27M Technologies AB
Diskettgatan 11
SE-583 35 LINKÖPING
SWEDEN

October 28, 2008

http://www.27m.com/
+46 13 21 81 20

Digital Television - MPEG/DVB

overview — slides

Leif Stensson

Important Terms

Some important terms that will be covered in this presentation:

- MPEG (Moving Pictures Expert Group standard)
- DVB (Digital Video Broadcast standard)
- Program / Service (layman's term "TV channel")
- Event (layman's term "TV program")
- Network vs Original Network
- Transponder / Multiplex (Mux)
- Transport Stream: a multiplexed collection of Elementary Streams
- Elementary Stream
- Packet
- Section
- Table
- Descriptor
- Descriptor Loop

The Signal Path

The data path from the source to the destination in a typical digital television scenario is:

- **Content sourcing**. Someone provides the actual sound and images.
- **Encoding**. The various forms of data (audio, video, subtitling, other data) are encoded into a binary form.
- **Multiplexing**. The separate components, called Elementary Streams, that make up a television service are joined, and multiple services that will be broadcast on the same transmission channel are also joined, producing a Transport Stream.
- **Modulation**. The digital data is converted to some kind of analogue signal.
- **Broadcast**. The analogue signal is sent out.
- **Reception**. The analogue signal is picked up via an antenna.
- **Demodulation**. The analogue signal is converted back into a digital data sequence representing the Transport Stream.
- **Demultiplexing**. The Transport Stream is broken up into the individual Elementary Streams that make it up.
- **Decoding/Parsing**. Some of the data from the Elementary Streams is extracted and processed.
- **Presentation**. Some of the extracted data is presented to the user.

MPEG

The MPEG standard (ISO13818) provides definitions for:

- Transport Layer Packets
- Data Transmission
- Video Encoding
- Audio Encoding
- Tables
- Descriptors
- Programs
- Program Clock Reference (PCR)
- a few Table types, where the three most important are:
  - Program Assocation Table (PAT) (0x00)
  - Conditional Access Table (CAT) (0x01)
  - Program Map Table (PMT) (0x02)
- some Descriptor types, such as:
  - `CA_descriptor` (0x09).
  - `ISO_639_language_descriptor` (0x0A).

MPEG — Streams

The Elementary Stream consists of a sequence of data sections, typically containing one kind of data, such as audio, video, Tables, or something else.

The Transport Stream (TS) consists of a sequence of Packets of one or more Elementary Stream multiplexed into one common stream. All Packets are tagged with a Packet ID (PID) which indicates which Elementary Stream the Packet belongs to. Since the PID is essentially the name of the Elementary Stream, the term "PID" is often casually used as a near-synonym for Elementary Stream.

A few PID values are reserved for special functions. In MPEG, PID 0 is defined to contain only Program Association Tables (PAT). The PAT serves as an index of all Programs that are present in the stream, giving each a Program Number and a PID indicating where the Program's Program Map Table (PMT) can be found.

The Program Map Table (PMT) is an index for an individual Program indicating which Elementary Streams belong to that Program. It also contains information about those Elementary Streams, and also some general information about the Program.

Table Format

The Tables that MPEG defines are all encoded as one or more Sections consisting of:

- a 64-bit header,
- the actual data for the part of the Table that is contained the Section, and
- a 32-bit checksum

The header consists of:

- an 8-bit field giving the Table type
- a 4-bit field with special functions
- a 12-bit length field
- a 16-bit Table Extension field,
- an 8-bit version field,
- an 8-bit current Section number field
- an 8-bit last Section number field

The Table Extension is usually an identifier for the object that the Table provides information about. For instance, the Table Extension of the PAT is the Transport Stream ID, and the Table Extension of the PMT is the Program Number.

With a 12-bit length field and an 8-bit Section number field, it is possible to make tables of up to 2^{12+8} bytes (including the header and checksum), meaning 1 megabyte. But some tables have further size restrictions on the size of individual sections (typically 2^{10} bytes), and the PMT had an additional restriction of only being allowed to have one Section.

The MPEG Table format is also used by DVB.

Descriptor Format

MPEG defines a format for Descriptors, which is also used by DVB. The Descriptors have a very simple general format:

- an 8-bit type tag field
- an 8-bit length field
- a data field consisting of as many bytes as indicated by the length field.

but the interpretation of the data field is defined separately for each Descriptor type.

Descriptors normally occur in lists called “Descriptor Loops”, which usually have a 12-bit length field indicating the length of all the Descriptors in the Loop, and then the actual Descriptors. But sometimes, when a Descriptor Loop occurs at the end of some data that already has a known length, the length field is skipped.

