented, error correction methods, so that packets may be synchronized to error correction blocks, and the physical layer of the system can aid the packet level synchronization process in the decoder. Another reason for the particular packet length selection is interoperability with the ATM format. The general philosophy is to transmit a single MPEG-2 transport packet in four ATM cells. There are, in general, several approaches to achieve this functionality. Section 8.5.4.2 includes a discussion of one possible approach.

The contents of each packet are identified by the packet headers. The packet header structure is layered and may be described as a combination of a fixed-length "link" layer and a variable-length adaptation layer. Each layer serves a different function similar to the link and transport layer functions in the OSI layered model of a communications system. In the digital television system, this link and adaptation level functionality are used directly for the terrestrial broadcast link on which the MPEG-2 transport bit stream is transmitted. However, in a different communications system, e.g., ATM, the MPEG-2 headers would not play a role in implementing a protocol layer in the overall transmission system. The MPEG-2 headers would be carried as part of the payload in such a case and would continue to serve as identifiers for the contents of the data stream.

8.2.1 The "link" layer

The link layer is implemented using a four byte header field. The format of the header field is described in greater detail in Section 8.4.1. Some of the important functions that are enabled by the header elements are described here.

8.2.1.1 Packet synchronization

Packet synchronization is enabled by the sync_byte, which is the first byte in a packet. The sync_byte has the same fixed, pre-assigned, value for all MPEG-2 bit streams. In some implementations of decoders the packet synchronization function is done at the physical layer of the communication link (which precedes the packet demultiplexing stage), in which case this sync_byte field may be used for verification of the packet synchronization function. In other decoder implementations this byte may be used as the primary source of information for establishing packet synchronization. The standard does not specify the details of the approach to be used to implement this function in a decoder but only provides the hooks in the bit stream to facilitate the function.

8.2.1.2 Packet identification

As discussed earlier, an important element in the link header is a 13 bit field called the PID or Packet ID field. This provides the mechanism for multiplexing and demultiplexing bit streams, by enabling identification of packets belonging to a particular elementary or control bit stream. As the location of the PID field in the header is always fixed, extraction of the packets corresponding to a particular elementary bit stream is very simple to achieve once packet synchronization is established by filtering packets based on PIDs. The fixed packet length makes for simple filter and demultiplexing implementations suitable for high speed transmission systems.

8.2.1.3 Error handling

Error detection is enabled at the packet layer in the decoder through the use of the continuity_counter field. At the transmitter end, the value in this field cycles from 0 through 15 for all packets with the same PID that carry a data payload (as will be seen later, the transport protocol allows the definition of packets that have no data payload). At the receiver end, under normal conditions, the reception of packets in a PID stream with a discontinuity in the continuity_counter value indicates that data has been lost in transmission. The transport processor at the decoder then signals the decoder for the particular elementary stream about the loss of data. This signaling approach is not included in the standard.

Because certain information (such as adaptation headers, time stamps, and program maps) is very important to the smooth and continuous operation of a system, the transport system has a means to increase the robustness of this information to channel errors by providing a mechanism for the encoder to duplicate packets. Those packets that contain important information can be duplicated at the encoder. At the decoder, the duplicate packets are either used if the original packet was in error or are dropped.

Semantics for identifying duplicate packets are described in the description of the continuity_counter.

8.2.1.4 Conditional access

The transport format allows for scrambling of data in the packets. Each elementary bit stream in the system can be scrambled independently. The Digital Television Standard specifies the descrambling approach to be used but does not specify the descrambling key and how it is obtained at the decoder. The key must be delivered to the decoder within a time interval of its usefulness. There is "private" data capacity at several locations within the transport stream where this key data might be carried. Two likely locations are 1) a separate private stream with it's own PID, or 2) a private field within an adaptation header carried by the PID of the signal being scrambled. The security of the conditional access system is ensured by encrypting the descrambling key when sending it to the receiver, and by updating the key frequently. There are no constraints imposed by this Standard on the number of keys that can be used and the rate at which they may be changed. As mentioned before, the key encryption, transmission, and decryption approaches are not a part of the Standard and could differ in different applications of the Digital Television Standard.

The EIA and NCTA have a Joint Engineering Committee which is working to standardize receiver interfaces to implement the key decryption and bit stream descrambling functions. This is described in more detail in Section 8.5.3.

Information in the link header of a transport packet indicates whether the payload in the packet is scrambled and if so, flags the key to be used for descrambling. The header information in a packet is always transmitted in the clear, i.e., unscrambled. The amount of data to be scrambled in a packet is variable depending on the length of the adaptation header. It should be noted that some padding of the adaptation field may be necessary for certain block mode algorithms. Conditional access is discussed in greater detail in Section 8.5.3.

Note that the general MPEG-2 transport definition provides the mechanism to scramble at two levels, within the PES packet structure and at the transport layer. Scrambling at the PES packet layer is primarily useful in the program stream (which is not supported in the Digital Television Standard), where there is no protocol layer similar to the transport layer to enable this function. In this system, scrambling will be implemented only at the transport layer.

8.2.2 The adaptation layer

The adaptation header in the MPEG-2 packet is a variable-length field. Its presence is flagged in the link level section of the header. The functionality of these headers is basically related to the decoding of the elementary bit stream that is extracted using the link level functions. Some of the functions of this layer that are important to the functioning of the digital television system are described here.

