

MPEG-2 : A TUTORIAL INTRODUCTION TO THE SYSTEMS LAYER

P A Sarginson

The MPEG-2 Systems Specification [1] describes how MPEG-compressed video and audio data streams may be multiplexed together to form a single data stream. This paper aims to introduce the jargon and the fundamental principles of the MPEG-2 Systems layer.

The Basics

Programme - Within a broadcast context, a 'programme' is usually taken to mean one of the many entertainment entities shown by a broadcaster during the course of a day such as "The 9 O'clock news" or "Eastenders". Confusingly, MPEG uses the word programme to mean a single broadcast service or channel. So, BBC1 is a programme as far as MPEG is concerned.

Elementary Stream - A programme comprises one or more elementary streams. An elementary stream is the name given to a single, digitally coded component of a programme, for example coded video or audio. The simplest type of programme is a radio service comprising a single audio elementary stream. A traditional television broadcast comprises three elementary streams; one carrying coded video, one carrying coded stereo audio and one carrying teletext data. Future television broadcasts are certain to comprise many more elementary streams. For example, there may be additional video elementary streams carrying the same picture at alternative resolutions and there may be a choice of audio and teletext elementary streams each in a different language.

The MPEG-2 Systems layer describes how elementary streams comprising one or more programmes are multiplexed together to form a single data stream suitable for digital transmission or storage.

The output of an MPEG-2 multiplexer (see figure 1) is a contiguous stream of 8-bit wide data bytes. There are no constraints on the data rate but clearly it must at least equal the total combined data rate of the contributing elementary streams. The multiplex may be fixed or variable data rate and may contain fixed or variable data rate elementary streams.

In addition to the elementary streams, a variety of additional information may be included in the multiplex:

- A system of time stamps is specified ensuring that related elementary streams are replayed in synchronism at a decoder.
- Tables of service information may be included detailing network parameters, details of the programmes within the multiplex and the nature of the various elementary streams.
- Support is provided for the control of scrambling for conditional access applied to one or more of the elementary streams though it should be noted that neither the scrambling algorithm nor the access control is specified by MPEG.
- Any number of additional 'private data' channels may be accommodated. Private data is a data stream whose content is not specified by MPEG. Such data streams may be used to carry data services such as teletext, additional service information specific to a particular network, commands intended to control modulation and network distribution equipment and any other type of data required by a particular application.

Peter Sarginson is with the BBC Research and Development Department, Kingswood Warren

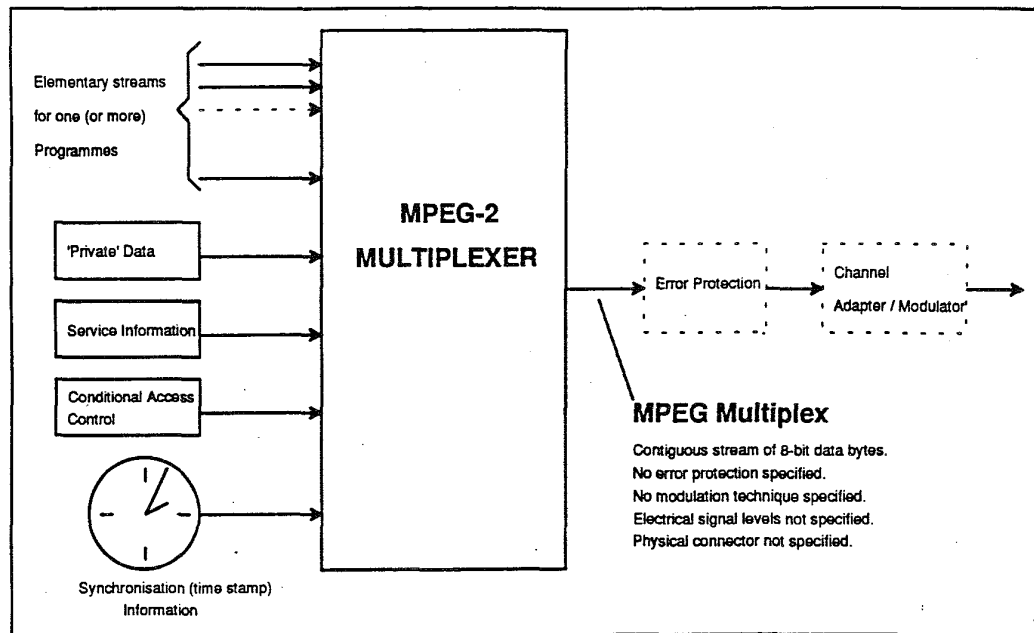


Figure 1 - An MPEG-2 multiplexer

There are some notable multiplex characteristics not specified by MPEG:

- There is no form of error protection within the multiplex. Error protection and the subsequent modulation of the MPEG-2 multiplex is chosen to suit the characteristics of the channel or storage medium and is not specified by MPEG.
- There is no electrical or physical specification for the MPEG-2 multiplex so a designer may use whatever signal levels or connector types best suit the application.

Programme Streams and Transport Streams

The MPEG-2 Systems Specification defines two alternative multiplexes. One is called a 'transport stream', the other is called a 'programme stream'. Each is optimised for a different set of applications.

The programme stream is based on the established MPEG-1 multiplex. Like the MPEG-1 multiplex, it can accommodate a single programme only and is intended for the storage and retrieval of programme material from digital storage media.

A programme stream is intended for use in error-free environments because it is rather susceptible to errors. There are two reasons for this. Firstly, the programme stream comprises a succession of relatively long and variable length packets. Each packet begins with a packet header. An error occurring in the packet header may cause the loss of the entire packet. As programme stream packets may contain many kilobytes of data, the loss of a single packet may represent the loss or corruption of an entire video frame. Secondly, the variability of the packet lengths means that a decoder cannot predict where one packet will finish and a new one will start. Instead it must read and interpret the packet length field contained in each packet header. If this packet length field is corrupted by an error the decoder will lose synchronism with the stream, again resulting in the loss of at least one packet.

