A METHOD OF ANALYZING MPEG DATA IN ENCAPSULATED STREAMS

F. Eugene Rohling
DVA Group, Inc.

Abstract

This paper describes a method of analyzing encapsulated binary data streams for the purposes of performing detailed message analysis. This method evolved from a general purpose analysis tool used to analyze radar data. It is now being applied to the analysis of MPEG-2 content and access control data delivered both in-band and out-of-band. It is particularly useful for compartmentalizing the details of sensitive control and encryption information within the MPEG data strea of an access control system..

The method allows users to describe encapsulated framed data, parsing a binary data stream, and generating human readable output that can be used to analyze and resolve problems. The template files can be tailored and customized to reveal varying levels of proprietary and confidential data within the binary stream.

INTRODUCTION

This paper identifies a solution that helps test and field engineers analyze complex MPEG data streams. It uses the familiar NAS access control service as an example of data that has been encapsulated four times when it is received within a headend system. Finally, it discusses the need for these tools as new technologies emerge.

This paper specifically discusses access control data. Many off-the-shelf tools exist for analyzing standard MPEG-2 video and DOCSIS services. However, access control systems are by their nature proprietary, and tools for looking at stream usage of Motorola Broadband DigiCipher, Scientific Atlanta

Power Key, and other access control streams are usually held close. This makes it difficult for an MSO to find problems in his local system, especially when he is responsible for operating it.

Encapsulated MPEG Data

The National Access Control Service (NAS) owned by Motorola Broadband and operated by AT&T (now Comcast) is an excellent example of MPEG encapsulated data. Figure 1 shows the various layers of MPEG data. First, the DigiCipher OOB data is encapsulated into MPEG private data message packets. When it arrives in the headend, data is then sent from the satellite receiving device (IRT) across Ethernet to the out of band modulator (OM). That is, the OOB data is carried as an encapsulated MPEG data stream within a HITS multiplex through the satellite system. [1]

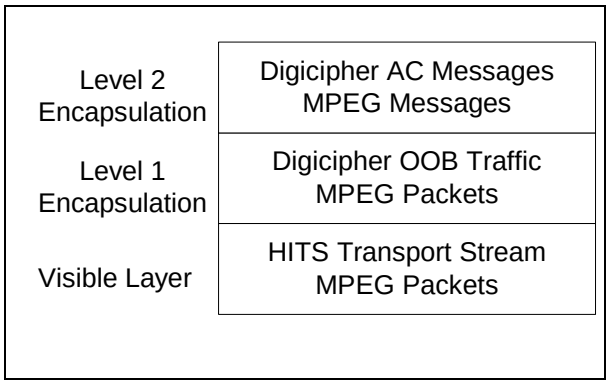

Figure 1 - NAS Encapsulation

A standard MPEG recording tool such as the DSTS by Logic Innovations allows you to record the data stream as it is received by the IRT. But if you want to recover only the data

seen by the set-top, then you must remove the HITS transport stream.

Several one-off tools have been built to detunnel the data, but they are all considered proprietary by the AC provider. Sometimes an MSO has legitimate reasons to determine if his access control system is operating properly or if he is receiving all the data his contract with NAS provides.

Similar problems exist with Motorola DAC based local access controllers. In this case, the problem becomes more urgent because the MSO is responsible for the operation of the DAC.

In many systems, access control data is encapsulated on a TCP/IP network and sent to a modulating device. Rather than data being MPEG encapsulated in MPEG, it is now MPEG encapsulated within IP. While good Ethernet tools exist, they to not provide utilities to integrate with MPEG tools. [1]

Compartmenting Data

To give the MSO the tools Motorola originally used to develop DigiCipher would be giving away the keys to their access control kingdom. But to give MSO's tools that help identify if code objects are spinning, or if TV Guide data is still online, or to identify if channel maps are being provided to their facility are all reasonable requests.

A legitimate need exists to compartment the visibility of MPEG access control implementation so legitimate users can visualize it operationally without compromising the access control system.

Processing Binary Data

Many processing programs exist for processing text. Unix has a wealth of tools such as awk, sed, grep, lex, and perl. But converting a 100 MByte file from binary to readable text becomes unwieldy when the result can generate many Gigabytes of data and take significant time to sort through and filter that data.

It is significantly less time consuming for analysts to process binary data and extract only the information they need to do their task.

HISTORY

The problem of analyzing a complex data stream that has been multiplexed into many layers is not unique to the cable or MPEG industries. Instrumentation systems during the 1980 to 1995 time frame commonly mixed and multiplexed dissimilar data from many sources within a telemetry or tape recorded data stream.

