# AN ENGINEER'S GUIDE TO CHAPTER 7 PACKET TELEMETRY TRANSPORT

**Richard W. Hoffman**

**GDP Space Systems**

747 Dresher Rd., Suite 125, Horsham, PA 19044
rhoffman@gdpspace.com

## ABSTRACT

Chapter 7 of IRIG106-17 defines the means of encapsulating packetized data within a PCM telemetry stream, ostensibly for transport from a platform to a processing location, via that platform's conventional means of PCM transmission. While providing a mechanism for bridging platforms via the telemetry stream, a myriad of use-cases evolve, adding varying degrees of complexity to an implementation. Understanding these use-cases, their challenges, and some of the potential solution methodologies helps to determine the best implementation for a given mission. This paper seeks to present some of these aforementioned points, some obvious, and others uncovered over the course of working with solutions-seekers, in an effort to help cultivate and shape the growing demand for packet telemetry transport bridging.

## INTRODUCTION

As packetized telemetry becomes more ubiquitous across a wider number of test environments, the requirement to de-serialize telemetry data streams has subsequently moved further up the processing chain, closer to the measurement instrumentation. Platform data backbones have evolved in a symbiotic manner, accommodating packetized, asynchronous data streams without the need for sophisticated, or purpose-built multiplexing front-ends. A challenge remains to be overcome, though, when bridging the physical gap between a packet transport backbone on a test platform and the transport and processing infrastructure which exists within the range environment. The relative ease with which data can be packetized and transported on the platform could, potentially, be seen as a waste of time and processing power since transmission from a physically independent platform to a packet transport network is going to require serialization of the data. Understanding this, then, is there any benefit to utilizing packetized data on the platform itself, or should the serial data streams simply be

transported in familiar ways and used to create packetized streams once of the platform, where those capable networks already exist?

## CHAPTER 7 OVERVIEW

Chapter 7 of IRIG 106 has, since the release of IRIG 106-15, defined a method of encapsulating and encoding packetized data within a PCM stream with the intent of using that PCM stream to transport the data frames through conventional PCM transmission mediums. This provides a means of bridging between data networks which support asynchronous, packetized data streams without having to substantially alter the underlying data packets themselves to accommodate transport.

At the most basic level, Chapter 7 accomplishes its intended purpose by way of a PCM stream of fixed data rate with a known frame pattern wrapped around packets, or portions of packets. The Chapter 4 PCM stream is, itself, a very familiar, commonly used method of transporting multiplexed data measurements and a detailed overview isn't within the scope of this document's discussion. Where Chapter 7 enhances current implementations is in the definition of the methods to handle well-defined packetized data types which can be reliably encoded and transported via that Chapter 4 PCM stream.

There are generally two device roles which are required to fulfill the requirements of the Chapter 7 encapsulation mechanism: That of the encoder, and that of the decoder. These terms are applied relative to the Chapter 7 encapsulation mechanism, meaning that the encoder captures incoming packetized data and multiplexes it into a constant bit rate PCM data stream. Both of these device roles are faced with challenges which drive the complexity of the implementations up, and the nature of the data networks which are being bridged add to this complexity themselves.

A typical method of inter-network PCM transport, RF transmission, further muddies the waters with regard to data integrity and reliability. Where low bit errors can be, to varying degrees, problematic in more traditional Chapter 4 streams, they can be catastrophically disruptive to the Chapter 7 process if critical packet data structures are corrupted by transmission errors. A decoding device which is processing IP packets from the Chapter 7 stream, for instance, can't be expected to determine the end location of a packet and the beginning of another if the location and or length fields of those packets are corrupted. In a serial stream of data, this effectively derails the process of reconstructing packets of data, even if the higher level Chapter 4 frame remains locked to.

In order to account for these possible errors, Chapter 7 uses Golay encoding to protect against and recover from low bit error rate conditions. This Golay encoding is used on both the elemental Chapter 7 transmission frame fields as well as the underlying data packet header fields and is robust enough to recover up to 3 erroneous bits in every 24 transmitted. Further discussion of Golay encoding is outside of the scope of this document and subsequent talk.

