

Universidad Técnica Federico Santa
María
Campus Casa Central
Valparaíso, Chile



“AVANCE ANÁLISIS DE PAQUETES MPEG-2 TS ”

Identificación De Frames

DESARROLLADO POR:

Sebastián Castillo Araya

CORREO :

sebastian.castilloar@sansano.usm.cl

FECHA:

08 Febrero 2023

Índice

1	Introducción	2
2	Paquetes MPEG-2 Transport Stream	3
3	Análisis usando Wireshark	6
3.1	Paquete PES únicamente con Payload	7
3.2	Paquete PES con Adaptation Field y Payload	9
4	Programa en C para identificar frames	11
5	Trabajo futuro	12
6	Conclusión	13
	Referencias	13

Índice de tablas

1	Cabecera TS.	3
2	Valores Adaptation Field Control.	3
3	Valores Frame type.	5

Índice de figuras

1	Paquete TS y cabecera.	3
2	Valor bits 01, solo payload.	4
3	Valor bits 11, adaptation field seguido de payload.	4
4	Paquete PES y cabecera.	4
5	Cabecera paquete PES.	4
6	Contenido paquete PES.	5
7	Cabecera paquete PES.	5
8	Captura de paquetes Wireshark.	6
9	Paquete TS solo con payload.	7
10	Cabecera paquete PES solo con payload.	7
11	Continuación cabecera PES solo con payload.	7
12	Contenido paquete PES con payload.	8
13	Cabecera paquete TS con adaptation field y payload.	9
14	Contenido campo Adaptation Field.	9
15	Cabecera paquete PES con AF y Payload.	9
16	Continuación cabecera PES con AF y Payload..	10
17	Contenido paquete PES con AF y Payload	10
18	Programa en C capturando e identificando paquetes MPEG2 TS.	11

1. Introducción

El presente informe es una recopilación de los avances obtenidos durante la investigación del estándar MPEG-2 Transport Stream (MPEG-2 TS). El objetivo es describir de forma general la estructura de los paquetes MPEG-2 TS para posteriormente identificar los tipos de frames presentes en los paquetes que corresponden a video.

Para esto en primer lugar se presenta contenido teórico sobre la composición de los paquetes y posteriormente se realiza una prueba práctica mediante uso Wireshark, script bash y programas en C para corroborar la información.

Debido a que el análisis está plenamente basado en la identificación de frames, se omiten varios tópicos presentes en el estándar MPEG-2, por lo tanto se recomienda al lector revisar las referencias si es que desea conocer más en profundidad el tema.

2. Paquetes MPEG-2 Transport Stream

La estructura de los paquetes MPEG-2 TS (también llamados TS) se encuentra definida en el estándar ISO/IEC 13818, el cual indica que un paquete TS se divide como máximo en secciones de 188 bytes y que para identificar a un transport stream se debe revisar la cabecera de tamaño 4 bytes del paquete TS (Tabla 1). Específicamente se corrobora si el primer byte del paquete, que corresponde al byte de sincronización, es 0x47 (Figura 1).

Parámetro	Tamaño
Sync byte	1 byte
Transport error indicator	1 bit
Payload Unit Start Indicator (PUSI)	1 bit
Transport priority	1 bit
PID	13 bits
Transport Scrambling Control	1 bit
Adaptation Field Control	2 bits
Continuity Counter	4 bits

Tabla 1: Cabecera TS.

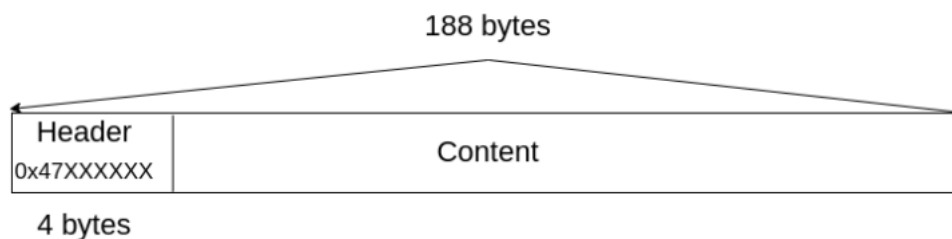


Figura 1: Paquete TS y cabecera.

