DEVELOPMENT OF NETWORK DATA AGGREGATOR FOR PAYLOAD FILTERING TO CONTROL TELEMETRY BANDWIDTH

Phil Ellerbrock, Charles Reyzer, Bob Zettwoch
The Boeing Company

Andy Kragick, Troy Troshynski
Avionics Interface Technologies

ABSTRACT

Emerging aircraft avionics and vehicle management communication systems have switched to higher data rate networks such as Ethernet and Fibre Channel, requiring Flight Test systems that can acquire data information from these networks. The data has moved onto higher speed networks, but the telemetry bandwidth has not increased, therefore producing a need to selectively capture data from within a packet, down to the bit level, for telemetry without capturing the entire message. Ethernet and Fibre Channel are transport protocols without rigid payload structure definitions as seen with MIL-STD-1553 or ARINC-429. Avionics traffic can be defined in any manner including dynamic length components or repeating structures that are difficult to define generically. AIT (Avionics Interface Technologies) developed the Airborne Network Data Aggregator (ANDA) unit for The Boeing Company with six different payload structure filter types to generically capture data out of network structures.

INTRODUCTION

The spectrum available for flight test telemetry transmission has been reduced significantly in the past decade and the future plans for spectrum usage does not look any better. The 2010 National Broadband Plan by the Federal Communications Commission (FCC) laid out a plan to “make 500 megahertz newly available for broadband use within the next 10 years, 300 megahertz of which should be made available between 225 MHz and 3.7 GHz for mobile use within the next 5 years” [1]. This combined with other applications encroaching in L band (1 to 2 GHz), S band (2 to 4 GHz), and C band (4 to 8 GHz), such as airport ground surveillance radars, ADS-B, medical devices, cellular technology and other technologies are putting pressure on the telemetry community to find ways of reducing bandwidth needs in a growing data environment. Adding to this problem test ranges are congested and often are competing for frequency spectra between executing test programs. While the ability to telemeter data is still currently viable the trend is an ever shrinking spectra such that the ability to transmit more data air to ground from a test aircraft is likely not viable.

In opposition to reducing transmission spectrum, the amount of data being produced by new aircraft platforms is expanding. Traditional digital bus technology such as MIL-STD-1553 had a maximum transmission rate of 1 Mbps per bus and ARINC-429 had a maximum rate of 100 kbps.

At these transmission speeds, the entire bus or even multiple buses could be captured by instrumentation systems and transmitted via telemetry to ground stations within the current telemetry bandwidth which, in many cases, is limited to 5 to 10 Mbps. A single, fully-loaded, digital bus on Gigabit Ethernet (1.25 Gbps) or Fibre Channel (1.0625 Gbps) can now only have about a half percent of their traffic transmitted in current bandwidths. Both Ethernet and Fibre Channel are also going to even higher speeds, which would result in even lower allowable percentages of bus data to telemeter, requiring even more filtering of the input bus to stay within telemetry bandwidth.

Additionally, any solution to this problem should be compatible and compliant to the Range Commander's Council – Telemetry Group's (RCC-TG) IRIG-106 standards [2] such that it could fly on government ranges, and it would need to be flexible enough to handle an ever growing list of data structures within the payload section of each protocol. As a military contractor, solutions have to reside within onboard systems capable of passing the environmental qualifications military aircraft requires which may also add power, weight, and size limitations to design solutions.

IRIG-106 ALIGNMENT

The IRIG-106 standards were created to ensure interoperability across government ranges. Government contracts often specify compliance to these standards. There are multiple sections within the standard that are used with handling telemetry. There are data format structures for telemetry in Chapter 4 (PCM), Chapter 7 (Packet TM), Chapter 8 (Digital Data Bus), Chapter 10 (data recording) Chapter 10.3 (Data streaming), Chapter 11 (Data Formatting) and Chapter 24 (TmNS). There are also data structure definitions in Chapter 9 (TMATS) and Chapter 23 (Meta Data Language or MDL).

