300 Welsh Road Building 3
Horsham, PA 19044-2294
215.657.5270
www.gdpspace.com

"Telemetry over Internet Protocol (TMoIP) – An Overview"

Gary Thom
GDP Space Systems, Horsham, PA, USA

June 21, 2017

Abstract

As telemetry ground stations are finally making the move toward network centric architectures, it is worth considering the lessons learned over the previous 10 years of designing, installing, troubleshooting and optimizing telemetry data distribution over IP networks. This paper discusses some of the architectural decisions to be made and some of the pitfalls to avoid in developing the next generation of networked telemetry ground stations. Critical issues such as latency, efficiency, data loss and Quality of Service are addressed, as well as techniques for troubleshooting these problems. Also included in the paper is a discussion of the effectiveness and efficiency of evolving packet formats.

Key Words: Internet, IP, TCP, UDP, TMoIP, network, PCM, Latency, packet loss.

Note: This is a living document that will be modified and expanded as new information and best practices become available.

1 Introduction

Companies like GDP Space Systems have been sending PCM data over packet switched networks for over a decade. In the beginning, there was very little interest in this new method of transporting PCM data. There were microwave links, fiber optic links, coax cable, and matrix switches. They all served the industry very well for many years.

During this time, IP networks have grown in usage, capacity and capability. In the data world, they have become ubiquitous, allowing data to be distributed worldwide in the blink of an eye. Reliability and redundancy are built in and provide guaranteed delivery of an infinite range of data.

The first non-traditional data type that began to move to IP networks was voice. The low cost of data transport over IP networks drove a cottage industry in toll bypass and low cost international voice traffic over IP networks. This was followed closely by video when video conferencing moved from circuit switched networks to the packet based IP networks and by the cable TV industry changing from RF video distribution over cable to packet based digital video transmission over coax and fiber.

This set the stage for sending PCM data over IP networks or Telemetry over IP (TMoIP).

2 Why do we want to use TMoIP

The motivation for moving to TMoIP was twofold: first, to find cost effective PCM data distribution and second, to provide reliable and robust PCM data distribution regardless of the destination. The global explosion of IP networking has provided a built in infrastructure with access to the most remote destinations. A wide variety of transport media for IP traffic provides ubiquitous connectivity, whether twisted pair, fiber optic cable, microwave links, satellite links, analog modems and cell phones, IP connectivity was everywhere.

This ubiquity and global deployment drove down the cost of networking components such as routers and switches. It provided dynamic routing and redundant paths, improving reliability and fault tolerance. The insatiable appetite for more data has resulted in ever increasing bandwidth availability. Today 10 Gigabit per second (Gbps) networks are becoming common place with 40 Gbps following quickly behind.

The result is a reliable, cost effective infrastructure for PCM data distribution, whether on private IP networks or globally via the public internet.

A byproduct of the move to IP is that with very little effort, the TMoIP gateway device could provide frame aligned packets of data. This proves to be useful for driving software decoms. These are software programs that run on standard computers that are capable of decommutating, processing and displaying telemetry data. By providing frame aligned packets of PCM data, the hardware frame synchronizer and serial to parallel converter in traditional hardware decoms can be eliminated. These software decoms provide a low cost alternative to purpose built hardware decoms for application where data rates and processing loads could be managed by a general purpose CPU.

Another side effect of the move to IP is that these packets of PCM data could very easily be captured by a computer and stored to disk in industry standard formats such as IRIG 106 Chapter 10. This provides some efficiency by converting all PCM data to IP at the edge of the network where it could then be easily distributed, decommutated or recorded.

3 Networking Basics

In order to understand the benefits and problems associated with TMoIP, it will be useful to have a brief review of networking basics. Networks use a number of stacked protocols for implementing data communications. Each layer of this protocol stack provides a different service or capability. There are two conceptual protocol stack models: the Internet Protocol Suite and the ISO 7 Layer Model. We will use the Internet Protocol Suite model in this paper. The model will be discussed in the following sections.

3.1 Link Layer

The Link Layer is the first layer of the protocol stack and consists of two sub-layers. The physical sub-layer is responsible for transmitting and receiving raw bits. This layer specifies the physical connectors, the electrical signals and the data coding used to send logical data from point to point. There are a variety of Physical Layer protocols that define network hardware technologies. Some common Physical

Layer protocols include, but are not limited to: 10Base-T, 100Base-T, 1000Base - SX for Ethernet and 802.11a, 802.11b and 802.11g for WI-FI. There are also many other common physical layers such as analog telephone modems, Digital Subscriber Line (DSL) modems and GSM or 4G mobile wireless modems.

The second sub-layer is the data link layer and is provided by the Ethernet protocol. It defined communications between devices on the same subnet. A subnet is the collection of all devices that are directly connected through a switch, hub or coax cable as in very early Ethernet networks. Each device has a Media Access Control (MAC) address and Ethernet packets are sent from device to device based on that MAC address.

An Ethernet packet, shown in Figure 1, consists of a preamble, a start of frame delimiter, an Ethernet packet header, the payload, a frame check sequence and an inter-packet gap. The preamble helps the receiver to synchronize to the data. The start of frame delimiter marks the end of the preamble and the start of the packet header. The packet header contains addressing and other information specific to the Ethernet protocol. The payload is the data being carried in the packet. The frame check sequence is a Cyclic Redundancy Check (CRC) code used to detect errors in transmission. The inter-packet gap provides idle time between packets before the next packet is transmitted. An optional 4-byte VLAN tag may be inserted after the MAC Source Address.

Figure 1- Ethernet Packet Format

<table>
<tr><td rowspan="3">Preamble</td><td rowspan="3">Start of frame delimiter</td><td>MAC destination</td><td>MAC source</td><td>Ethertype or length</td><td rowspan="2">Payload Data</td><td>Frame check sequence</td><td rowspan="3">Inter-packet gap</td></tr>
<tr><td colspan="3">Packet Header</td><td>Trailer</td></tr>
<tr><td colspan="5">Ethernet Frame</td></tr>
<tr><td>7 bytes</td><td>1 bytes</td><td>6 bytes</td><td>6 bytes</td><td>2 bytes</td><td>46–1500 bytes</td><td>4 bytes</td><td>12 bytes</td></tr>
</table>

3.2 Internet Protocol Layer

A single subnet has pretty limited scope, so to broaden connectivity between devices the Internetworking Protocol (IP) was developed. There are two versions of the IP protocol: IPv4 and IPv6. The IP packet is carried within the payload of the Ethernet packet.

3.2.1 IPv4

The IPv4 packet consists of an IP header, shown in Figure 2, which provides IP layer addressing as well as other IP protocol information.

Figure 2 - IPv4 Packet Format

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>09</td><td>08</td><td>07</td><td>06</td><td>05</td><td>04</td><td>03</td><td>02</td><td>01</td><td>00</td><td>W</td></tr>
<tr><td colspan="4">Version</td><td colspan="4">Hdr Len</td><td colspan="8">Type of Service</td><td colspan="16">Total Length</td><td>1</td></tr>
<tr><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td><td>2</td></tr>
<tr><td colspan="8">Time to Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td><td>3</td></tr>
</table>

Source Address	4
Destination Address	5
Payload Data (0 to 1480 bytes)	

The IPv4 protocol provides a higher level addressing scheme based on four bytes represented as decimal numbers separated by periods (IE 192.168.100.25). A Subnet Mask is used to identify the size and address range of the subnet. For example, for the above address, if the subnet mask was 255.255.255.0, that would indicate that the subnet could contain 254 devices with addresses from 192.168.100.1 to 192.168.100.254. The first and last addresses have special meaning and do not represent actual devices. IPv4 addresses were originally divided into a number of classes to accommodate different size networks and other special purposes, as shown Table 1.

Table 1 - IPv4 Address Classes

Address Class	Address Range
Class A Address	0.0.0.0 - 127.255.255.255
Class B Address	128.0.0.0 - 191.255.255.255
Class C Address	192.0.0.0 - 223.255.255.255
Multicast Address	224.0.0.0 - 239.255.255.255
Reserved	240.0.0.0 - 247.255.255.255

A process called Address Resolution, using the Address Resolution Protocol (ARP) translates the IP address of a device on the local subnet to its Ethernet MAC address so that packets can be sent from device to device. This protocol allows an ARP table to be created in the device which links IP address to the MAC address for the devices in the subnet.

