A

Dissertation

On

Dynamic Troubleshooting Of Specific ETR-290 Alarms in a MPEG Transport Stream

Submitted in Partial fulfillment of the requirement

For the award of Degree of

MASTER OF TECHNOLOGY

(VLSI DESIGN & EMBEDDED SYSTEM)

Submitted By:

ANUJ PANCHAL

University Roll No: 07/VLSI/09

Under the Guidance of:

Mr RAJESH ROHILLA

(ASSOCIATE PROFESSOR)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY, DELHI

(FORMERLY DELHI COLLEGE OF ENGINEERING)

2009-2011

DELHI TECHNOLOGICAL UNIVERSITY, DELHI
Department of Electronics & Communication Engineering

CERTIFICATE

This is certified that the dissertation entitled "**Dynamic Troubleshooting Of Specific ETR-290 Alarms in a MPEG Transport Stream**" is a work of Anuj Panchal (University Roll No. 07/VLSI/09), a student of Delhi Technological University. This work was completed under my direct supervision and guidance and forms a part of the Master of Technology (VLSI Design and Embedded System) course and curriculum. He has completed his work with utmost sincerity and diligence.

The work embodied in this major project has not been submitted for the award of any other degree to the best of my knowledge and belief.

Rajesh Rohilla
Associate Professor
ECE department
Delhi Technological University
Delhi-110042

ACKNOWLEDGEMENT

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my project guide **Mr. Rajesh Rohilla,** Associate Professor, Department of Electronics and communication, Delhi Technological University, for his invaluable guidance, encouragement and patient reviews. His continuous inspiration only has made me complete this dissertation. Without his help and guidance, this dissertation would have been impossible. He remained a pillar of help throughout the project.

I am deeply thankful to **Prof. Rajiv Kapoor**, H.O.D., Electronics and Communication Engineering and **Prof. Ashok Bhattacharyya** (former H.O.D.) for their motivation and inspiration, at the same time I am very thankful to the entire faculty and staff members of Electronics & Communication Engineering Department for their direct or indirect help, cooperation, love and affection.

At last, I am grateful to my parents,classmates and my friends for their moral support all the time. They have been always around to cheer me up in the odd times of this work.

Anuj Panchal
(07/VLSI/09)

ABSTRACT

The current scenario of Transport stream Monitoring and Analyzing is such that we are doing the analysis of the Transport Stream at the output end i.e. the Streams are analyzed at the receiver end and then the Troubleshooting is done by manually doing the required change and therefore is not dynamic.

In this project the work has been done to analyze the output Transport stream just after the multiplexer output, and make changes to the transport stream after analyzing the stream for a specific ETR 290, Priority 1 alarm . All this is done without manual interception.

For the real-time environment , the generated stream after Satellite Transmission are received back via a Ku,C band antenna and then are analyzed..

In this project the real time environment is simulated in the Multiplexing code itself .The Analyzer monitors the Multiplexed stream for a specific ETR-290 alarm and will make necessary changes in the repetition frequency of the that specific PID packet.

The specific ETR-290 alarm we have taken here is PAT_error.

The entire process takes place within the same Block wherein we don’t have to Increase or Decrease the Frequency of reception of a PAT packet by manually inserting the time frame.

Here the Analyzer itself measure the PAT packet frequency and itself makes the adjustment whether to increase or decrease the frequency of that PAT packet by looping back to the multiplexer which generate PAT Packet..

TABLE OF CONTENTS

Page No.

Chapter 1

Introduction

1.1 Introduction

The systems part of this literature addresses the combining of one or more elementary streams of video and audio, as well as other data, into single or multiple streams which are suitable for storage or transmission. The Systems coding follows the syntactical and semantic rules imposed by the Specification ISO 13818-1 and provides information to enable synchronized decoding of decoder buffers over a wide range of retrieval or receipt conditions. The System coding shall be specified in two forms: the **Transport Stream** and the **Program Stream**. Each is optimized for a different set of applications. Both the Transport Stream and Program Stream defined in this literature follows the coding syntax as specified in ISO 13818-1 which is necessary and sufficient to synchronize the decoding and presentation of the video and audio information, while ensuring that data buffers in the decoders do not overflow or underflow.

Information is coded in the syntax using time stamps concerning the decoding and presentation of coded audio and visual data and time stamps concerning the delivery of the data stream itself. Both stream definitions are packet-oriented multiplexes. The basic multiplexing approach for single video and audio elementary streams is illustrated in Figure 1. The video and audio data is encoded as described in ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3. The resulting compressed elementary streams are packetized to produce **PES packets**. Information needed to use PES packets independently of either Transport Streams or Program Streams may be added when PES packets are formed.

This information is not needed and need not be added when PES packets are further combined with system level information to form **Transport Streams** or **Program Streams**. This systems standard covers those processes to the right of the vertical dashed line.

1.2 Basic Multiplexing approach

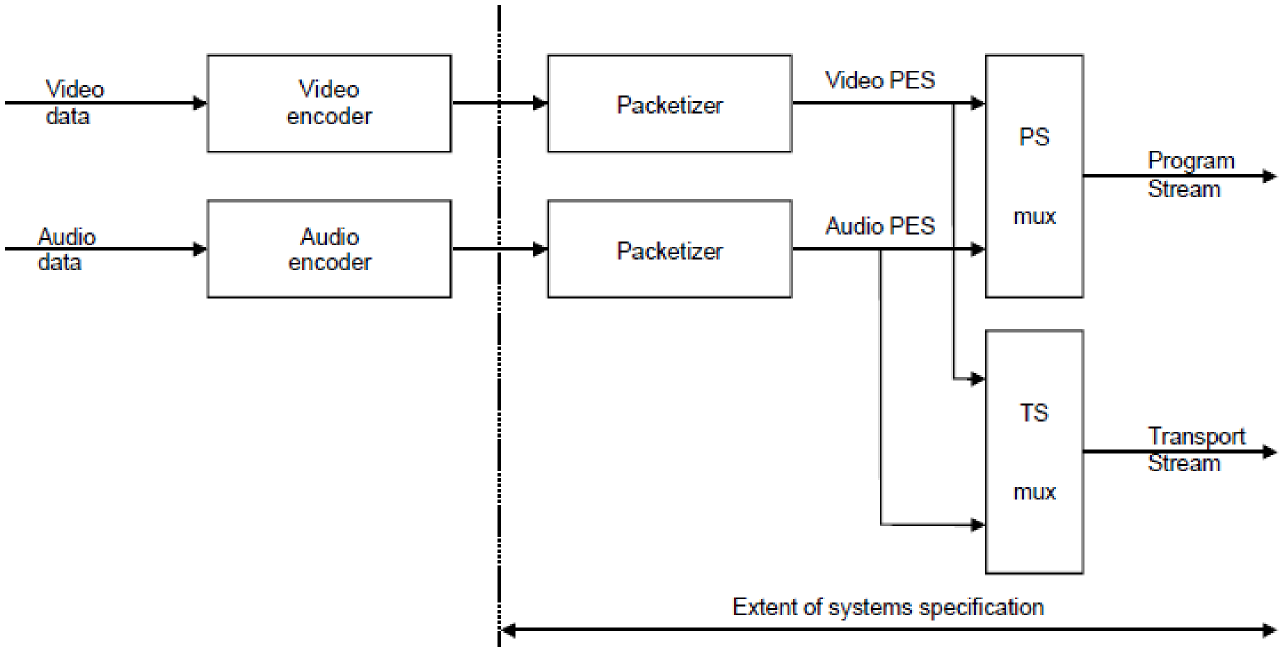


Fig 1: Basic Multiplexing Approach

1.3 Transport Stream

The Transport Stream is a stream definition which is tailored for communicating or storing one or more programs of coded data according to ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3 and other data in environments in which significant errors may occur. Such errors may be manifested as bit value errors or loss of packets. Transport Streams may be either fixed or variable rate. In either case the constituent elementary streams may either be fixed or variable rate. The syntax and semantic constraints on the stream are identical in each of these cases. The Transport Stream rate is defined by the values and locations of Program Clock Reference (PCR) fields, which in general are separate PCR fields for each program. There are some difficulties with constructing and delivering a

Transport Stream containing multiple programs with independent time bases such that the overall bit rate is variable.

The Transport Stream may be constructed by any method that results in a valid stream. It is possible to construct Transport Streams containing one or more programs from elementary coded data streams, from Program Streams, or from other Transport Streams which may themselves contain one or more programs.

1.4 Program Stream

The Program Stream is a stream definition which is tailored for communicating or storing one program of coded data and other data in environments where errors are very unlikely, and where processing of system coding, e.g. by software, is a major consideration.

Program Streams may be either fixed or variable rate. In either case, the constituent elementary streams may be either fixed or variable rate. The syntax and semantics constraints on the stream are identical in each case. The Program Stream rate is defined by the values and locations of the System Clock Reference (SCR) and mux_rate fields. A prototypical audio/video Program Stream decoder system is depicted in Figure 2. The architecture is not unique– system decoder functions including decoder timing control might as equally well be distributed among elementary stream decoders and the channel specific decoder – but this figure is useful for discussion. The prototypical decoder design does not imply any normative requirement for the design of an Program Stream decoder. Indeed non-audio/video data is also allowed, but not shown.

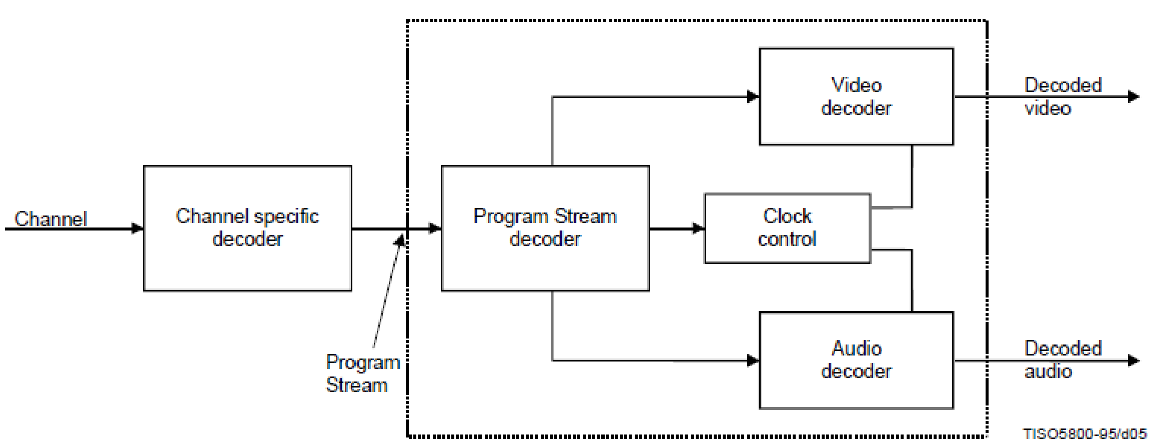


Fig 2: A protypical Decoder for a Program stream

1.5 Conversion between Transport Stream and Program Stream

It may be possible and reasonable to convert between **Transport Streams** and **Program Streams** by means of PES packets.

PES packets may, with some constraints, be mapped directly from the payload of one multiplexed bit stream into the payload of another multiplexed bit stream. It is possible to identify the correct order of PES packets in a program to assist with this if the program_packet_sequence_counter is present in all PES packets.