8.2.2.1 Synchronization and timing

Synchronization of the decoding and presentation process for the applications running at a receiver is a particularly important aspect of real-time digital data delivery systems such as the digital television system. Because received data is expected to be processed at a particular rate (to match the rate at which it is generated and transmitted), loss of synchronization leads to either buffer overflow or underflow at the decoder, and as a consequence, loss of presentation and/or display synchronization. The problems in dealing with this issue for a digital compressed bit stream are different from those for analog NTSC. In NTSC, information is transmitted for the pictures in a synchronous manner, so that one can derive a clock directly from the picture synchronizing signal. In a digital compressed system the amount of data generated for each picture is variable (based on the picture coding approach and complexity), and timing cannot be derived directly from the start of picture data. Indeed, there is really no natural concept of synchronism between display and transmission (that one is familiar with in NTSC) in a compressed digital television bit stream.

The solution to this issue is to transmit timing information in the adaptation headers of selected packets, to serve as a reference for timing comparison at the decoder. This is done by transmitting a sample of a 27 MHz clock in the program_clock_reference (PCR) field, which indicates the expected time at the completion of the reading of that field from the bit stream at the transport decoder. The phase of the local clock running at the decoder is compared to the PCR value in the bit stream at the instant at which it is obtained, to determine whether the decoding process is synchronized. In general, the PCR from the bit stream does not directly change the phase of the local clock but only serves as an input to adjust the clock rate. Exceptions are during channel change and insertion of local programming. As mentioned earlier, the nominal clock rate in the decoder system is 27 MHz. A point to note here is that the Standard only specifies the means of transmitting synchronization information to a receiver but does not specify the implementation of the sync recovery process. Note also that the audio and video sample clocks in the decoder system are locked to the system clock derived from the PCR values. This simplifies the receiver implementation in terms of the number of local oscillators required to drive the complete decoding process, and has other advantages such as rapid sync acquisition.

Details of the format for the PCR are given in Section 8.4.2.1. Note that in this implementation the encoder and decoder system clocks are set completely independently of the modem clock. This makes for a clean separation of functionality when implementing the two subsystems, and leads to simpler interfaces. This also makes it simpler for the transport system to interface to modems that are used for transmission on other media such as cable television, DBS, computer networks, etc.

8.2.2.2 Random entry into the compressed bit stream

Random entry into the application bit streams such as video and audio is necessary to support functions such as program acquisition and program switching. Random entry into an application is possible only if the coding for the elementary bit stream for the application supports this functionality directly. For example, the video bit stream supports random entry through the concept of intraframes (or I-frames) that are coded without any

prediction, and can therefore be decoded without any prior information. The beginning of the video sequence header information preceding data for an I-frame could serve as a random entry point into a video elementary bit stream. In general, random entry points should also coincide with the start of PES packets where they are used, e.g., for video and audio. The support for random entry at the transport layer comes from a flag in the adaptation header of the packet that indicates whether the packet contains a random access point for the elementary bit stream. In addition, the data payload of packets that are random access points starts with the data that forms the random access point of entry into the elementary bit stream itself. This approach allows packets to be discarded directly at the transport layer when switching channels and searching for a resynchronization point in the transport bit stream, and also simplifies the search for the random access point in the elementary bit stream once transport level resynchronization is achieved.

A general objective is to have random entry points into the programs as frequently as possible, to enable rapid channel switching.

8.2.2.3 Local program insertion

The transport system supports insertion of local programs and commercials, by use of flags and features dedicated to this purpose in the transport packet Adaptation Header. This important issue has been discussed at great length within MPEG. The syntax allows local program insertion to be supported and its performance to improve as techniques and equipment are developed around these syntax tools. The use of these syntax elements will need to be within some imposed constraints to ensure proper operation of the video decoders. There may also need to be constraints on some current common broadcast practices, imposed not by the transport, but rather by virtue of the compressed digital data format.

The functionality of program segment insertion and switching of channels at a broadcast head-end are quite similar, the difference being in the time constants involved in the splicing process, and also in the fact that in program segment insertion the bit stream is switched back to the original program at the end of the inserted segment, while in the channel switching case one most likely switches over to yet another program at the end of the splice. There are other detailed issues related to the hardware implementation that may differ for these two cases, including input source devices and buffering requirements. For example, if local program insertion is to take place on a bit stream obtained directly from a network feed, and if the network feed does not include place-holders for local program insertion, the input program transport stream will need to be buffered for the duration of the inserted program segment. If the program is obtained from a local source, e.g., a video server or a tape machine, it may be possible to pause the input process for the duration of the inserted program segment. Neither of these is an issue for channel switching.

8.2.2.3.1 Systems level view

There are two layers of processing functionality to address when doing local program insertion. The lower layer functionality is related to splicing transport bit streams for the individual elements of the program. The higher level functionality is related to

coordination of this process between the different elementary bit streams which make up the program transport stream. Figure 8.3 illustrates the correct approach to implement program insertion.