Regardless of the method of transmission, whether traditional PCM (Chapter 4), PCM interspersed with asynchronous packet data (Chapter 7), or the provisions for dynamic spectrum sharing (Chapter 27 and 28), the quantity of data from aircraft network sources has long since exceeded the bandwidth available for telemetry. For analog and digital bus (MIL-STD-1553 and ARINC-429) data acquisition systems exist that can perform data selection suitable to reduce the amount of data telemetered to the transmission channel provided. However for aircraft Ethernet and Fibre Channel data now present on aircraft systems there is a driving need for a data selection solution to pass a subset of data off a bus. In order to do this, the data has to be selectively thinned out to selected words for telemetry.

The selection of words from the aircraft network sources requires a description of these parameters as telemetry attributes. These attributes are required to dictate how the data is acquired, processed, and telemetered. The IRIG 106-17 standards currently describe these attributes either in Chapter 9 (TMATS) or Chapter 23 (MDL). As of 2015, when Boeing was developing the requirements for a network data aggregator, TMATS could not support defining a variable length message in the S-records and MDL was not yet an adopted standard.

In the absence of an industry standard capable of defining the multitude of data structures, a protocol agnostic definition would be the next best option when aggregating selected data from network traffic. A protocol agnostic solution is ideal as this means the embedded device doesn't need to understand the structure of the data in the network packet payloads but only needs instructions on a byte offset basis as to what to do with data. This removes a need to define all traffic and only define the minimum information necessary to capture a particular sequence of bits. An onboard system needs to perform without adding significant latency during data selection and without significant computational resources to fit within a package that can survive the environment of tactical military aircraft (power, weight, and size limitations). Minimal data structure definition supports these challenges.

TYPES OF DATA STRUCTURES

Boeing has experience with many different types of avionics busses used in Flight Testing. On the military side, historical usage is with MIL-STD-1553. The current trend is moving over to more commercially derived bus architectures which have included IEEE 802.3 (Ethernet) and ANSI 272-2003 (Fibre Channel). One of the challenges with moving into this new domain of upper level protocol is the lack of defined structure. MIL-STD-1553 has a limited number of data words allowed in a message such that the command word allows great control to be levied by instrumentation systems on capturing selected messages or words for telemetry.

At the Ethernet level, there is only a datagram which is a large box in which to stuff whatever is desired to be sent. This could be data in traditional word slots ala 1553, or could be video, or could be a mixture of different pieces of information. Fibre Channel is similar to Ethernet in that there is just a bucket of data being transported. The data_field within the Fibre Channel data frame can hold anything. The same data structure could be used both for Fibre Channel or Ethernet transport. Both protocols have the ability to fragment messages into multiple frames or packets.

There are design considerations for embedded systems to optimize data structures for speed; however, data systems need to be more general from a cost perspective in order to support multiple different types of avionics boxes. The ability to pull information out of a data structure is highly dependent on the data structure and its complexity. This sets-up competing needs for the avionics box designers trying to minimize traffic for speed and for the instrumentation acquisition designers who need some level of consistency such that new firmware isn't required to capture traffic for each new avionics box transmitting on a bus.

Many examples from across multiple programs and platforms have been combined and abstracted into 4 generalized examples to highlight the necessity for a network aggregator solution that has the capability to inspect a packet inside the payload level and make decisions based on that.

In the following data structures the terminology listed here will be used for communication purposes even though these terms don't exist in the nomenclature of the protocols but will ease discussion.

- Message: A reassembled IP datagram (UDP or TCP) or a reassembled Fibre Channel Exchange
    - A message can be broken down into a header and submessage
- Message Header: A section within the message that contains key identification information
- Keyword: A sequence of bits that uniquely identify a message
- Submessage: A way of segmenting a message to contain different types of information
    - A submessage can consist of a header, blocks, or data
- Submessage header: A section within the submessage that contains identifying information
- Submessage blocks: A section of data that can occur multiple times
- Submessage data: Consecutive bits that are desired for data selection

The first data structure example is "Fixed Length, Fixed Format" with a structure diagram below in Figure 1.