Certain other information necessary for conversion, e.g. the relationship between elementary streams, is available in Tables and headers in both streams. Such data, if available, shall be correct in any stream before and after conversion.

1.6 Packetized Elementary Stream

Transport Streams and **Program Streams** are each logically constructed from **PES packets**, as indicated in the syntax definitions in . PES packets shall be used to convert between Transport Streams and Program Streams; in some cases the PES packets need not be modified when performing such conversions. PES packets may be much larger than the size of a Transport Stream packet.

A continuous sequence of PES packets of one elementary stream with one stream ID may be used to construct a PES Stream. When PES packets are used to form a PES stream, they shall include Elementary Stream Clock Reference (ESCR) fields and Elementary Stream Rate (ES_Rate) fields.

The PES stream data shall be contiguous bytes from the elementary stream in their original order. PES streams do not contain some necessary system information which is contained in Program Streams and Transport Streams.

Examples include the information in the Pack Header, System Header, Program Stream Map, Program Stream Directory, Program Map Table, and elements of the Transport Stream packet syntax.

The PES Stream is a logical construct that may be useful within implementations of this Recommendation | International Standard; however, it is not defined as a stream for interchange and interoperability.

Applications requiring streams containing only one elementary stream can use Program Streams or Transport Streams which each contain only one elementary stream. These streams contain all of the necessary system information. Multiple Program Streams or Transport Streams, each containing a single elementary stream, can be constructed with a common time base and therefore carry a complete program, i.e. with audio and video.

1.7 Timing model

Systems, Video and Audio all have a timing model in which the end-to-end delay from the signal input to an encoder to the signal output from a decoder is a constant. This delay is the sum of encoding, encoder buffering, multiplexing, communication or storage, demultiplexing, decoder buffering, decoding, and presentation delays. As part of this timing model all video pictures and audio samples are presented exactly once, unless specifically coded to the contrary, and the inter-picture interval and audio sample rate are the same at the decoder as at the encoder. The system stream coding contains timing information which can be used to implement systems which embody constant end-to-end delay. It is possible to implement decoders which do not follow this model exactly; however, in such cases it is the decoder's responsibility to perform in an acceptable manner.

All timing is defined in terms of a common system clock, referred to as a System Time Clock. In the Program Stream this clock may have an exactly specified ratio to the video or audio sample clocks, or it may have an operating frequency which differs slightly from the exact ratio while still providing precise end-to-end timing and clock recovery.

In the Transport Stream the system clock frequency is constrained to have the exactly specified ratio to the audio and video sample clocks at all times; the effect of this constraint is to simplify sample rate recovery in decoders.

1.8 Conditional access

Encryption and scrambling for conditional access to programs encoded in the Program and Transport Streams is supported by the system data stream definitions. Conditional access mechanisms are not specified in this Report .

1.9 Synchronization

Synchronization among multiple elementary streams is accomplished with Presentation Time Stamps (PTS) in the Program Stream and Transport streams. Time stamps are generally in units of 90 kHz, but the System Clock Reference (SCR), the Program Clock Reference (PCR) and the optional Elementary Stream Clock Reference (ESCR) have extensions with a resolution of 27 MHz. Decoding of N-elementary streams is synchronized by adjusting the decoding of streams to a common master time base rather than by adjusting the decoding of one stream to match that of another.

The master time base may be one of the N-decoders' clocks, the data source's clock, or it may be some external clock.

Each program in a Transport Stream, which may contain multiple programs, may have its own time base. The time bases of different programs within a Transport Stream may be different.

Because PTSs apply to the decoding of individual elementary streams, they reside in the PES packet layer of both the Transport Streams and Program Streams.

End-to-end synchronization occurs when encoders save time stamps at capture time, when the time stamps propagate with associated coded data to decoders, and when decoders use those time stamps to schedule presentations.

Synchronization of a decoding system with a channel is achieved through the use of the SCR in the Program Stream and by its analogue, the PCR, in the Transport Stream.

The SCR and PCR are time stamps encoding the timing of the bit stream itself, and are derived from the same time base used for the audio and video PTS values from the same program.

Since each program may have its own time base, there are separate PCR fields for each program in a Transport Stream containing multiple programs. In some cases it may be possible for programs to share PCR fields, **Program Specific Information (PSI),** for the method of identifying which PCR is associated with a program. A program shall have one and only one PCR time base associated with it.

CHAPTER- 2

TS and PES Packet Details and Information used for decoding at Receiver

2.1a The Packetized Elementary Stream (PES)

All elementary streams in MPEG are first packetized in variable-length packets called PES packets. The packets, which primarily have a length of 64 kbytes, begin with a PES header of 6 bytes minimum length.

The first 3 bytes of this header represent the “start code prefix”, the content of which is always 00 00 01 and which is used for identifying the start of a PES packet.

The byte following the start code is the “stream ID” which describes the type of elementary stream following in the payload. It indicates whether it is, e.g. a video stream, an audio stream or a data stream which follows.

After that there are two “packet length” bytes which are used to address the up to 64 kbytes of payload. If both of these bytes are set to zero, a PES packet having a length which may exceed these 64 kbytes can be expected.

The MPEG decoder then has to use other arrangements to find the PES packet limits, e.g. the start code. After these 6 bytes of PES header, an “optional PES header” is transmitted which is an optional extension of the PES header and is adapted to the requirements of the elementary stream currently being transmitted.

It is controlled by 11 flags in a total of 12 bits in this optional PES header.

These flags show which components are actually present in the “optional fields” in the optional PES header and which are not. The total length of the PES header is shown in the “PES header data length” field.

The optional fields in the optional header contain, among other things, the “Presentation

Time Stamps" (PTS) and the "decoding time stamps" (DTS) which are important for synchronizing video and audio.

At the end of the optional PES header there may also be stuffing bytes. Following the complete PES header, the actual payload of the elementary stream is transmitted which can usually be up to 64 kbytes long or even longer in special cases, plus the optional header.

In MPEG-1, video PES packets are simply multiplexed with PES packets and stored on a data medium . The maximum data rate is about 1.5 Mbit/s for video and audio and the data stream only includes a video stream and an audio stream.

This "Packetized Elementary Stream" (PES) with its relatively long packet structures is not, however, suitable for transmission and especially not for broadcasting a number of programs in one multiplexed data signal.

In MPEG-2, on the other hand, the objective has been to assemble up to 6, 10 or even 20 independent TV or radio programs to form one common multiplexed MPEG-2 data signal. This data signal is then transmitted via satellite, cable or terrestrial transmission links. To this end, the long PES packets are additionally divided into smaller packets of constant-length.

From the PES packets, 184-byte-long pieces are taken and to these another 4-byte-long header is added (Fig 3), making up 188-byte-long packets called "transport stream packets" which are then multiplexed.

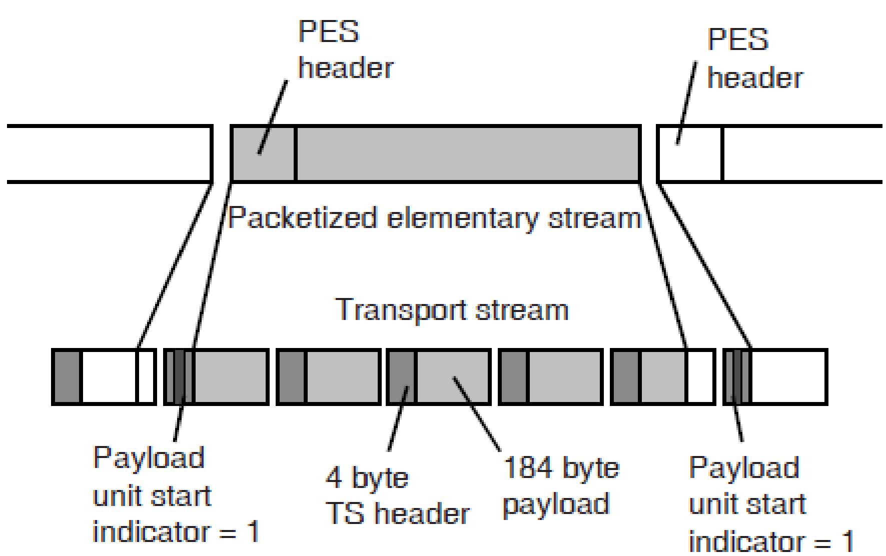
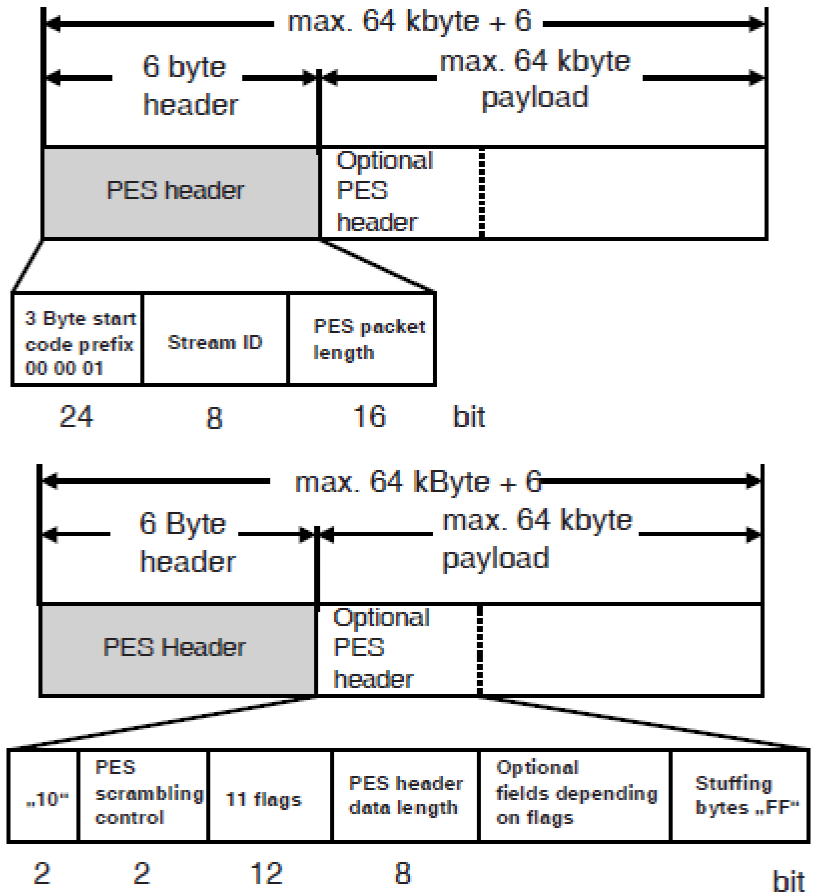


