



## CHAPTER 7

### Packet Telemetry Downlink

Acronyms .....	7-iii
7.1 Packet Telemetry .....	7-1
7.2 Minor Frame Format .....	7-1
7.2.1 Minor Frame Sync Word .....	7-1
7.2.2 Minor Frame Data Words .....	7-1
7.2.3 Golay Code Protection .....	7-2
7.3 Asynchronous Packet Multiplexing .....	7-2
7.3.1 Standard Packet Encapsulation .....	7-2
7.3.2 Low-Latency Packet Encapsulation .....	7-3
7.3.3 Packet Encapsulation Structure .....	7-4
7.4 Packet Format .....	7-5
7.4.1 Fill Packet .....	7-6
7.4.2 Application-Specific Packet .....	7-6
7.4.3 Test Counter Packet .....	7-6
7.4.4 Chapter 10 Packet .....	7-7
7.4.5 Raw Ethernet Media Access Control Frame Packet .....	7-8
7.4.6 Ethernet Internet Protocol Packet .....	7-8
7.4.7 iNET TmNS Packet .....	7-8
7.5 Randomization, Encryption, and Error Correction .....	7-9

#### Table of Figures

Figure 7-1. Minor Frame Illustration as a Series of Bytes .....	7-1
Figure 7-2. Golay Code Encoding and Decoding .....	7-2
Figure 7-3. Overview of the Packet Encapsulation Mechanism .....	7-3
Figure 7-4. Packet Encapsulation Mechanism with Low-Latency Packets .....	7-3
Figure 7-5. Minor Frame Structure .....	7-4
Figure 7-6. Minor Frame Header, Unprotected Part .....	7-4
Figure 7-7. Minor Frame Header, Golay Code Protected Part .....	7-4
Figure 7-8. Packet Structure .....	7-5
Figure 7-9. Packet Header, Protected Bytes .....	7-5
Figure 7-10. Chapter 10 Packet with Protected Header .....	7-7
Figure 7-11. Protected Part of the Chapter 10 Header .....	7-7
Figure 7-12. Unprotected Part of the Chapter 10 Header .....	7-8
Figure 7-13. iNET TmNS Packet Structure .....	7-8
Figure 7-14. TmNSDataMessageHeader Golay Coded Part .....	7-9
Figure 7-15. TmNSDataMessageHeader Unprotected Part .....	7-9

This page intentionally left blank.

## Acronyms

FCS	frame check-sum
IAW	in accordance with
iNET	integrated Network Enhanced Telemetry
IP	internet protocol
LLP	low-latency packet
MAC	media access control
PCM	pulse code modulation
TmNS	Telemetry Network System

This page intentionally left blank.

## CHAPTER 7

### Packet Telemetry Downlink

This standard defines a pulse train structure in accordance with (IAW) [Chapter 4](#) to transport variable-length known-format packets. This standard defines the method to incorporate Chapter 10 packets, integrated Network Enhanced Telemetry (iNET) Telemetry Network System (TmNS) packets, and Ethernet data packets into the pulse code modulation (PCM) stream.

#### 7.1 Packet Telemetry

Packets are inserted asynchronously into a PCM stream minor frame. Data information is encapsulated in type-specific variable-size packets that support multiplexing and telemetering of different types of packets simultaneously in a single PCM stream. Packet types such as Chapter 10 packets, TmNS data messages, and Ethernet data are identified in the packet header.

#### 7.2 Minor Frame Format

The minor frame is a fixed-length PCM frame. Transmission is most significant bit first.

##### 7.2.1 Minor Frame Sync Word

The minor frame uses a 32-bit sync word. The sync word shall be 0xFE6B2840 if no error correction is used. A sync word 0x1ACFFC1D is used if the optional Reed-Solomon error correction is applied to the PCM stream.

##### 7.2.2 Minor Frame Data Words

The size of data words is 8-bit (referenced hereafter as byte). All included structures are byte aligned and their placement in the minor frame is big-endian.

The number of data words in the minor frame size shall be  $N \times 223$  bytes, where  $N$  is between 1 and 8. This length supports the optional Reed-Solomon error correction without additional overhead.

The minor frame structure is presented as a series of serial bytes in [Figure 7-1](#).

	7	0
1	SYNC WORD (bits 31..24)	
2	SYNC WORD (bits 23..16)	
3	SYNC WORD (bits 15..8)	
4	SYNC WORD (bits 7..0)	
5	DATA BYTE 1	
	...	
	...	
$4+N*223$	DATA BYTE $N*223$	

Figure 7-1. Minor Frame Illustration as a Series of Bytes

### 7.2.3 Golay Code Protection

A single-bit transmission error may cause excessive data loss in packet telemetry. If the error occurs in identification or structure length fields, it can lead to misinterpretation of the packet or a loss of a series of packets.