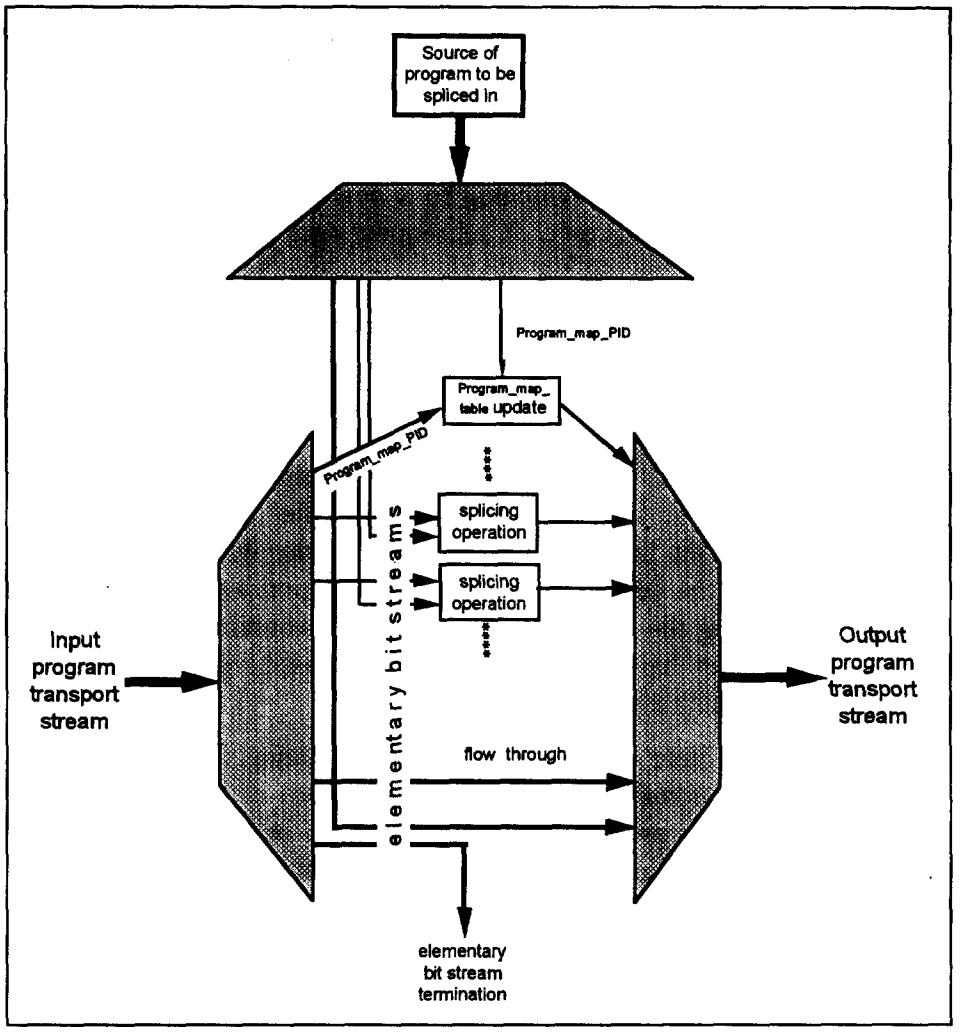

Figure 8.3. Example program insertion architecture.

The first step for program insertion to take place is to extract (by demultiplexing) the packets, identified by the PIDs, of the individual elementary bit streams that make up the program that is to be replaced, including the bit stream carrying the program_map_table. Once these packets have been extracted, as illustrated in Figure 8.3, program insertion can take place on an individual PID basis. If applicable, some packets may be passed through without modification. There is also the flexibility to add and drop elementary bit streams.

8.2.2.3.2 Basics of elementary bit stream insertion

The interface for elementary bit stream insertion is at the transport layer of the protocol. This means that bit stream insertion always takes place in units of transport packets. The primary features enabling local elementary bit stream insertion are the

discontinuity_indicator field and the splice_countdown field in the transport header. The discontinuity_indicator signals the decoder that the PCR is changing to a new time base. This simply informs the decoder that the change in the bit stream is not due to an error in the channel, but rather is intended by the program provider. The implication for the decoder is that it should continue normal decoding, and it is the encoder's responsibility to make sure that the bit stream has been constructed in a compliant manner that will not cause decoders to crash due to overflow or underflow.

The splice_countdown field in the adaptation header is used to signal that a subsequent packet is the point for switching to a new bit stream. The count-down is a positive number which decrements on each subsequent packet of that service.

The Digital Television Standard places some constraints on the transport encoding which are more stringent than the MPEG-2 requirements. One constraint is that a video PES header is followed immediately by a video access unit. This will speed acquisition. The first packet in an insertion will contain the PCR value, with the PCR discontinuity bit set to "1" to inform the decoder that a splice has occurred. The first payload in the stream will begin with a PES Header, which will have a PTS resident, so that the decoder can determine the display time immediately. Because the PES header also has the data_alignment_indicator set, the first data following the header will be the start of the video sequence. Consequently, the decoder has all the information available to begin decoding immediately after receiving the beginning of the spliced program segment. (In general, an MPEG-2 stream does not have these constraints imposed, and hence does not have guaranteed performance at the splice points.)

It is important to note that the process of facilitating frequent opportunities for splicing in a program bit stream is not within the control of the transport layer of the system. The transport only provides the mechanism of implementing the splice itself. Hence the provision of opportunities for inserting program segments requires control over the source coding algorithms used to encode the audio and video.

An important reference for examples of program insertion is contained in Annex L of ISO/IEC 13818-1.

8.3 Higher level multiplexing functionality

As described earlier, the overall multiplexing approach can be described as a combination of multiplexing at two different layers. In the first layer program transport streams are formed by multiplexing one or more elementary bit streams at the transport layer, and in the second layer the program transport streams are combined (using asynchronous packet multiplexing) to form the overall system. The functional layer in the system that contains both this program and system level information is called the PSI or Program Specific Information.