If the destination IP address is not in the local subnet, the device will send the packets to the MAC address of a Gateway. The Gateway or router is a device that connects one subnet to another. A routing table is typically created in the device to simplify the routing of packets to the local subnet or to one or more gateway devices. The routing table identifies IP address ranges for the local subnet as well as the ranges serviced by the gateway devices. The ARP table is then used to convert the gateway IP address to its MAC address. Remote destinations can be reached by going through a series of router hops until finally the packet enters the destination subnet and then the destination device.

An additional level of address abstraction can be achieved by using host names. A host name is a text name that is linked to an IP address. A Domain Name Server can be queried with a host name, which then returns the related IP Address of the device. A hierarchy of DNS servers provides for global address resolution.

A device usually needs to be configured with its IP Address, the Subnet Mask and a Gateway IP Address. This can be done it two different ways. The first method is static or manual configuration where the information is entered into the device by an operator. This is the most common method and has the advantage of knowing what devices are on the network and what IP addresses are assigned to them. The second method uses the Dynamic Host Control Protocol (DHCP) to automatically assign the local address information as well as some additional configuration information. While this may seem to be a

simpler solution, it has the drawback of not being able to find a specific device without using host names and the DNS system. It is also not favored by private networks that want to tightly control what devices are on the network and the addresses that they use.

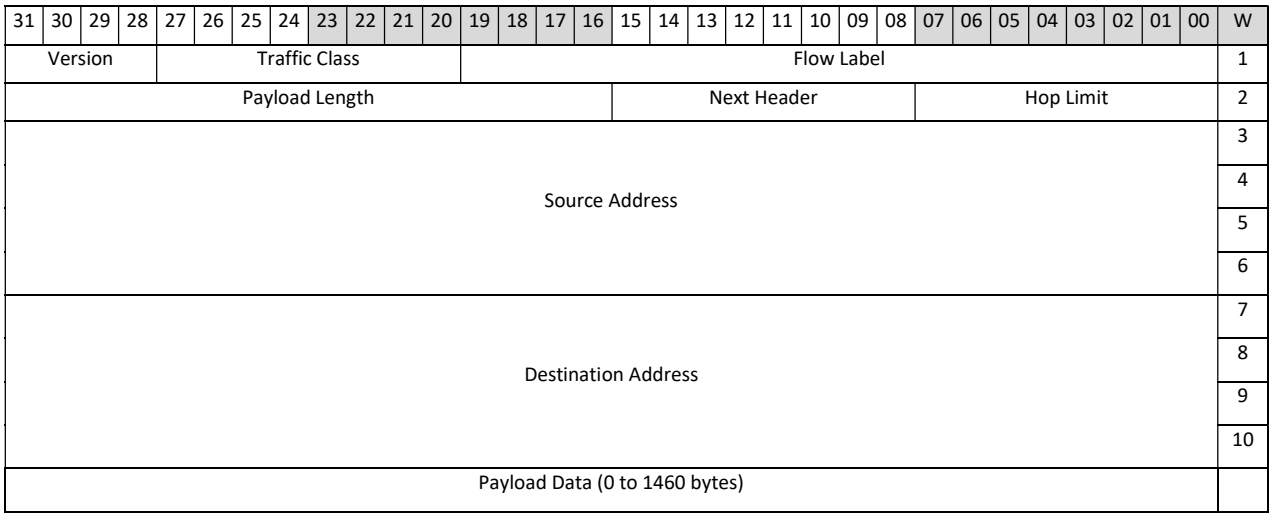
Multicast addresses are in the range: 224.0.0.0 to 239.255.255.255. This range is divided into subgroups: Reserved Multicast Addresses, Well Known (registered) Multicast Addresses and Dynamically Allocated Multicast Addresses. Reserved Multicast Addresses should not be used. Well Known Multicast Addresses are registered with IANA and are for specific protocols or applications. Dynamic Multicast Addresses can be used for any application. Multicast will be discussed further in a later section.

3.2.2 IPv6

The main motivation for a new IP protocol was the need for more IP addresses due to the explosion of internet devices. There was a concern that we were running out of routable IP addresses. This was temporarily slowed by the use of Network Address Translation (NAT) which could be used to hide any number of local IP addresses behind a single public routable IP address. There were two approaches to solving this problem. The first was to just expand the IP address fields and the second was to create a completely new protocol that expanded the IP address fields and also fixed some of the problems with IPv4. The later approach was selected and resulted in the IPv6 protocol.

The IPv6 packet consists of an IP header, shown in Figure 3, which provides IP layer addressing as well as other IP protocol information. IPv6 packet headers are larger than IPv4 packet headers. While the IPv6 address is four times the size of the IPv4 address, the IPV6 packet is only twice as large as the IPv4 header, growing from 20-bytes to 40-bytes.

Figure 3 - Ipv6 Packet Format



This header is significantly simpler than the IPv4 header. The difference is that headers can be stacked in order to provide additional protocol information.

The IPv6 address is 128 bits (16 bytes) long and is formatted as eight 4-hexidecimal digit numbers separated by colons (":"), for example: 2001:0111:0222:0333:0000:0000:0000:0001. To simplify use, a shorthand has been defined. Strings of 0000 groups can be removed. The location of the removed zeros is indicated by the "::" and this can only be done once in the address. So the above address can be

written as 2001:0111:0222:0333::0001. Furthermore, leading zeros can also be removed, so the address can be further reduced to 2001:111:222:333::1.

An IPv6 interface is allowed to have more than one IPv6 address. Every interface has a Link-Local address which begins with the prefix fe80::/10 (that is the first 10 bits are defined to be 1111 1110 10. The link-local address is only valid on the local subnet and routers do not forward packets with link-local source or destination addresses. The remaining bits can be set in a variety of ways. These link-local addresses can be configured automatically without the need for DHCPv6. One such way is to set the lower 64-bits (the Interface ID) based on a modification of the interfaces Ethernet MAC address. The modification is fairly simple. The global flag (7th bit) of the MAC address is inverted and the value fffe is inserted between the 3rd and 4th byte of the MAC address. For example the MAC address 14:18:57:a3:e9:c4 is converted to Interface ID 1618:57ff:fea3:e9c4. The other bits are typically set to 0, thus the full link-local address would be fe80:0000:0000:0000:1618:57ff:fea3:e9c4 or fe80::1618:57ff:fea3:e9c4. The link-local address can also be generated by using a random Interface ID or a manually assigned Interface ID. During initialization a Duplicate Address Discovery processes is used to assure that the Link-local address is unique.

Of more interest is the Global Unicast Address, and an interface may have more than one Global Unicast Address assigned to it. This is similar to the public address in IPv4. The Global Unicast Address is split into three parts. The low order 64-bits are the Interface ID and represent a device on the local network. The second part is the Subnet ID which allows a local network to be separated into multiple subnets. This is variable in length and can be from 0 to 64 bits as represented by the Subnet ID Length field typically found in the IPv6 setup. If the Subnet ID is less than 64 bits, the remaining bits are the Global Routing Prefix. Typically, the Subnet ID length is set to 16 bits. The Global Routing Prefix identifies the local network on the global internet.

DHCPv6 can be used to automatically assign Global Unicast Addresses for an interface in the same way that DHCP works for IPv4. DNSv6 can be used to convert host names to IPv6 addresses in the same way that DNS works for IPv4.

Multicast addresses in IPv6 use the prefix ff00::/8. So any address that begins with “ff” is a multicast address. The next byte contains flags that indicate if the multicast address is well-known or dynamic and the scope of the multicast group. Limiting the scope of the multicast address solves one of the problems of IPv4 multicast traffic which is how far should it be routed. IPv6 multicast scope bits allow a multicast traffic to be limited, for example, to a single interface (interface-local), to a single subnet (link-local) or global. Multicast will be discussed further in a later section.

3.3 Transport Layer

Above the IP Layer is the Transport Layer. This layer provides two types of data delivery services. The first is User Datagram Protocol (UDP) and the other is the Transmission Control Protocol (TCP). These provide different types of service as described in the following sections and are carried in the payload of the IP packet.

3.3.1 UDP

UDP provides a connectionless, best effort delivery service. Packets are sent from the source to the destination without any connection setup handshaking. The protocol does not provide for

acknowledgment and retransmission so that any lost packets are lost forever. The benefit of this protocol is that it only requires unidirectional communications; it is low latency since buffering for retransmission is not required; it is stateless, allowing it to easily scale to a large number streams and it has a small efficient header. The UDP packet is shown in Figure 4.