Una vez se ha identificado el paquete TS se debe revisar que el campo PUSI tenga su bit en 1, ya que este indica el inicio de un nuevo Packetized Elementary Stream (PES) que a la vez indica la presencia de carga útil.

Revisado el campo PUSI se continúa con el campo Adaptation Field Control, ya que este indica si el paquete tiene o no tiene Adaptation Field.

Valor bit	Significado
01	No hay campo de adaptación, solo payload
10	Solo campo de adaptación, no hay payload
11	Campo de adaptación seguido de payload
00	Reservado para otros usos

Tabla 2: Valores Adaptation Field Control.

De la tabla 2 se observa que los valores 01 (Figura 2) y 11 (Figura 3) indican presencia de carga útil, por lo tanto son esos los valores a buscar.

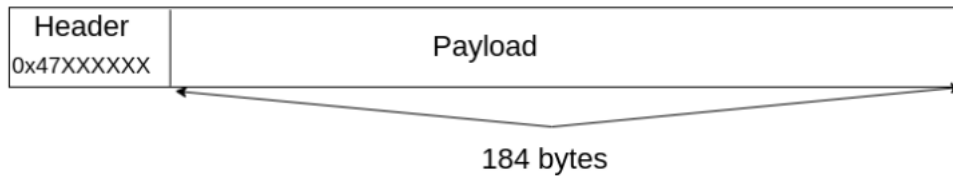


Figura 2: Valor bits 01, solo payload.

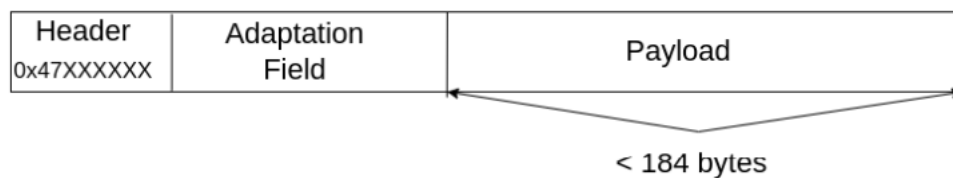


Figura 3: Valor bits 11, adaptation field seguido de payload.

En el caso que haya presencia de Adaptation Field, se debe saber cuanto tamaño abarca este campo para ubicar el inicio de la carga útil. El primer byte siguiente a la cabecera indica el tamaño del campo adaptation field (únicamente cuando corresponda). La forma para encontrar el comienzo de la carga útil es tomar la posición del byte que indica el tamaño del campo y sumar el valor de este.

Una vez ubicada la carga útil, la cual corresponde al paquete PES encapsulado en el TS (Figura 4), se debe revisar la cabecera PES.

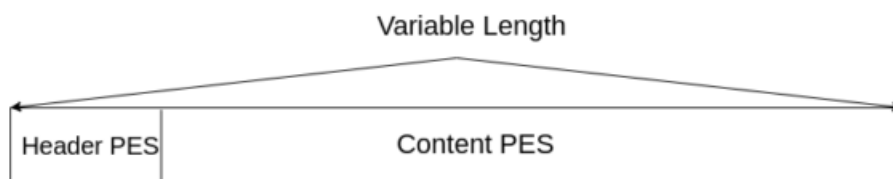


Figura 4: Paquete PES y cabecera.

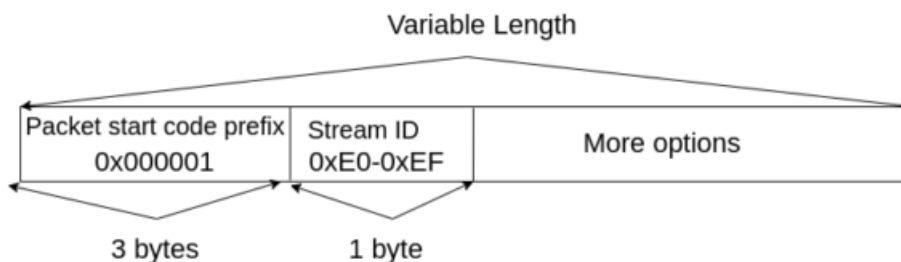


Figura 5: Cabecera paquete PES.