The transport stream was devised for multi-programme applications such as broadcasting and a single transport stream can accommodate many independent programmes. It comprises a succession of 188 byte long packets called 'transport packets'. The use of short, fixed-length packets means that the transport stream is not as

susceptible to errors as the programme stream. Further, each 188-byte long packet is easily given additional error protection by processing it through a standard error protection process such as Reed-Solomon encoding. The improved error resilience of the transport stream means that it has a better chance of surviving the error-prone channels to be found, for example, in a broadcast environment.

It might seem that the transport stream is clearly the better of the two multiplexes with its increased error resilience and ability to carry many simultaneous programmes. However, it should be remembered that the transport stream is a more sophisticated multiplex than the programme stream and is consequently more difficult to create and to demultiplex. It is also less like the well-established MPEG-1 multiplex.

Presentation Units and Access Units

Figure 2 shows an uncompressed digital video sequence being MPEG coded to a target data rate of 5 Mbit/s. Each picture in its uncompressed form is termed a 'presentation unit'. The coder compresses each presentation unit to give a coded picture which is termed an 'access unit'. Note that video access units are not all the same size. Their size depends on whether they represent an 'I', 'P' or 'B' picture (see [2]) and also on how difficult the picture was to code.

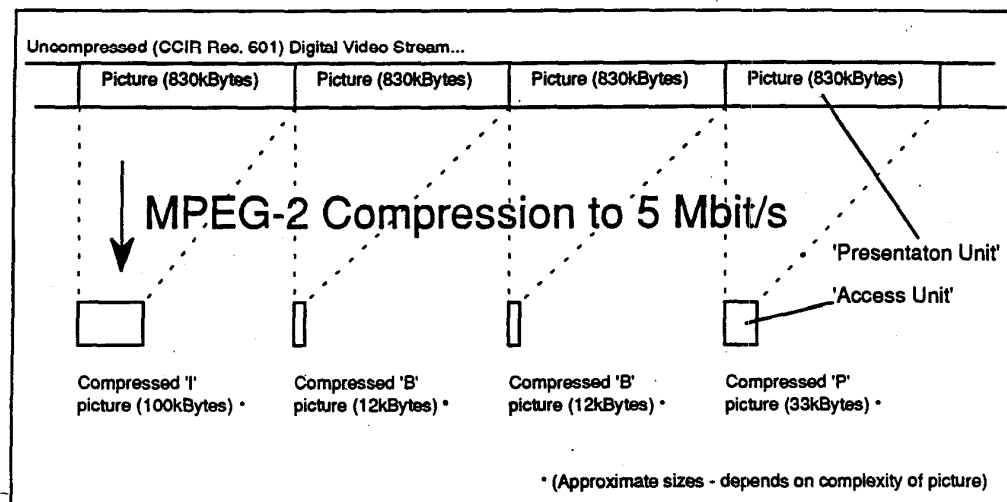


Figure 2 - The compression of 'presentation units' to yield 'access units'

The result of MPEG encoding a video sequence is a succession of video access units and it is this succession of access units that comprises the video elementary stream.

Similarly, the result of MPEG encoding audio is a succession of audio access units and it is this succession of access units that comprises an audio elementary stream. Each audio access unit typically contains a few tens of milliseconds of compressed audio.

Packetised Elementary Streams

The next stage in the creation of either MPEG-2 multiplex is to convert each elementary stream into a 'Packetised Elementary Stream' (PES). A Packetised Elementary Stream consists entirely of 'PES-packets' as shown in figure 3.

A PES-packet consists of a header and a payload. The payload simply consists of data bytes taken sequentially from the original elementary stream. There is no requirement to align the start of access units and the start of

PES-packet payloads. Thus a new access unit may start at any point in the payload of a PES-packet and it is possible for several small access units to be contained in a single PES-packet.

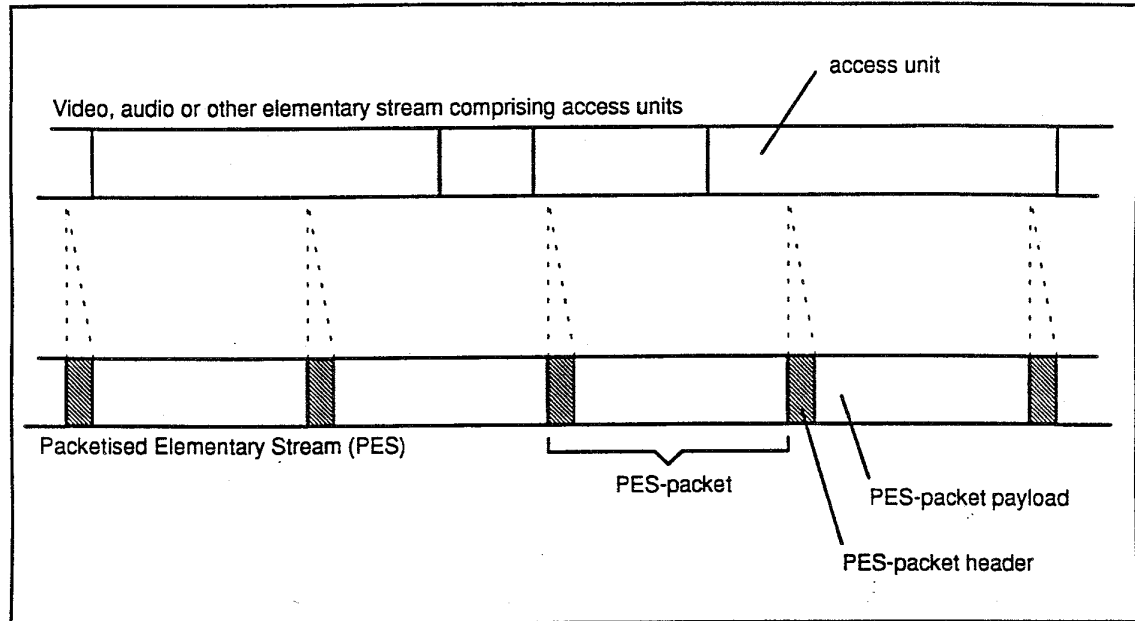


Figure 3 - Conversion of an elementary stream to a Packetised Elementary Stream

PES-packets may be of variable length subject to a maximum length of 64 kBytes. (There is one exception to this and that is in the case of a video packetised elementary stream when carried in a transport stream where PES-packets may be of unbounded length.) The designer of an MPEG-2 multiplexer may choose to exploit this flexibility in a number of ways. Possibilities include simply using a fixed PES-packet length or varying the length in order to align the start of access units with the start of PES-packet payloads.