8.3.1 Single program transport multiplex

A program transport stream[12] is formed by multiplexing individual elementary bit streams (with or without PES packetization) that share a common time base. As the elementary streams are multiplexed, they are formed into transport packets and a control bit stream that describes the program (also formed into transport packets) is added. The elementary bit streams, and the control bit stream (also called the elementary stream map in Figure 8.4), are identified by their unique PIDs in the link header field. The organization of this multiplex function is illustrated in Figure 8.4. The control bit stream contains the program_map_table that describes the elementary stream map. The program_map_table includes information about the PIDs of the transport streams that make up the program, the identification of the applications (e.g., audio, video, etc.) that are being transmitted on these bit streams, the relationship between these bit streams, etc. The details of the program_map_table syntax and the functionality of each syntax element are given in Section 8.4.5. The identification of a bit stream carrying a program_map_table is done at the system layers to be described next.

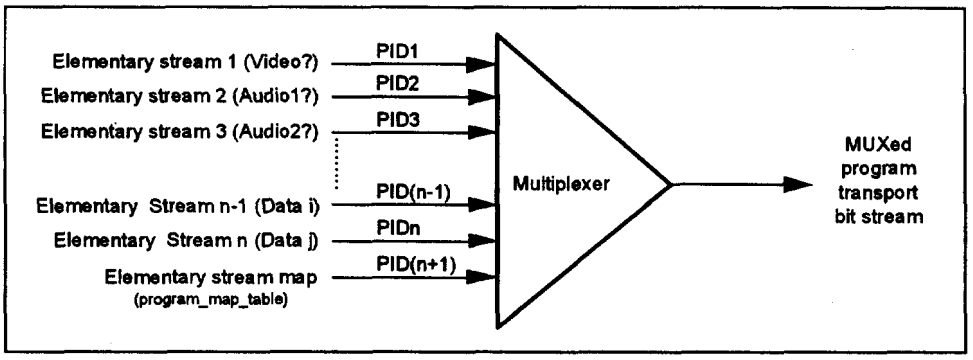

Figure 8.4. Illustration of the multiplex function to form a program transport stream.

The transport syntax allows a program to be comprised of a large number of elementary bit streams, with no restriction on the types of applications required within a program. For example, a program transport stream does not need to contain a single video or audio bit stream; it could be a data "program". On the other hand, a program transport stream could contain multiple related video and audio bit streams, as long as they share a common time base. The data applications that can be carried are flexible; the only constraint is that there should be an appropriate stream_type ID assignment to enable recognition of the application corresponding to the bit stream by a decoder. The list of application types that will be supported in the initial digital television system are given in Chapter 7. Note that the initial selection of applications does not limit the future. (Indeed, it is quite impossible for one to anticipate all possible future applications!)

---

[12] The terminology can be confusing. The term **program** is analogous to a channel in NTSC as discussed in Section 8.1.2. The term **program stream** refers to a particular bit stream format defined by MPEG but not used in the Digital Television Standard. **Program transport stream** is the term used to describe a transport bit stream that has been generated for a program.

Note that, for the different elementary bit streams that make up a program, the link level functions are carried out independently without program level coordination. This includes functions such as PID manipulation, bit stream filtering, scrambling and descrambling, definition of random entry packets, etc. The coordination between the elements of a program is primarily controlled at the presentation (display) stage based on the use of the common time base. This common time base is imposed by having all elementary bit streams in a program derive timing information from a single clock, and then by transmitting this timing information via the PCR on one of the elementary bit streams that constitute the program. The data for timing of presentation is present in the elementary bit stream for each individual application.

8.3.2 System multiplex

The system multiplex is the process of multiplexing different program transport streams. In addition to the transport bit streams (with the corresponding PIDs) that define the individual programs, a system level control bit stream with PID=0 is defined. This bit stream carries the program_association_table that maps program identities to their program transport streams. The program identity is represented by a number in the program_association_table. A program corresponds to what has traditionally been called a channel, e.g., PBS, C-SPAN, etc. The map indicates the PID of the bit stream containing the program_map_table for a program. Thus, the process of identifying a program and its contents takes place in two stages: first one uses the program_association_table in the PID=0 bit stream to identify the PID of the bit stream carrying the program_map_table for the program, and then in the next stage one obtains the PIDs of the elementary bit streams that make up the program from the appropriate program_map_table. Once these steps are completed the filters at a demultiplexer can be set to receive the transport bit streams that correspond to the program of interest. This information can also be used to implement a program guide function in the receiver as discussed in more detail in Section 7.2.

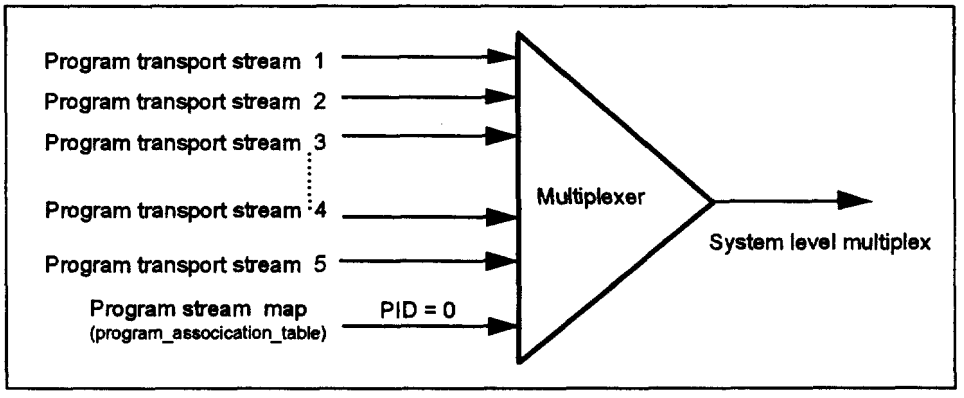

Figure 8.5. Illustration of the multiplex function to form the system level bit stream.