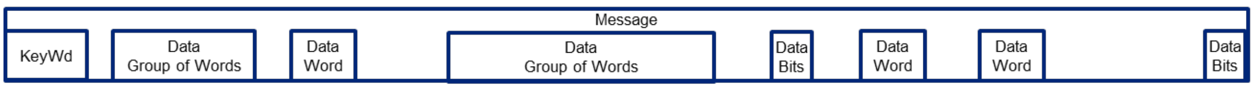

Figure 1: Fixed Length Fixed Format Data Structure

This structure is the closest related to a 1553 message. In the 1553 message, a command word identifies the remote terminal and subaddress. The Keyword (shown as "Keywd" in the diagram) is an equivalent for message identification. The ANDA can utilize the Keyword and then go on further to do data selection on Words or bits. Possible data selections are shown as "Data Group of Words", "Data Word", and "Data Bits." This structure is simple and can be defined with simple byte offsets since, in every occurrence of a message, the same bit locations mean the same thing.

The second data structure example is "Repeating Blocks without Offsets" with a structure diagram below in Figure 2.

Figure 2: Repeating Blocks without Offsets Data Structure

This structure is composed of 2 main sections (Fixed, and Dynamic). The fixed sections are the boxes "KeyWd" and "Non Repeating Block (Optional)". The dynamic sections are the submessages.

ANDA uses the Keyword (shown as "KeyWd") to determine if the message is desired for data selection or not. The next part of the fixed section ("Non Repeating Block (Optional)" is a fixed length block of data that may be there for some messages but may not for other messages; however, it would be consistently present or absent for each occurrence of a message.

A submessage is composed of a block count (shown as "# Blks") and a variable number of repeating blocks (shown as "Blk"). A repeating block is the same size within a submessage but a repeating block may be a different size for different submessages. The size of the repeating block is the same for each occurrence of each submessage. Each occurrence of the message could have a different number of repeating blocks for each submessage which is why a block counter is required to dynamically parse through the message. The "D" boxes represent a set of data to be captured for telemetry. This may be a few words or an entire block of data. Since the number of repeating blocks is dynamic and variable, the ANDA has to be able to limit the number of block captures in telemetry while passing all of them through for recording. The number of submessages that may occur could be different between different messages but has to be consistent between occurrences of the same message since there are no identification headers in the submessages. The starting location of each submessage is dynamic based on the number of repeating blocks in the previous submessage, and there is no offset location provided so the ANDA has to traverse the message dynamically to get to later submessages.

The third data structure example is "Repeating Blocks with Offsets" with a data structure diagram below in Figure 3.

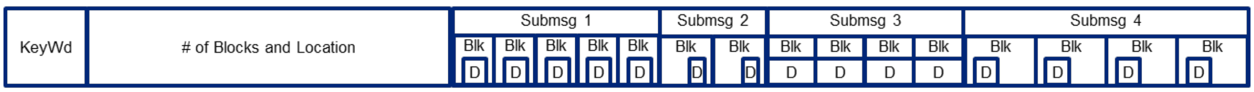

Figure 3: Repeating Blocks with Offsets Data Structure

This structure is very similar to the second structure (Repeating Blocks without Offsets). The main differences are that the block counters have been pulled out of the submessages and put into the beginning fixed structure and the dynamic offset locations for each submessage have been added. This has the benefit of being able to "jump" straight to the desired submessage.

The last data structure example is "Repeating Blocks with Headers" with a data structure diagram shown below in Figure 4.

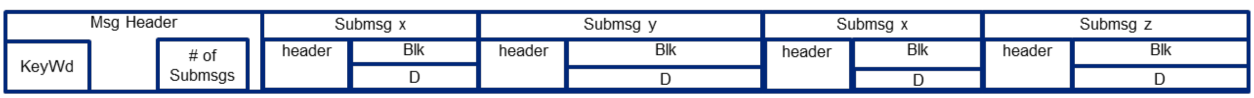

Figure 4: Repeating Blocks with Headers Data Structure

This structure has a message header that has the Keyword (shown as "KeyWd") and a count of how many submessages are in the message (shown as "# of Submsgs"). Following the header is each submessage. A submessage has a header that identifies which submessage is following and a size of the submessage. Since there are no offset locations provided and submessages do not have to show up in a predicable sequence, this header information is required to dynamically traverse through the message to find the submessage containing the words that are desired for data-selection. A submessage could occur multiple times. In order to find the correct blocks, ANDA has to be able to dynamically traverse and use both the Keyword and Submessage header for data-selection determination.