There are a number of common packetized data transport networks in use in the testing environment, but the most common and probably most widely understood is the ubiquitous Ethernet/IP network. Figure 1 demonstrates a typical use-case where an IP network is in place on an airborne test platform, simply and efficiently providing the backbone for acquisition devices to get their data packaged and sent along downstream. The figure below illustrates how the packets are placed into a serial data stream and transmitted via RF to a ground-based receiver, which is then given the task of regenerating the constituent packets and sending them into the existing network. The network is now responsible for the proper handling of those packets.

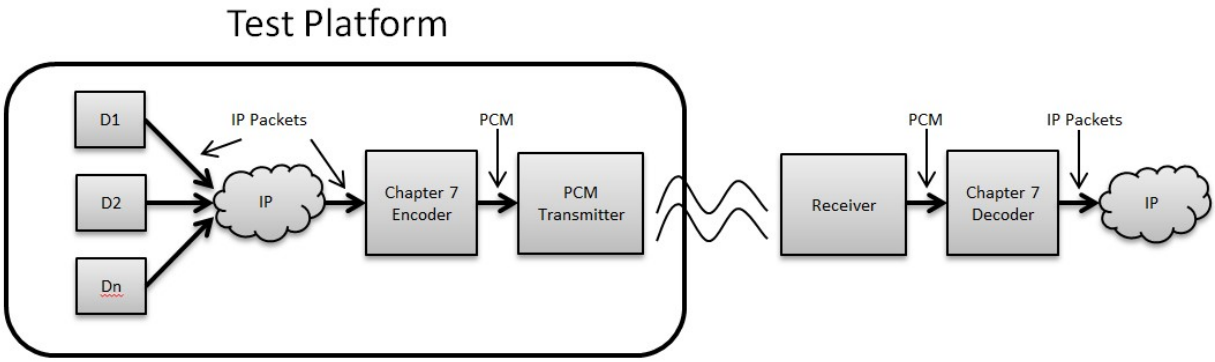


Figure 1 – Typical Chapter 7 IP Bridge

The well-defined and well-understood nature of the IP network, for instance, allows for both implementers and users to rely on these tried-and-true technologies to ensure efficient, simple handling of complex matters like routing, data delivery assurance, and security.

## USE-CASE EXAMINATION

Chapter 7 explicitly defines the methodology for handling four specific, industry defined packetized data types, as well as a generic “Application Specific” data type which allows for proprietary, vendor-, or mission-specific data packets to be encapsulated by the same

hardware. Chapter 7 references other sections of the IRIG 106 standard to detail the high-level packet details with the aforementioned callouts to header sections and fields which require additional processing in order to encode them for transmission and reconstruction reliability. The following sections will discuss the various considerations and challenges of different mission requirements for Chapter 7 technology, as well as some of the more nebulous applications of the specification.

There are some considerations that broadly apply to all of the data types and use-cases which will be discussed in their respective sections. Generally speaking, implementers should be aware that the packet formats which are supported by Chapter 7 typically have different ranges of acceptable values for fields like packet length or variable length fields. Where this is the case, the underlying packet will often have a length that may not fit nicely into the PCM stream's frame. The Chapter 7 specification provides for this case with the use of a packet fragmentation scheme. Underlying packets may be broken into smaller fragments in order to fit them within the PCM telemetry frame. The PCM telemetry frame itself may also be required to transport "normal" PCM telemetry interspersed among the Chapter 7 packet fragments. Reconstructing the packets and the "base" PCM telemetry from their constituent frames requires buffering and processing of potentially large numbers of packet fragments. As is the case with any buffering-intensive application, end-to-end latency understandably increases as the buffering needs do.