Fig 3: Transport Stream Packet

2.2a PES Packet



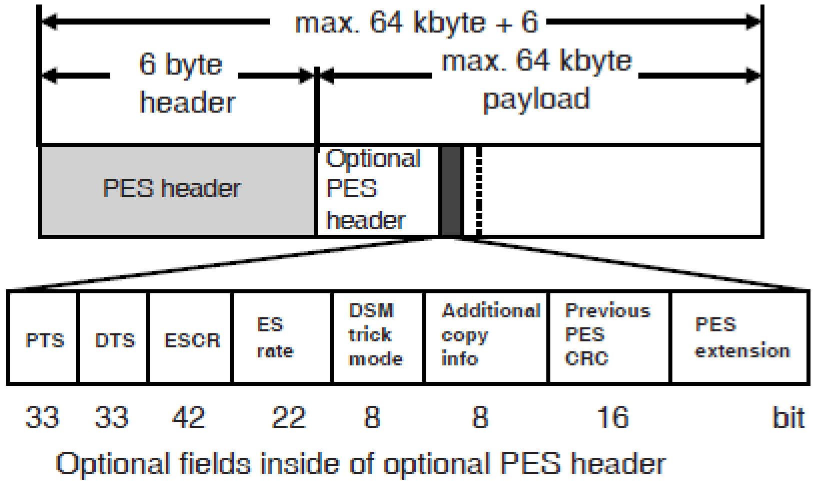


Fig 4: PES Packet Detail

2.3a Semantic definition of fields in PES packet

packet_start_code_prefix – The packet_start_code_prefix is a 24-bit code. Together with the stream_id that follows it constitutes a packet start code that identifies the beginning of a packet. The packet_start_code_prefix is the bit string

'0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – In Program Streams, the stream_id specifies the type and number of the elementary stream as defined by the stream_id Table 1. In Transport Streams, the stream_id may be set to any valid value which correctly describes the elementary stream type as defined in Table 2. In Transport Streams, the elementary stream type is specified in the Program Specific Information .

stream_id	Note	stream coding
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 … 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

Table :1

PES_packet_length – A 16-bit field specifying the number of bytes in the PES packet following the last byte of the field. A value of 0 indicates that the PES packet length is neither specified nor bounded and is allowed only in PES packets whose payload consists of bytes from a video elementary stream contained in Transport Stream packets.

PES_scrambling_control – The 2-bit PES_scrambling_control field indicates the scrambling mode of the PES packet payload. When scrambling is performed at the PES level, the PES packet header, including the optional fields when present, shall not be scrambled (Table 2).

Value	Description
00	Not scrambled
01	User-defined
10	User-defined
11	User-defined

Table :2

Transport Stream contents and parameters

2.4b Multiplexed MPEG-2 transport stream packets

An MPEG-2 transport stream contains the 188-byte-long transport stream packets of all programs with all their video, audio and data signals. Depending on the data rates, packets of one or the other elementary streams will occur more or less frequently in the MPEG-2 transport stream.

For each program there is one MPEG encoder which encodes all elementary streams, generates a PES structure and then packetizes these PES packets into transport stream packets. The data rate for each program is usually approx. 2 - 7 Mbit/s but the aggregate data rate for video, audio and data can be constant or vary in accordance with the program content at the time. This is then called “statistical multiplex”.

The transport streams of all the programs are then combined in a multiplexed MPEG-2 data stream to form one overall transport stream (Fig.5) which can then have a data rate of up to about 40 Mbit/s.

There are often up to 6, 8 or 10 or even 20 programs in one transport stream. The data rates can vary during the transmission but the overall data rate has to remain constant. A program can contain video and audio, only audio (audio broadcast) or only data, and the structure is thus flexible and can also change during the transmission.

To be able to determine the current structure of the transport stream during the decoding, the transport stream also carries lists describing the structure, so-called “tables”.

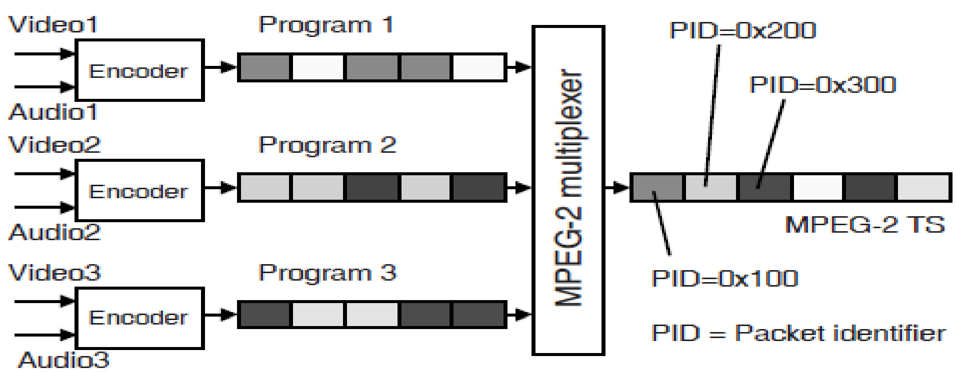


Fig 5: Multiplexed Transport Stream Packets

2.5b The MPEG-2 Transport Stream Packet

The MPEG-2 transport stream consists of packets having a constant length (Fig.6). This length is always 188 bytes, with 4 bytes of header and 184 bytes of payload. The payload contains the video, audio or general data.

The header contains numerous items of importance to the transmission of the packets. The first header byte is the “sync byte”. It always has a value of 47hex (0x47 in C/C++ syntax) and is spaced a constant 188 bytes apart in the transport stream.

It is quite possible, and certainly not illegal, for there to be a byte having the value 0x47 somewhere else in the packet.

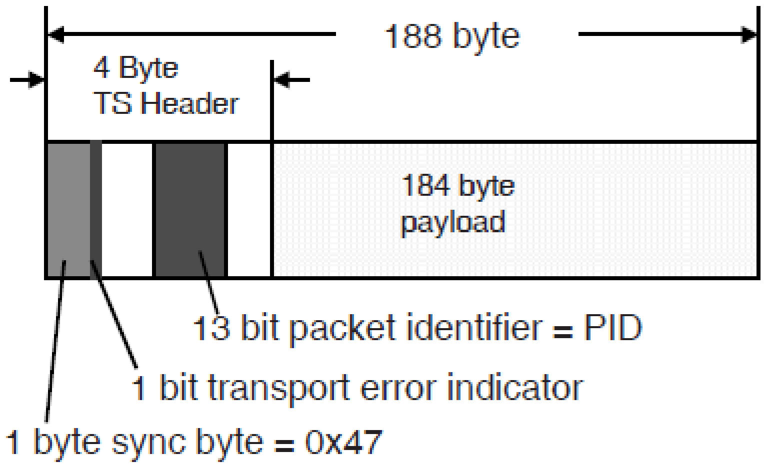


Fig 6: 188 byte Transport Stream Packet

The sync byte is used for synchronizing the packet to the transport stream and it is its value plus the constant spacing which is being used for synchronization. According to MPEG, synchronization at the decoder occurs after five transport stream packets have been received. Another important component of the transport stream is the 13 bit-long "packet identifier" or PID for short. The PID describes the current content of the payload part of this packet. The hexadecimal 13 bit number in combination with tables also included in the transport stream show which elementary stream or content this is.

It may be necessary occasionally to transmit more than 4 bytes of header per transport stream packet. The header is extended into the payload field in this case. The payload part becomes correspondingly shorter but the total packet length remains a constant 188 bytes. This extended header is called an "adaptation field" (Fig. 7).

The other contents of the header and of the adaptation field are shown below , and will be discussed thouroghly in the following chapter .