Figure 4 shows the fields comprising a PES-packet header. The first four bytes (i.e. the 'PES-packet start code prefix' plus the 'stream_id') comprise the PES-packet start code. This combination of 32 bits is guaranteed not to arise in the packetised elementary stream other than at the start of a PES-packet. (There is a general exception to this; the contents of private data is not constrained by MPEG and as such it may contain emulations of the PES-packet start code).

The stream_id byte distinguishes PES-packets belonging to one elementary stream from those of another within the same programme. MPEG specifies the permitted values for this field including 32 available for audio elementary streams and 16 for video elementary streams.

Flags 1 and Flags 2 are 'indicator bits' which indicate the presence or absence of the various optional fields that may be included in a PES-packet header. These optional fields convey additional information about the packetised elementary stream such as whether scrambled or not, relative priority, copyright information, and an optional error check field for the packet.

Of particular importance are the two most significant bits of 'flags 2' marked 'P' and 'D' in figure 4. When set, these indicate the presence of a 'Presentation Time Stamp' (PTS) field and a 'Decoding Time Stamp' (DTS) field respectively within the PES-packet header. Time stamps are the mechanism provided by the MPEG-2 systems layer to ensure correct synchronism between related elementary streams at a decoder.

The 'PES header data length field' is the last of the mandatory bytes in the PES-packet header. Its value gives the number of bytes of optional PES-packet header data present in the PES-packet header before the first byte

of the PES-packet payload is reached. (There are 25 optional PES-packet header fields which between them may total nearly 200 bytes of additional data).

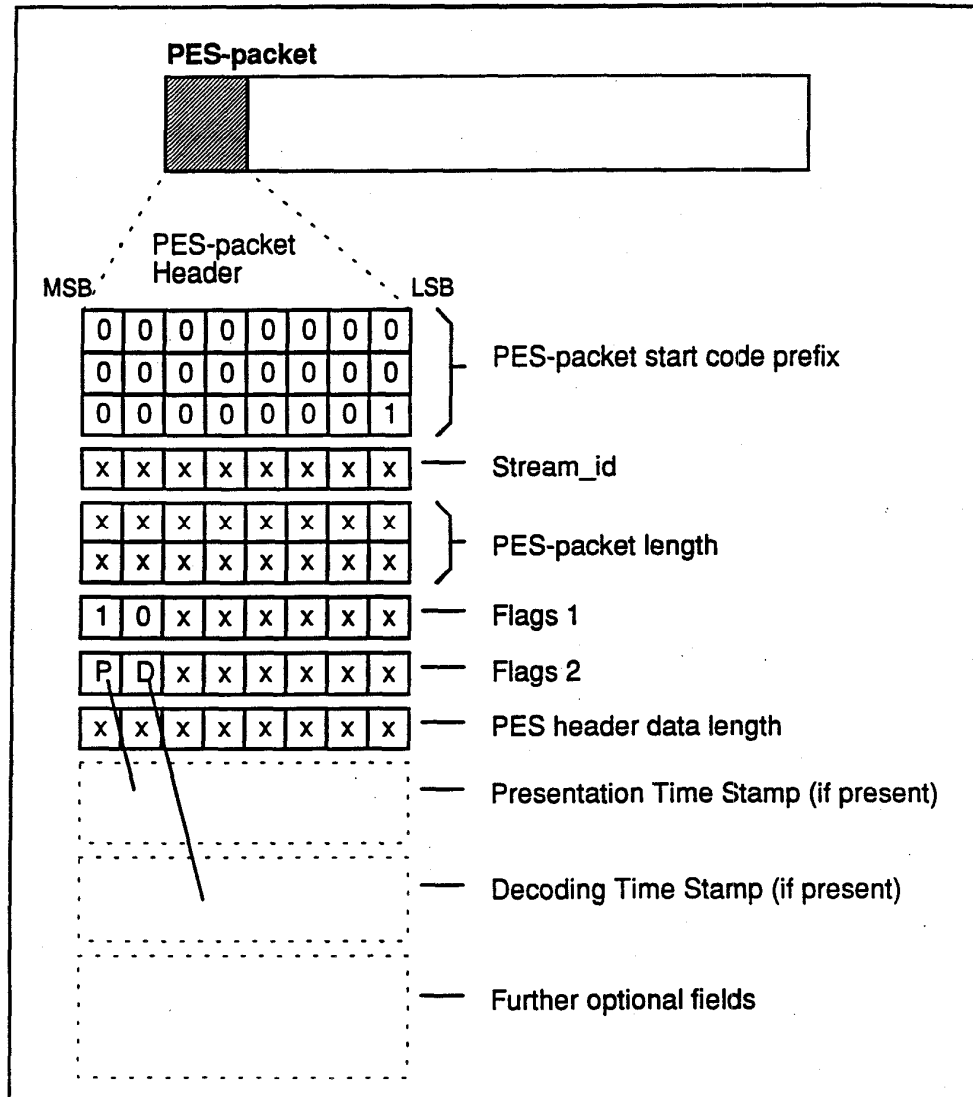


Figure 4 - A PES-packet header

The Programme Stream multiplex

In a programme stream, PES-packets derived from the contributing elementary streams are organised into 'packs' (see figure 5). A pack comprises a 'pack-header', an optional 'system-header' and any number of PES-packets taken from any of the contributing elementary streams in any order. There is no constraint on the length of a pack except that a pack header must occur at least every 0.7 seconds within the program stream as the pack header contains important timing information. The system header contains a summary of the characteristics of the programme stream such as its maximum data rate, the number of contributing video and audio elementary streams and further timing information. A decoder may use the information contained in a system header to determine whether it is capable of decoding the programme stream or not.