The system layer of multiplexing is illustrated in Figure 8.5. Note that during the process of system level multiplexing, there is the possibility of PIDs on different program streams being identical at the input. This poses a problem because PIDs for different bit streams need to be unique. A solution to this problem lies at the multiplexing stage, where

some of the PIDs are modified just before the multiplex operation. The changes have to be recorded in both the program_association_table and the program_map_table. Hardware implementation of the PID reassignment function in real time is helped by the fact that this process is synchronous at the packet clock rate. The other approach, of course, is to make sure up front that the PIDs being used in the programs that make up the system are unique. This is not always possible with stored bit streams.

Note that the architecture of the bit stream is scalable. Multiple system level bit streams can be multiplexed together on a higher bandwidth channel by extracting the program_association_tables from each system multiplexed bit stream and reconstructing a new PID=0 bit stream. Note again that PIDs may have to be reassigned in this case.

Note also that in all descriptions of the higher level multiplexing functionality, no mention is made of the functioning of the multiplexer and multiplexing policy that should be used. This function is not a part of the Standard and is up to individual designers. Because its basic function is one of filtering, the transport demultiplexer will function on any digital television bit stream regardless of the multiplexing algorithm used.

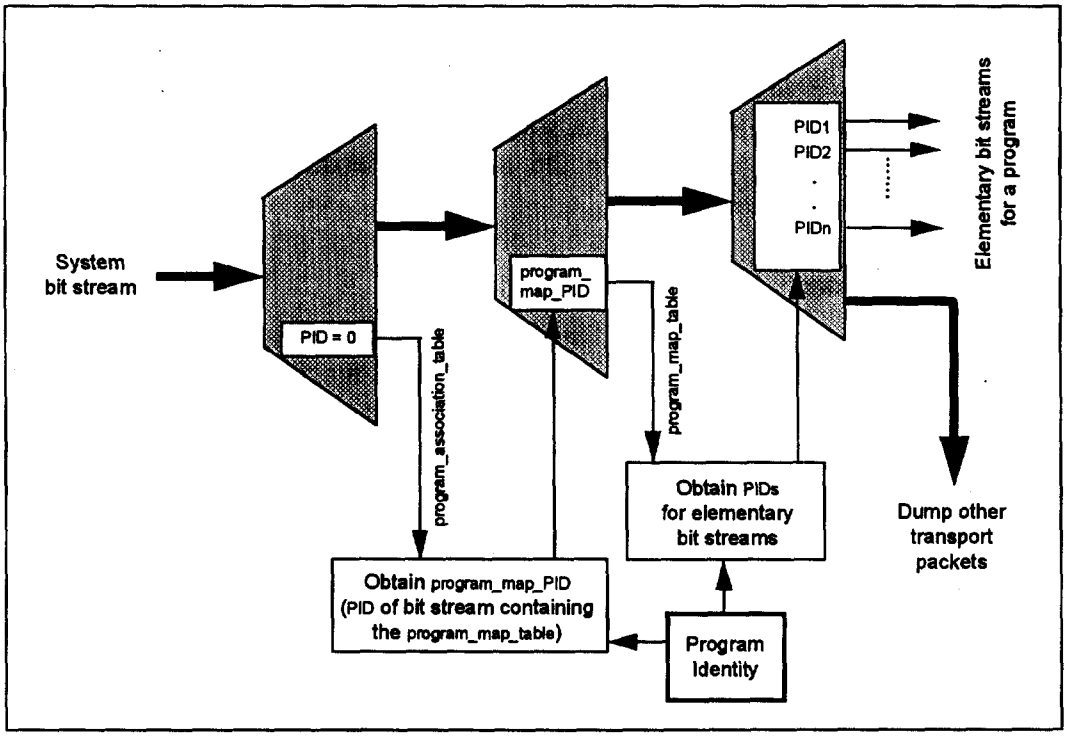

Figure 8.6. Illustration of transport demultiplexing process for a program.

Figure 8.6 illustrates the entire process of extracting elementary bit streams for a program at a receiver. It also serves as one possible implementation approach, although not the most efficient. In practice the same demultiplexer hardware could be used to extract both the program_association_table and the program_map_table control bit streams. This also represents the minimum functionality required at the transport layer to extract any application bit stream (including those that may be private).

Note that once the packets are obtained for each elementary bit stream in the program, further processing to obtain the random entry points for each component bit stream, to achieve decoder system clock synchronization, to obtain presentation (or decoding) synchronization, etc., needs to take place before the receiver decoding process reaches normal operating conditions for receiving a program.

It is important to clarify here that the layered approach to defining the multiplexing function does not necessarily imply that program and system multiplexing should always be implemented in separate stages. A hardware implementation that includes both the program and system level multiplexing within a single multiplexer stage is allowed, as long as the multiplexed output bit stream has the correct properties as described in this document and defined in Annex C of the Digital Television Standard.