The Link to Radars

A good example was a radar instrumentation system developed for the F-15, F-16, and B-1 aircraft by Lockheed Georgia under the Advanced Radar Test Bed (ARTB) program. The requirements for that system required it to visualize and record traffic from up to four MIL-STD-1553 data bus streams, up to four streams of telemetry data, several custom low, medium, and high speed data streams at an aggregate rate of up to 12 Mbytes/sec. This was a feat for the 1989 designed system. They also required the system to be versatile and instrument any of five radars on the three aircraft. The requirements finally required time stamping the data to +/- 10 microseconds.

High Speed Analysis Becomes Key

Instrumenting the aircraft, multiplexing data, recording data, and time tagging data was straightforward. Much of it was

performed in hardware. But the system proved that reducing and analyzing the data became a significant labor intensive task. U.S. Air Force engineers likened the task of finding a needle in a hay stack.

This system evolved into the bench top Radar Instrumentation System (RIM-68) developed by Flexible Engineering Resources, Inc. (FER). This company developed a method of encapsulating the data in a common format and a method of parsing the data at high speeds so a small number of parameters could be visualized in both text and graphic format. The method was coined "MAcq" for Modular Acquisition.

MACQ FILTERING [3]

The "macq_filter" program performed the analysis side of this task was called the "MAcq_filter". It analyzed data for both real-time and post processing. It used "filters" that described the encapsulated nature of the data stream to both extract and process the stream into either human readable form or into derivative streams for off-the-shelf graphic programs to process as shown in Figure 2.

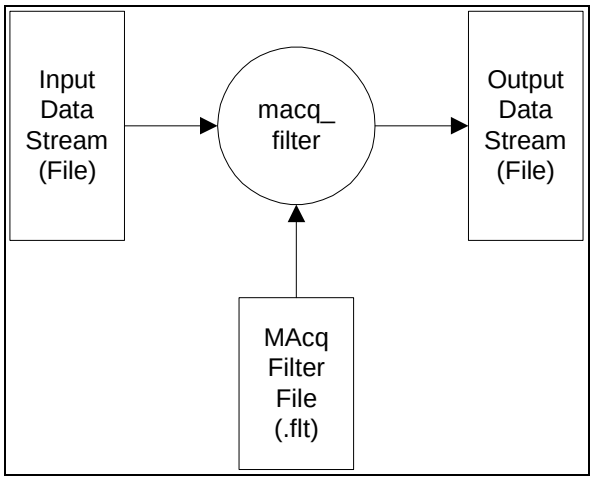

Figure 2 - MAcq Filter Process

Processing Frames of Data

The MAcq filter input description was designed to process nested frames of variable length data in a serial data stream. Figure 3 shows the format of the filter file. Note that the format of the filter file allows recursion. That is, optional filter frames can be nested within a top level scope frame to create the same data recursion effect often found with software recursion. This is the primary benefit of applying MAcq filters to encapsulated data problems.