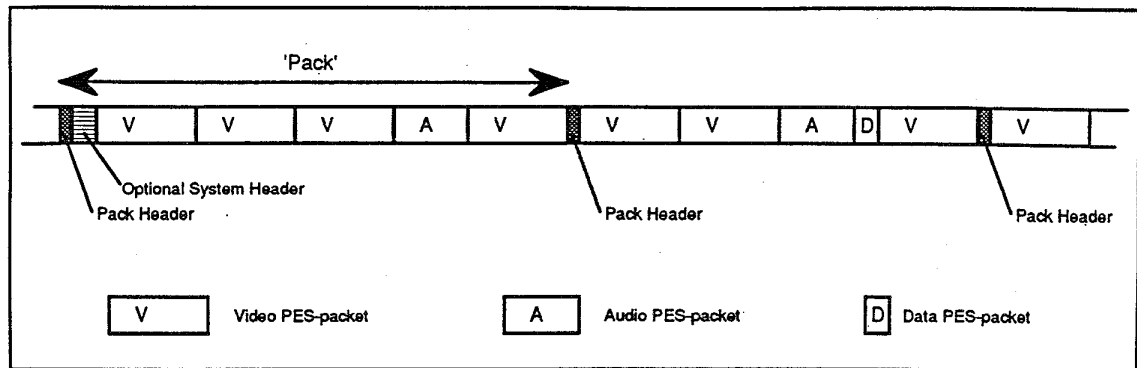


Figure 5 - Structure of the MPEG-2 'Programme Stream' multiplex

The Transport Stream multiplex

The transport stream multiplex consists entirely of short, fixed length transport packets. A transport packet (see figure 6) is always 188 bytes long. It comprises a 4-byte header followed by an 'adaptation field' or a 'payload' or both. In a transport stream, the PES-packets from the various elementary streams are each divided among the payload parts of a number of transport packets.

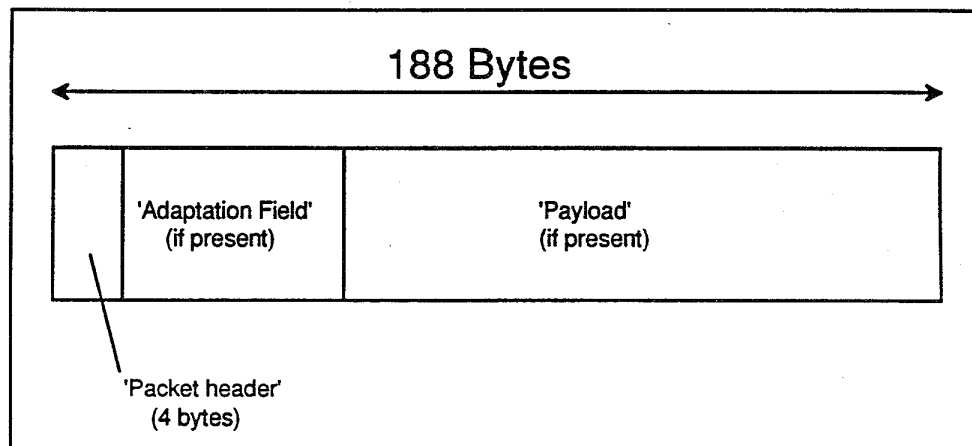


Figure 6 - The structure of a transport packet

Figure 7 shows how each PES-packet is divided into the payloads of a number of transport packets. The process is subject to two constraints:

1. The first byte of each PES-packet must become the first byte of a transport packet payload.
2. Only data taken from one PES-packet may be carried in any one transport packet.

A PES-packet is unlikely to exactly fill the payloads of an integer number of transport packets. As shown in figure 7, it will often be the case that there is insufficient bytes to completely fill the payload of the final transport packet. So as not to contravene the two constraints of the previous paragraph, the excess space is deliberately wasted by including an adaptation field of the appropriate length in this transport packet. This wastage can be minimised through careful choice of PES-packet length. This also provides an argument for the use of long PES-packet lengths as this will ensure that a greater proportion of transport packets are completely filled.

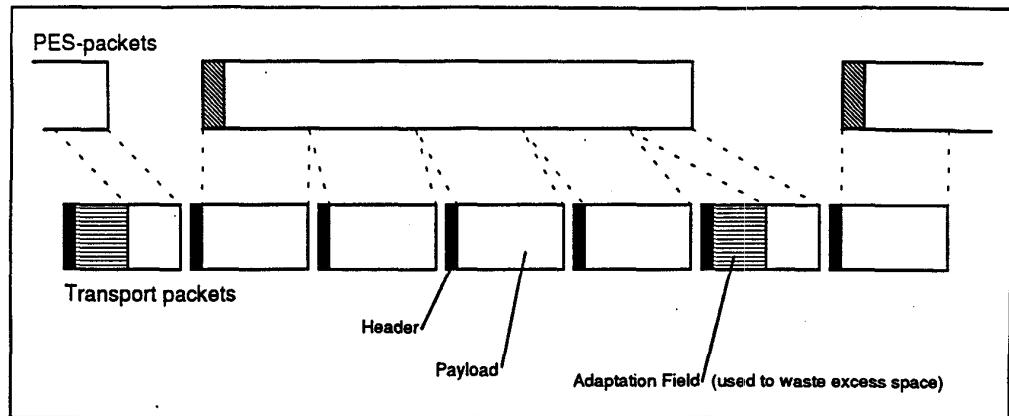


Figure 7 - Dividing a PES-packet into a number of transport packets

All the packetised elementary streams that are to be multiplexed together are converted to transport packets in this way and the resulting transport packets are then output sequentially to form an MPEG-2 transport stream. Transport packets containing service information and 'null' transport packets used to soak-up any spare capacity may also appear in the transport stream. There are no constraints on the order in which transport packets appear within the multiplex except that the chronological order of packets belonging to the same elementary stream must be preserved.