It is likely that as the Digital Television Standard is implemented, certain approaches to the multiplexing process will evolve and become standard industry practice. As appropriate, SMPTE or other standardizing organizations may document these approaches and their related interfaces as voluntary standards or recommended practices.

8.4 The transport format and protocol

This Section defines the syntax elements for the transport layer bit stream. All syntax elements need to be recognized at some level in a digital television receiver. Most syntax elements trigger a response in the transport decoder. A few are present for interoperability with MPEG-2.

8.4.1 Link level header

Figure 8.7 shows the link level header with the functionality assigned to each bit. Some of these have been discussed earlier. These general functions may not all be used on the terrestrial broadcast channel, but those that are not used are useful for transmitting the same bit stream over other links, including cable links, computer networks, etc. In short, they provide interoperability features. The remainder of this Section explains the particular usage of these syntax elements.

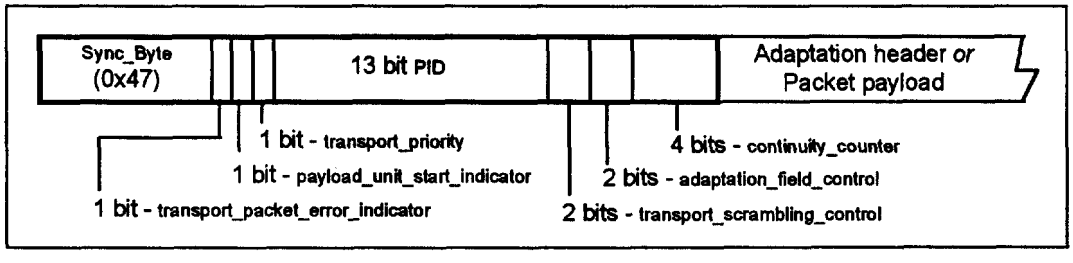

Figure 8.7. Link header format for the transport packet.

The transport_packet_error_indicator is available to signal errors from the modem to transport demultiplexer. If this bit is set the payload of the packet should not be used. As the digital television system does not support prioritized delivery, the transport_priority flag is ignored in the receiver. However this bit could be set during the packetization

process, if the transport layer is also used to simultaneously feed a prioritized delivery system. The payload_unit_start_indicator is used to signal that the packet payload contains the start of a structure which encapsulates the payload data. For audio and video data, this is the PES packet, and for PSI data this is the start of a table.

The adaptation_field control bits signal whether or not an adaptation header field is present and if so, whether or not it is accompanied by a payload in the same transport packet. No adaptation headers are allowed for transport packets containing PSI information other than for signaling with the discontinuity_indicator that the version_number (Section 2.4.4.4 of ISO/IEC 13818-1) may be discontinuous. This restriction allows the PMT bandwidth to be limited to a reasonable level without significantly restricting the opportunities and locations for private data to be transmitted.

The transport_scrambling_control indicators have been defined for the digital television system as shown in Table 8.1.

Table 8.1 Transport Scrambling Control Indicators

transport_ scrambling_ control	Function
00	packet payload not scrambled
01	not scrambled, state may be used as a flag for private use defined by the service provider.
10	packet payload scrambled with "even" key
11	packet payload scrambled with "odd" key

8.4.2 Adaptation level headers

The presence of the adaptation field is signaled by the adaptation_field_control bits of the link level header as described before. The adaptation header itself consists of information useful for higher level decoding functions. The header format is based on the use of flags to indicate the presence of the particular extensions to the field.

The header starts with a fixed-length 2-byte component that is always present (if an adaptation header is transmitted). The format is shown in Figure 8.8.

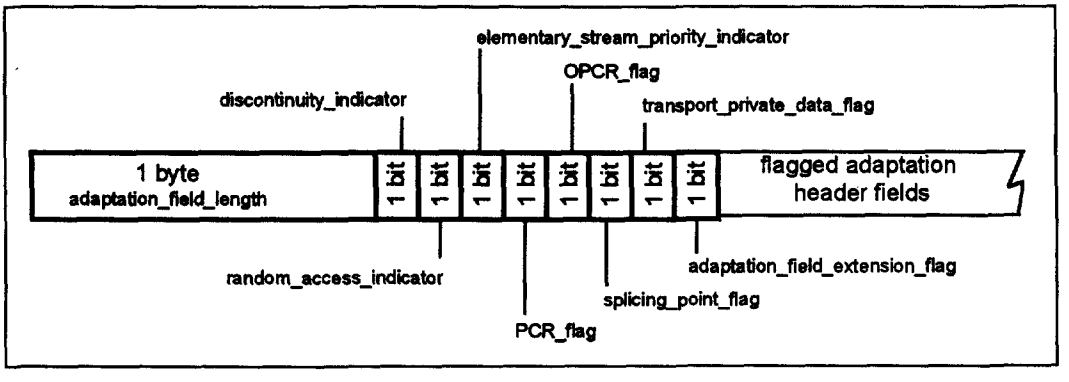

Figure 8.8. Format for the fixed-length component of the adaptation header.

The adaptation_field_length specifies the number of bytes that follow it in the adaptation header. The adaptation header may include stuffing bytes after the last adaptation header component field. (Stuffing bytes have a value of 0xFF and are not interpreted at the decoder.) If stuffing bytes are included, the adaptation_field_length includes them in the count. The value in the adaptation_field_length field can also be used by the decoder to skip over the entire adaptation header, and to directly advance to the data payload in the packet if desired.