Figure 4 - UDP Packet Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	W
Source Port Number																Destination Port Number																1
Packet Length																Checksum																2
Payload Data (0 to 1476 bytes)																																

The port numbers are used to associate the packet with a process at the source or destination. The IP address identified the destination device for a packet. The port number identifies a destination process on the destination device for the packet. There are several types of port numbers, as shown in Table 2. For example, port 80 is used for HTTP service, so a sender who knows the IP address of an HTTP server, can use the well-known HTTP port number to send a packet to the HTTP server process.

Table 2 – Port Numbers

Port Number	Description
0 to 1023	Well Known ports for common services
1024 to 49151	Registered ports for specific services
49152 to 65535	Dynamic ports for any use

The packet length defines the total size of the UDP packet including both the header and payload. This is useful for very small packets which do not meet the minimum Ethernet packet size of 64 bytes, since these packets must be padded out to the minimum size. The checksum provides a data integrity check for the packet. It is optional in IPv4 and mandatory in IPv6. If it is not used, it is set to zero. At the receiving device, if the checksum test fails, the entire packet is discarded.

3.3.2 TCP

TCP provides a connection oriented, guarantee delivery service. A connection is established between the source and destination by exchanging three handshake messages called SYN, SYN-ACK and ACK. This allows both ends to establish a connection at the same time. In order to provide guaranteed delivery, TCP uses an acknowledge/re-transmission protocol with a sliding window. The window is the number of bytes that can be transmitted but not yet acknowledged. The sender can send packets until the number of un-acknowledged bytes fills the window. As bytes are acknowledged, more bytes can be sent. This improves throughput over protocols that acknowledge every packet. An acknowledgement does not need to be sent for every packet, it only needs to be sent for the last byte received. This acknowledges all prior un-acknowledged bytes.

In order to support the acknowledge/re-transmission protocol, the TCP header, shown in Figure 5, includes fields for the Sequence Number, Acknowledgement Number, Window Size and various flags. It also includes a checksum that protects the header and the payload.

Figure 5 - TCP Packet Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	W
Source Port Number																Destination Port Number																1
Sequence Number																																2
Acknowledgement Number																																3
Offset				Flags												Window Size																4
Checksum																Urgent Pointer																5
Payload Data (0 to 1464 bytes)																																

One drawback to this protocol is that buffering is required in both the sender and the receiver that is at least as large as the window size. This buffering holds bytes in the sender in case they need to be retransmitted and it holds bytes in the receiver in case retransmitted bytes need to be inserted before them.

When congestion occurs and packets are lost, the protocol will provide for retransmission, however, a delay is provided before the retransmission occurs. As more packets need to be retransmitted, the delay is increased in order to reduce the throughput of the TCP stream.

Finally, TCP provides a stream of bytes. Packets are transmitted at random sizes based on various conditions including timeouts and the retransmission window space available. The result of this is that bytes received at the destination need to be parsed for the start of the information content. Applications that use TCP need provide some method of identifying the start of the information element. Many internet applications use ASCII keywords to simplify the process.

3.4 Application Layer

At the Application Layer, an additional packet format is defined with specific information for transmitting PCM data over the IP network. One of the most contentious areas of TMoIP is the format of the TMoIP header. In the absence of a standard, many early adopters developed their own TMoIP header. This resulted in a large number of different header formats. Since then, there have been several attempts to standardize the header formats but competing requirements has not made this a very successful process.

3.4.1 IRIG 218-10 Packet Format

The Range Commanders Council (RCC) has developed the IRIG 218 standard, the latest version having been published in 2010. This is a very efficient TMoIP header consisting of only four bytes as shown in Figure 6. It is derived from the Pseudo-Wire protocol defined by the IETF.

Figure 6 - IRIG 218-10 Packet Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

300 Welsh Road Building 3
Horsham, PA 19044-2294
215.657.5270
www.gdpspace.com

Res	L	R	M	Res	Length	Sequence Number
Payload Data						

Six bits of the header are reserved (Res) and set to zeros. Four bits (L R, M) carry alarm information. The alarm bits are a carryover from circuit switched telephony over IP defined in the Pseudo-Wire standard. These bits are remapped from forward and reverse alarm indication to a local and remote error indication which is very poorly defined. The Length field is used only when the payload is less than 64 bytes. If the Length field is zero, then the payload length is determined from lower level protocols. In reality, this field is not needed at all because the length can always be determined from the IP or UDP layers. This is the result of not understanding the minimum Ethernet packet size requirement of 64 bytes. The real requirement is that the minimum Ethernet packet size is 64 bytes. Accounting for the 20-byte Ethernet header results in a minimum Ethernet payload of 46 bytes. If the IP/UDP packet headers and payload are less than 46 bytes, which is allowed, then padding is added to the Ethernet packet to fill out the payload to the required 46 bytes. This has no effect on the IP or the UDP length information which will always reflect the correct payload size. So the only real useful information in the IRIG-218-10 header is the sequence number. This field allows the receiving device to determine if packets are lost or if they arrive out of order.

The IRIG 218-10 packet format does not contain any information that would be useful in reconstructing the PCM output at the receiving device. The standard makes a reference to using the IETF RFC 1889 Real Time Protocol (RTP) to provide clock recovery support for TMoIP. The required RTP header fields are shown in **Error! Reference source not found.**Figure 7 and the IRIG 218-10 packet is placed in the payload of the RTP packet.

Figure 7 - RTP Packet Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	W
Ver		P	X	CSRC Count				M	Payload Type							Sequence Number																1
Timestamp																																2
SSCR Identifier																																3
IRIG 218-10 Header																																
Payload Data																																

While RTP does provide a 4-byte Timestamp, none of the other information is relevant. There is another 2-byte sequence number, a 7-bit Payload Type, a 1-bit Marker bit, a 4-bit CSRC Count, a 1-bit eXtention flag, a 1-bit Padding flag and a 2-bit Version. The CSRC is optional so the count would be zero, the extensions are not relevant, so the X bit would be zero and padding is not required, so the P bit would be zero. It seems to be a waste of space for the value received and an opportunity for confusion. For example, there is no definition for the format or resolution of the timestamp. This will lead to interoperability problems as different vendors use different time bases and resolutions. For this reason, many vendors of TMoIP products have developed their own TMoIP packet formats which they feel better serve the application.

The IRIG 218-10 packet format is really only designed for the PCM-to-PCM applications. It does not provide good support for the PCM-to-Computer applications. This is because there is no information in the header that indicated that the payload is frame aligned or the status of the frame and sub-frame synchronization functions.

3.4.2 IRIG 106 Chapter 10 Packet Format

IRIG 106 Chapter 10 is a telemetry data packetization format for recorders. However, it also specifies a UDP data distribution capability. This format is convenient because it allows telemetry data to be sent directly to software decommutation and display software in a standardized format.

The packetization first begins with a UDP transfer Header as shown in Figure 8**Error! Reference source not found.**. The Version is fixed at '1111'. If the Type is '0000', the UDP packet contains a one or more full Chapter 10 packets and the header consists only of the first four bytes. If the Type is '0001', the Chapter 10 packet is segmented across multiple UDP packets and an additional 8-bytes are added to the header to aid in recombining the segments.

Figure 8 - IRIG 106 Chapter 10 UDP Transfer Packet Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	W
UDP Transfer Sequence Number																								Type				Version				1
Reserved								Channel Sequence Number								Channel ID																2
Segment Offset																																3
Payload Data																																

The IRIG 106 Chapter 10 data then consists of Computer Generated Packets, Time Data Packets and Data Format Packets. Each of these Packets has a 24-byte Packet Header, an optional 12-byte Timestamp, a Packet Body and a Packet Trailer that contains Filler data and a 1-byte to 4-byte Checksum. The Payload Body begins with a 4-byte Channel Specific Data Word followed by data. Of particular interest for this paper are the PCM Data Packet types. The details of these formats are out of scope for this paper, but the basic concepts are useful to understand.

There are three types of PCM Data Packets: Packed, Unpacked and Throughput. Packed and Unpacked modes are frame aligned to either a 16-bit or 32-bit boundary so that the first bit of the minor frame is the first bit of the packet. Each minor frame in the packet is preceded by an Intra-Packet Timestamp and Intra-Packet Header. In Packed mode, the minor frame data is contiguous through the packet regardless of the word size. Fill is added at the end of the minor frame to meet the 16-bit or 32-bit alignment. In Unpacked mode, each word in the minor frame is put in a separate 16-bit or 32-bit word and zero filled as needed. In Throughput mode, there is no attempt to align the minor frames with the packets and the data is contiguous through the end of the packet.

As you can see, the IRIG 106 Chapter 10 packetization adds significant overhead to the PCM data.