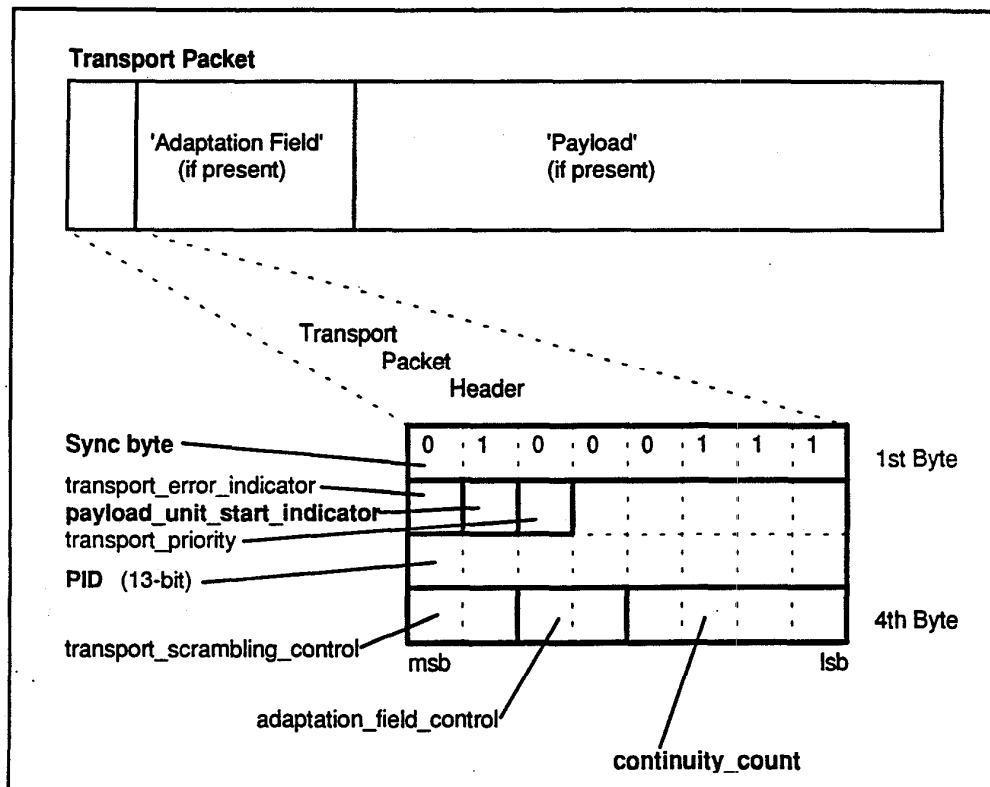


Figure 8 - The structure of a transport packet header

The transport packet header

A transport packet begins with a 4-byte header, the structure of which is shown in figure 8. Of the various fields it contains, four are particularly important:

- The first byte of the packet header is a 'sync-byte' having the value 47 (hex). This value is not unique within a transport packet and can quite naturally occur in other fields of the transport packet. However, the fact that a sync byte will occur every 188 bytes within a transport stream enables a suitably designed decoder to lock onto this repetition and hence to identify the start of each new transport packet.
- A single transport stream may carry many different programmes each comprising many packetised elementary streams. The 13-bit 'Packet Identifier' (PID) field is used to distinguish transport packets containing data from one elementary stream from those carrying data from other elementary streams. Of the 2^{13} possible values, 17 are reserved for special purposes. This leaves 8,175 values that may be assigned to the various elementary streams and represents the maximum number of elementary streams that can be accommodated in a single transport stream. It is the responsibility of the multiplexer to ensure that each elementary stream is awarded a unique PID value. Other than this, MPEG does not constrain which PID value is assigned to a particular elementary stream.
- The 'payload_unit_start_indicator' is set to show that there is something special about the first byte of the payload part of the transport packet. For example, the bit is set if the first byte of the payload is also the first byte of a PES-packet.
- Finally, the 'continuity count field' is incremented between successive transport packets belonging to the same elementary stream. This enables a decoder to detect the loss or gain of a transport packet and to hopefully conceal the errors that might otherwise result from such a loss.

Programme Specific Information

In a transport stream, each transport packet is tagged with an appropriate PID value indicating to which elementary stream its payload belongs. There may be many elementary streams comprising many different programmes. How can a decoder determine which elementary streams belong to each programme ? The answer is to include additional information within the transport stream to explicitly state the relationship between the available programmes and the PID values of their component elementary streams. Such information is called 'Programme Specific Information' (PSI) and must be present in every transport stream.

The Programme Specific Information specified for the MPEG-2 systems layer comprises four types of table:- The 'Program Association Table' (PAT), the 'Program Map Table' (PMT), the 'Network Information Table' (NIT) and the 'Conditional Access Table' (CAT). Each of these tables may be carried as the payload of one or more transport packets and hence accommodated in a transport stream.

Every programme carried in a transport stream has a Programme Map Table associated with it. This table gives details about the programme and the elementary streams that comprise it. Figure 9 shows a possible programme map table. From this table, a decoder can determine that the coded video elementary stream is being carried in transport packets tagged with PID = 726 and that an English language audio elementary stream is available from transport packets tagged with the PID value 57.

The basic programme map table may be embellished with some of the many 'descriptors' specified within the MPEG-2 systems specification. They convey further information about a programme or its component elementary streams. The descriptors include video encoding parameters, audio encoding parameters, language identification, pan-and-scan information, conditional access details, copyright information and so on. A broadcaster or other user may define additional private descriptors if required.