DATA SELECTION OVERVIEW

To address the growing tension between increasing data load requirements and restricted telemetry bandwidth, Avionics Interface Technologies (AIT) developed the Airborne Network Data Aggregator (ANDA). The ANDA provides hardware and comprehensive software tools to enable

capture of specific data within the open-ended payload structure of high data rate networks such as Ethernet and Fibre Channel and direct this data-selection to telemetry (TM).
The ANDA serves a dual purpose in the avionics flight test network. First, the ANDA provides intelligent selection of data targeted for the limited available TM bandwidth. Second, it provides a point of data aggregation from several high-speed serial (Fibre Channel & Ethernet) streams so that this data may be forwarded to a flight test data recorder.

The ANDA data-selection function is accomplished via a Multi-Level Data Select Logic (M-LDSL) which selects data to be sent to a Current Value Table (CVT) which is the upstream data source for TM scheduling and transmission to onboard telemetry. ANDA operation is controlled via AIT-developed software tools that include data configuration and compilation, a graphical user interface, and the ability to create APIs for aircraft Interface Control Documents (ICD). A significant feature of the ANDA software toolset is the ability to program offline without the requirement to be connected to the unit itself.

MULTI-LEVEL DATA SELECT LOGIC (M-LDSL)

As noted above, a data aggregator must be able to identify and inspect selected packets inside the aircraft network's data payload and make predetermined data-selection decisions based on its content. To accomplish this task, the ANDA employs Multi-Level Data Select Logic, a three-level hierarchy that exploits the payload structure of the datagram is shown below in Figure 5

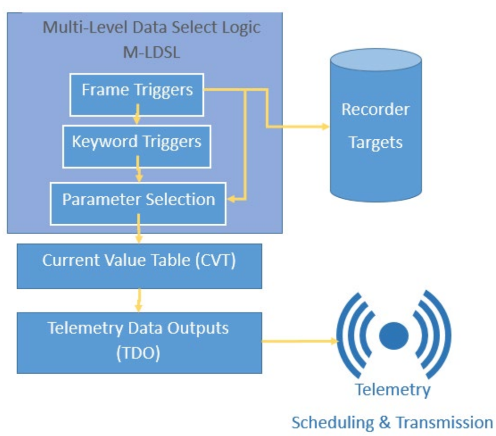

Figure 5: Multi-Level Data Select Logic Diagram

The three-level hierarchy consists of:

1. Frame Triggers, which identify data source and destination.
2. Keyword Triggers, which identify specific avionics application messages.
3. Data Selection, which identifies specific data parameters (submessages) within the application message.

The ANDA accommodates the generalized data structures described above using a frame-based data structure that includes an address header followed by one or more messages, each including a Message ID. The content of the individual message includes an arbitrary number of submessages or parameters.

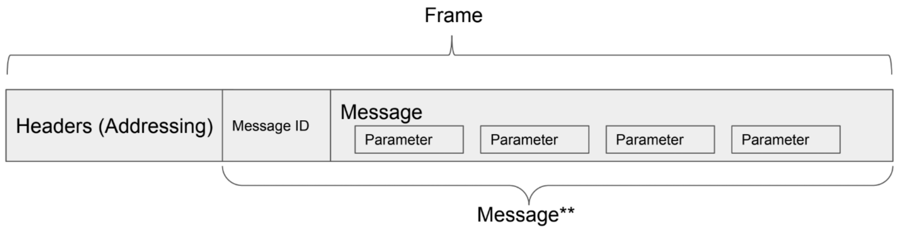

Figure 6: Generalized Data Structure

M-LDSL: FRAME TRIGGERS

Frame triggers may be used to identify data based on the source and destinations avionics applications or system. Based on frame header addressing information identifying source and destination addresses, frame triggers allow specific frames to be selected (or blocked) from TM and/or data recorders.

The ANDA handles up to 63 frame trigger modules per input port. Each frame trigger module may trigger off any field in the Ethernet or Fibre Channel frame header.

M-LDSL: KEYWORD TRIGGERS

The functions of keyword triggers are:

1. Identify specific avionics application messages;
2. Filter and route the type or class of data content within a message payload;
3. Forward that content to a specified entry point in the data selection level of the M-LSDL, which extracts the actual data destined for TM.