3.5 Special Packet Header formats

GDP has been very flexible in implementing special header formats at the request of its customers. Typically, these special header formats are supporting some special data processing application in the computer that is receiving the TMoIP data. These headers may contain special status information or have specific formatting and positioning for the header information. A common change is the timestamp format or the addition of bit and frame sync status.

4 TMoIP Applications

There are two primary applications for the use of TMoIP. These are PCM-to-PCM data distribution over IP networks and PCM-to-Computer distribution over IP networks.

In the PCM-to-PCM data distribution application, shown in Figure 9, we are using the IP network to replace traditional PCM transport mechanisms such as copper cable (coax or twisted pair) links, microwave links, fiber optic cable links and matrix switches. In these applications, we start with PCM as serial data and clock signals. This is applied to the TMoIP ingress device which converts the serial data to parallel, formats the data into TMoIP packets and then outputs them over the network using TCP or UDP transport. At the receiving end, the TMoIP egress device receives the TMoIP packets, strips off the various packet headers and converts the parallel data back to serial data and clock signals.

The output clock generation function in the egress device is a complicated function. If there are timestamps in the TMoIP header, the device can use them to set a course output clock rate. However, since the timestamp generator and the output clock reference are not synchronized, the course output clock rate will not be exact. So the egress device must use the output buffer level to provide fine control of the output clock rate. In cases where there are downstream bit synchronizers or other data rate sensitive devices, corrections in the output clock rate must be small enough that they do not disrupt the operation of those downstream devices. If timestamps are not present, inter packet times must be measures and used to estimate a data rate or an output clock rate must be manually configured in the egress device. In either of these latter cases, fine control of the data rate based on buffer level must be performed. This is further complicated in the presence of packet loss. Those lost packets must be accounted for or replaced in the buffer level monitoring process.

Figure 9 - PCM to PCM Data Distribution

In the PCM-to-Computer distribution application, shown in Figure 10, we are using the IP network as a convenient input interface to a computer. The computer would typically be running a recording and/or decommutation application program. This usage has gained a lot of traction for applications where the data rates, numbers of PCM streams and processing requirements are sufficiently low enough that they can be supported by the processing power of the computer CPU. In addition, an Ethernet input provides a very simple and generic data interface to an application which eliminates the need for special hardware interfaces and operating specific device drivers. Also eliminated is the need to regenerate an output clock.

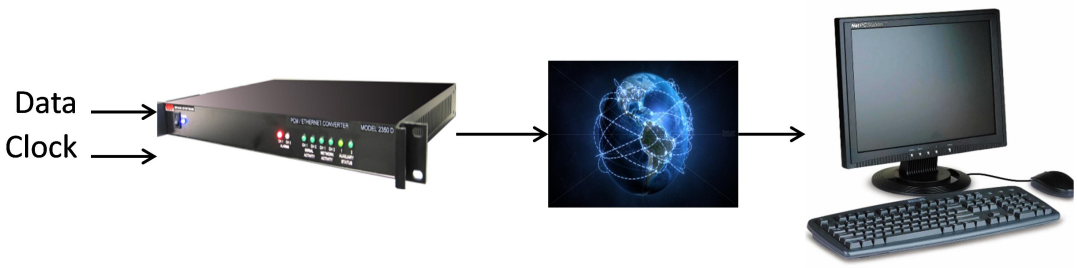


Figure 10 – PCM-to-Computer Data Distribution

5 TMoIP Network Design Decisions

5.1 IPv4 vs IPv6

One of the first choices to make in sending PCM data over an IP networks is which version of Internet Protocol will be used. IPv4 is the most common choice and currently the most widely supported IP protocol. It has been in use for several decades and most networking people are intimately familiar with designing and configuring IPv4 networks. IPv4 does a very good job of providing internetwork routing and supports the desired higher level data delivery services.

However, there is currently a move to implement and deploy IPv6 based networks. One major reason for promoting IPv6 is the IP address space. IPv4, which uses 32-bit addresses, is running out of globally routable addresses. IPv6 provides 128-bit addresses which should be sufficient for the foreseeable future. But the benefit of IPv6 is not just in the IP address size; IPv6 also makes other significant improvements over IPv4 that may be useful in TMoIP networks. These improvements include:

- Devices can have multiple addresses. These may have local scope or global scope.
- Auto-configuration of IPv6 addresses eliminates the need for DHCP, although DHCPv6 is available if desired.
- Network Address Translation devices are no longer needed because of the large address space. Private network protection can still be performed using firewalls.
- Simpler header format. Even though the header is larger due to the larger IP addresses, there are fewer fields in the header.
- Better multicast routing
- Simplified, more efficient routing using the Flow Label field. This is specifically designed for real time streaming data and helps to prevent out of order packets because all traffic from a source with the same label is routed the same way.
- Quality of service (QoS) support using Traffic Class field.
- Built-in authentication and privacy support using header extensions.
- Flexible options and extensions.

5.2 TCP vs UDP

Another choice to make in sending PCM data over an IP networks is the use of TCP or UDP transport mechanism. TCP provides a guaranteed delivery service that assures that the data being sent will arrive at the destination. This is achieved by implementing an acknowledgment and retransmission system. Data is sent up to some “window size”. At that point, no more data can be sent until some of the previously transmitted data has been acknowledged. The window size determines how much data must be buffered in the transmitter and receiver to allow retransmission of non-acknowledged data. The longer the round trip transit time between sender and receiver, the larger this window will need to be or the longer the sender will need to wait for acknowledgements. This buffering leads to increased latency as the buffer must be large enough and contain enough data so that it will not run dry while waiting for retransmission of missing data.

A second feature of TCP is that it provides a byte stream. You would think this could be useful in sending a stream of PCM data, however, this has some issues that make parsing a stream a little more difficult. Data sent to a TCP socket in the sending device does not necessarily end up in a single packet. The byte stream is divided into packets based on the TCP window size, timeouts and other criteria. So, the start of a TMoIP packet may no longer be aligned with the start of a TCP packet. Finally, the receiving device may not pass individual received TCP packet payloads to the receiving process, but may buffer up data based on read size or various timeouts. These features require the TCP byte stream to be parsed byte-by-byte or have pointers to find the next TMoIP packet header.

Finally, TCP service only provides a point-to-point connection. It requires communications in both directions in order to set up the connection even if you are only transmitting data in one direction. There is a connection setup process (SYN) that takes place and that must be completed successfully before any data is transmitted. This is not an issue in typical IP network environments, but there are some applications where bidirectional communications is not available. Some examples are: 1) RF transmission of TMoIP data or 2) transmission of TMoIP traffic through an “optical diode” for security reasons.

UDP does not provide a guaranteed delivery service. There is no connection process. Data is “fire and forget”. There is no acknowledgement that packets have been received and no retransmission of lost packets. The benefit of this service is that it is very low latency. No buffering is required. The down side is that lost packets are lost forever, however, in modern networks that are not congested, there is very little or no packet loss.

The second benefit of UDP is that it is a datagram service. That is, the data sent to a UDP socket in the sending device is sent intact within a single UDP packet (assuming the size does not exceed the MTU limit). If the data being sent is all the same size, then the UDP packets will all be the same size. Data delivered by the receiving device to the receiving process will be a complete UDP payload. If the sending device aligns a TMoIP header with the start of the UDP packet, the receiving device will receive that TMoIP header at the beginning of the received data. This significantly simplifies parsing of the stream.

Finally, UDP is unidirectional. It will work through RF links and optical diodes. It also does not require the receiving device to be ready when transmission starts. Receivers can come on-line later and the routing process will begin sending packets.

While the quick reaction may be to use TCP for TMoIP applications because of the guaranteed delivery of error free data, most practical applications use UDP for the following reasons:

1) Low latency
2) Simplified parsing of datagrams
3) Availability of multicast service
4) Does not require bi-directional connections

Most purpose built TMoIP networks are engineered to have sufficient bandwidth to support the desired PCM traffic level. In addition, these networks are built using switches and routers that that have sufficient aggregate throughput to support the traffic level. For these reasons, UDP packet loss is very rare. If the network is under-designed to the point where UDP packets are being dropped due to congestion, TCP may exacerbate the condition by increasing the congestion due to retransmission.

5.3 Multicast vs Unicast

UDP can operate in two ways: unicast or multicast. Unicast allows a packet to be sent from one sender to one specific receiver. This is a very simple, very straight forward method of passing data. For many TMoIP applications, this is the normal mode of distributing PCM data. The sender decides on the destination of the packets.