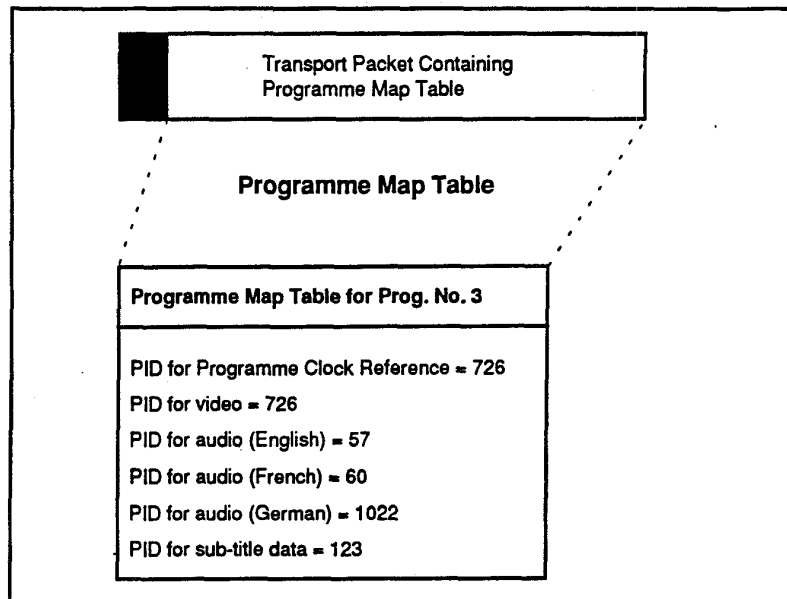


Figure 9 - Representation of a Programme Map Table

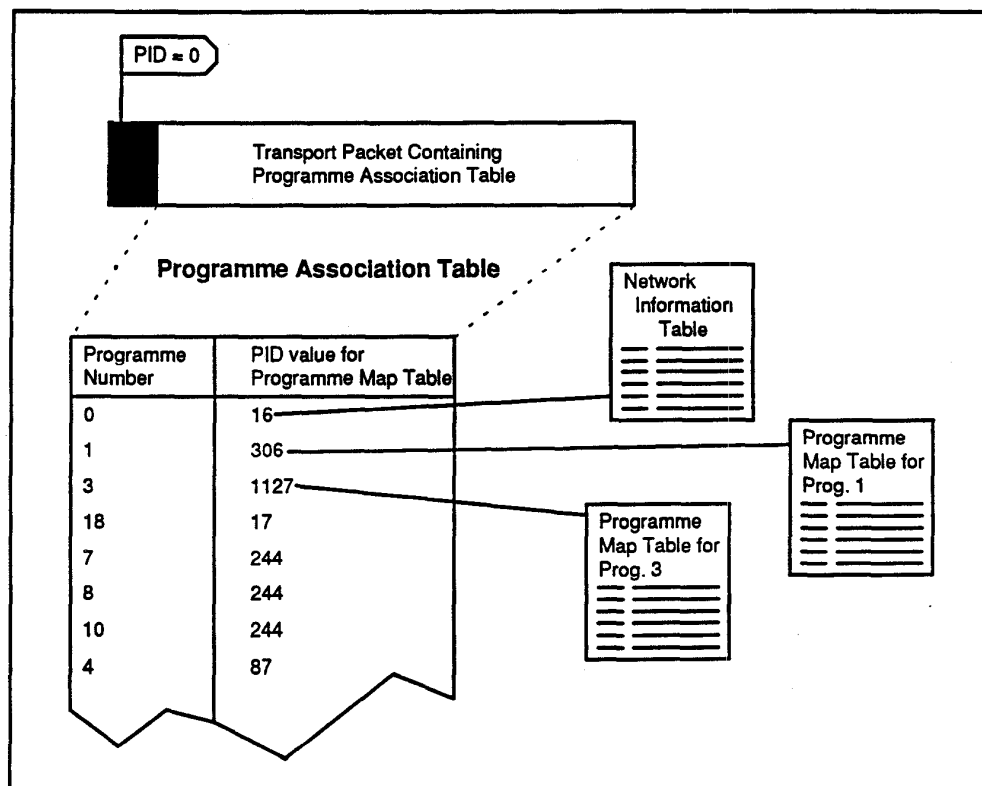


Figure 10 - A Programme Association Table

A complete list of all the programmes available in a transport stream is maintained in the Programme Association Table. This table is easily found as it always has the PID value 0. Each programme is listed along with the PID value of the transport packets that contain its Programme Map Table. For example, in the Programme Association Table of figure 10, the Programme Map Table for programme 3 is found in transport packets with PID value 1127.

It is permitted for a single Programme Map Table to contain the details of more than one programme. This is of particular advantage when each programme definition is short as the combined Programme Map Table may still fit within the payload of a single transport packet which is more efficient than the use of a separate transport packet for each of the programme definitions.

The programme number zero has special meaning within the Programme Association Table. It is used to point the way to the Network Information Table. This table is optional and its contents is private (i.e. defined by the broadcaster / user and not MPEG). Where present, the table is intended to provide information about the physical network carrying the transport stream such as channel frequencies, satellite transponder details, modulation characteristics, service originator, service name and details of alternative networks available.

Finally we have the Conditional Access Table. If any elementary streams within a transport stream are scrambled, then a Conditional Access Table must be present. The table provides details of the scrambling system(s) in use and provides the PID values of transport packets that contain the conditional access management and entitlement information. The format of this information is not specified within the MPEG-2 Systems specification as it depends on the type of scrambling system employed.

Programme Specific Information is also defined for use in the programme stream multiplex. As a programme stream may only carry a single programme, all elementary streams present in the multiplex must belong to the same programme. A table called a 'Programme Stream Map' is defined for use in the programme stream and states the type (audio, video, other) of information carried in each elementary stream. It may be embellished with MPEG-defined or private 'descriptors' to provide virtually limitless information about the programme or any of its elementary streams.