DVB

The DVB standard (basic document EN 300 468) extends the MPEG standard by providing definitions for:

- Services
- Networks
- Bouquets
- Events
- Time and Date
- several Table types, such as:
  - Network Information Table (NIT) (0x40, 0x41)
  - Service Description Table (SDT) (0x42, 0x46)
  - Bouquet Association Table (BAT) (0x4A)
  - Event Information Table (EIT) (0x4E…0x6F)
  - Time Date Table (TDT) (0x70)
  - Running Status Table (RST) (0x71)
  - Time Offset Table (TOT) (0x73)
- several more Descriptor types, such as:
  - `network_name_descriptor` (0x40).
  - `bouquet_name_descriptor` (0x47).
  - `service_descriptor` (0x48).
  - `linkage_descriptor` (0x48).
  - `short_event_descriptor` (0x4D).
  - `extended_event_descriptor` (0x4E).
  - `teletext_descriptor` (0x56).
  - `AC-3_descriptor` (0x6A).

The term Service that appears in DVB is “the same thing” as the Programs in MPEG, except that DVB extends the concept by adding new attributes to the Service that a plain MPEG Program doesn’t have (such as a Running Status, a user-readable name, and information about what Original Network the Service belongs to).

Using MPEG and DVB together

- The **Program Association Table** provides the master index of all Programs/Services on a transponder.
- For each Program/Service, a **Program Map Table** provides basic information about the composition of that Program/Service.
- The **Network Information Table** provides general information about the low-level network, sometimes also including information about other networks.
- The **Service Description Table** provides information about individual Services, extending the basic MPEG Program information.
- The **Time Date Table** and **Time Offset Table** provides information about wall-clock time in the network.

MPEG and DVB together: Channel Search

Channel searching is a task that any Set-Top Box must typically do before it can show any Services, and it is perhaps that task that most directly summarizes the practical relation between several important MPEG and DVB concepts. The exact procedure for a channel search can vary a bit, but the following steps are typical:

1. find a transport stream (tune to a broadcast frequency),
2. wait for the PAT to appear on PID 0 to determine what Programs are available,
3. for each Program, wait for the PMT for that program to appear on the PMT PID listed in the PAT,
4. for each Program, search the SDT for information about the Services that corresponds to the Program (and in particular, find out what kind of Service it is, and what it's name is),
5. search the NIT for information about other Transport Streams, and for each such stream that has not already been searched, repeat the channel search procedure for each such Transport Stream.

Example: PAT — Swedish Terrestrial MUX1

Here is an example from a live broadcast of the Swedish terrestrial MUX1, where the raw data of the Program Association Table was:

```
00 b0 4d 03 fd c3 00 00 00 00 e0 10 14 3c f4 3c
13 ba f3 ba 13 ce f3 ce 15 a4 f5 a4 04 d8 e4 d8
16 08 f6 08 03 f2 e3 f2 16 d0 f6 d0 16 a8 f6 a8
05 0a e5 0a 05 00 e5 00 03 66 e3 66 ff fe e7 12
0b c2 eb c2 0b cc eb cc 0b d6 eb d6 e8 ca 31 cb
```

The first eight bytes are the Table header, which decodes as follows:

00	Table Type for PAT
b0 4d	Size: 0x04d bytes after the length field
03 fd	Table Extension: 0x03fd (TSID)
c3	Table version information
00 00	This section is number 0, last section is number 0

The last four bytes are a 32-bit CRC checksum. The rest is a list of 32-bit entries containing the following information:

Program 0(0x0)	PID=16(0x10)
Program 5180(0x143c)	PID=5180(0x143c)
Program 5050(0x13ba)	PID=5050(0x13ba)
Program 5070(0x13ce)	PID=5070(0x13ce)
Program 5540(0x15a4)	PID=5540(0x15a4)
Program 1240(0x4d8)	PID=1240(0x4d8)
Program 5640(0x1608)	PID=5640(0x1608)
Program 1010(0x3f2)	PID=1010(0x3f2)
Program 5840(0x16d0)	PID=5840(0x16d0)
Program 5800(0x16a8)	PID=5800(0x16a8)
Program 1290(0x50a)	PID=1290(0x50a)
Program 1280(0x500)	PID=1280(0x500)
Program 870(0x366)	PID=870(0x366)
Program 65534(0xfffe)	PID=1810(0x712)
Program 3010(0xbc2)	PID=3010(0xbc2)
Program 3020(0xbcc)	PID=3020(0xbcc)
Program 3030(0xbd6)	PID=3030(0xbd6)

Note 1: Program 0 is not a real program; it's used to indicate the PID where network information is broadcast.