UDP also provides a multicast service where one sender can send to multiple receivers. This is a very effective and efficient mechanism to provide a similar capability as a non-blocking matrix switch where one input can be connected to multiple outputs. But a second feature of multicast TMoIP service is that senders can send without knowing the destination. This is very effective in those cases where multiple PCM streams need to be made available to everyone and the receivers will decide which PCM streams they want to receive.

There are two parts to sending multicast packets. First, the sending device will use a multicast address as the destination address of the packet. Second, the receiving devices must JOIN the multicast group represented by the multicast address. By joining the multicast group, a receiving device indicates to the network elements (routers and switches) that packets for that group should be sent to that device. IPv4 devices use the Internet Group Management Protocol (IGMP) protocol to join a multicast group, while IPv6 devices use the Multicast Listener Discovery (MLD) protocol.

5.4 Private Networks vs Shared Networks

A Private network is one that is specifically allocated to transmitting PCM data. No other data flows are present. As that name implies, a Shared network is shared between PCM data and other data types. The type of network used will depend on the criticality of the data. If the purpose is for casual quick look of some PCM stream and the data rate is modes, a Shared network may be perfectly suitable. In other cases, a Private network may not be possible because the public internet may need to be used. Obviously, in these cases, you will have less control over the quality of service and reliability of the PCM data transmission. The extreme case of a shared network is the public internet.

However, if the transmission of PCM data is mission critical, it is recommended that a Private network be built. This will isolate the PCM traffic from any other less critical but potentially disruptive traffic. If a

completely physically private network is not possible, it may be possible to approximate a Private network by using Virtual LANs (VLAN) or Multi-Protocol Label Switching (MPLS) tunnels. Virtual LANs separate different traffic flows using VLAN tags, an optional field in the Ethernet Layer header. Network switches then use these tags to segregate the traffic into different virtual networks, preventing traffic received from a device on one VLAN from being sent to a device on another VLAN. Communications between VLANs can take place at Inter-VLAN routers which is a function present in Layer 3 switches. Generally, the VLAN tag is applied at the ingress to the network (input of the first switch) and removed at the egress from the network. This makes VLAN traffic separation transparent to the connected devices. Conversely, some devices allow the VLAN tags to be specified at the input and output of the device. In this case, the ingress and egress switches would need to be programmed to pass the VLAN tags.

Multi-Protocol Label Switching (MPLS) is another mechanism used to create a private tunnel through a public network. A label is inserted between the Ethernet and IP layers. A route from source to destination is first established based on meeting bandwidth and latency criteria. If a node (router) can support the requested bandwidth and latency, a next hop route is associated with the label for that node. Once the destination is reached, there will be a path from source to destination through the intermediate nodes that can reserve and support the requested bandwidth. When a packet is received that contains a label, the next hop is determined from the label table, bypassing normal routing decisions and quickly forwarding the packet.

Finally, sending sensitive data over public networks raises certain security issues. There are many options for encrypting the TMoIP stream. If TCP is used, SSL/TLS is available to provide encryption in a standard way. If UDP is used, there are several methods to provide encryption in an unreliable delivery environment. Datagram Transport Layer Security (DTLS) has been defined to address this gap in a way that approximates TLS. DTLS, however, has the drawback of implementing acknowledge and retransmission of missing datagrams. ITU-T H.235.6 defines streaming encryption for audio and video that can be readily adapted to TMoIP without acknowledgement and retransmission.

5.5 Data and Management Traffic Separation

Separation of Data traffic and Management traffic into different networks provides two advantages in TMoIP systems. First, it removes the computer systems that perform the management (setup, configuration and status) function from the same network that is being used for the TMoIP data. As mentioned earlier, this has the benefit of removing a lot of spurious operating system generated network traffic from the data network. This traffic has the potential to increase network jitter resulting in higher latencies.

The second advantage is security. Often, the TMoIP network will be carrying classified data which must be segregated from other data. Eliminating computer systems from that network reduces vulnerability and simplifies the Information Assurance certification of the network.

5.6 Quality of Service Functions

There are a variety of Quality of Service mechanisms that can be used to improve the performance of the TMoIP network.

5.6.1 Traffic Classes

Both IPv4 and IPv6 provide fields in their headers for marking the packets with a Class of Service. In IPv4, this is the Type of Service (TOS) field also called the Differentiated Services Code Point (DSCP) field. In IPv6, it is the Traffic Class field. These fields are used to identify specific traffic flows that should have special handling applied as the packets pass through the network. For example, all normal traffic could be given a Class of 0, Voice over IP traffic could be given a Class of 6 and Telemetry over IP traffic could be given a Class of 7. In this way, Voice and Telemetry traffic can be given special handling.

Setting the traffic class alone, does not provide any improved quality of service. You must also configure the routers in the network to treat the classes differently. For example, traffic classes 6 and 7 can be given Expedited Forwarding treatment and class 0 can be given Assured Forwarding. Expedited Forwarding puts those packets into very short, high priority queues. Whereas Assured Forwarding packets are put into longer, low priority queues. The router will output packets from the high priority queues before outputting packets from the low priority queues. This helps to minimize latency and jitter. It also helps to avoid dropped packets as network congestion grows. Assured Forwarding is typically used for low priority TCP packets, knowing that these packets will eventually be retransmitted if dropped.

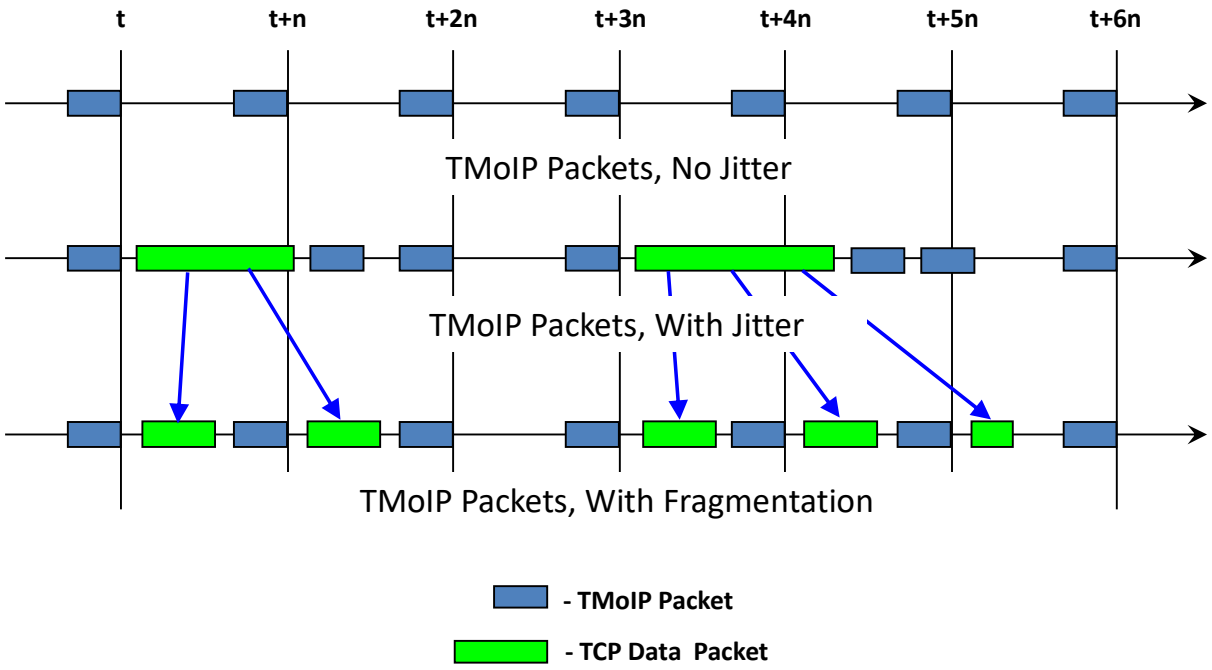
There are many other features that can be programmed in the routers that determine how different classes of service are treated. There are different queueing algorithms: First-in-First Out, Weighted Fair Queueing, and Class Based Weighted Fair Queueing. These are used for prioritizing the traffic flows. Next, there are Congestion Avoidance tools that attempt to predict congestion and then discard packets in order to avoid congestion. These include: Random Early Detection and Weighted Random Early Detection. In both cases, impending network congestion is detected when queues are nearing their full level. When this is detected, packets from lower priority queues are dropped. Dropped UDP packets instantly reduce the required bandwidth because there is no retransmission. Dropped TCP packets cause the sender to take two actions. First, they delay their retransmission as more packets are dropped, and second, they decrease their acknowledgement window sending fewer packets between acknowledgements. Both of these actions work to reduce the required bandwidth to support the TCP traffic.