Time Stamps and Clock References : the basics

Figure 11 shows a simple decoder capable of extracting and replaying video and audio from either a programme stream or a transport stream. The switches are rather clever and identify the transport packets in a transport stream (or PES-packets in a programme stream) that contain data from the wanted video or audio elementary streams. The switch only permits data bytes from the wanted elementary stream to pass into the buffer. Any bytes that are not a part of the original elementary stream are not allowed to pass. As each buffer fills, so the access units comprising the original elementary stream accumulate. Remember that a video access unit is simply a single coded picture. When instructed to do so, the video decoder takes a complete video access unit from the buffer, decodes it and displays the decoded picture on the screen. Similarly, the audio decoder can decode each audio access unit to provide some milliseconds of audio.

How does a decoder know when to decode each access unit ? The answer is that a time stamp explicitly tells a decoder the exact time at which an access unit should be removed from the buffer and decoded. A decoder making correct use of the time stamps may rest assured that it is decoding access units in synchronism with the other elementary streams comprising a programme and that its buffer will never under or overflow.

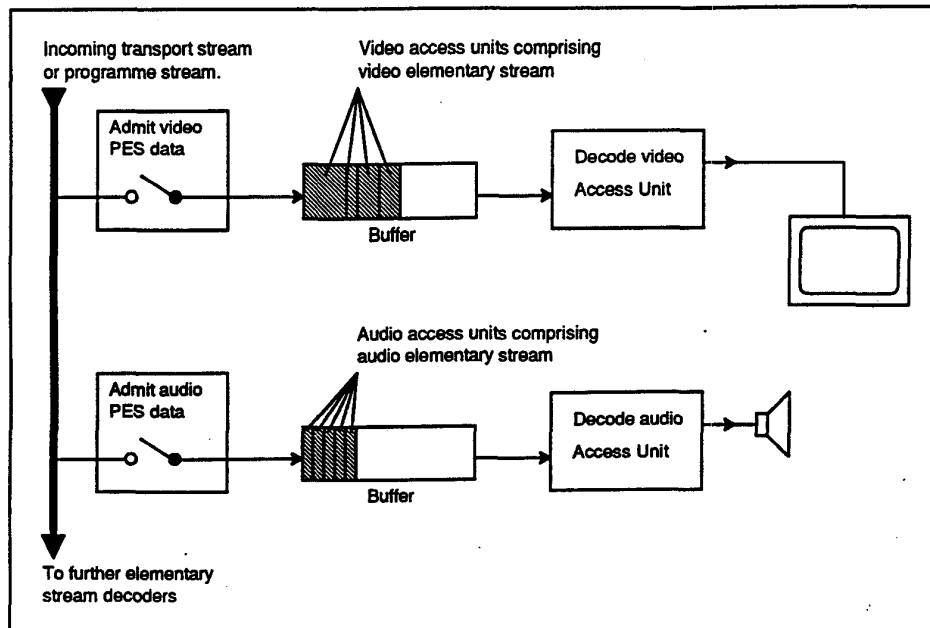


Figure 11 - A possible decoder.

A time stamp is a value representing a time. One function of the multiplexer is to assign time stamps to the access units as they are produced by a coder. In figure 12, the video coder has generated three access units and they have been assigned the time stamps 10:27, 10:28 and 10:29 respectively. So in this (rather low frame rate) example, the first access unit is to be decoded at 27 minutes past 10, the second at 28 minutes past 10 and the third at 29 minutes past 10.

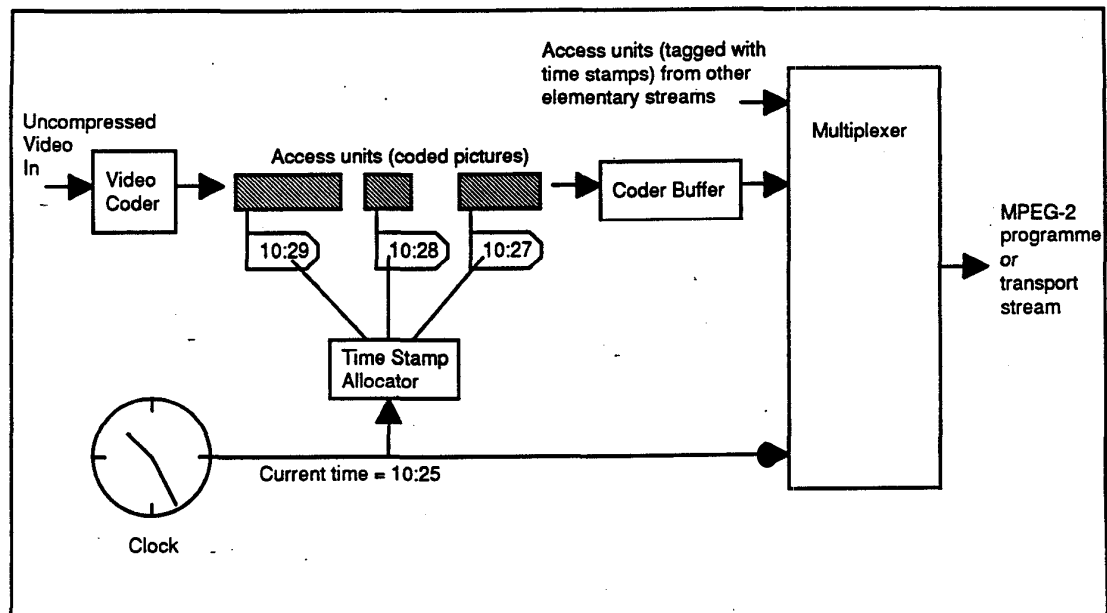


Figure 12 - Time stamps are allocated to access units by the multiplexer

In order to allocate time stamps, the coder has to "know" what the current time is. The current time is maintained by an accurate clock located in the multiplexer. Whenever a new access unit is generated by the coder, the current time is inspected and a time stamp based on the current time is assigned to the new access

unit. Note that the time stamp is not simply equal to the current time. It cannot be. Remember that the time stamp represents the time at which the access unit will be decoded at the decoder. This clearly must occur at some time in the future. So when a time stamp is first allocated to an access unit by the multiplexer it will always be deliberately offset into the future with respect to the current time.