Note 2: the Swedish operator, Teracom, has chosen to use the same number for PID and Program Number for most of the programs, so Program 870 has it's PMTs on PID 870. But not all operators do this, and it isn't possible to do it generally, since the Program Number is a 16-bit value, but the PID is only a 13-bit value.

Example: PMT — Swedish Terrestrial MUX1

Continuing with the same recorded data as for the PAT in the previous example, this was the raw data of the PMT on PID 1010, for Program 1010:

```
02 b0 34 03 f2 ed 00 00 e3 fb f0 00 06 e3 ec f0
07 56 05 73 77 65 09 00 03 e3 fa f0 00 06 e3 f9
f0 0c 6a 01 00 52 01 bd 05 04 41 43 2d 33 02 e3
fb f0 00 a1 d9 4a 10
```

The first eight bytes are the Table header:

02	Table Type for PMT
b0 34	Size: 0x034 bytes after the length field
03 f2	Table Extension: 0x03f2 (Program Number)
ed	Table version information
00 00	This section is number 0, last section is number 0

After the header, the first two bytes (`e3 fb`) are the PCR PID, but only the lowest 13 bits are used, giving the value: 0x03fb.

After the PCR PID, there is a Program Descriptor Loop, with first 4 unused bits, then a 12-bit length field, and then the Descriptor data. But in this case, the data is `f0 00`, giving a value of 0 for the length field, meaning no Descriptor data is present.

The remaining data is a list of Elementary Streams, which decodes to the following information:

```
ESPID=1004(0x3ec), type=6(0x6)
    teletext_descriptor
        (RAW DATA: 56 05 73 77 65 09 00)
ESPID=1018(0x3fa), type=3(0x3)
ESPID=1017(0x3f9), type=6(0x6)
    AC-3_descriptor
        (RAW DATA: 6a 01 00)
    stream_identifier_descriptor, tag: 189(0xbd)
        (RAW DATA: 52 01 bd)
    registration_descriptor
        (RAW DATA: 05 04 41 43 2d 33)
ESPID=1019(0x3fb), type=2(0x2)
```

Example: NIT — Swedish Terrestrial MUX1

Continuing with the same recorded data as for the previous examples, the Network Information Table (NIT) on PID 16 contained the following four parts of data: the header, the Network Descriptor Loop, the Transport Stream Descriptor Loop, and the final 32-bit checksum.

```
40 f0 e5 31 a2 c3 00 00
                        f0 5a 5f 04 4f 54 56 00
4a 10 03 fd 22 f1 ff fe 02 00 00 00 00 45 49 54
2b bf 4a 0c 04 56 22 f1 11 bc 09 04 00 10 95 00
4a 15 04 56 22 f1 12 5c 09 0d 00 02 78 09 94 36
00 00 00 01 01 10 01 4a 0c 04 57 22 f1 12 66 09
04 08 00 46 00 40 0d 54 65 72 61 63 6f 6d 5f 4d
75 78 5f 31
            f0 7e 03 fd 22 f1 f0 78 5f 04 00 00
00 29 5a 0b 03 1c 82 40 1f 81 12 ff ff ff ff 41
21 03 66 01 16 d0 01 16 a8 01 0b c2 01 0b cc 01
0b d6 01 05 00 01 05 0a 01 15 a4 01 16 08 01 04
d8 01 83 40 16 d0 c0 01 16 a8 c0 60 16 08 c0 02
15 a4 c0 61 04 d8 c0 63 03 66 c0 62 0b c2 c3 21
0b cc c3 22 0b d6 c3 23 05 00 c0 5b 05 0a c0 5c
03 f2 40 00 13 ce 40 00 14 3c 40 00 13 ba 40 00
ff fe 40 00
            44 d5 71 2f
```

The header is similar to other tables, but the Table type is `40`: NIT-actual (meaning NIT for the current network).

The first descriptor of the Network Descriptor Loop consists of the data `5f 04 4f 54 56 00`. The first byte in a Descriptor is always the Descriptor tag, and the second byte is always the length of the rest of the Descriptor (after the first two bytes), so we know that this Descriptor is 6 bytes long by the fact that the value of the second byte is 4, and $2 + 4 = 6$.

The Descriptor tag of the first descriptor is `5f`, which makes that descriptor a `private_data_specifier_descriptor`.

Similarly, the last descriptor of the Network Descriptor Loop consists of the data `40 0d 54 65 72 61 63 6f 6d 5f 4d 75 78 5f 31`. Here, the Descriptor tag is `40`, making it a `network_name_descriptor`, giving the network name: "`Teracom_Mux_1`". (`54` is ASCII "T", `65` is ASCII "a", and so on.)

