TS Manipulator
Requirement Specifications &
High Level Design Document

ECE-612 Project Phase I
George Mason University

Version: 2.1
Revision: 2
Date: 11/02/2008
Author  Mr. Pranav Varsekar

Copyright Notice:
No Copyright.
# Document Revisions History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Author(s)</th>
<th>Reviewed by</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mr. Pranav Varsekar</td>
<td></td>
<td>10/22/2008</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>2</td>
<td>Mr. Pranav Varsekar</td>
<td></td>
<td>11/02/2008</td>
<td>Source Code Updated</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1 Introduction........................................................................................................................................... 4  
1.1 PURPOSE......................................................................................................................................... 4  
1.2 INTENDED AUDIENCE AND READING SUGGESTIONS................................................................. 4  
1.3 REFERENCES.................................................................................................................................... 4  
2 OVERALL REQUIREMENTS...................................................................................................................... 5  
2.1 ASSUMPTIONS....................................................................................................................................... 5  
2.2 PRODUCT FEATURES...................................................................................................................... 5  
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS........................................................................... 5  
3 SYSTEM FEATURES............................................................................................................................... 6  
3.1 FUNCTIONAL REQUIREMENTS......................................................................................................... 6  
4 HIGH LEVEL DESIGN............................................................................................................................. 7  
4.1 OVERALL SETUP OF THE SYSTEM................................................................................................. 7  
4.2 DATA AND CONTROL GRAPH...................................................................................................... 8  
4.3 DATA POOL DIAGRAM................................................................................................................... 9  
4.4 SOFTWARE MODULE..................................................................................................................... 10  
4.5 PROCESSES AND THREADS........................................................................................................... 11  
4.6 HARDWARE & COMMUNICATION INTERFACE........................................................................... 11  
5 REAL TIME CHALLENGES.................................................................................................................... 12  
6 TEST INFRASTRUCTURE REQUIREMENTS........................................................................................ 12  
7 STATUS OF THE PROJECT................................................................................................................... 12  
8 APPENDIX A : GLOSSARY..................................................................................................................... 13

LIST OF FIGURES

1 OVERALL SYSTEM SETUP................................................................................................................... 7  
2 DATA AND CONTROL GRAPH........................................................................................................... 8  
3 DATA AND CONTROL GRAPH........................................................................................................... 9
1 Introduction

1.1 Purpose
This document describes the software requirement specifications and High Level Design of 'TS Manipulator', which aims to modify a MPEG2-TS in-line. The requirement specification is independent of hardware / software platform being used. This specification describes functionality and interfaces of the 'TS Manipulator' and it also covers optional or additional features, which can be incorporated.

1.2 Intended Audience and Reading Suggestions
This document is prepared as a Project of Real Time Embedded system class however the individuals intending to work in MPEG2-TS domain can use it if required.
The document is divided in three sections, first section lays out the Requirement Specifications of the Software system, Second section describes the High Level design and implementation details of the software, and the Third section exhibits the progress so far done on the project.

Project Scope :
The TS Manipulator will be implemented on Linux running onX86 platform. The initial release will incorporate two manipulations over MPEG2-TS, namely, Service blocker (PID blocker) and TS level encryptor. The input and output interfaces will be IP based (UDP/TCP).

NOTE : It is assumed that the person reading this document has basic understanding of the MPEG2 TS infrastructure.

1.3 References

- ISO/IEC 13818 Generic coding of moving pictures and associated audio information.
- ETSI documentation of DVB-S and DVB-C
- www.mpeg.org
2 Overall Requirements

2.1 Assumptions

- The software will be realized on a platform which has at least two IP interfaces ie. one for input and another for output.
- The maximum bit rate of the input and output data stream is 40 Mbits/S
- The IP channel has 100 MBps or better bandwidth.

2.2 Product Features

The following is the summary of salient features provided by the MPEG-2TS Manipulator.

- The software should be able to receive MPEG2-TS data streamed on UDP (DVB over IP) on one of the IP interfaces of the hardware.
- The software should be able to parse the system tables of the DVB and display the related information on the UI.
- UI should be capable enough to display the related information efficiently at the same time it should allow the user to enter manipulation parameters.
- The software should be capable enough to sustain the high bit rate data while manipulating TS, the packet drop incidences should be as minimum as possible.
- The output bit rate of TS should be equal to input bit rate of the TS with 0.5% maximum variation or it will result in malfunctioning of the modulators.