How far into the future a time stamp should be offset is a subject in itself and depends on a number of factors including the size of the coder and decoder buffers and the bit rate at which the elementary stream is carried in the multiplex. The offset must be sufficiently large to allow sufficient time for the access unit to pass through the coder buffer, be conveyed to the decoder via the multiplex and be fully written into the decoder buffer.

In order to interpret the time stamps, the decoder requires its own accurate clock to determine the appropriate time to decode a particular access unit. It is essential that the decoder's clock is in exact synchronism with the clock at the multiplexer. This is achieved by including regular samples of the current time according to multiplexer's clock in the multiplex. The decoder uses these samples to check and correct the decoder clock.

Time Stamps and Clock References : the details

The clocks used at the multiplexer and decoder do not measure time in hours and minutes but in units of 27 MHz expressed as a 42-bit binary number and there is no requirement that the clocks are related to Greenwich Mean Time.

In a programme stream which can carry only a single programme, the clock at the multiplexer is called the 'System Clock'. Access units in all the elementary streams of the programme are assigned time stamps based on this system clock. Regular samples of the system clock are carried in the programme stream to enable the decoder clock to maintain synchronism with the system clock. These samples are called 'System Clock References' (SCR) and they are encoded in optional fields in the pack headers of the programme stream. A System Clock Reference must appear in the programme stream at least once every 0.7 seconds.

The alternative MPEG-2 multiplex, the transport stream, may contain many independent programmes. Each programme has its own independent clock called a 'Programme Clock' which need not be synchronised with the clocks of other programmes. (It is permitted for programmes to share Programme Clocks). Access units comprising a programme are assigned time stamps based on the associated programme clock. Samples of each programme clock called 'Programme Clock References' (PCR) are carried in the transport stream. They enable a decoder to synchronise its clock with the programme clock of the programme to be decoded. A Programme Clock Reference for each Programme Clock must appear in the transport stream at least every 0.1 seconds. Programme clock references are actually encoded in the adaptation fields of transport packets. Exactly which transport packets are carrying the programme clock reference values for a particular programme may be found in the programme map table defining the programme. (See "PID for Programme Clock Reference" in figure 9).

Note the potential confusion here. In a programme stream, the clock is called a 'System Clock' and a sample of its value is called a 'System Clock Reference'. In a transport stream the clocks are called 'Programme Clocks' and samples of their values are called 'Programme Clock References'. Apart from these differences in naming, there is very little difference between the clocks and their function in either the programme stream or transport stream.

Time stamps themselves are 33-bit binary values expressed in units of 90 kHz.

There are in fact two types of time stamp. The first and most fundamental is the 'Presentation Time Stamp' (PTS). It specifies the time at which an access unit should be removed from the decoder buffer, decoded and presented to the viewer. MPEG assumes that this can be performed instantaneously. In practice, the transfer of data into the decoder and the decoding process itself is certain to take some time and it is the responsibility of the decoder designer to compensate for these practicalities.

For many types of elementary stream, Presentation Time Stamps are the only time stamps that will be needed. However, in the significant case of an elementary stream carrying MPEG coded video then a second type of time stamp called a 'Decoding Time Stamp' (DTS) may be required.

A decoding time stamp specifies the time at which an access unit should be removed from the decoder buffer and decoded, but not presented to the viewer. Instead, the decoded picture is temporarily stored for presentation at a later time. Such treatment is necessary for the 'I' and 'P' picture types of an MPEG coded video sequence (See [2]).

A Decoding Time Stamp never occurs on its own and is always accompanied by a Presentation Time Stamp. Here, the Presentation Time Stamp specifies the time at which the decoded picture is eventually released from temporary storage and is presented to the viewer. Thus the Presentation Time Stamp will always be greater (i.e. further in to the future) than its associated Decoding Time Stamp.

It is not necessary to allocate time stamps to every access unit. A decoder will usually know in advance the rate at which access units are to be decoded and it is sufficient to provide the occasional time stamp simply to ensure that the decoding process maintains long-term synchronism. The constraint specified by MPEG is that a time stamp should occur at least every 0.7 seconds in a video or audio packetised elementary stream.

In either type of MPEG-2 multiplex, the time stamps are carried as optional fields in the headers of PES-packets (as shown in figure 4). If an access unit has a time stamp associated with it then the time stamp is encoded in the header of the PES-packet in which that access unit commences.

Summary

Two alternative multiplexes are specified for the MPEG-2 systems layer. The 'programme stream' is biased towards the storage and replay of a single programme from a digital storage device while the 'transport stream' is intended for the simultaneous delivery of a number of programmes over potentially error-prone channels. Both multiplexes facilitate the inclusion of Programme Specific Information detailing the programme(s) and elementary streams present. Both multiplexes use a system of time stamps and clock references to ensure the synchronous replay of related elementary streams and correct buffer behaviour at a decoder. There are many optional syntax elements and many opportunities to include private (user-defined) syntax extensions enabling either multiplex to be optimised to suit a particular application.

Acknowledgement

I would like to thank the BBC for permission to publish this paper.

References

- [1] ISO/IEC, "Generic Coding of Moving Pictures and Associated Audio: Systems", (MPEG-2 Systems Specification), ISO/IEC 13818-1, November 1994
- [2] ISO/IEC, "Generic Coding of Moving Pictures and Associated Audio: Video", (MPEG-2 Video Specification), ISO/IEC 13818-2, May 1994