For example, a keyword trigger would identify a message containing critical low-speed sensor data such as airspeed or altitude that would be selected for routing to telemetry output, while a message containing video would be ignored.

Based on a given value in a specific field within the message payload, keyword triggers allow the user to select and route the payloads of identified messages—the message ID—of any word in the message. Keyword triggers operate on messages and may be used to identify the message based on a bit field located anywhere in the message.

M-LDSL: DATA SELECTION

Data selection—the lowest level of the Multi-level Data Select hierarchy— is the logic that extracts specific avionics application message parameters—the bit and byte fields contained within the message payload. Once extracted, this information is forwarded to the Current Value Table (CVT). The keyword trigger identifies the data selection entry point into the CVT. In turn, the CVT provides the data parameter source for TM scheduling and transmission.

At this point in the data extraction process, submessages exclude timestamp and frame header information, consisting only of payload data, including up to a 32-bit message ID that defines the particular format of the data (e.g., airspeed, altitude, etc.). Data arriving at the data selection level is processed serially.

CURRENT VALUE TABLE (CVT)

The CVT is the source database for telemetry processing and scheduling. The ANDA processor creates the output bit stream from the contents of the CVT. Status information may also be inserted as desired. Via its LAN and management interfaces, the ANDA allows a "quick look" at message parameter contents via CVT access and read operations.

The CVT accommodates up to 8192 entries. Each data entry may be up to 128 bits in length. Since each CVT entry size matches the RAM word length there is no need for double buffering, reducing latency. An example Fibre Channel packet is shown captured into CVTs and then passed into TM below in Figure 7

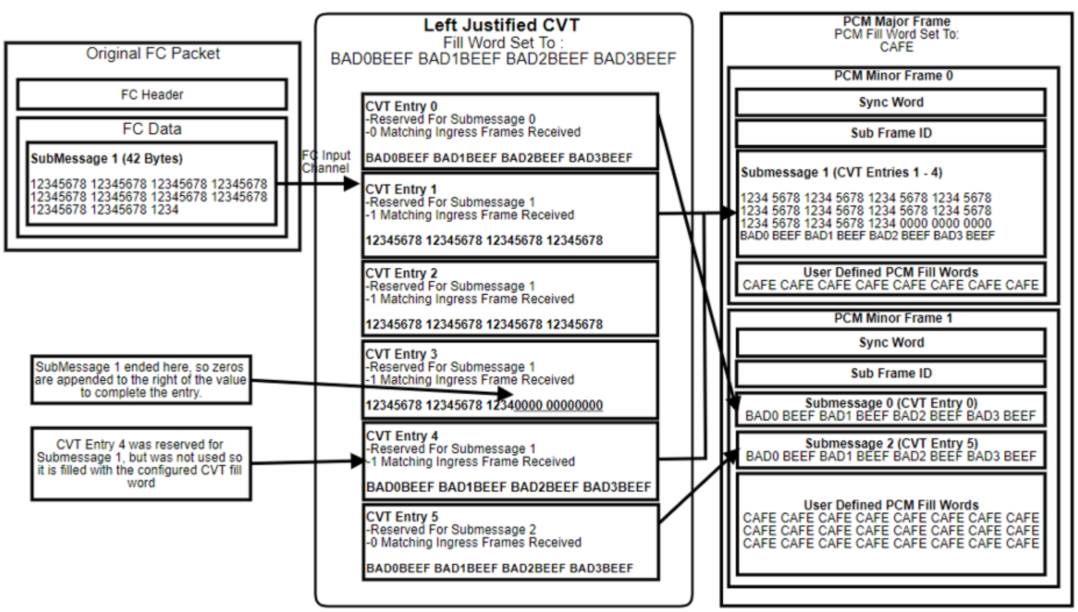

Figure 7: Data Selection Diagram

TELEMETRY DATA OUTPUTS (TDO)

The ANDA TDO is a TM frame formatter that reads CVT data and outputs it in PCM or packetized telemetry (TmNS) format to comply with the IRIG-106 standard. It then processes a circular buffer of user-programmed instructions. The TDO module also includes a user-programmed "minor cycle time" to control the output rate and supports time synchronization to IRIG-B DC or IEEE 1588 master clocks.