For Quality of Service mechanisms to be effective there should be an end-to-end policy which is implemented in each network device that the data will transit through. If not, there is the potential that the desired handling will fall apart in the unprotected segments, and quality of service will not be maintained.

5.6.2 Packet Fragmentation and Interleaving

For those cases where the TMoIP data cannot be separated from common Ethernet data, Packet Fragmentation and Interleaving can help improve performance. Packet fragmentation and interleaving is a mechanism that helps to reduce the jitter of high priority, low latency real time data streams such as TMoIP. It works in the routers by chopping up large TCP packets into smaller pieces. This prevents the UDP traffic from having to wait while large packets are being sent, minimizing the packet arrival time jitter. This does not affect the TCP traffic which just spreads the byte stream over more packets. This is shown in Figure 11Figure .

Figure 11- Packet Fragmentation



5.7 Bandwidth Considerations

It is critical to assure that the volume of telemetry data to be transmitted over the TMoIP network does not exceed the capacity of the network. This is easier if the network is a TMoIP-only network and is not shared with other traffic. But even in this case, care must be taken to account for the packetization overhead when determining the aggregate bandwidth requirements. Even after accounting for the overhead, it is important to limit traffic to about 85% of the wire speed to account for transmitter and receiver recovery time and intermediate network equipment processing loads. Modern network devices are designed to operate at wire speeds, however, they often have packet processing limitations. Small packets at very high data rates may overload the devices ability to process and pass data.

In those cases where the network is shared with other Ethernet traffic, it is difficult to estimate the impact of the non-TMoIP data. This data tends to be bursty and unrestricted in data rate. Generic Ethernet traffic such as HTTP, FTP, SMTP, POP is very intermittent and unpredictable. This type of traffic can cause serious issues with the reliability of TMoIP traffic if the aggregate data rates are approaching wire speeds or the data rate limitations of intermediate network equipment.

6 TMoIP Problems and Issues

6.1 Overhead

All of the functionality provided by the various network layers comes at the price of overhead. The payload data is the information that we want to send. Each layer wraps the payload data with header and possibly trailer bytes. This then becomes the payload for the next layer which again adds header

and trailer bytes, and so on. Whether it is video, telemetry or email, it is all burdened with this overhead. The Ethernet packet header consists of 18 bytes. In addition, each Ethernet packet has a preamble / start of frame delimiter of 8 bytes and an inter-packet gap of 12 bytes. The IPv4 header consists of 20 bytes. The UDP header size is 8 bytes and the TCP header size is 20 bytes. This is shown in Table 3. These are the minimum header sizes and they do not contain any optional bits such as the VLAN tag, which will further increase the header sizes. Obviously, payload size has a tremendous impact on the efficiency of TMoIP. The Ethernet packet payload is limited to the Maximum Transmission Unit (MTU). The maximum MTU is 1500 bytes, however, this may be set lower in some networks. The IP header and UDP/TCP header size must be subtracted from the MTU in order to determine how much payload the UDP or TCP packets can carry.

Table 3 - IPv4 Overhead (in bytes)

Preamble, SOF and Inter-packet Gap	Ethernet Header Size	IPv4 Header Size	UDP Header Size	TCP Header Size	TMoIP Header Size	Total Overhead	Maximum Payload Size	Minimum Overhead
20	18	20	8		4	70	1472	4.7%
20	18	20		20	4	82	1460	5.6%

The IPv6 header contains 20 bytes. The effect on overhead is shown in Table 4.

Table 4 – Ipv6 Overhead (in bytes)

Preamble, SOF and Inter-packet Gap	Ethernet Header Size	IPv6 Header Size	UDP Header Size	TCP Header Size	TMoIP Header Size	Total Overhead	Maximum Payload Size	Minimum Overhead
20	18	40	8		4	90	1448	6.2%
20	18	40		20	4	102	1436	7.1%

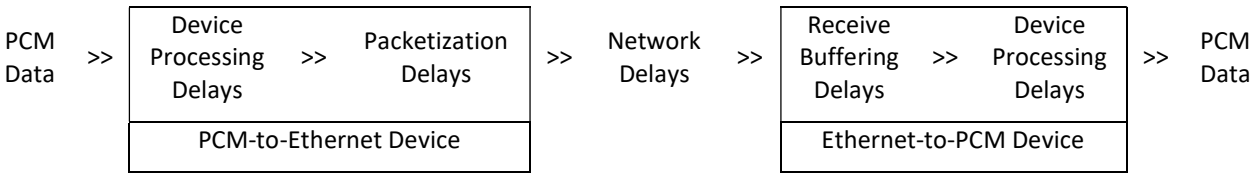
The payload size is one parameter that we have control over in producing TMoIP packets. Making the payload larger will reduce the overhead percentage and increase the transmission efficiency at the expense of latency. Conversely, making the payload smaller will increase the overhead percentage and decrease the transmission efficiency, but the latency will be reduced.

Jumbo Frames allow the Ethernet packet payload to expand beyond the 1500 byte limit up to 9000 bytes. However, these are illegal Ethernet packets and network equipment must specifically support the use of Jumbo Frames. If they do not, packets may be truncated or chopped up into smaller packets as they pass through various network devices.

6.2 Latency

Latency is the delay from the time a bit enters the PCM-to-Ethernet device until that same bit leaves the Ethernet-to-PCM device. Everyone wants zero latency; however, some latency is unavoidable. The good news is that to a certain extent it is controllable. End-to-end latency is made up of several components as shown in Figure 12.

Figure 12 - TMoIP Delay Chain



Device Processing Delays are vendor specific and depend on serial-to-parallel / parallel-to-serial conversion, DMA buffering, packet processing and network stack delays. Efficient design will minimize these delays, but they are out of the control of the user.

Packetization Delays are often user controllable, but there is a trade-off. Data is buffered while enough data to fill a packet is received. Once the packet is filled, the packet can be sent. So making smaller packets will minimize that delay. This comes at the expense of efficiency because the smaller the packets, the larger the impact of packet header overhead, thus the larger the effective data rate will be. For example, converting a 1 Mbps PCM stream to 100 byte IPv4 UDP packet will have an 800 microsecond packetization delay, but will require 50% overhead resulting in an effective data rate of 1.5 Mbps on the wire.

Network delays are the time it takes a packet to traverse the network from source device to destination device. This delay is not only made up of the travel time of the electrical signals, but also the delays through the intervening network devices (switches, routers, etc). These devices each have their own processing, packetization and buffering delays, but they also have queueing delays which become significant as the network becomes congested. Applying Quality of Service mechanisms in these devices can minimize some of these delays.

Receive Buffering Delays are similar to Packetization Delays in that the entire packet must be received before any of the data can be processed. This is because a Frame Check Sequence at the end of the Ethernet packet must be checked before the packet is released to the waiting processes. Additional buffering may also be required at the receiving end to compensate for network jitter. Jitter will be discussed later, but essentially, this additional buffering assures that continuous data will be available for output even though the traffic on the network may be bursty. In networks that are congested or have a large amount of non-TMoIP traffic, jitter compensation buffering will need to be larger.

As a result of all of these components, it is very difficult to say exactly what the latency will be. However, assuming the network is well designed and not in a congested state, all of the device and network delays should be fairly constant, allowing the user to control latency by specifying the packet size and jitter buffering.

6.3 Skew

Skew is the channel-to-channel difference in the latencies. If this difference is great, then PCM events that occur at the same time at the source will no longer occur at the same time at the destination. Like latency, users always want zero skew, and like latency, some skew is unavoidable. Supposed that you have two PCM data streams, one at 5 Mbps and one at 32 Kbps. Supposed that you also want to maximize efficiency so you use large packets for both streams. There will be a significant difference in the latency of the two streams resulting in a very large skew. Now a user can attempt to compensate for this manually by adjusting the packet size of each stream, but this becomes difficult if the data rates change often, or especially if they are not know at the time the link is configured. However, there are two automatic ways to minimize skew.

One method is to timestamp each TMoIP packet at the source, to synchronize all TMoIP devices to a common time base, to identify the largest latency of all channels of interest, to use that latency as an offset to the current time for establish a playout time, and then to playout the TMoIP data when the timestamp matches the playout time. IRIG B, Network Time Protocol (NTP) and IEEE-1588 Precision Time Protocol (PTP) are mechanisms that can be used to synchronize all of the devices to a common time base. Determining the longest latency and communicating that to all devices is where things get a little more difficult.