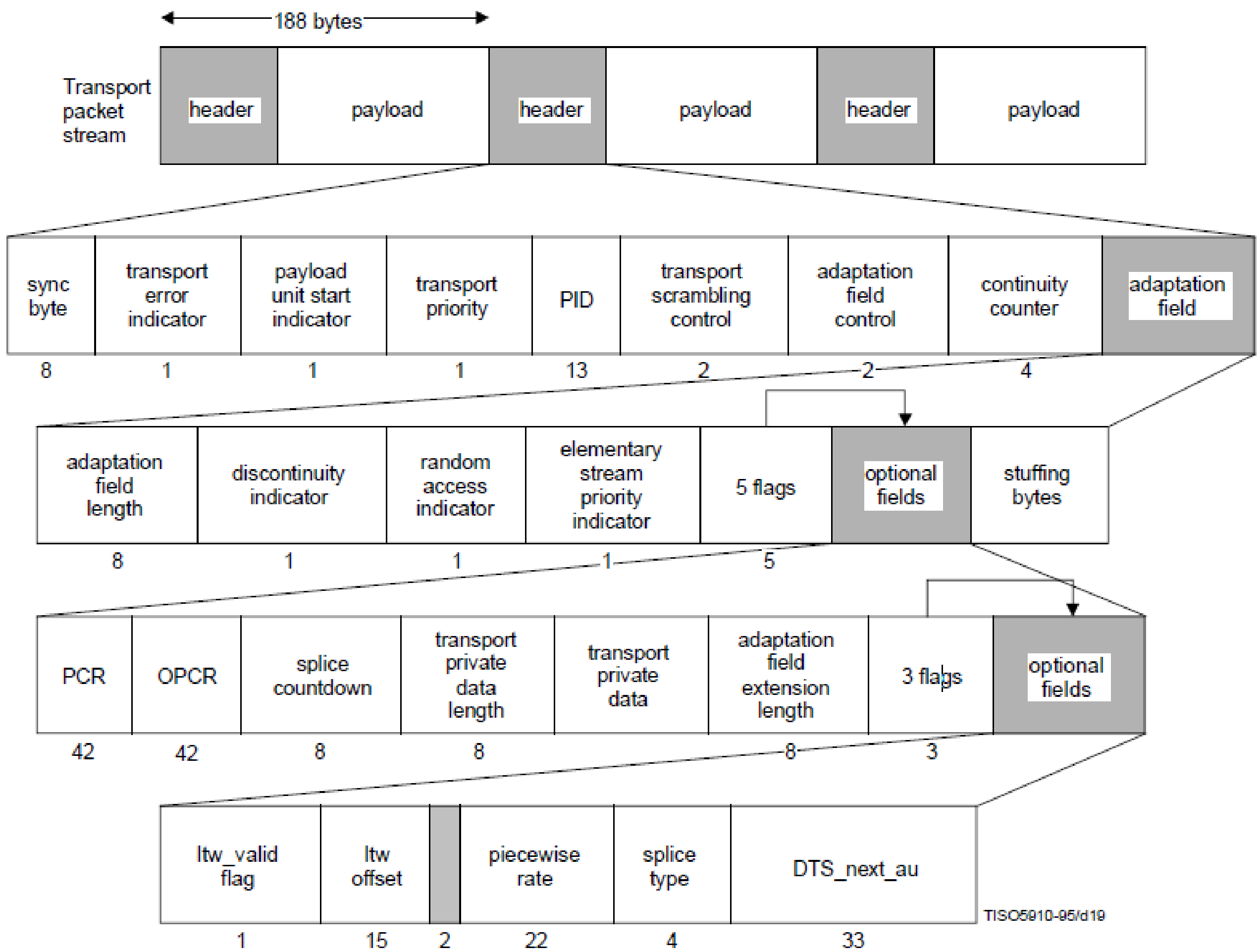


Fig 7: The MPEG-2 Transport Stream Packet detail

2.6b Basic Information in Transport stream Packet required at decoder Level

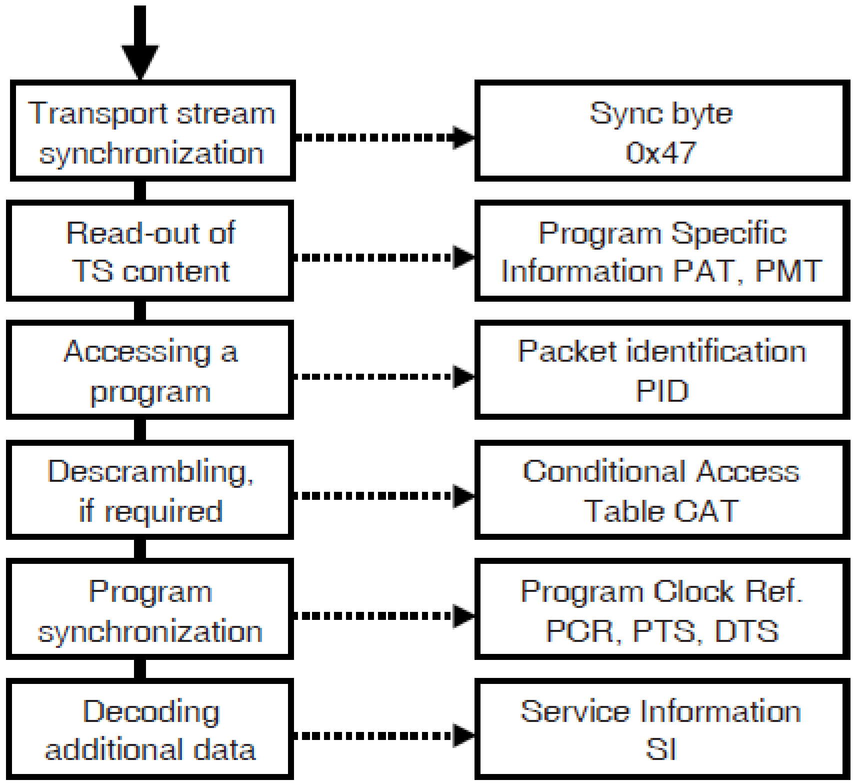


Fig 8: Basic Information in Transport stream Packet required at decoder Level

2.7b Information for the Receiver

In the following paragraphs, the components of the transport stream which are necessary for the receiver will be considered as shown in (Fig 9)

Necessary components means in this case: What does the receiver, i.e. the MPEG decoder, need for extracting from the large number of transport stream packets with the most varied contents exactly those which are needed for decoding the desired program?

In addition, the decoder must be able to synchronize correctly to this program.

The MPEG-2 transport stream is a completely asynchronous signal and its contents occur in a purely random fashion or on demand in the individual time slots. There is no absolute rule which can be used for determining what information will be contained in the next transport stream packet.

The decoder and every element on the transmission link must lock to the packet structure.

The PID (packet identifier) can be used for finding out what is actually being transmitted in the respective element. On the one hand, this asynchronism has advantages because of the total flexibility provided but there are also disadvantages with regard to power saving. Every single transport stream packet must first be analysed in the receiver .

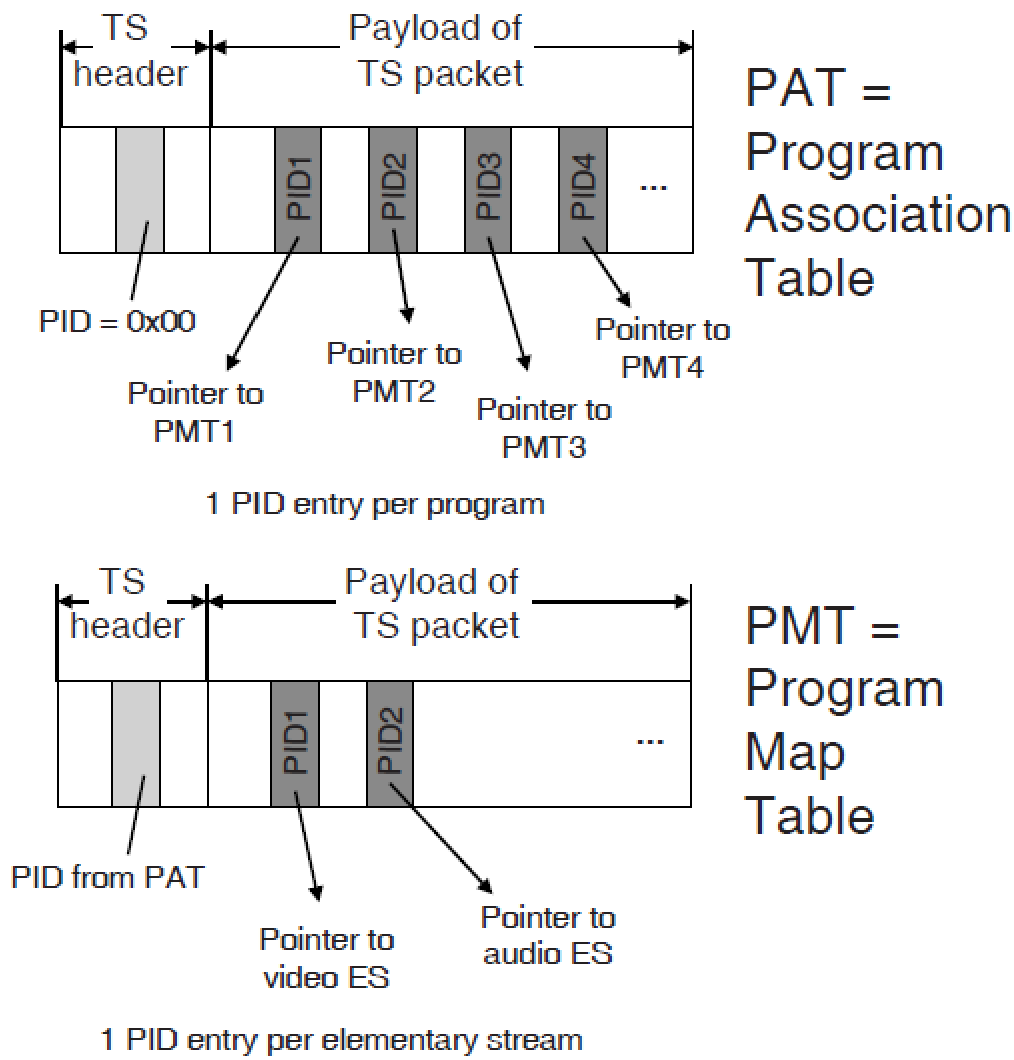


Fig 9: Information for the Receiver

2.8b Synchronizing to the Transport Stream

When the MPEG-2 decoder input is connected to an MPEG-2 transport stream, it must first lock to the transport stream, i.e. to the packet structure.

The decoder, therefore, looks for the sync bytes in the transport stream. These always have the value of 0x47 and always appear at the beginning of a transport stream packet. They are thus present at constant intervals of 188 bytes. These two factors together, the constant value of 0x47 and the constant spacing of 188 bytes, are used for the synchronization.

If a byte having a value of 0x47 appears, the decoder will examine the positions of n times 188 bytes before and after this byte in the transport stream for the presence of another sync byte. If there is, then this is a sync byte. If not, then this is simply some code word which has accidentally assumed this value. It is inevitable that the code word of 0x47 will also occur in the continuous transport stream. Synchronization will occur after 5 transport stream packets and the decoder will lose lock after a loss of 3 packets (MPEG-2 Standard) .

The number and the structure of the programs transmitted in the transport stream is flexible and open. The transport stream can contain one program with one video and audio elementary stream, or there can be 20 programs or more, some with only audio, some with video and audio and some with video and a number of audio signals which are being broadcast. It is, therefore, necessary to include certain lists in the transport stream which describe the instantaneous structure of the transport stream. These lists provide the so-called "**program specific information", or PSI** in short (Fig.9.).

They are tables which are occasionally transmitted in the payload part.

The first table is the **"Program Association Table" (PAT).**

- This table occurs precisely once per transport stream but is repeated every 0.5 sec..
- This table shows how many programs there are in this transport stream. Transport stream packets containing this table have the value zero as packet identifier (PID) and can thus be easily identified. In the payload part of the program association table, a list of special PIDs is transmitted.
- There is exactly one PID per program in the program association table (Fig. .).
- These PIDs are pointers, as it were, to other information describing each individual program in more detail. They point to other tables, the so-called "**Program Map Tables" (PMT).**

“**Program Map Tables” (PMT).**

- The program map tables, in turn, are special transport stream packets with a special payload part and special PID.
- The PIDs of the PMTs are transmitted in the PAT. If it is intended to receive, e.g. program No.3, PID no. 3 is selected in the list of all PIDs in the payload part in the program association table (PAT). If this is, e.g. 0x1FF3 . the decoder looks for transport stream packets having PID = 0x1FF3 in their header. These packets are then the program map table for program no.3 in the transport stream. The program map table, in turn, contains PIDs which are the PIDs for all elementary streams contained in this program (video, audio, data).

Since there can be a number of video and audio streams in a Transport Stream - the viewer must select the elementary streams to be decoded. Ultimately he will select exactly 2 PIDs - one for the video stream and one for the audio stream, resulting e.g. in the two hexadecimal numbers PID1 = 0x100 and PID2 = 0x110.
PID1 is then e.g. the PID for the video stream to be decoded and PID2 is the PID
for the audio stream to be decoded.
From now on, the MPEG-2 decoder will only be interested in these transport stream packets, collect them, i.e. demultiplex them and assemble them again to form the PES packets.
It is precisely these PES packets which are supplied to the video and audio decoder
in order to generate another video-and-audio signal.

2.9b ETR-290 alarms

The Measurement Group in the DVB Project has defined numerous measurements on the MPEG-2 transport stream within its Measurement Guidelines ETR 290. According to ETR 290, the errors to be detected by means of these measurements were graded into three levels of priority:
Priority 1, 2, 3.

MPEG-2 transport stream errors:

• Priority 1 - no decidability

• Priority 2 - partially no decodability

• Priority 3 - errors in the supplementary information/SI

If there is a Priority 1 error, there is often no chance to lock to the transport stream or even to decode a program. Priority 2, in contrast, means that there is partially no possibility of reproducing a program faultlessly. The presence of a category 3 error, on the other hand, only indicates errors in the broadcasting of the DVB service information. The effects are then dependent on how the set-top box used reacts. Apart from the category 3 errors, all measurements can also be applied with the American ATSC standard where comparable analyses can be made on the PSIP tables.

The following measurements on the MPEG-2 transport stream are defined in the DVB Measurement Guidelines ETR 290: Fig 10.

Measurement	Priority
TS_sync_loss	1
Sync_byte_error	1
PAT_error	1
PMT_error	1
Continuity_count_error	1
PID_error	1
Transport_error	2
CRC_error	2
PCR_error	2
PCR_accuracy_error	2
PTS_error	2
CAT_error	2
SI_repetition_error	3
NIT_error	3
SDT_error	3
EIT_error	3
RST_error	3
TDT_error	3
Undefined_PID	3

Fig 10

CHAPTER- 3

PROPOSED WORK

3.1 Introduction

The current scenario of Transport stream Monitoring and Analyzing is such that we are doing the analysis of the Transport Stream at the output end i.e. the Streams are analyzed at the receiver end and then the Troubleshooting is done by manually doing the required change and therefore is not dynamic.
In this project the work has been done to analyze the output Transport stream just after the multiplexer output , and make changes to the transport stream after analyzing the stream for a specific ETR 290 ,Priority 1 alarm . All this is done without manual interception .
For the real-time environment , the generated stream after Satellite Transmission are received back via a Ku,C band antenna and then are analyzed..
In this project the real time environment is simulated in the Multiplexing code itself .The Analyzer monitors the Multiplexed stream for a specific ETR-290 alarm and will make necessary changes in the repetition frequency of the that specific PID packet.
The specific ETR-290 alarm we have taken here is PAT_error.