This is why a self-correcting coding called extended binary Golay code (hereafter called simply Golay code) is applied to structure-critical elements in the PCM stream. This additional coding allows protecting the packet identification and packet length information and supports correction of up to 3-bit transmission errors in a 24-bit sequence. This is accomplished by encoding 12-bit words into 24-bit words. The coding and decoding of the Golay code is illustrated in [Figure 7-2](#).

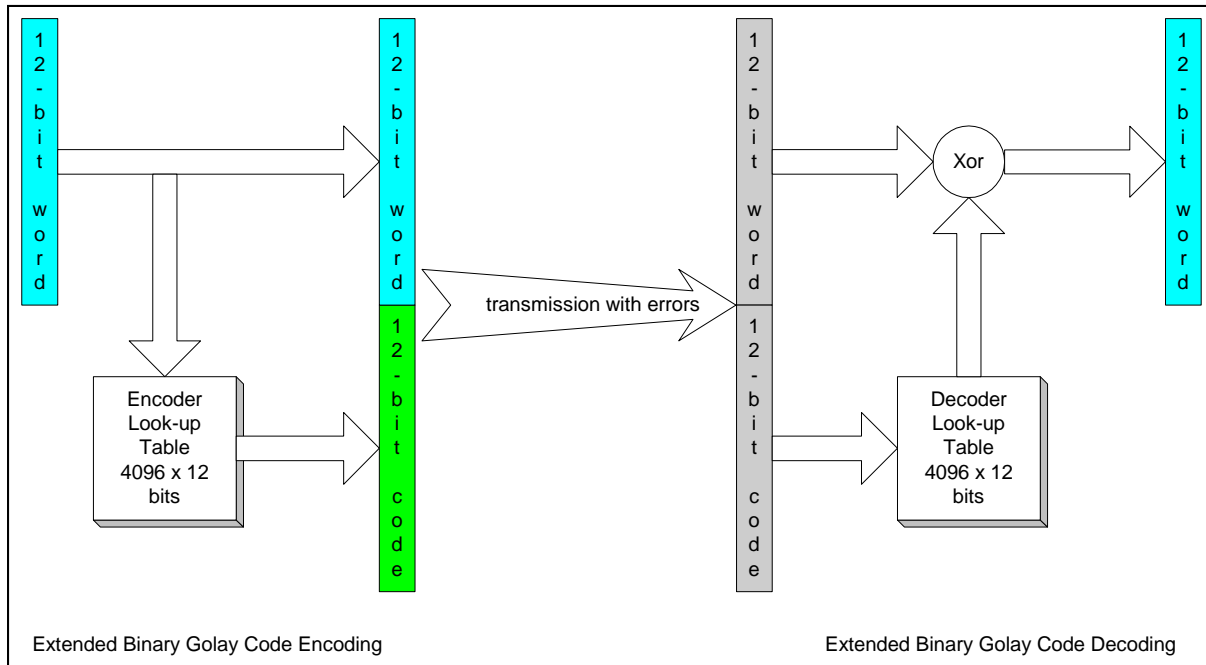


Figure 7-2. Golay Code Encoding and Decoding

Golay encoding shall be carried out IAW [Appendix Q](#).

The Golay code protection method is applied on the structure-critical elements that are explicitly indicated in the following paragraphs. Elements such as structure length and type information fields are Golay code protected.

## 7.3 Asynchronous Packet Multiplexing

The minor frame contains asynchronously inserted packets. To keep the overhead low, the size of the packets can be longer than the minor frame – so one packet may span over several minor frames. Packets are transmitted seamlessly; a new start of packet must follow immediately after the last byte of a packet.

### 7.3.1 Standard Packet Encapsulation

In order to find the start of a packet, a fixed-length minor frame header is placed at the beginning of the minor frame. The minor frame header contains an offset to the first byte of the first packet in the minor frame – provided there is at least one start of packet in this minor frame.

It is not necessary to have one or more start of packet in every minor frame. One packet may span over multiple minor frames (see Packet N+1 in [Figure 7-3](#)). When a packet spans multiple minor frames, the start of the packet exists only in the first minor frame (including the packet header and any content headers). The continuation parts of the packet in the consecutive minor frame(s) are not considered as the start of a packet, so there will be neither packet header included, nor will be offset to the first start of a packet stored for these parts in the minor frame header. The overview of the packet encapsulation mechanism is shown in [Figure 7-3](#).