Another approach is to attempt to control all of the latencies to a common value. Assuming the device and network latencies are fixed, this amounts to controlling packet size and buffering for each channel of interest. This is made a little more complicated it the different PCM streams are at different data rates. The GDP TMoIP devices have a latency control mode that allows the user to specify the desired latency. By setting this value to be the same for all channels, the units will set the packet size appropriately based on the measured PCM input data rate and the selected jitter buffering. In this way, it does not matter what the data rate is of the PCM streams that are connected, all streams will experience the same latency. No configuration of the receiving devices is required nor is timing distribution, and this approach adapts to changing data rates.

Skew control is further complicated when there are multiple source or multiple destination devices across which the streams are assigned. In these cases, time must be accurately distributed to all of the devices in the system.

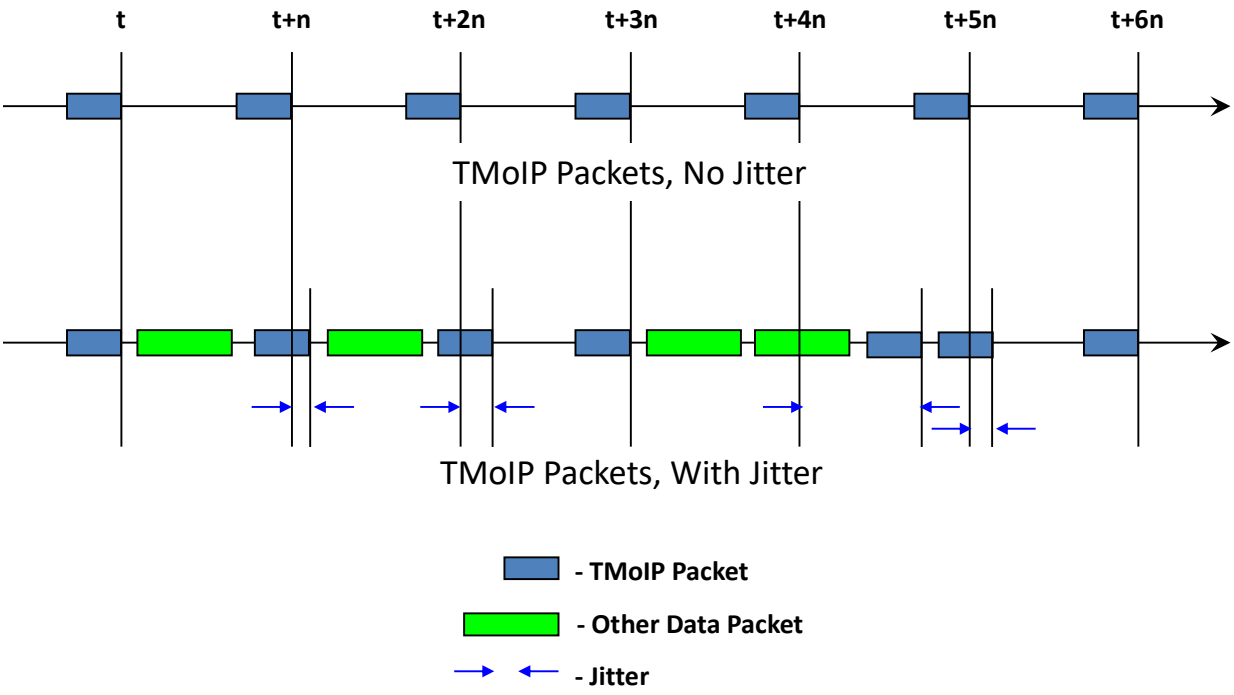
6.4 Jitter

PCM data has a constant, continuous data rate. Every bit time, a new bit of information is available and the bit clock is generally at a constant stable rate. Once the PCM data is packetized and sent onto the network, this is no longer the case. The packets containing the PCM data are sent at higher data rates in short bursts. This is shown at the top of Figure . If there is no other traffic on the network, then the packets can be sent at a regular rate and will reach the destination at that same rate, making it easy to convert back to a constant bit rate serial PCM output.

But when other traffic is present on the network, the TMoIP packets may not be able to be sent at a regular rate. They may need to wait for other packets to be completed or network equipment may be queueing up packets from different sources. As a result, when packets arrive at the destination, they have jitter as shown at the bottom of Figure 13. This can be especially troublesome when small TMoIP

packets are used to reduce latency and the network has large TCP packets from web browsing, email, or file transfers.

Figure 13 - Packet Arrival Time Jitter



The problem with jitter is that when data is needed for the parallel to serial conversion process, the next packet may not yet have been received. The serial PCM steam is a continuous stream of data. Any break in that stream will result in loss of bit sync or decom lock and errors in the resulting data. To compensate for this, buffering must be added in the receiving device and sufficient data must be held in the buffer to assure that data will always be available for playout while the device is waiting for additional packets to arrive. The amount of buffering needed will depend on the magnitude of the jitter. Typically, the minimum jitter buffer is two packets so that while one is being output, the second can be received. As jitter increases, the number of packets in the buffer needs to increase, and this further increases the latency.

6.5 Packet Loss

Packet Loss occurs when packets are sent over the network and they do not reach the destination. There are many reasons for packet loss which will be discussed in this section.

The first reason is physical. In this case there is a bad, damaged or out of spec cable. This would also include damaged connectors which do not seat properly and damaged or faulty network equipment. In these cases, the result may be bit errors in the data transmission. The Ethernet layer contains a CRC check, which if it fails causes the entire packet to be dropped. So you do not get bit errors over the network, just missing data.

The next common cause is incorrectly configured Ethernet data rate and duplex settings. This is very commonly misunderstood. If two devices are connected together using an Ethernet cable, both devices must be configured the same way. The auto settings for the network interface means auto-negotiate the data rate and the duplex. If both devices are set to auto-negotiate, then they each advertise their supported data rates and duplex and then select the highest common rate. Today, this tends to be pretty reliable method of configuring the interfaces. If both devices are manually configured, they must be configured the same. This seems obvious, but is the cause of many networking problems. If the data rates are set differently, there will be no communications between the devices. If the duplex is set differently, there may appear to be a connection. Pings may work properly; some data may be received successfully. However, the communications may be unreliable and may result in dropped packets due to collisions, see Table 5. However, if one device is set to auto-negotiate and the other device is set manually, the negotiations will fail. In that case, the auto configured device will default to 10Mbps and Half Duplex. If the manually configured device is set to Full Duplex, a duplex mismatch will occur with the resulting dropped packets. Some devices will provide auto-detection for the data rate when negotiation fails. This will provide a data rate match. However, there is no auto-detection for the duplex.

Table 5 - Duplex Mismatches

		Device A Duplex Setting		
		Auto	Half	Full
Device B Duplex Setting	Auto	OK	OK	BAD
	Half	OK	OK	BAD
	Full	BAD	BAD	OK

These first two causes of packet loss are easily corrected. The next cause is a little more difficult. That is packet loss due to congestion. Congestion is the condition where more data is passing through a network device than the device is able to handle. In this case, the device will discard packets in order to prevent buffer overflow. The best way to avoid congestion is by understanding the total data requirements for the network and assuring that it does not exceed the capability of the network equipment. Having a 100 Mbps Ethernet network does not mean that you can transmit 100 Mbps of PCM data. In addition to the packetization overhead previously described, there may be other data being sent on the network. One example is the ARP resolution packets and any other higher level protocols. This is especially true if there are computers on the network which may be trying to connect to DNS servers, WINS servers, file servers, and other devices such as printers. This traffic is taking place even when you think the computer is idle. In addition, if it is not a TMoIP-only network, there will be other “normal” network activity such as web browser (HTTP), email (SMTP/POP), file transfer (FTP) and other data traffic. In practice, in a well-designed TMoIP-only network, you may be able to get to 85% of the wire speed.

Packet loss not only affects the integrity of the output of the Ethernet-to-PCM data stream, it can also affect the output clock rate. Typically, the Ethernet-to-PCM device will monitor buffer fullness in order to provide fine control of the output clock to prevent buffer overflow or underflow. If packets are dropped, this can throw that mechanism out of whack resulting in abrupt changes in the output clock rate. To prevent this, the missing packets must be accounted for in the buffer measurement process.

7 Troubleshooting

There are two primary categories of problems that may affect TMoIP networks, shown in Table 6. The first category is connectivity problems. In this case, packets sent from the source do not reach the destination. There are several causes of connectivity problems. In these cases, a route does not exist between the source and the destination. This may be the result of a physical break in the network path, either due to cabling or architecture. This may be the result of mis-configuration of firewall or network address translation (NAT) functions which make destinations unreachable even though the physical connection is possible. Similarly, having the source and destination on different VLANs with no intervening VLAN router will make the destination unreachable. And finally, if the traffic is using multicast, multicast traffic may be blocked by routers or switches.