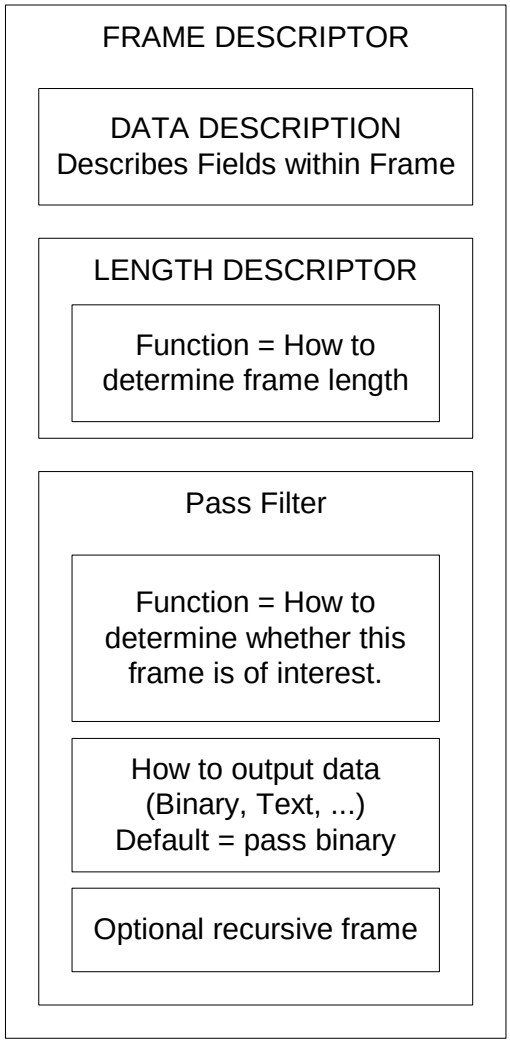

Figure 3 - Filter File Format

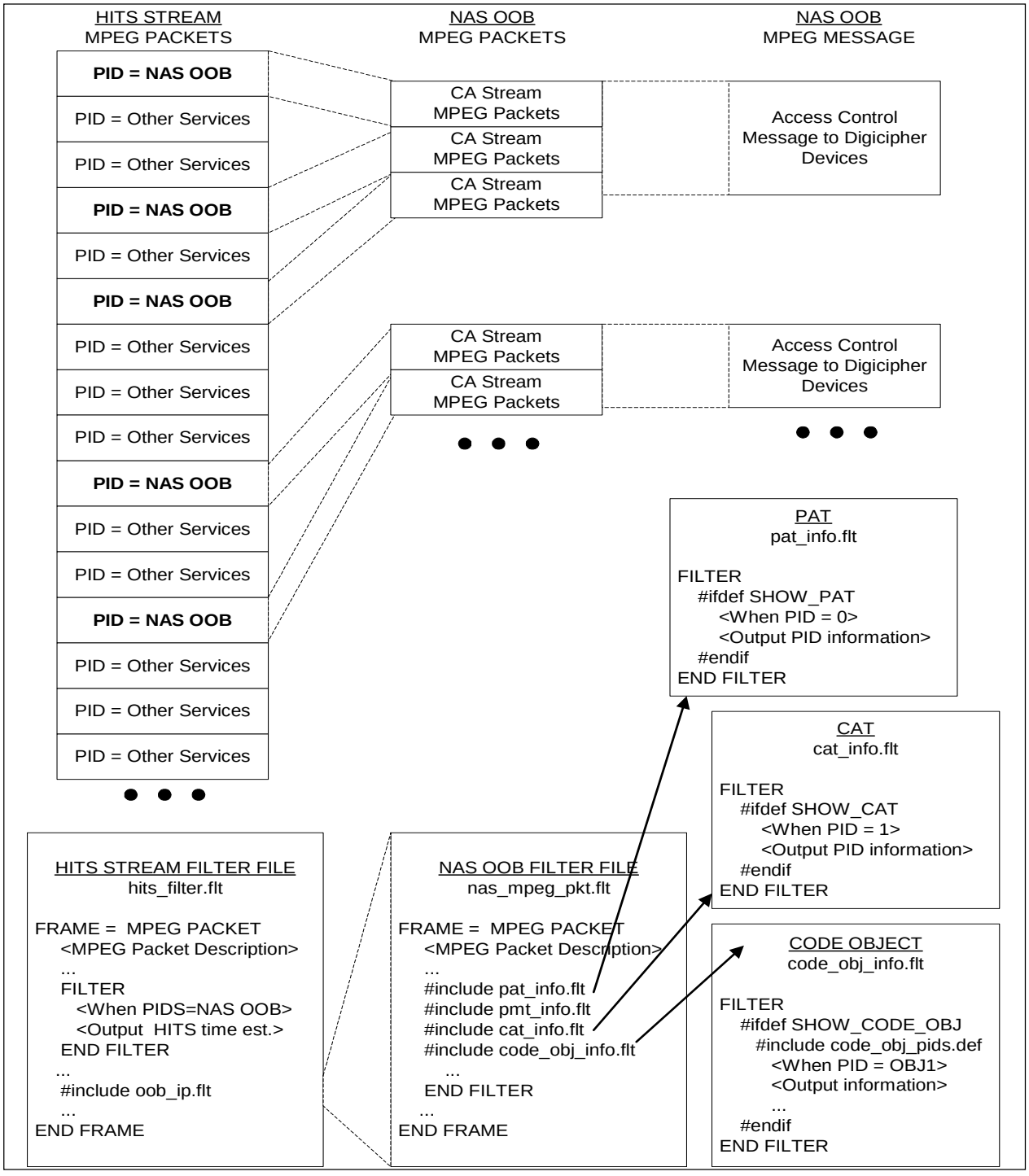

Figure 4 - Representing Encapsulation

Data Description

Each of the fields within a frame must be defined. The data description block within the filter identifies fields of data, such as the packet sync (47 Hex), the continuity counter, or the PID fields of an MPEG packet.

Variable Length Data

While MPEG packets are fixed format (188 bytes or 204 bytes), UDP / IP data is not. The length, however, can be readily determined from the contents of the UDP packet. Note the length clause contains a function used to establish the length of the arbitrary frame.

Selecting Data to be Processed

One or more pass filters look at frame headers and establish whether data needs to be passed. For MPEG data, the pass filter would likely select PIDS. For UDP data, it might select UDP source and/or destination ports.

Once data is selected, it is then processed. The output section defines what data is to be output. Output can be formatted text such as:

PID=234 TIME=88:99

or it can be binary data. Outputting binary data is quite useful for simple extraction of encapsulated data. That is, if all you want are the MPEG packets from a NAS IP OOB stream going to an OM-1000, you simply detunnel the UDP packets to that device.