Para la cabecera del paquete PES (Figura 5) se debe corroborar que el campo Packet start code prefix tenga el valor de 0x000001 y el campo Stream ID tenga un valor entre 0xE0 y 0xEF pues esto indica que el paquete PES corresponde a video. Respecto al campo More Options este contiene campos opcionales de tamaño variable por lo que es un poco más complicado identificar el inicio de la carga útil. Para profundizar más en este aspecto revisar sección Wireshark.

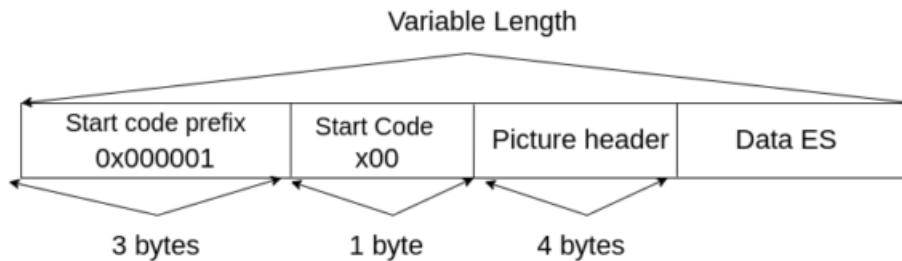


Figura 6: Contenido paquete PES.

Ya revisada la cabecera se procede a analizar el contenido del paquete PES (Figura 6). Para este paso se debe corroborar que el campo Start code prefix tenga el valor 0x000001, Start Code tenga el valor de x00 pues esto indica que el contenido del paquete son frames, y por último se debe revisar el campo Picture Header (Figura 7).

byte 0								byte 1								byte 2								byte 3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Temporal sequence number								Frame type								Video buffering verifier delay								---							

Figura 7: Cabecera paquete PES.

El campo Picture Header (Figura 7) tiene un campo llamado Frame Type el cual indica que tipo de frame es el que contiene el paquete (Tabla 3).

Valor bit	Significado
01	Frame tipo I
10	Frame tipo P
11	Frame tipo B

Tabla 3: Valores Frame type.

Identificando los valores de bits es posible identificar qué tipo de frame corresponde al paquete MPEG2-TS.

3. Análisis usando Wireshark

Para el análisis utilizando Wireshark se procede a transmitir video en formato MPEG2-TS vía UDP. Para esto se utiliza la herramienta ffmpeg en Linux ejecutando el siguiente comando (también se puede encontrar como script bash en el [repositorio de códigos](#)).

```
ffmpeg -f v4l2 -framerate 25 -video_size 640x480 -i /dev/video0 \
-c:v mpeg2video -pix_fmt yuv420p -s 720x480 -aspect 16:9 -streamid 0:180 \
-b:v 2298k -maxrate 2298k -minrate 2298k -bufsize 2298k \
-f mpegts -muxrate 2700000.0 udp://10.2.51.11:12345?pkt_size=188;
```

El comando ejecutado permite transmitir la webcam integrada de un notebook en formato MPEG2-TS con destino a una dirección IP vía UDP.

Mientras se ejecuta la transmisión en segundo plano, se comienza la captura de paquetes a través de Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
121	0.060643763	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
122	0.060668355	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
123	0.060684380	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
124	0.060710400	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
125	0.060726198	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
126	0.060751048	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
127	0.060766242	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
128	0.060790398	10.2.51.136	10.2.51.11	MPEG TS	232	NULL packet
129	0.060805768	10.2.51.136	10.2.51.11	MPEG PES	232	video-stream [MP2T fragment of a reassembled packet]
130	0.060830582	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]
131	0.060846554	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]
132	0.060875496	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]
133	0.060886884	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]
134	0.060907028	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]
135	0.060934060	10.2.51.136	10.2.51.11	MPEG TS	232	[MP2T fragment of a reassembled packet]

Figura 8: Captura de paquetes Wireshark.

La figura 8 muestra como se ve la captura de los paquetes transmitidos por el comando ejecutado. Wireshark identifica automáticamente el tipo de paquete por lo que facilita la ubicación de los mismos. También se logra observar entre todos los paquetes MPEG TS un paquete MPEG PES, el cual indica el inicio de un nuevo paquete.

Para comprobar lo presentado en la sección anterior se proceden a analizar dos paquetes MPEG PES, uno con campo Adaptation Field y otro únicamente con Payload.