Table 6- TMoIP Problems

- TMoIP Problems
  - Connectivity Issues
    - No Physical Route
    - Firewall or NAT misconfiguration
    - VLAN isolation
    - Blocked multicast traffic
  - Dropped Packet Issues
    - Physical Problems
    - Duplex mismatch
    - Congestion
    - Excessive data rate

To troubleshoot these types of problems, utilities such as Ping or Tracert can be useful. Some TMoIP devices provide these utilities within the device. If the device does not, then a laptop can be substituted for the source device in order to use the utilities. Ping can then be used to walk from source to each intermediate network device in order to determine where the route is blocked. At that point the configuration of that network device can be investigated. There may be cases where the Ping is successful, but there still is not connectivity from source to destination for TMoIP traffic. In this case, it is likely that a Firewall is blocking the UDP traffic but passing ICMP traffic which is used for the Ping utility. Additionally, a switch or router may be blocking multicast traffic, but passes unicast UDP and ICMP traffic.

The second category of problems that affect TMoIP networks is dropped packets. In this case, there is connectivity, and some data gets from source to destination, however, some packets are lost which results in data errors at the destination. This may be the result of a physical problem such as a damaged cable, connector or electrical interface. Another possible cause of packet loss is duplex mismatch which was previously discussed. Network congestion may also result in dropped packets, and is also discussed in a previous section. There may also be excessive packet jitter which in some cases may cause buffer underflow in the Ethernet-to-PCM device which results in data loss. And finally, exceeding the network data rate will result in lost packets. While this seems obvious, it may not always be easy to identify. This is because intervening network segments or connections may be operating at a lower data rate than expected. For example an intermediate segment may be configured for 10Mbps or a wide area network connection may be going through a low data rate telecom connection.

These conditions can cause very intermittent errors which can occur anywhere in the network chain and may be very difficult to track down. Troubleshooting these problems is equally difficult. Checking duplex and data rate settings along the network path is an easy task. Checking for congestion within network devices may be done by checking the device statistics. While checking these conditions in private networks may be relatively easy, it may be much more difficult in public networks where you do not have control or access to intermediate network devices. Physical problems are very difficult to find. A cable may be crushed or bent in such a way that the impedance is affected. A break in ground signal may affect common mode noise rejection. In improperly polished or crushed fiber optic cable may decrease signal quality. Bent pins and pushed out pins may result in intermittent connections. These will all lead to dropped packets and can only be found through careful examination of cables, connections and equipment in the network path.

8 Example Systems

This section provides descriptions of a couple examples of TMoIP systems.

The first example, shown in Figure 14, is a simple microwave replacement where eight channels of PCM data needed to be sent from Point A to Point B. The challenge here was that the customer wanted to be able to connect any PCM stream to any channel regardless of data rate and have the resulting PCM streams maintain their original channel-to-channel timing relationship (skew). The customer did not want to have to configure the channels before use, but essentially wanted the link to act like a coax cable. This was achieved by establishing a target latency for all the channels. The data rate of each channel is measured at the PCM-to-Ethernet device and used to control the packet size in order to achieve the desired latency target. The system successfully achieved the customer's requirements.

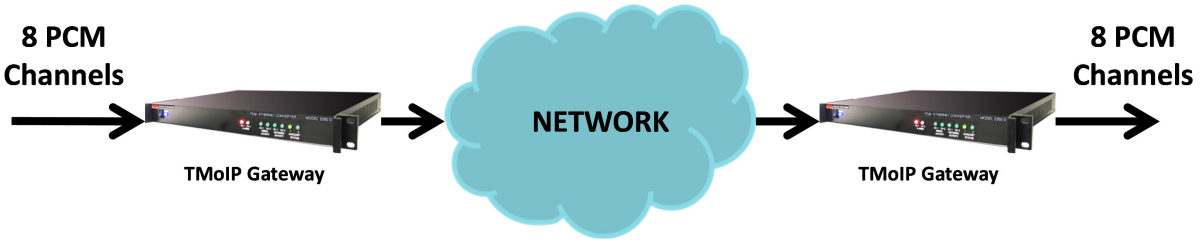


Figure 14 - Microwave Replacement Application

The second example, shown in Figure 15, is a very large TMoIP distribution and switching system. There are twenty remote sites that acquire PCM date over RF links. At the remote location, the PCM data is put onto the IP network either directly from TMoIP capable receivers or from legacy or third party receivers using TMoIP gateway devices. The system includes 80 channels of best source selection and can output best source streams to three separate control rooms for decommutation, processing and display. Key to this application is a software package that provides central control of the entire system. The software provides four high level functions: System Configuration, Mission Definition and Activation, System Status and Report Generation. The System Configuration function provides for the addition of remote sites, control rooms, TMoIP sources, Best Source Selectors and TMoIP destinations. Multicast addresses for all streams are auto assigned and managed by the software. All source and destination channels are named and abstracted from the hardware. Mission Definition and Activation allows

sources to be combined into best source groups. Best source groups can then be sent to control rooms. When the mission is activated, best source selector resources are automatically allocated to the mission and PCM data is routed by configuring TMoIP destination devices with the appropriate multicast group to receive. The System Status display shows the quality, data rate and activity of each stream used in the mission. When the mission is completed, the Report Generator function creates a time correlated report of channel activity, channel quality and best source selection results for all streams for the duration of the mission or a subset of mission time.

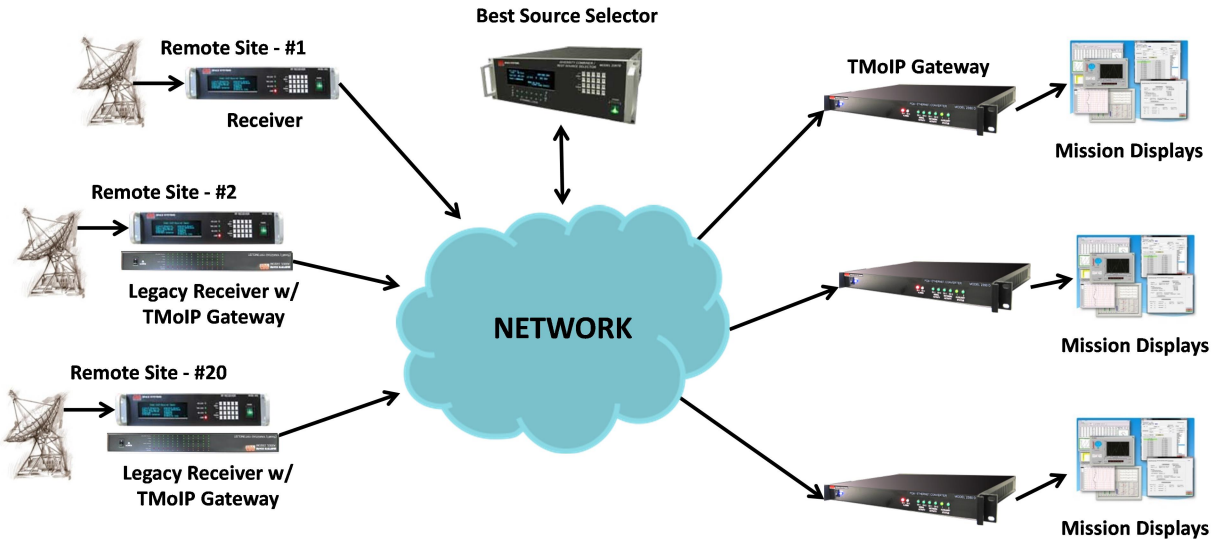


Figure 15 - Large TMoIP System Application

9 Conclusions

IP networks provide a very efficient and effective means of distributing real-time low latency PCM data as long as attention is paid to the network architecture, packetization and buffering. Commercial use of IP networking is driving down the cost and driving up the bandwidth of networking devices. Today's network devices can be used to build very robust and reliable IP systems for TMoIP transport.

The wider use of TMoIP requires telemetry engineers to become proficient in network configuration and troubleshooting.

Currently, IRIG 218 attempts to standardize the TMoIP protocol. However, in many cases, the requirements are not explicitly stated, resulting in spotty acceptance and limited interoperability. IRIG 218 will be soon undergoing revision and the hope is that some of this real world experience can be taken into account in producing a more widely accepted standard. The wider use of TMoIP requires telemetry engineers to become proficient in network configuration and troubleshooting.