The TDO places the current timestamp at the start of the message and then inserts data from the CVT in 16-bit packages. The 16-bit package is the smallest size: 1-bit items will consume 16 bits in the output stream; 18-bit items would consume 32 bits, and so forth.
TDO data may be routed to two identical PCM interfaces as well as to available Gigabit Ethernet outputs for use by data recorders.

USER SOFTWARE TOOLS

ANDA hardware is supported by a System Setup and Configuration software utility (SSU) centered on user-readable XML.
Key elements of the SSU include:

1. A configuration graphical user interface for normal create/edit/save functions. The configuration data is saved in the form of XML documents. An XML command line configuration compiler provides the ability to program and compile optimized binary code to be loaded into the ANDA configuration memory. This allows the definition and compilation of configurations without the requirement to be connected to the ANDA hardware, so the compiled configurations may be loaded to the unit at a later time.
2. A configuration library for creating APIs for C#, export/load is available to import data programmatically from the aircraft Interface Control Documents or database.

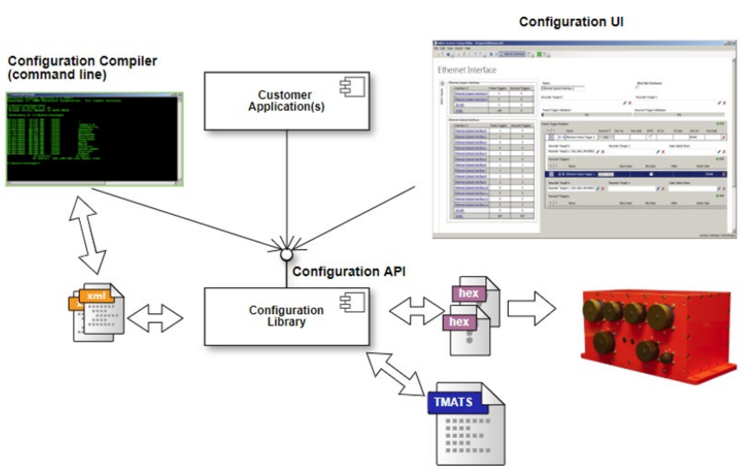

Figure 8: User Software Diagram

Setup configurations defined by the SSU are stored in XML format and enables a variety of user operations, including:

1. Definition of Ethernet and Fibre Channel data selection and filtering;
2. Telemetry frame builder to define data format for PCM or TmNS outputs;
3. Compiled code may be transferred TMATS outputs, which may be deployed to ground stations for telemetry data interpretation.

SSU also "modularizes" configuration information, separating network administrative parameters (e.g., IP addresses, UDP ports, etc.) from the core messaging parameter configurations. This modularity ensures that changes to network parameters do not invalidate the overall configuration setup.

The ANDA is supported by a Health and Status Display (HSD) application, which may be hosted in any Windows-based (7/8/10) environment with a LAN interface. A primary function of the SSH is to load programmed configurations into the non-volatile memory of the ANDA. The SSH also enables real time capture, telemetry, and recorder statistics and counters as well as viewing of parameters from the CVT. The parameter data from the CVT may be selected, logged and viewed in user-defined Engineering Units.

CONCLUSIONS

For the foreseeable future as avionics digital bus architectures continue to move to faster bit rates and the RF spectrum allocations continue to be constrained, there will be a need to utilize a Network Data Aggregator to capture and data select for telemetry requirements. A significant challenge regarding the data structures could be overcome if there were new standards created for data structures to reside within the Ethernet or Fibre Channel protocols such that standard methods could be created aiding the vendor space in development that was seen for MIL-STD-1553, ARINC-429, and IRIG-106 Chapter 10/11.

REFERENCES

1. Federal Communication Commission (2010). *Spectrum Analysis: Options for Broadband Spectrum.* (OBI Technical Paper No. 3). Washington, DC: U.S. Government Printing Office. Retrieved from https://www.fcc.gov/general/national-broadband-plan/spectrum-analysis-paper.pdf
2. Range Commanders Council (2017). *Telemetry Standards* (IRIG106-17). Retrieved from https://www.wsmr.army.mil/RCCsite/Pages/Publications.aspx