The entire process takes place within the same Block wherein we don’t have to Increase or Decrease the Frequency of reception of a PAT packet by manually inserting the time frame.
Here the Analyzer itself measure the PAT packet frequency and itself makes the adjustment whether to increase or decrease the frequency of that PAT packet by looping back to the multiplexer which generate PAT Packet..

3.2 Missing or Errored Program Association Table (PAT) (PAT_error)

The program structure, i.e. the composition of the MPEG-2 transport stream is variable or, in other words, open. For this reason, lists for describing the current transport stream composition are transmitted in special TS packets in the transport stream.

The most important one of these is the Program Association Table (PAT) which is always transmitted in transport stream packets with PID=0 and Table ID=0.

If this table is missing or is errored, identification, and thus decoding, of the programs becomes impossible.

In the PAT, the PIDs of all Program Map Tables (PMTs) of all programs are transmitted. The PAT contains pointer information to many PMTs.

A set top box will find all necessary basic information in the PAT. A PAT which is missing, is transmitted scrambled, is error or is transmitted not frequently enough will lead to an error message "PAT error".

The PAT should be transmitted free of errors and unscrambled every 500 ms at a maximum.

A PAT error occurs when

- the PAT is missing,
- the repetition rate is greater than 500 ms**.**
- the PAT is scrambled.
- the table ID is not zero.

In this project we have focused on controlling the repetition rate

3.3 Project Stages

1) To make a Transport Stream with least of parameters using coding(C)

2) Analyze the frequency of the PAT packet if it is overshooting or is under the desired Value.

3) Use PID controller to change the PAT packet frequency by providing the loopback within the multiplexer

3.3.1 Project Stage 1

To Generate aTransport Stream

Transport Stream coding layer allows one or more programs to be combined into a single stream. Data from each elementary stream are multiplexed together with information that allows synchronized presentation of the elementary streams within a program.

A Transport Stream consists of one or more programs. Audio and video elementary streams consist of access units.

Elementary Stream data is carried in PES packets. A PES packet consists of a PES packet header followed by packet data. PES packets are inserted into Transport Stream packets. The first byte of each PES packet header is located at the first available payload location of a Transport Stream packet. The PES packet header begins with a 32-bit start-code that also identifies the stream or stream type to which the packet data belongs. The PES packet header may contain decoding and presentation time stamps (DTS and PTS).

The PES packet header also contains other optional fields. The PES packet data field contains a variable number of contiguous bytes from one elementary stream.

Transport Stream packets begin with a 4-byte prefix, which contains a 13-bit Packet ID (PID), defined in Table . The PID identifies, via the Program Specific Information (PSI) tables, the contents of the data contained in the Transport Stream packet. Transport Stream packets of one PID value carry data of one and only one elementary stream.

The PSI tables are carried in the Transport Stream. There are four PSI tables:

• Program Association Table;
• Program Map Table;
• Conditional Access Table;
• Network Information Table.

Transport Stream packets may be null packets. Null packets are intended for padding of Transport Streams. They may be inserted or deleted by re-multiplexing processes and, therefore, the delivery of the payload of null packets to the decoder cannot be assumed.

Transport Stream Algorithm

```
// TRANSPORT STREAM:
MPEG_transport_stream() {
do {
transport_packet()
} while (nextbits() = = sync_byte)

}
```

Transport Stream packet layer

```
transport_packet()
{
sync_byte
transport_error_indicator
payload_unit_start_indicator
transport_priority
PID
transport_scrambling_control
adaptation_field_control
continuity_counter
```

```
if(adaptation_field_control = = '10' || adaptation_field_control = = '11'){
adaptation_field()
}
if(adaptation_field_control = = '01' || adaptation_field_control = = '11') {
for (i = 0; i < N; i++){
Data byte
}

}

}
```

Semantic definition of fields in Transport Stream packet layer

sync_byte – The sync_byte is a fixed 8-bit field whose value is '0100 0111' (0x47). Sync_byte emulation in the choice of values for other regularly occurring fields, such as PID, should be avoided.

transport_error_indicator – The transport_error_indicator is a 1-bit flag. When set to '1' it indicates that at least 1 uncorrectable bit error exists in the associated Transport Stream packet. This bit may be set to '1' by entities external to the transport layer. When set to '1' this bit shall not be reset to '0' unless the bit value(s) in error have been corrected.

payload_unit_start_indicator – The payload_unit_start_indicator is a 1-bit flag which has normative meaning for Transport Stream packets that carry PES packets or PSI data . When the payload of the Transport Stream packet contains PES packet data, the payload_unit_start_indicator has the following significance:

a '1' indicates that the payload of this Transport Stream packet will commence with the first byte of a PES packet and

a '0' indicates no PES packet shall start in this Transport Stream packet.

If the payload_unit_start_indicator is set to '1', then one and only one PES packet starts in this Transport Stream packet.

When the payload of the Transport Stream packet contains PSI data, the payload_unit_start_indicator has the following significance: if the Transport Stream packet carries the first byte of a PSI section, the payload_unit_start_indicator value shall be '1', indicating that the first byte of the payload of this Transport Stream packet carries the pointer_field.

If the Transport Stream packet does not carry the first byte of a PSI section, the payload_unit_start_indicator value shall be '0',indicating that there is no pointer_field in the payload. For null packets the payload_unit_start_indicator shall be set to '0'.

The meaning of this bit for Transport Stream packets carrying only private data is not defined in this Specification.

transport_priority – The transport_priority is a 1-bit indicator. When set to '1' it indicates that the associated packet is of greater priority than other packets having the same PID which do not have the bit set to '1'. The transport mechanism can use this to prioritize its data within an elementary stream. Depending on the application the transport_priority field may be coded regardless of the PID or within one PID only. This field may be changed by channel specific encoders or decoders.

PID – The PID is a 13-bit field, indicating the type of the data stored in the packet payload.

PID value 0x0000 is reserved for the Program Association Table .

PID value 0x0001 is reserved for the Conditional Access

PID values 0x0002 – 0x000F are reserved. PID value 0x1FFF is reserved for null packets.

Refer Table 3

Table: PID value discription

Value	Description
0x0000	Program Association Table
0x0001	Conditional Access Table
0x0002	Transport Stream Description Table
0x0003-0x000F	Reserved
0x00010 ... 0x1FFE	May be assigned as network_PID, Program_map_PID, elementary_PID, or for other purposes
0x1FFF	Null packet
NOTE – The transport packets with PID values 0x0000, 0x0001, and 0x0010-0x1FFE are allowed to carry a PCR.	

Table 3

transport_scrambling_control – This 2-bit field indicates the scrambling mode of the Transport Stream packet payload.

The Transport Stream packet header, and the adaptation field when present, shall not be scrambled. In the case of a null packet the value of the transport_scrambling_control field shall be set to '00'

3.2.2 Project Stage 2

Program association Table

The Program Association Table provides the correspondence between a program_number and the PID value of the Transport Stream packets which carry the program definition. The program_number is the numeric label associated with a program

Program_association_section()

{

table_id

section_syntax_indicator

'0'

```
reserved
section_length
transport_stream_id
reserved
version_number
current_next_indicator
section_number
last_section_number
for (i = 0; i < N; i++) {
program_number
reserved
if (program_number = = '0') {
network_PID
}
else {
program_map_PID
}
}
CRC_32 }
```

Calculating the frequency of repetition of PAT pid

The PAT PID in a transport stream is identified by 13 bit PID value .For the packet to be identified as the PAT Packet the 13 bit PID value has to be 0.The logic to calculate the PAT frequency used is to calculate the no of times the PID value 0 is received and adding the time difference between each PID 0 received .

The total time achieved is then divided by the Number of Packets of PAT detected.

3.2.3 Project Stage 3

Implementing PID Control for PAT repetition frequency Error correction

PID (Proportional, Integral, Derivative) control is a widely-used method to achieve and maintain a process set point. The *process* itself can vary widely, ranging from temperature control in thousand gallon vats to speed control in miniature electric motors to position control of an inkjet printer head, and on and on.

Here we have used the PID controller for the Variation in repetition in PAT frequency .

The PID control equation may be expressed in various ways, but a general formulation is:

Drive = kP*Error + kI*Σ Error + kD * dP/dT

Error = Difference between the Actual value of the PAT Repetition frequency and the Desired PAT Repetition frequency.

dP/dT is the rate of change of PAT Repetition frequency.

Σ Error is the summation of Previous error values.

The proportional coefficient kP, the integral coefficient kI, and the derivative coefficient kD are *gain* coefficients which *tune* the PID equation to the particular process being controlled.

Tuning methods: Not all of the terms in the PID equation are necessarily used. Fitting the PID approach to a particular control problem involves *tuning*; deciding which terms to include, and determining what the gains should be for those terms