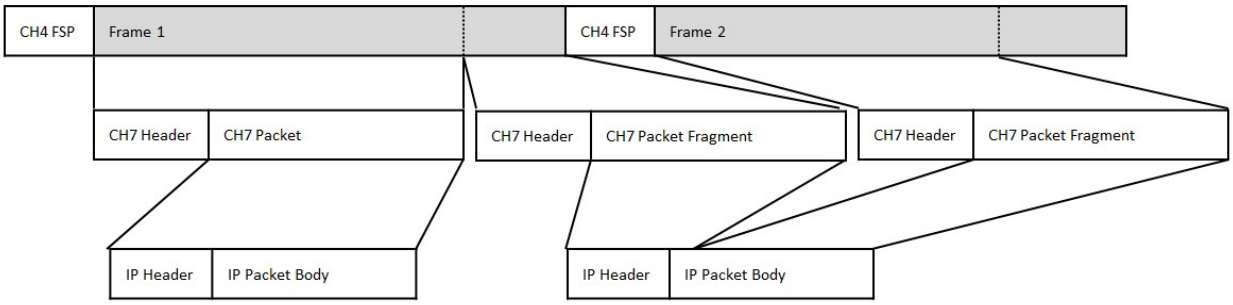


Figure 2 – IP Packet Fragmentation

In addition to the fragmentation and interleaving of PCM telemetry data, the underlying data networks are oftentimes handling asynchronous streams which are, by nature, "bursty"; at times the data from the networks may exceed the bit rate of the Chapter 7 transport, and at other times may, unless accounted for, produce a PCM stream with no data transitions. Chapter 7 attempts a "multiplex and fill" paradigm which requires significant buffering on the encoding side, as well as significant buffering and processing on the decoding side.

Having previously mentioned some of the concerns about processing and buffering time, it warrants mentioning that there is a mechanism by which the writers of Chapter 7 intended to handle time-sensitive packets within the specification: The Low-Latency Encapsulation Packet. The low-latency packet is intended to acquired and processed immediately following the discernment of frame sync status on the Chapter 7 encapsulation frame. Low-latency packets are wholly contained in a single encapsulation frame; that is, there is no fragmentation of low-latency packets. Low-latency packets immediately follow the encapsulation frame sync pattern and are therefore guaranteed to be processed before any packets contained in the Chapter 7 encapsulation frame, even if subsequent packets are fragments of a larger packet which had arrived in a previous encapsulation packet and had begun processing.

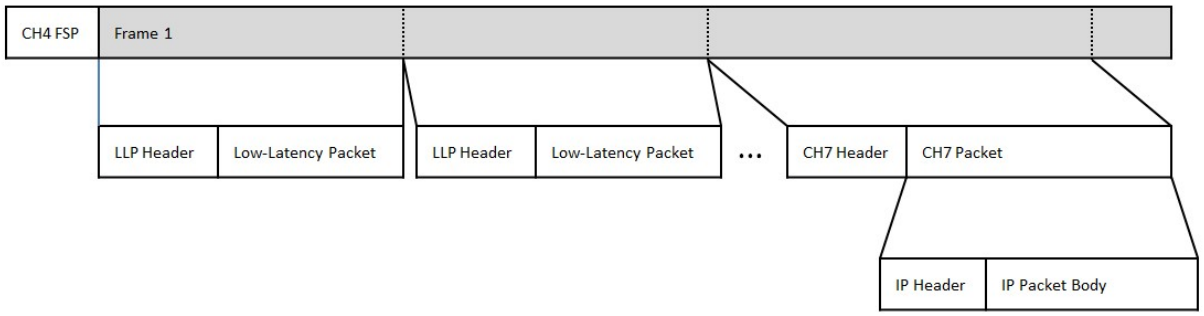


Figure 3 – Low-Latency Packet Mechanism

A rudimentary knowledge of the basic elements and functions of some header fields will be assumed for the following sections of this discussion. Ethernet and IP addressing can be a complicated matter to understand, as packets progress through a network, and so some elaboration may be required. Where this is the case, the specifics as they apply to the intended use for the Chapter 7 functionality will be the focus.