3.1. Paquete PES únicamente con Payload

La figura 9 muestra la cabecera del paquete TS. Se observa como el campo Sync Byte tiene el valor 0x47 tal como se mencionó anteriormente. También se logra ver como el campo Adaptation Field Control tiene los valores 01, el cual indica únicamente carga útil en el paquete.

```

▶ User Datagram Protocol, Src Port: 45098, Dst Port: 12345
▼ ISO/IEC 13818-1 PID=0xb4 CC=3
  ▼ Header: 0x4740b413
    0100 0111 .... = Sync Byte: Correct (0x47)
    .... 0... .. = Transport Error Indicator: 0
    .... .1.. .... = Payload Unit Start Indicator: 1
    .... .0. .... = Transport Priority: 0
    .... ...0 0000 1011 0100 .... = PID: Unknown (0x00b4)
    .... ..00.. .... = Transport Scrambling Control: Not scrambled (0x0)
    .... ..01 .... = Adaptation Field Control: Payload only (0x1)
    .... ..0011 = Continuity Counter: 3
[MPEG2 PCR Analysis]

```

Figura 9: Paquete TS solo con payload.

La figura 10 muestra como el prefijo tiene valor 0x000001 y que el stream id está dentro del rango de los valores correspondiente a video.

```

▼ Packetized Elementary Stream
  prefix: 000001
  stream: video-stream (0xe0)

```

Figura 10: Cabecera paquete PES solo con payload.

La figura 11 muestra los campos opcionales de la cabecera PES.

```

▼ PES extension
  length: 0
  1... .. must-be-one: True
  .0.. .... must-be-zero: False
  ..00 .... = scrambling-control: not-scrambled (0)
  .... 0... priority: False
  .... .0.. data-alignment: False
  .... ..0. copyright: False
  .... ...0 original: False
  1... .. pts-flag: True
  .1.. .... dts-flag: True
  ..0. .... escr-flag: False
  ...0 .... es-rate-flag: False
  .... 0... dsm-trick-mode-flag: False
  .... .0.. additional-copy-info-flag: False
  .... ..0. crc-flag: False
  .... ...0 extension-flag: False
  header-data-length: 10
▶ PES header data: 310027ddf1110027af11

```

Figura 11: Continuación cabecera PES solo con payload.

Finalmente la figura 12 muestra el contenido útil y se observa como los primeros 4 bytes coinciden con los valores que indican presencia de fotograma. Para identificar qué tipo de fotograma es, se analiza el 6to byte y sus bits en posición 3-4-5. Es decir:

$$0x97 \rightarrow 10010111 \rightarrow 010 \rightarrow \text{Frame.tipo.P}$$

```
PES data: 000001000097fffb80000001b58
```

Figura 12: Contenido paquete PES con payload.

3.2. Paquete PES con Adaptation Field y Payload

La figura 13 muestra la cabecera del paquete TS que a diferencia del anterior, este indica que tiene Adaptation Field y Payload. Además también se observa un campo que antes no estaba presente, el cual indica el largo en bytes de Adaptation Field.

```
ISO/IEC 13818-1 PID=0xb4 CC=15
  Header: 0x4740b43f
    0100 0111 ..... = Sync Byte: Correct (0x47)
    ..... 0..... = Transport Error Indicator: 0
    ..... .1... = Payload Unit Start Indicator: 1
    ..... ..0. = Transport Priority: 0
    ..... ...0 0000 1011 0100 ..... = PID: Unknown (0x00b4)
    ..... ..00.. = Transport Scrambling Control: Not scrambled (0x0)
    ..... ...11.... = Adaptation Field Control: Adaptation Field and Payload (0x3)
    ..... ..1111 = Continuity Counter: 15
[MPEG2 PCR Analysis]
Adaptation Field Length: 7
```

Figura 13: Cabecera paquete TS con adaptation field y payload.

Debido a la presencia de Adaptation Field se hacen presente más variantes asociados el paquete (Figura 14).

```
Adaptation Field
  0... .. = Discontinuity Indicator: 0
  .0... .. = Random Access Indicator: 0
  ..0. .... = Elementary Stream Priority Indicator: 0
  ...1 .... = PCR Flag: 1
  .... 0... = OPCR Flag: 0
  .... .0.. = Splicing Point Flag: 0
  .... ..0. = Transport Private Data Flag: 0
  .... ...0 = Adaptation Field Extension Flag: 0
Program Clock Reference: 0x0000000000a9f2f90
```

Figura 14: Contenido campo Adaptation Field.