Storing Data

The MAcq filter allows "scratchpads" to be used to temporarily store data. This initially became very useful when analyzing F-16 radar data.

APPLYING MACQ TO MPEG DATA

DVA Group began a research program in 2002 known as "Crown Royal" or CR to identify whether MAcq could be used to parse MPEG data and generate text output files.

Processing Frames of Data

Figure 4 shows how MAcq filter files can be used to describe and process the NAS satellite transport stream and extract the conditional access table (CAT). This shows a simple case of extracting OOB messages.

Need for Storage

Note that MPEG packets contain MPEG messages, and that MPEG messages can span multiple MPEG packets. When analyzing an MPEG stream in the general case, MPEG messages on multiple PIDs may interleave themselves in the temporal sequence of the MPEG stream. The MAcq scratchpad is useful for this case.

However, the MAcq implementation only allows statically defined scratchpads. This was fine for only detunneling OOB data, but was not adequate for cross PID correlation problems. As such, the general case of providing a general PID storage for detunneling MPEG messages was not adequate. Indexed scratchpads need to be added to the MAcq filter syntax.

Compartmenting Knowledge

In this context, compartmentalization refers to the Department of Defense (DoD) style security compartmentalization used during the cold war. That is, everything is on a "need-to-know" basis.

Access control providers have been reticent to only provide necessary information outside (and often inside) their corporate control. Providing MSOs and vendors with too much detail places the acces control provider at risk, and makes the MSO vulnerable to attack.

The MAcq filter provides a method of only providing information on a "need-to-know" basis. That is, filters that describe MPEG formatted information, or that simply announce the presence of a channel map, code object, or conditional access table may be appropriate for an MSO to obtain. However, the details of conditional access, especially key exchanges can be hidden by simply omitting the filters that are not needed.

PUTTING IT ALL TOGETHER

Engineers in the cable industry have many tools at their disposal. Many off-the-shelf products will parse Ethernet and IP packets, and others parse MPEG packets. Use of MAcq should take advantage of the strengths of existing tools.

Analyzing Local Access Control Data

Local AC data is often encapsulated on an Ethernet IP network. Off-the-shelf tools such as Etherpeek and the Unix tcpdump utility provide historical recording of Ethernet IP network in text or binary form. To make sense of the MPEG packets, however, requires the content to be detunneled.

The MAcq_filter can be used to detunnel the MPEG packets and put them in a form that MPEG analyzers can use. They can then be analyzed in native MPEG forms.

The same solution addresses instrumentation of systems in which video is transported across an Ethernet IP network. Many new MPEG re-multiplexors are being introduced that accept video streams across IP networks.

Using with Unix Pipes

Visualizing the delivery of code objects, VOD content, channel maps, and other necessary components of a cable system requires a tool that can output data in graphical form.

The macq_filter has been used in the radar community to visualize its effectiveness. The tool filters, processes, and then streams selected data in both real-time and playback instances into off-the-shelf 3-dimensional analysis tools.

The same can be applied to monitoring the OOB data within a headend. That is, MAcq can filter and process the access control stream and stream data into commercially (and sometimes free) third party software tools that display arbitrary bar graphs. This can be used to build tools that show code objects, channel maps, and other access control data as a percentage of bandwidth.

Work to Date

DVA Group has successfully used the original macq_filter program for simple tasks. The original program worked because 188 byte packets were long word aligned. It enabled analysis of PID distribution, continuity counts, and extraction of PIDS in binary form. It also allowed an encapsulated IP layer to be extracted from a given PID in an MPEG transport stream.

But extracting an OOB stream encapsulated within IP data could not be performed without being able to parse frames in byte word alignment.

SUMMARY

We have proven the underlying technology behind the macq_filter tool can help fill the gaps in commercial MPEG analysis tools. DVA Group continues to evolve the filter tool so it properly supports the needs of embedded cable systems in the future.

MANY THANKS

The author extends his appreciation to Michael Adams and all the people who helped write “OpenCable Architecture”. By showing a top level view of how Motorola Broadband and Scientific Atlanta conditional access systems work, we can discuss real world cable industry applications in a public forum.

REFERENCES

1. Michael Adams, “OpenCable Architecture”, 2000, Cisco Press.

2. Dr. Keith Montierth Jr, Todd Jahng, Lisa Chiang, Greg Rohling, Gene Rohling, “Three Dimensional Data Visualization System for the F-16 Program”, JAWS S3 Conference, June 10, 1997.

3. Flexible Engineering Resources, Inc. “MAcq User Manual”, September 13, 1997.