The presence of additional adaptation header fields is indicated by the state of the last five single bit flags shown in Figure 8.8 (with a value of '1' indicating that a particular field is present). The three flags at the beginning do not result in extensions to the adaptation header. The elementary_stream_priority_indicator is not used in the system, and, if set, would be ignored by the decoder.

As mentioned earlier, the other components of the adaptation header appear based on the state of the flags shown in Figure 8.8. The order in which these components appear in the bit stream is the same as the order of the flags. Based on the type of adaptation header information being conveyed, the data in these fields may be either fixed-length or variable-length. These fields are described in detail next.

8.4.2.1 The PCR and OPCR fields

The use of the PCR has been described in detail in Section 8.2.2.1. The PCR is used for synchronization of the system decoding process. Under certain circumstances this field may have to be modified during the transmission process. The PCR will be transmitted at least once every 100 milliseconds.

The original program clock reference (OPCR) indicates the intended time of arrival of the last byte of the original_program_clock_reference_base at the target decoder for a single program. This field is not modified during transmission and may be used for recording and playback of single programs. The OPCR is not used by the receiver in the decoding process.

8.4.2.2 The splice_countdown field

This field is useful for local program insertion as described in Section 8.2.2.3.2. The splice_countdown field indicates the number of packets, with the same PID as the current packet, that remain in the bit stream until a splicing point packet. It thus signals to the receiver the point at which a switch from one program segment to another may occur. Similarly, at the end of an inserted program segment, the count in this field signals the receiver when the inserted segment will end and when the original program will resume. The splicing point packet is defined as the packet containing a point in the elementary bit stream from which point onwards data can be removed and replaced by another bit stream, so that the resulting transport bit stream is valid according to MPEG-2 rules.

8.4.3 PSIs and the pointer_field

The program_association_table and the program_map_tables, as mentioned in Section 8.3, describe the organization of a multiplexed bit stream and are a part of the PSI layer of the digital television system. PSI tables, in general, are transmitted in the appropriate bit stream sequentially, without any gap between the tables. This implies that tables do not necessarily start at the beginning of a transport packet. This also implies that in order to decode specific tables, there needs to be some indication of where they begin in the bit stream. This functionality is achieved with the pointer_field. The pointer_field, if present, is the first byte of the payload of a packet (after the link and adaptation headers). The pointer_field is present in the packet if a PSI table begins in the packet, an event which is signaled at the link level, by setting the payload_unit_start_indicator to '1'. The pointer_field indicates the number of bytes that follow it before the start of a PSI table. As an example a pointer_field value of 0x00 indicates that a new PSI table begins immediately following it.

8.4.4 The program_association_table

As discussed in Section 8.3.2, the program_association_table is transmitted as the payload of the bit stream with PID=0 and describes how program numbers that are associated with programs, (e.g., PBS, C-SPAN, etc.) map on to bit streams containing the program_map_tables for these programs. The program_association_table may be transmitted as multiple program_association_segments with each segment having a maximum length of 1024 bytes. The transport decoder can extract individual table segments from the bit stream in whatever order it desires.

The variable-length component of the table consists of program_count number of fixed-length entries corresponding to each program, and stuffing_bytes (to make up the program_association_segment_length).

The program identity '0' is reserved for the network_PID, i.e., the PID of the bit stream carrying information about the configuration of the overall system. As discussed in Chapter 7, the Network PID is available for use by broadcasters to provide information relating to the programs currently being transmitted, and also for programs that will be transmitted at future times. For all other program identities, the program_map_PID is the PID of the bit stream containing the program_map_table for the particular program.

8.4.5 The program_map_table

As discussed in the previous Section, the program_map_table is transmitted as the payload of the bit stream with PID=program_map_PID (as indicated in the program_association_table). The program_map_table carries information about the applications (e.g., video, audio, etc.) that make up programs. The format of the first 8 bytes is the same and has similar functionality as that for the program_association_table. The similarity in format is intended to facilitate simple software decoding of the headers.

8.4.6 Descriptors

To describe certain characteristics of the program or the elementary bit stream, descriptors are transmitted in the program_descriptor and the elementary_stream_descriptor

fields. In general, each program_descriptor and the elementary_stream_descriptor can consist of a number of individual descriptor field elements transmitted sequentially.

Two factors need to be considered in order to use descriptors. In the first place, there has to be a mechanism for indicating the presence of the descriptors. In the PSI tables that have been described, this functionality is achieved by the length field that precedes the descriptor, with a value of zero indicating that no descriptor is present. A second function is the identification of the descriptor itself. This is achieved within the descriptor header, which consists of a one byte descriptor_tag field followed by a one byte descriptor_length field that specifies the number of bytes in the descriptor following the descriptor_length field. The set of valid descriptor_tags is the same as that defined for MPEG-2. Additional descriptor_tags specific to the Digital Television Standard are defined in Section 5.7.2 of Annex C to the Standard.

8.4.7 The PSI paradigms and constraints

The time required to acquire programs is of concern in the digital television system design as NTSC users are accustomed to a rapid response time from the receivers when they switch channels. The approach used to ensure rapid access is a combination of defining a repetition rate for transmission of different PSI bit stream elements and assigning PID values to bit streams that are related to the program number under consideration. The program paradigms are such that recognition of the paradigm helps speed up the acquisition of programs in the receiver. A receiver that is unaware of the program paradigms will also be able to acquire the program, except that it could take a longer time to complete the acquisition process.