## RAW ETHERNET

I've chosen to discuss support for "Raw Ethernet MAC Frame" data first because this data type is a fundamental portion of the IP and Chapter 11 packets which are supported with additional layers of processing. A basic Ethernet frame structure is represented in the figure below.

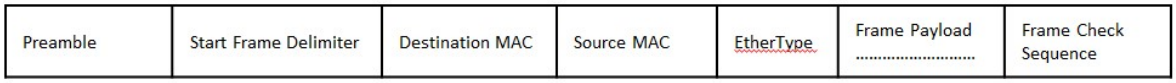


Figure 4 – Ethernet Frame

As an Ethernet frame proceeds from network node to network node, the destination MAC address is changed to reflect the MAC address of the next node in the chain. In cases where there isn't a means of bi-directional communication for the network devices, or some other means of address resolution(Ethernet ARP), an Ethernet frame will be unable to route through the network to its ultimate destination. The Ethernet frame from the acquisition device must have the destination MAC address of the Chapter 7 encoding device or a generic broadcast MAC address in order for the packet to be routed through network switches/routers on the platform network and arrive at the encoding device to be encapsulated and subsequently transmitted. In a Chapter 7 Ethernet frame application, it falls to the decoding device, post-transmission, to make sense of the end-point for the Ethernet frames. The determination of the intended end point is likely a user configuration parameter, but the lower-level MAC address must be populated with a broadcast MAC address or a MAC address as determined via ARP by the decoder. The good news is that, once the Ethernet frame has been decoded, there are numerous well-defined, standardized methods for determining the MAC address of the next hop in the network, downstream from the decoder.

Having elaborated on some of the challenges of handling raw Ethernet frames, the process of handling IP packets expands on those concepts and presents some additional points of consideration.

## INTERNET PROTOCOL (IP) PACKETS

The increasing footprint of telemetry-over-IP (TMoIP) devices and networks in the range testing environment provides an obvious transition point(s) for Chapter 7 capable devices to facilitate. In a modern, TMoIP capable testing environment, very little traffic is handled at the Ethernet layer, and is instead further wrapped with IP protocol headers in order to make application layer access of the packets more controllable. The Chapter 7 specification does not delve into the protocol layers below IP, which are the typical levels at which data will be accessed downstream, but both TCP and UDP are tacitly supported. Both protocols present unique challenges, but TCP is potentially so complicated as to be declared impractical.

Transmission Control Protocol involves a bi-directional stream of data and synchronization packets between two mutually involved end points in a network. In order for this mechanism to work in a Chapter 7 context, each Chapter 7 device would be required to accommodate encode and decode functionality and a Chapter 4 PCM stream would need to be provided so that protocol packets could be exchanged over an additional transmission medium. The protocol processing overhead is considerable, and concerns about transmission latency quickly begin to render TCP practically untenable.