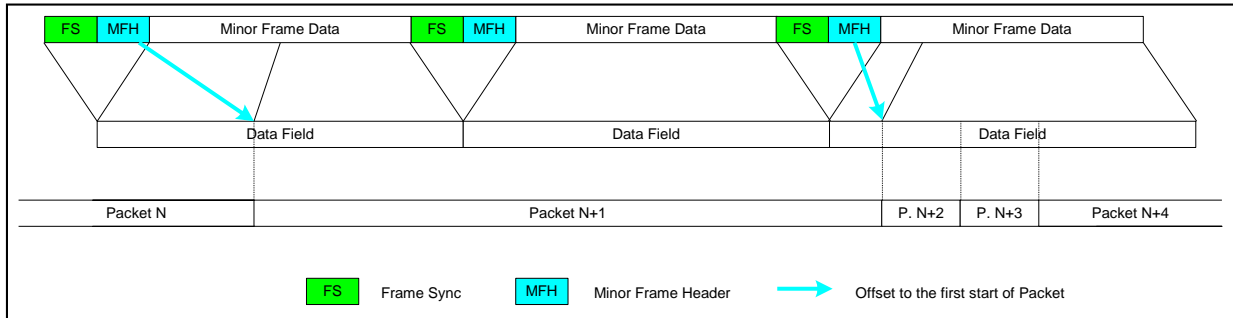


Figure 7-3. Overview of the Packet Encapsulation Mechanism

### 7.3.2 Low-Latency Packet Encapsulation

The transmission of a long packet may cause too long latency for some critical data; therefore, a low-latency packet (LLP) mechanism is provided, allowing the insertion of one or more packets with low-latency requirements within the transmission of a long packet. The interrupted long packet is resumed immediately after the LLP part of the minor frame.

One or more LLPs are allowed to be placed in the minor frame immediately following the minor frame header. An LLP end byte immediately follows each LLP. The LLP end byte identifies if more LLPs follow or if this LLP is the last LLP in the minor frame.

The LLPs shall not span multiple minor frames. The size of the LLP and its following LLP end byte together shall be less than or equal to the remaining space in the minor frame.

The minor frame structure with LLPs is shown in [Figure 7-4](#).

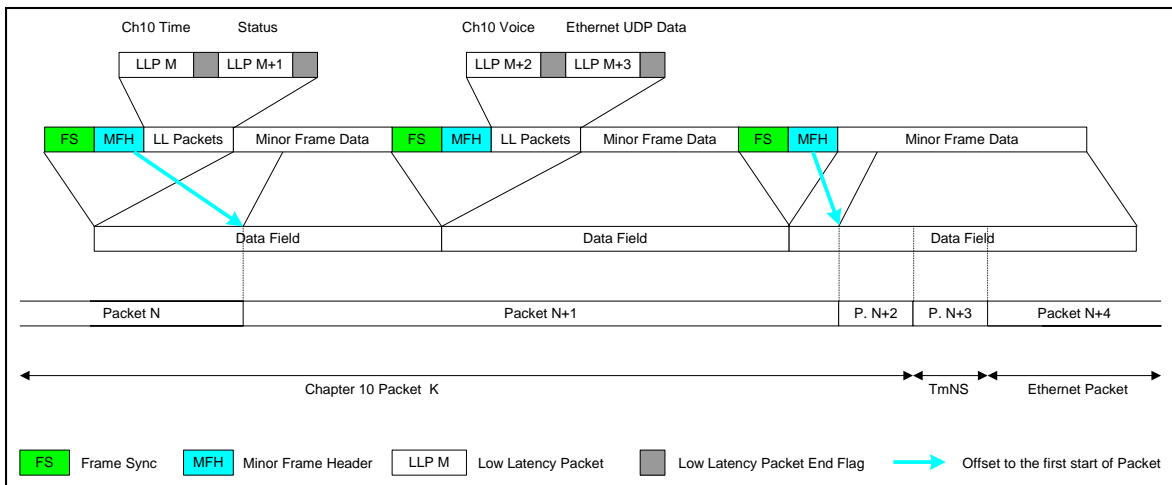


Figure 7-4. Packet Encapsulation Mechanism with Low-Latency Packets

Data belonging to the normal packets continues in the next minor frame immediately after the header and any optional LLPs. Please note: the offset in the minor frame header is not necessarily pointing immediately after the LLP.

### 7.3.3 Packet Encapsulation Structure

The final structure can be seen in [Figure 7-5](#).