2.3 Design and Implementation Constraints

General constraints are listed below for more constraints specific to the platform please refer to appendix.

- Input and output bit rate of the data stream can not be more than 40 Mbits/S. To determine exact maximum supported bit rate, bench marking has to be done on the fully functional software.
- All the parsing and encapsulation is done in software, so the overall CPU consumption will be very high. For better performance and minimum data loss run the software without X-Server or X-Server with Minimum processes running.
## 3 System Features

**Functional Requirements**

<table>
<thead>
<tr>
<th>Requirement code</th>
<th>Functional requirement</th>
<th>List of anticipated error condition</th>
<th>Corresponding response</th>
</tr>
</thead>
<tbody>
<tr>
<td>F001</td>
<td>Receiver the DVB-IP multicast / unicast at given IP address and Port.</td>
<td>Loss of data due to very high incoming bit rate.</td>
<td>Parsing and manipulation of TS is distributed to various dedicated sets of threads/processes. Bench marking to be done for maximum supported input bit rate</td>
</tr>
<tr>
<td>F002</td>
<td>Parsing of Various system Tables of the TS and its display on the UI</td>
<td>Parsing is very CPU centric, so it may result in data loss due to lack of CPU BW available to TS manipulator threads.</td>
<td>TS system table information is redundant most of the times, so don't parse system tables if their version number has not changed.</td>
</tr>
<tr>
<td>F003</td>
<td>QT or Text base UI</td>
<td>Replacing existing TS packets with NULL packets is CPU centric and can result in data drop. Manipulation of PAT is tricky.</td>
<td>Set efficient run time priorities of the threads. Bench mark for maximum input data rate supported.</td>
</tr>
<tr>
<td>F004</td>
<td>Service blocker (PID blocker)</td>
<td>Encryption is CPU centric and can result in data drop.</td>
<td>Set efficient run time priorities of the threads. Bench mark for maximum input data rate supported.</td>
</tr>
<tr>
<td>F005</td>
<td>TS Packet encryptor.</td>
<td>Sending Encryption key over TS is an issue.</td>
<td>Set efficient run time priorities of the threads. Bench mark for maximum input data rate supported.</td>
</tr>
</tbody>
</table>
4 High Level Design Documentation

4.1 Overall Setup of the System

The overall setup of the TS Manipulator is as shown in the diagram:

The System has following setup:
- A DVB-IP streamer which streams DVB over UDP on a predefined multicast IP and port.
- ‘TS Manipulator’ software installed on a Linux machine which is having two Ethernet ports.
- The IP channel is assumed to have 100MB/S or better bandwidth, so that it can sustain the higher media bit rates.
- A DVB-IP player installed in the same network.
4.2 Data & Control Graph

Where,

1 : Input Thread, Receives DVB-IP Packets and writes it into TS-Pool
2 : Software Demux Thread, Parses TS and Filters A/V & Sys. Table Packets
3 : A/V Packet Manipulator.
4 : System Tables Manipulator
5 : Modified TS Pool Writer
6 : Output Thread, Transmits DVB-IP Packets

C : If the TS Packet is a A/V Packet
!C : If the TS Packet is a System Table Packet
4.3 Data Pool Diagram

The Data Pool Diagram is a graphical representation of the flow of the input media data through various internal buffers. This representation is independent of any involved processes.

**Pool sizes calculation:**
Assuming 40 Mbits/S (5MB/S) data rate of the input TS.

- Input TS Pool : 30 MB
- Output TS Pool : 30 MB
- A/V TS Packet Pool : 25 MB
- Modified A/V TS Packet Pool : 25 MB
- System Table PID Pool : 5 MB
- Modified System Table PID Pool : 5 MB
4.4 Software Modules

1. TS Pool Reader/Writer:
   TS Pool has following characteristics:
   - Byte align data read write.
   - Single Reader single Writer interface.
   The C code of the TS Pool module is given in Section 7.

2. TS Packet Pool Reader/Writer:
   TS Packet Pool has following characteristics:
   - TS Packet size (188 bytes) align data.
   - Multiple Writers (in this case two) Single reader interface.

3. UDP Receiver/Transmitter:
   This module performs following tasks:
   - Receive DVB-IP data and write it in input TS Pool.
   - Read TS data from output TS Pool and Multicast it over IP as DVB_IP.