8.4.7.1 The program paradigms

The program paradigms provide a simple mechanism for quickly recognizing the programs of interest and further identifying the transport bit streams containing elements of the program thus enabling quick acquisition. The basic philosophy is to select PIDs for the transport bit stream that are related to a program number. The program number within a single transport stream serves as the complete identification of the program.

Television programs can only be assigned program numbers 1 to 255. Non-television programs are also not allowed to use these particular program numbers. This allows the receiver to filter the television programs quickly by looking at the MSB of the program number. This process is further aided by specifying television programs to be described only within section 0 of the program_association_table.

We further define:

base_PID= program number* 0010h.

In pseudo C code:

```
base_PID = program number << 4
```

The paradigm to identify the transport bit streams containing certain elements of the program is defined as shown in Table 8.2. Table 8.3 illustrates an example for program 52.

Table 8.2 Transport Bit Streams

Name	PID Definition	Description
PMT_PID	base_PID+0x0000	PID for the bit stream containing the program_map_table for the program.
Video_PID	base_PID+0x0001	PID for the bit stream containing the video for the program.
PCR_PID	base_PID+0x0001	Implies the video bit stream also carries the PCR values for the program
Audio(A)_PID	base_PID+0x0004	PID for the bit stream containing the primary audio for the program. This may be either a complete main audio service (CM) as defined in Section 6.3 of Annex B or a main audio service, music and effects (ME) as defined in Section 6.4 of Annex B.
Audio(B)_PID	base_PID+0x0005	When the Audio (A) bit stream contains a complete main audio service (CM), the Audio(B)_PID shall not appear in the transport stream. When the Audio (A) bit stream contains a main audio service, music and effects (ME), the Audio (B) bit stream contains the principle program dialogue (D) associated with the ME service carried by Audio (A).
Data_PID	base_PID+0x000A	PID for the bit stream containing the data for the program.

Table 8.3 Example for Program 52 (0x0034)

Name	PID value
base_PID	0x0340
PMT_PID	0x0340
Video_PID	0x0341
PCR_PID	0x0341
Audio(A)_PID	0x0344
Audio(B)_PID	0x0345
Data_PID	0x034A

This paradigm enables immediate access to services defined in the paradigm once the program number is known. The program_map_table must be decoded to obtain the PIDs for services not defined by the paradigm but included within the program (such as additional audio service types or a second data channel). Because it is intended that the audio bit streams carried by Audio (A) and Audio (B) will be simultaneously decoded and added together in the receiver, only one audio service, either a single complete main (CM) channel or a main, music and effects (ME) plus one dialogue (D) channel, can be defined within the program paradigm. Access to all other audio services, for example, a second complete main (CM) in another language or the dialogue (D) channels carrying languages other than the principle language, will require that the program_map_table be decoded to obtain the PIDs for these services. According to the program paradigm, every 16th PID is a PMT_PID and may be assigned to a program. If a PMT_PID is assigned to a program by the program paradigm, the next 15 PIDs after that PMT_PID are reserved for elements of that program and shall not be otherwise assigned.

Additional constraints include:

1. Only one program is described in a PSI transport bit stream corresponding to a particular PID value. A transport bit stream containing a program_map_table may not be used to transmit any other kind of PSI table (identified by a different table_id).
2. Adaptation headers shall not occur in transport packets of the PMT_PID for purposes other than for signaling with the discontinuity_indictor that the version_number (Section 2.4.4.5 of ISO/IEC 13818-1) may be discontinuous.

8.4.7.2 Repetition rates

The maximum spacing that is allowed between occurrences of section 0 of the program_association_table is 100 ms.

The maximum spacing allowed between occurrences of a program_map_table containing television program information is 400 ms.

8.5 The PES packet format

The PES Packet may be generated by either the application encoder or the transport encoder; however, for the purposes of explanation the PES encoding is assumed to be a transport function. As described before, some elementary bit streams, including the compressed video and compressed audio, will go through a PES layer packetization process prior to transport layer packetization. The PES header carries various rate, timing, and data descriptive information, as set by the source encoder. The PES packetization interval is application dependent. The resulting PES packets are of variable-length with a maximum size of 2^{16} bytes, when the PES packet length field is set to its maximum value. This value is set to zero for the video stream, indicating that the packet size is unconstrained and that the header information cannot be used to skip over the particular PES packet. Note also that the PES packet format has been defined to also be of use as an input bit stream for Digital Storage Media (DSM) applications. Although the DSM format will not be used for broadcast applications, some of the PES header fields related to the DSM functions are also described in this Section. Note that the ability to handle input bit streams in the DSM format is not essential for a receiver, but may be useful for VCR applications.

Note that the format for carrying the PES packet within the transport layer is a subset of the general definition in MPEG-2. These choices were made to simplify the implementation of the digital television receiver and to also help error recovery.

A PES packet consists of a PES_packet_start_code, PES header flags, PES packet header fields, and a payload (or data block), as shown in Figure 8.9. The payload is created by the application encoder. The packet payload is a stream of contiguous bytes of a single elementary stream. For video and audio packets, the payload is a sequence of access units from the encoder. The access units correspond to the video pictures and audio frames.