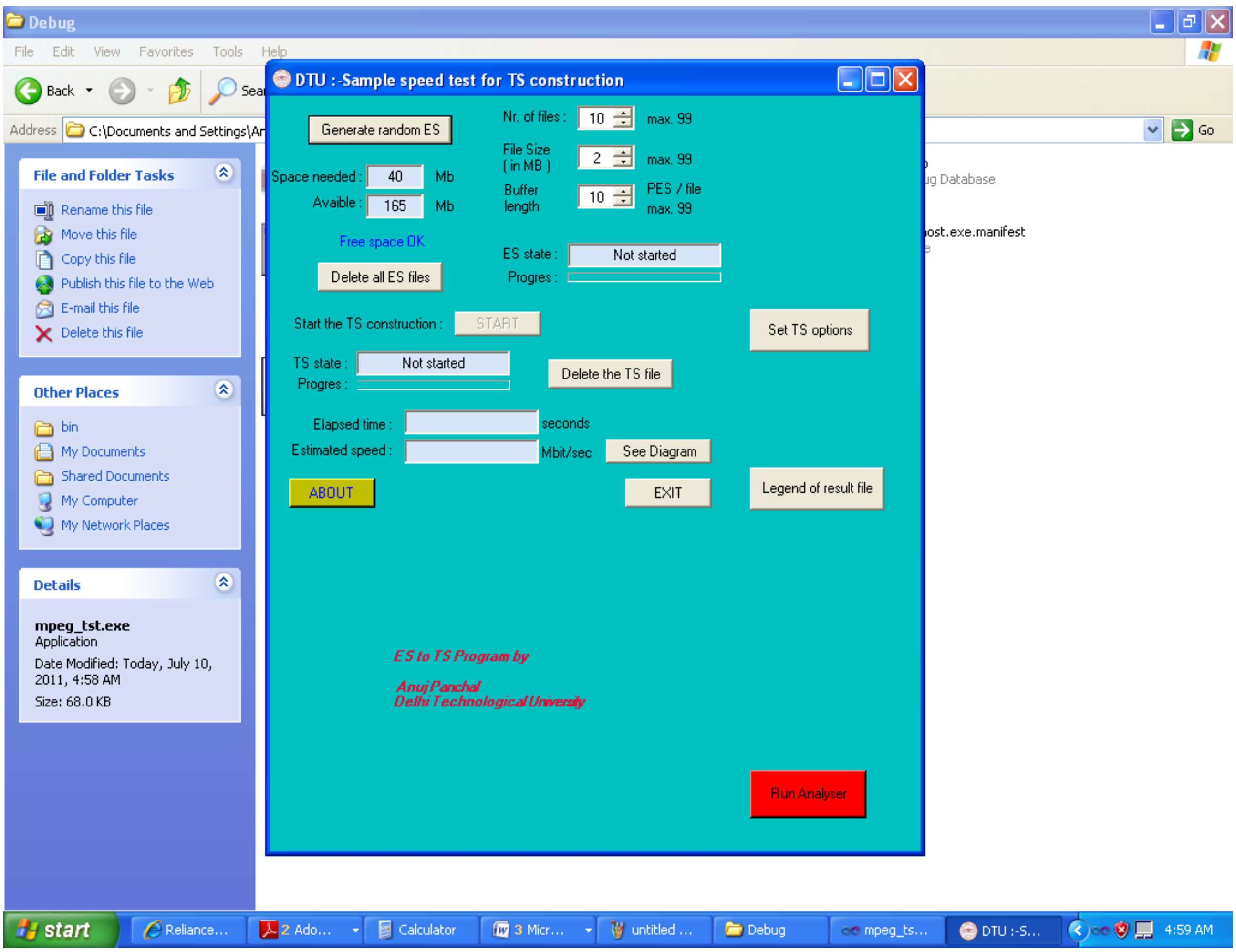
pseudocode to do PID:

```
Start:
     Error = Setpoint - Actual
     Integral = Integral + Error
     Derivative = Last - Actual
     Drive = (Error*kP) + (Integral*kI) + (Derivative*kD)
     Last = Actual
     wait(dt)
GOTO Start
```

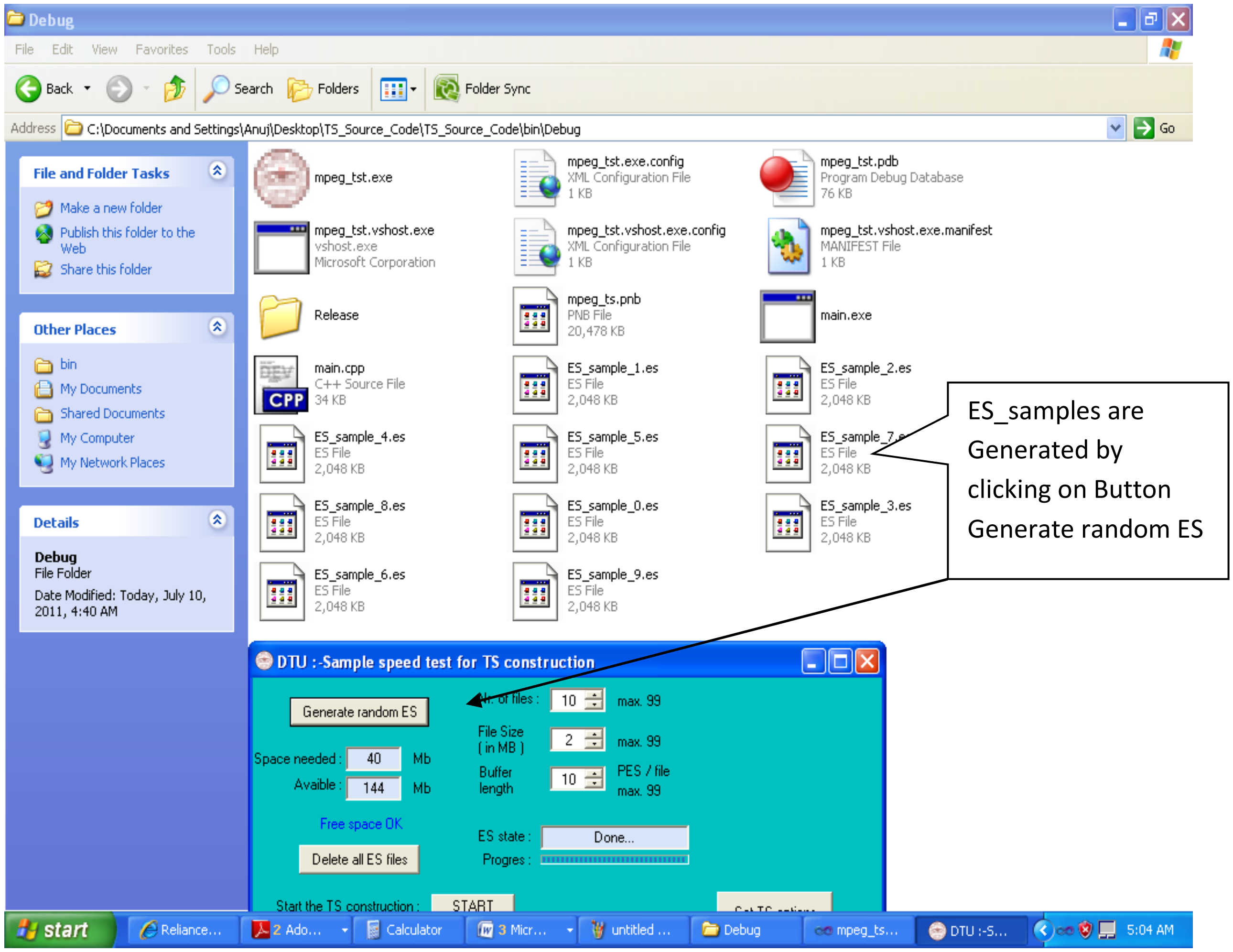
Chapter 4

RESULT ACHIEVED

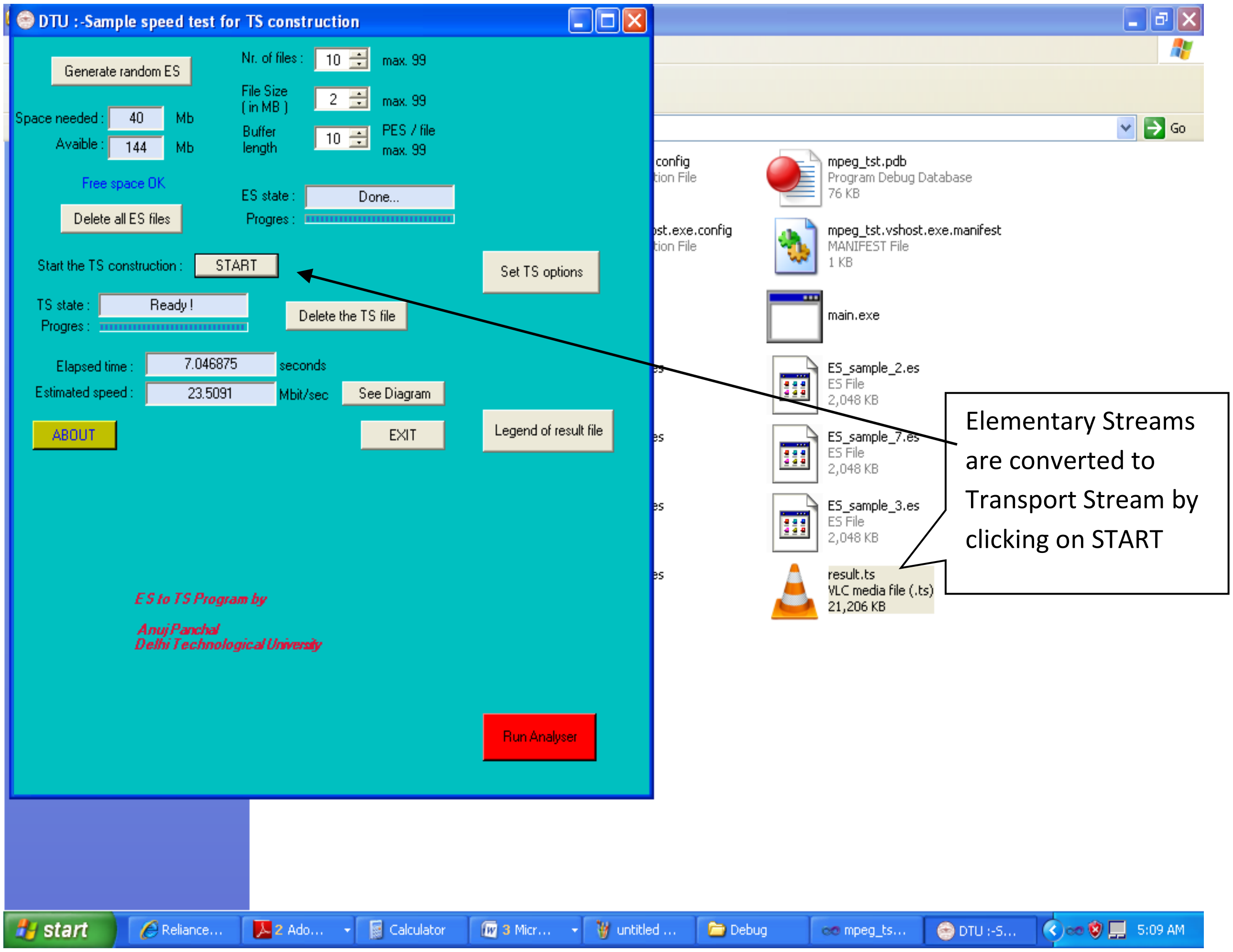
4.1 GUI based transport stream generator with analysing and troubleshooting options