La figura 15 al igual que el paquete anterior cumple con el prefijo y el stream id correspondiente a video.

```
Packetized Elementary Stream
  prefix: 000001
  stream: video-stream (0xe0)
```

Figura 15: Cabecera paquete PES con AF y Payload.

La figura 16 muestra las demás opciones de la cabecera PES. La estructura del paquete en este punto del análisis es básicamente la misma respecto a la del paquete anterior.

```
▼ PES extension
  length: 0
  1... .... must-be-one: True
  .0.. .... must-be-zero: False
  ..00 .... = scrambling-control: not-scrambled (0)
  .... 0... priority: False
  .... .0.. data-alignment: False
  .... ..0. copyright: False
  .... ...0 original: False
  1... .... pts-flag: True
  .1.. .... dts-flag: True
  ..0. .... escr-flag: False
  ...0 .... es-rate-flag: False
  .... 0... dsm-trick-mode-flag: False
  .... .0.. additional-copy-info-flag: False
  .... ..0. crc-flag: False
  .... ...0 extension-flag: False
  header-data-length: 10
▶ PES header data: 3100290cd1110027ddf1
```

Figura 16: Continuación cabecera PES con AF y Payload..

La forma para identificar el tipo de frame en el contenido del paquete PES (Figura 17), es la misma ya realizada. Se analiza el 6to byte y sus bits en posición 3-4-5. Es decir:

$$0xd7 \rightarrow 11010111 \rightarrow 010 \rightarrow \text{Frame.tipo.P}$$

```
PES data: 0000010000d7ffffb80000001b5811ff34
```

Figura 17: Contenido paquete PES con AF y Payload

4. Programa en C para identificar frames

Para la manipulación de paquetes MPEG2 TS se desarrolla un programa en C, el cual levanta un servidor UDP capaz de recibir paquetes e identificar frames. El programa puede ser descargado en el [repositorio de códigos](#).

Para poner a prueba el programa se ejecuta el comando utilizado en la sección de Wireshark con destino al servidor levantado por el programa en C.

```
Frame type: P - Payload
Frame type: P - Payload
Frame type: P - Payload y adaptation field
Frame type: P - Payload
Frame type: P - Payload y adaptation field
Frame type: P - Payload
Frame type: P - Payload
Frame type: P - Payload y adaptation field
Frame type: P - Payload
Frame type: P - Payload
```

Figura 18: Programa en C capturando e identificando paquetes MPEG2 TS.

La figura 18 muestra el resultado que se obtiene a partir del uso del programa, como se puede observar se identifican frames tipo P tanto para paquetes que tienen únicamente payload como para aquellos que tienen payload y adaptation field.

5. Trabajo futuro

Tal como se mencionó en un inicio, este documento es solo una recopilación de lo investigado y no un informe robusto como tal. Queda por investigar el por qué se reconocen únicamente frames tipo P, quizá existen más formas para identificar frames que se están omitiendo. También falta investigar acerca de cuando los paquetes PES llevan grupos de imágenes, puede que entre ellas existan más frames por identificar. Finalmente tomando en cuenta lo anterior queda por hacer más robusto el código en C para que este sea capaz de identificar correctamente todos los tipos de frame.

6. Conclusión

En este documento se mostró cómo se compone un paquete MPEG2-TS y cómo identificar los tipos de fotogramas cuando corresponde. Además para respaldar la información se analizaron los paquetes con Wireshark para contrastar la información. Finalmente se presenta un programa capaz de manipular paquetes e identificar tipos de fotograma.

Referencias

- [1] Estándar MPEG-1 ISO/IEC 11172-2 de 1995.
- [2] Estándar MPEG-2 ISO/IEC 13818-1 de 1997.
- [3] An introduction to MPEG-TS.[Fuente.](#)
- [4] Digital Video and HD. Charles Poynton.
- [5] A Guide to MPEG Fundamentals and Protocol Analysis. Tektronix.
- [6] Flujo de transporte MPEG-2.[Fuente.](#)
- [7] MPEG Headers Quick Reference.[Fuente.](#)
- [8] Packetized Elementary Stream Headers.[Fuente.](#)