4. MPEG2-TS Demultiplexer and Modifier:
   This is the key module in the system. This module performs following tasks:
   - Read TS data form input TS Pool.
   - Parse the TS data to filter A/V PIDs and System table PIDs (eg. PAT, PMT).
   - Display the available services in the given TS on the UI.
   - Encrypt the TS packets of user defined PIDs and write them in to output TS Pool.
   - Replace the TS packets of user defined PIDs with NULL packet and write them into output TS Pool.
   - Edit the PAT to reflect the modifications, packetize the PAT into TS packet and insert it in the output TS Pool.

5. User Interface:
   For the first phase the UI would be a simple Text interface which will display following information:
   - Available services in the TS
   - User defined Service to be encrypted or to be nullified.
   - Input bit rate
   - Output bit rate.
   For the second face the UI will be designed in QT which will display the above mentioned information.

4.5 Processes And Threads

The task of the software is distributed in following threads:

1. TS Receiver Thread:
   This thread performs following tasks,
   - Reception of DVB-IP data streamed on predefined multicast IP and Port.
   - Write received TS packets in in Input TS Pool.

2. Software Demux Thread:
This thread performs following tasks,

- Read TS Packets from Input TS Pool.
- Parse the TS for System Tables and pass the available services to UI thread for display on UI.
- Determine the type of the TS Packet.
- If the TS Packet is A/V Packet, write it into A/V TS Packet Pool, else write the TS packet into System Table PID Pool.

3. **UI Thread**:
   This thread performs following tasks:
   - Take the Input and Output multicast IP and Port from the user.
   - Initiate all the other threads. If the other threads are already running then reset them.
   - Display available services in the TS as parsed by Software demux thread.
   - Take input from the user to determine which services to be blocked or encrypted.

4. **TS Manipulator Thread**:
   This thread performs following tasks,
   - Query UI thread to determine which PIDs to be manipulated.
   - If a service has to be blocked, replace the all the TS packets corresponding to that service NULL TS packet and write the packet into Modified A/V TS Pool. All other TS packets are copied as it is.
   - If a service has to be encrypted, encrypt payload of all the TS packets corresponding to that service with a predetermined KEY and copy the packet into Modified A/V TS Pool. All other packets are copied as it is.
   - Modify the PAT to reflect the changes done to a TS, convert the PAT into TS Packets and write them into Modified System Table PID Pool. Replace the TS packet corresponding to the PMT of the blocked service with a NULL packets and write them into Modified System Table PID Pool. All other TS packets of System Tables are copied as it is.

5. **Output TS Generator Thread 1**:
   - This thread reads the Modified A/V TS Pool and writes the TS packet into Output TS Pool.

6. **Output TS Generator Thread 1**:
   - This thread reads the Modified System Table PID Pool and writes the TS packet into Output TS Pool.

7. **TS Transmitter Thread**:
   - This thread reads the Output TS Pool and Multicast it on a predetermined IP and Port.

4.6 **Hardware & Communications Interfaces**

The required Hardware and Communication interfaces are as follows:

- X86 Computer with clock 1GHz or better.
- Memory 1GB or more.
- Ethernet 1: For DVB-IP input stream.
- Ethernet 2: For DVB-IP output stream.

**NOTE**: However, for the first face, only one interface will be used for input and output.
5 Real Time Challenges

The Software has to address following Real Time requirements:

- Maintaining output data rate equal to input data rate.
- TS Parsing and manipulation is a very CPU centric operation, preventing data loss is an extremely critical task.

6 Test Infra-Structure Requirements

For Module Testing:

- A VLC player installed computer for streaming the TS.
- A Linux computer to run the TS Manipulator software.
- A VLC installed computer to play the modified TS.

NOTE: However, for the first face, all three tasks will be done on a single computer running Linux.

7 Status Of The Project

The status of the software modules is as follows:

1. TS Pool Reader/Writer
   - Fully functional
   - Done
2. TS Packet Pool Reader/Writer
   - Single Reader Writer done
   - Partial
3. UDP Receiver/Transmitter
   - Fully functional
   - Done
4. MPEG2-TS Demultiplexer and Modifier
   - Design done
   - Partial
5. User Interface
   - Design initiated
   - Not Done
# Appendix A: Glossary

- **MPEG**: Motion Pictures Expert Group.
- **DVB**: Digital Video Broadcasting
- **DVB-S**: Digital Video Broadcasting over Satellite Medium
- **DVB-C**: Digital Video Broadcasting over Cable Medium
- **DVB-IP**: Digital Video Broadcasting over IP Medium
- **TS**: Transport Stream.
- **PAT**: Program association Table.
- **PMT**: Program Map Table.