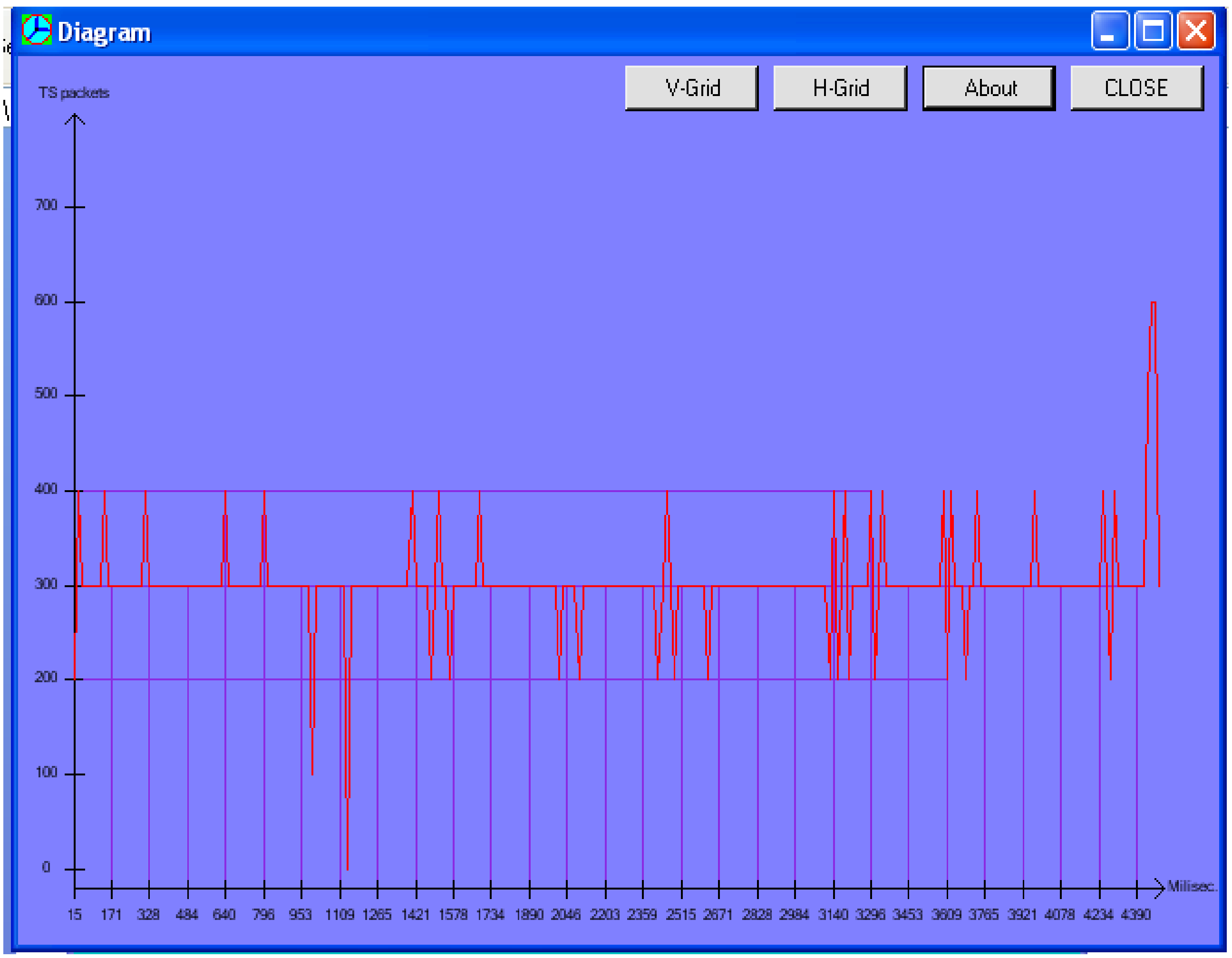
4.2 Generating Elementary Streams



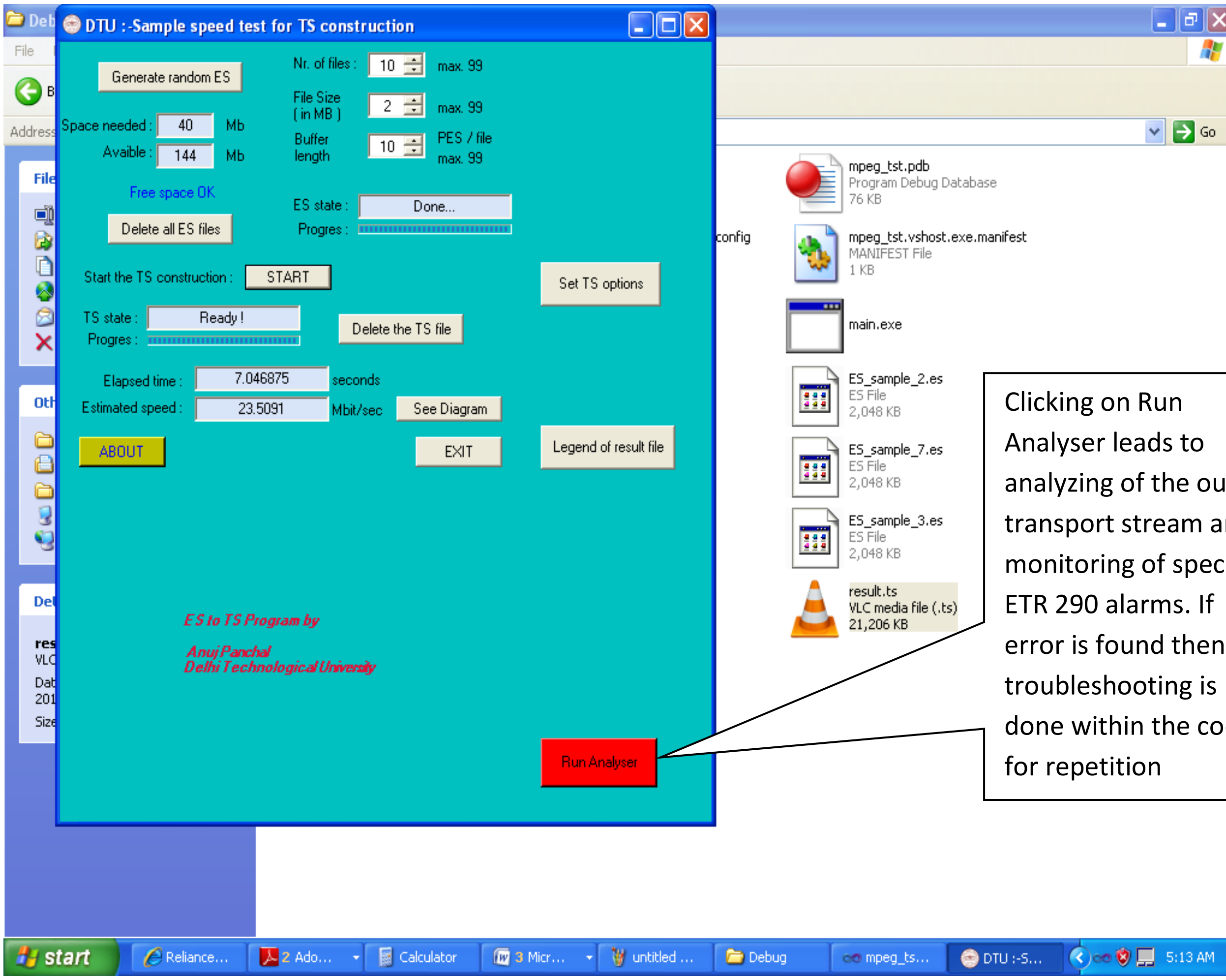
4.3 Generating Transport Stream



4.4 Transport Stream packet generation Diagram w.r.t time in milliseconds. It's a kind of efficiency diagram which show approximately how many TS packets were produced during a period of time. So , if the first value on your horizontal axe ("Milliseconds") it's 30 and the corresponding value on the vertical axe ("TS packets") it's 800 , this means that in the first 30 milliseconds of the simulation 800 TS Packets were produced. Accordingly, if the second value on the horizontal axe it's 54 and the corresponding value on the vertical axe it's 1000, this means that between milliseconds 30 and 54 there were 1000 TS Packets produced, so the system generated 1000 TS Packets in 24 milliseconds. The measurement it's done by taking the time after each group of 100 TS packets. This might slow up the process but it's more appropriate to the real conditions. The "spikes" noticed on the diagram are the cause of the system internals (This program isn't the only one using the CPU, RAM and HDD) ,hence variations are expected.