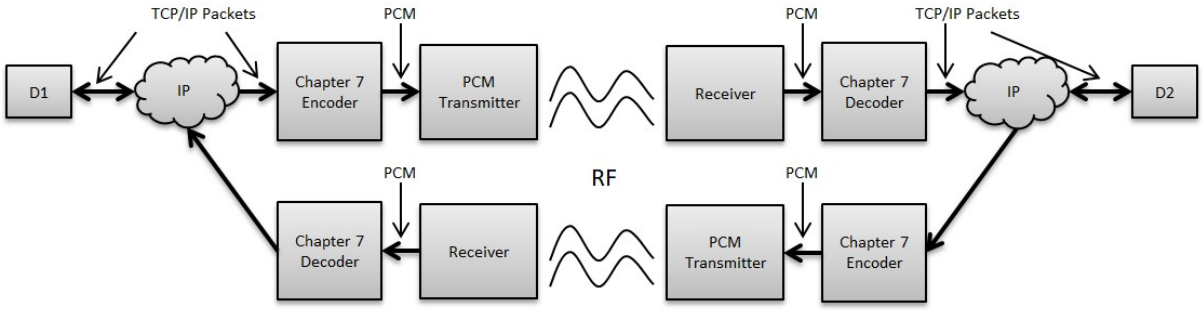


Figure 5- Conceptual TCP/IP Implementation

User Datagram Protocol (UDP), on the other hand, is the standard by which TMoIP devices currently handle data and is a more natural application for applying additional Chapter 7 functionality. UDP provides no mechanism for data delivery assurance, so while this potentially results in more inconsistent data delivery, there is no need to facilitate bi-directional communication in order to establish a data stream. Route determination, similar to the discussion from the Ethernet section above, is still a concern, but the addition of the IP packet fields provide several means of working through these concerns. Additionally, the IP layer presents several different application-specific options for how to use the Chapter 7 decoding devices.

UDP packets can be sent in a point-to-point (unicast) or point-to-multi-point (multicast) configuration. In order to unicast data, and endpoint must have a MAC address to associate with its IP address, as determined by the IP network. As with Ethernet address resolution, above, this typically requires bi-directional communication, which isn't likely in a typical Chapter 7 application. In a simple implementation of a solution to this problem, a TMoIP stream is directed to a Chapter 7 encoder, with a routable MAC address and IP address present on the platform's IP network. The Chapter 7 encoder is responsible for encapsulating the IP packets in the Chapter 4 stream, and providing that stream to a transmission device. The Chapter 7 decoder then utilizes a user-provided mapping parameter to determine the intended endpoint for the TMoIP stream, determines the MAC address and IP address of that endpoint,

and alters the underlying IP packet fields so that the TMoIP stream can successfully route through the network.

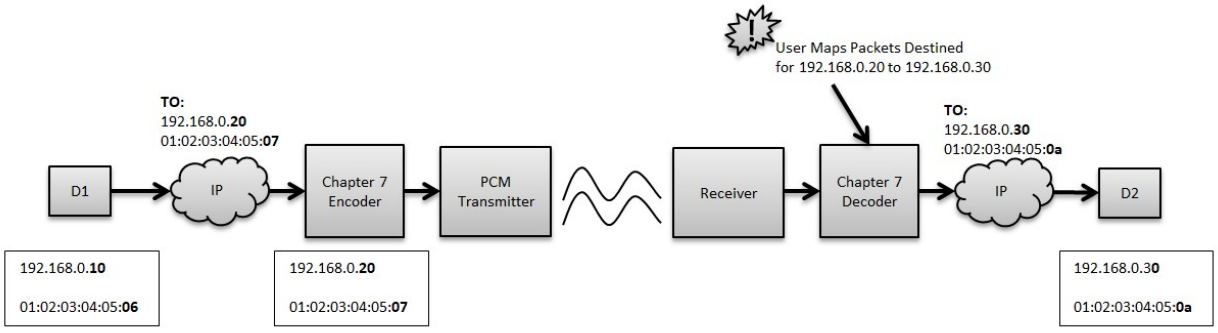


Figure 6 – Chapter 7 IP Bridge with Address Re-Mapping

The above use-case, while potentially complicated in its implementation, is a fair representation of a typical test range application for Chapter 7-TMoIP functionality. There are several other means of addressing the packets at the front-end, including IP multicast and broadcast, which don't require address mapping in order to facilitate data delivery, but both are also less manageable, by nature, and present additional layers of complication.

## CHAPTER 11 PACKETS

Chapter 11 packets present an opportunity to examine two potential applications for Chapter 7 decoding devices. In the both applications, a Chapter 7 encoding device is providing a PCM stream of encapsulated Chapter 11 packets for a decoding device to process. The intended purpose for the data, downstream, drives the implementation of the decoding device's outputs.

In the first figure, below, the PCM stream, which now contains decoded, "native" Chapter 11 packets, is fed back into a Chapter 11 capable decomm unit; this decomm unit could just as easily be a Chapter 11 recorder device.