Example: NIT (continued)

The remaining part of the raw NIT data is the Transport Stream Loop. In this case, it only contains one Transport Stream entry, which decodes as follows:

```
private_data_specifier_descriptor
    RAW DATA: 5f 04 00 00 00 29
terrestrial_delivery_system_descriptor, f=522000 kHz
    RAW DATA: 5a 0b 03 1c 82 40 1f 81 12 ff ff ff ff
service_list_descriptor
  SID=870(0x366), type=1(Digital TV)
  SID=5840(0x16d0), type=1(Digital TV)
  SID=5800(0x16a8), type=1(Digital TV)
  SID=3010(0xbc2), type=1(Digital TV)
  SID=3020(0xbcc), type=1(Digital TV)
  SID=3030(0xbd6), type=1(Digital TV)
  SID=1280(0x500), type=1(Digital TV)
  SID=1290(0x50a), type=1(Digital TV)
  SID=5540(0x15a4), type=1(Digital TV)
  SID=5640(0x1608), type=1(Digital TV)
  SID=1240(0x4d8), type=1(Digital TV)
    RAW DATA: 41 21 03 66 01 16 d0 01 16 a8 01 0b c2 01 0b cc
              01 0b d6 01 05 00 01 05 0a 01 15 a4 01 16 08 01
              04 d8 01
non-standard descriptor, tag 0x83, len=64
   RAW DATA: 83 40 16 d0 c0 01 16 a8 c0 60 16 08 c0 02 15 a4
             c0 61 04 d8 c0 63 03 66 c0 62 0b c2 c3 21 0b cc
             c3 22 0b d6 c3 23 05 00 c0 5b 05 0a c0 5c 03 f2
             40 00 13 ce 40 00 14 3c 40 00 13 ba 40 00 ff fe
             40 00
```

The descriptor with tag `83` is not defined by the basic DVB standard, but it is de facto the standard way of transmitting Logical Channel Number (LCN) information. But not all operators use this.

Example: SDT — Swedish Terrestrial MUX1

The Service Description Table (SDT) provides DVB information about individual Services. The Transport Stream used in the previous examples contained a fairly large SDT, so we'll settle for just looking at one part of it. Between the header and the checksum, the SDT contains a list of Service entries, consisting of a few header bytes and then a Descriptor Loop with information about the Service. As an example, the entry for Service 1240 was:

```
                      04 d8 fd 80 29 48 1c 01 13
53 76 65 72 69 67 65 73 20 54 65 6c 65 76 69 73
69 6f 6e 06 53 56 54 32 34 20 5f 04 00 00 00 14
f1 03 00 63 00
```

The contents of the Descriptor Loop for this Service decodes to:

```
service_descriptor, service_type=1(Digital TV)
  provider: "Sveriges Television"
  service: "SVT24 "
    RAW DATA: 48 1c 01 13 53 76 65 72 69 67 65 73 20 54 65 6c
              65 76 69 73 69 6f 6e 06 53 56 54 32 34 20
private_data_specifier_descriptor
    RAW DATA: 5f 04 00 00 00 14
non-standard descriptor, tag 0xf1, len=3
    RAW DATA: f1 03 00 63 00
```

Note: for some reason, the operator has put a space at the end of the Service name.

Example: Descriptors

The `service_descriptor` that appeared in the SDT example is a typical example of a DVB descriptor. The raw data was:

```
48 1c 01 13 53 76 65 72 69 67 65 73 20 54 65 6c
65 76 69 73 69 6f 6e 06 53 56 54 32 34 20
```

which decodes as:

48	Tag value 0x48: service_descriptor
1c	Length: 0x1c bytes after the length field
01	Service type is 1: Digital Television
13	Length of Provider Name
53 76 ... 6f 6e	Text: "Sveriges Television"
06	Length of Service Name
53 56 54 32 34 20	Text: "SVT24 "

The internal structure of the data (after the length field) is defined separately for each descriptor, in this case by the standard document EN 300 468.

As an of an MPEG descriptor (defined by ISO13818), we can look at the `ISO_639_language_descriptor`. The only examples of such a descriptor in the Swedish terrestrial MUX1 (that has been used in the other examples) is the following raw data:

```
0a 04 74 74 72 03
```

which decodes to:

0a	Tag value 0x0a: ISO_639_language_descriptor
04	Length: 4 bytes after the length field
74 74 72	Language code: "ttr"
03	Audio type is 3: Visually Impaired Commentary

Normally, the three-letter language code should be an ISO639 language code, but in this case, Swedish television uses what they call "talande textremsa" (speaking subtitle). It is not very good use of the standard to use the three-letter text field to indicate this kind of information, since the Audio Type field already informs us in a standardised way that this is commentary for the visually impaired. (So a more correctly use of the descriptor would have been to code it as `0a 04 73 77 65 03`, where the language code is "swe" for Swedish.)