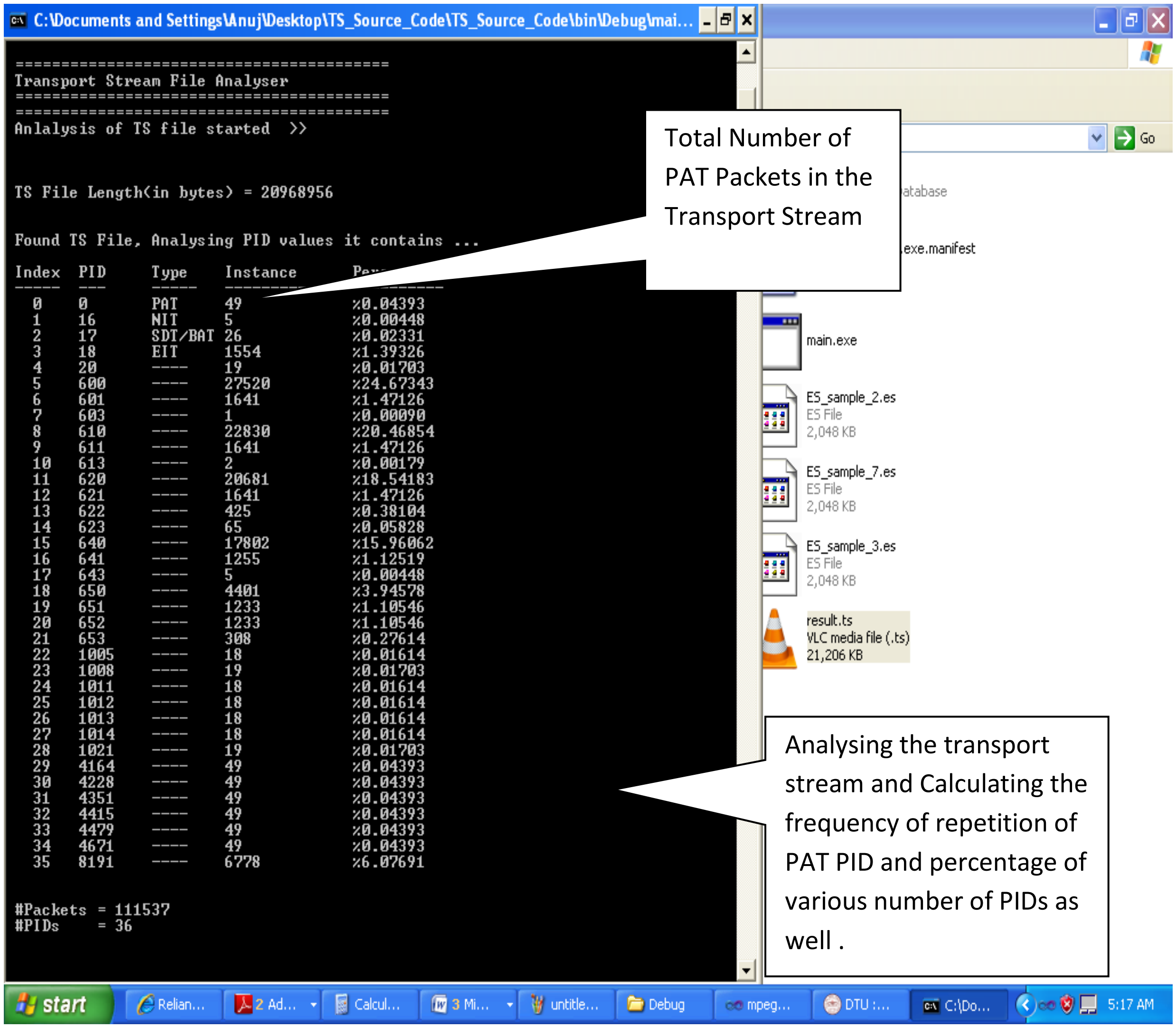


4.5 Running Analyzer

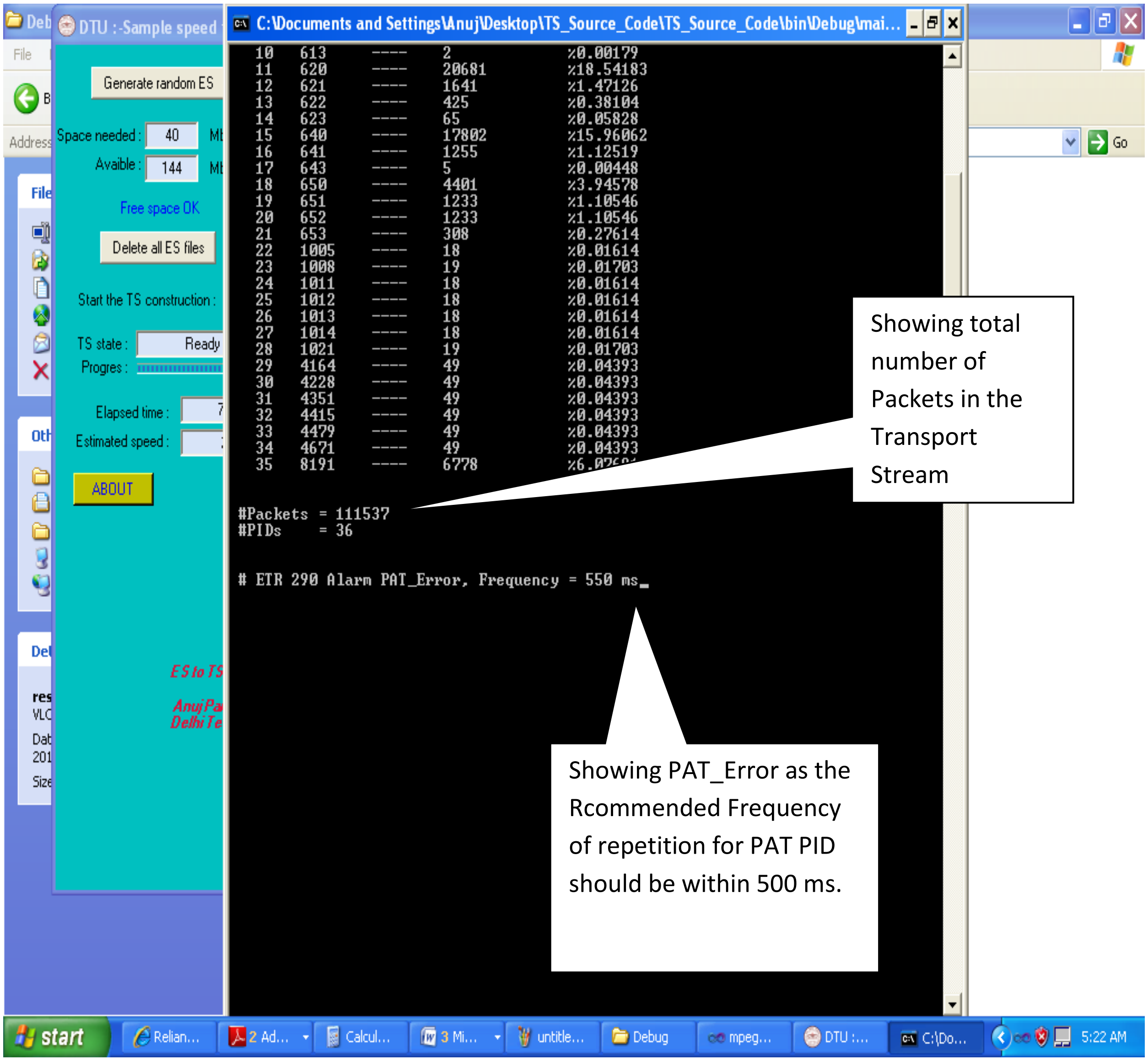


Clicking on Run Analyser leads to analyzing of the output transport stream and monitoring of specific ETR 290 alarms. If error is found then troubleshooting is done within the code for repetition

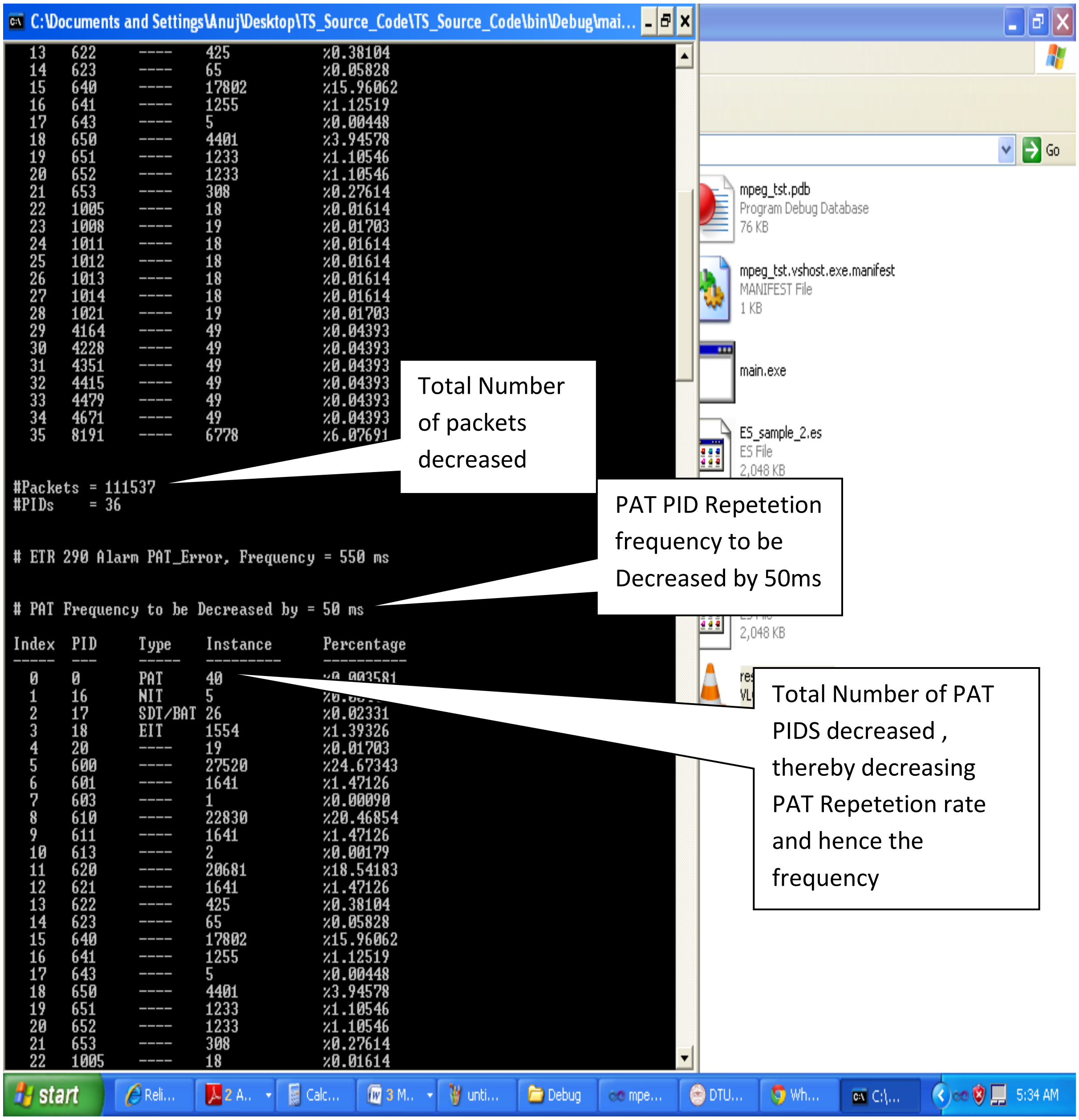
4.6 Transport Stream Analysing And ETR_290 alarm Identification



4.7 Showing Specific ETR-290 Alarm , PAT_Error



4.8 Troubleshooting PAT_Error By Decreasing The Frequency of Repetetion of PAT PID



Chapter 5

THESIS SUMMARY

5.1 Thesis Summary

Dynamic troubleshooting of a specific ETR-290 alarm in a Transport Stream has been done successfully by the means of giving a feedback path to the multiplexer code .

The advantages of this project is that there is no user interference required to change the frequency of the PAT packets. Here we used PID controller to calculate the difference between the actual and the desired PAT Packet frequency and take the action itself .

5.2 Future Direction

The future prospect could be to implement the same logic in troubleshooting the alarms in a real time environment . This means the environment where the error in the Transport Stream is introduced due to the Transmission channel . The Transmission channel can be a Satellite Transmission link or an IP Link.

5.3 Conclusion

We have successfully made the project , the real time dynamic environment has been simulated on PC for the proper functioning of troubleshooting of PAT_Error which is a ETR 290 Priority 1 alarm and results in no decoding of Audio and Video components. The project has been scheduled in two levels , the first level was generation of Transport Stream by multiplexing the elementary streams generated by the software. The generation of Transport stream was successfully achieved as per the ISO standard 13818 -1.The second level was to made the Transport Stream Analyzer which will monitor the Frequency of PAT PID and make an Error Correction as per the DVB Standard.

The second stage was successfully achieved. The Error Detection and Correction is successfully done . The PAT frequency deviation has been successfully corrected by the PID Controller . The PAT Packet frequency Deviation in the Transport Stream file has been detected and corrected by the Algorithm according to the ETR 290 standard.

Bibliography

1] R. Bardelli *C.* Benzi F. Fresia N. Lettieri, Monitoring the Transmission Quality of Digital Television: EUTELSAT's Experience and the TESEUS System (Television Surveillance System for EUTELSAT Satellites, 0-7803-6451-1/2000 IEEE

2] Chuanfei Luo, Jun Sun, Monitoring and Troubleshooting in Operational IP-TV System, 0018-9316© 2007 IEEE

3] *Marwan Krunzt, Ron Sass$, and Herman Hughe,* Statistical Characteristics and Multiplexing of MPEG Streams, 0743-166W95/ 1995 IEEE

4] INTERNATIONAL STANDARD ISO/IEC 13818-1 Information technology — Generic coding of moving pictures and associated audio information: Systems

5] INTERNATIONAL STANDARD ISO/IEC 13818-2 Information technology — Generic coding of moving pictures and associated audio information: Video

6] http://www.csharp-station.com/

7] http://www.functionx.com/csharp/

8] http://csharptutorial.com/mastering-csharp-compiler/

9] http://www.cprogramming.com/

10]ETSI TR 101 290 V1.2.1 (2001-05) *Technical Report* Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems

11]ETSI ETR 290 TECHNICAL May 1997 REPORT Source: EBU/CENELEC/ETSI-JTC Reference: DTR/JTC-00DVB-22

12] Springer Series on Signals and Communication Technology

13] Walter Fischer Digital Video and Audio Broadcasting Technology

11] [FISCHER4] Fischer, W.: MPEG-2 Transport Stream Syntax and Elementary Stream Encoding,

Seminar Documentation, Rohde&Schwarz, Munich, 2001

12] [FISCHER3] Fischer, W.: DVB Measurement Guidelines in Theory and Practice, Seminar Documentation, Rohde&Schwarz, Munich, 2001