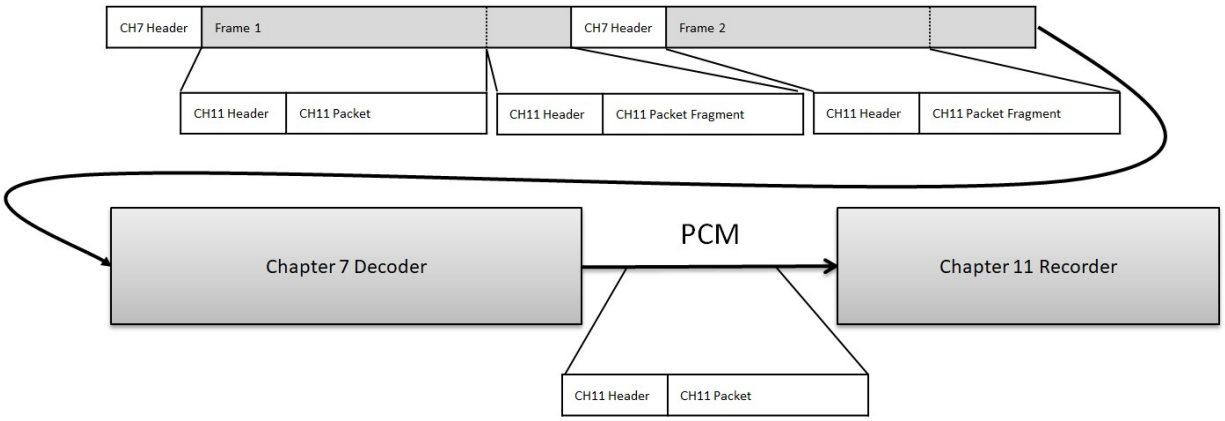


Figure 7 – Chapter 7 Decoder to Chapter 11 Recorder

In a second possible application for the decoding device, the Chapter 11 packets are wrapped with IP transmission headers and sent to IP network connected decomm workstations. This functionality is colloquially called a Chapter 7 Gateway function, and is a convenient means of providing a bridge for Chapter 11 packets onto an IP network for “re-storage” and processing.

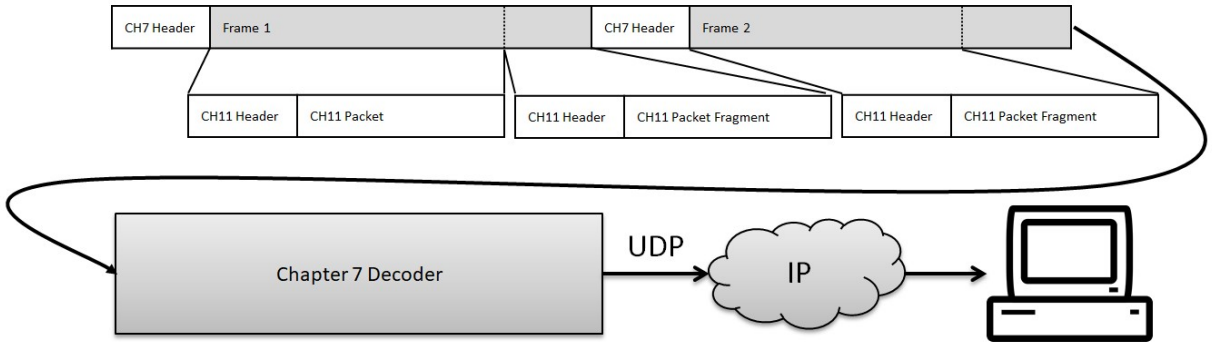


Figure 8 – Chapter 7 UDP/IP Gateway

## CONCLUSION

Chapter 7 is an emergent specification that helps to provide solutions for requirements to bridge different networks over traditional PCM mediums. The challenges faced by implementers are considerable and the specification is not intended to specifically address how each user requirement should be met, but the fundamental blocks are present for implementations to address those requirements as they are more clearly defined. As an increasing number of users move to adopt the technology and the vendor market moves to accommodate this, additional use-cases will arise, presenting further challenges and opportunities for refinement.