FRAME SYNC
MINOR FRAME HEADER, Unprotected Part (1 byte)
MINOR FRAME HEADER, Golay Code Protected Part (3 bytes)
(OPTIONAL) LLP 1 (variable size)
(OPTIONAL) LLP END BYTE 1 (1 byte)
...
(OPTIONAL) LLP M (variable size)
(OPTIONAL) LLP END BYTE M (1 byte)
PACKET PART 1 (variable size)
...
PACKET PART N (variable size)

Figure 7-5. Minor Frame Structure

- a. Frame Sync (bits 31-0). The minor frame sync word is according to Subsection [7.2.1](#).
- b. Minor Frame Header Unprotected Part. The minor frame header has a one-byte-long unprotected part. This byte is static for each PCM stream. The minor frame header, unprotected part can be seen in [Figure 7-6](#).

7	6	5	4	3	2	1	0
Stream ID				Reserved		Version	

Figure 7-6. Minor Frame Header, Unprotected Part

- Stream ID (bits 7..4). The stream ID can identify up to 16 different streams. Its usage is application-specific.
- Reserved (bits 3..2). These bits are reserved, and shall be set to 0.
- Version (bits 1..0). These bits are coding the Chapter 7 version:

00: Version 1  
 01: reserved  
 10: reserved  
 11: reserved

- c. Minor Frame Header Golay Code Protected Part. A minor frame header immediately follows the minor frame sync. The size of the minor frame header is 12 bits, coded and transmitted as a 24-bit Golay code word; it occupies the first 3 bytes of the minor frame. The minor frame header, Golay code protected part, [Figure 7-7](#), is structured as follows.

11	10	9	8	7	6	5	4	3	2	1	0
LL		Offset to First Packet Header (bits 10.. 0)									

Figure 7-7. Minor Frame Header, Golay Code Protected Part



- **LL: LLP Exists (bit 11)**
    - = 1 indicates if the minor frame data part contains one or more optional LLPs and the closing LLP end byte.
    - = 0 means that no LLP and no LLP end byte exists.
  - **Offset to First Packet Header (bits 10..0).** These bits provide a byte offset to the first byte of the first start of packet in this minor frame – provided a start of packet exists in this minor frame. The value is relative to the first data byte following the packet header (the value of 0 represents the first byte following the header.)  
If there is no start of packet in this minor frame, all bits shall be set to 1 (binary 1111111111).
- d. **LLP 1..M (Optional).** If one or more LLPs exist, the first LLP is placed immediately after the minor frame header.
- e. **LLP End Byte 1..M (Mandatory after every optional LLP).**  
= 0xFF indicates that another LLP immediately follows this byte.  
= 0x00 indicates there are no more LLPs in this minor frame. The byte following this end byte is the first byte of the packet part 1 (except the LLP end byte is placed at the last byte of the minor frame).
- f. **Packet Part 1..N.** Packet part 1..N can be a middle or end segment of a packet started in the previous minor frames, a full packet, or a starting segment of the packet that continues in the following minor frames.
- No gaps are allowed between the packet parts. If no packets are available for transmission, fill packets shall be inserted into the minor frame.

#### 7.4 Packet Format

The packet consists of a packet header and a packet data part as shown in [Figure 7-8](#).

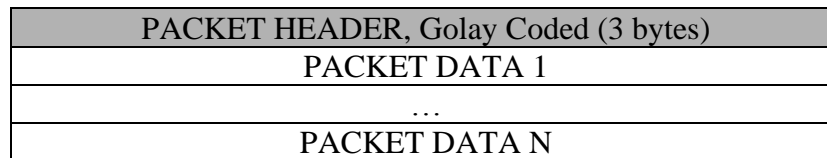


Figure 7-8. Packet Structure

- a. **Packet Header.** The size of the packet header is 24 bits and coded as 2 x 12 bit parts. It is coded and transmitted on 2 x 24-bit Golay code words. The order is first the bits 23..12, then the bits 11..0. Packet header, protected bytes can be seen in [Figure 7-9](#).

23	22	21	20	19	18	17	16	15	14	13	12
Reserved		Content				Fragment		Length (15..12)			
11	10	9	8	7	6	5	4	3	2	1	0
Length (11..0)											

Figure 7-9. Packet Header, Protected Bytes

The packet header consists of the following.

- Reserved (bits 23..22). These bits are reserved and shall be set to 00.
- Content: Packet Content (bits 21..18). These bits are identifying the content of the packet. The following values assigned:

- 0000: Fill Packet
- 0001: Application Specific Packet
- 0010: Test Counter Packet
- 0011: Chapter 10 Packet
- 0100: Raw Ethernet Media Access Control (MAC) Frame Packet
- 0101: Ethernet Internet Protocol (IP) Packet
- 0110: iNET TmNS Packet
- 0111 – 1111: reserved

- Fragment: Packet Fragmentation. (bits 17..16).

- 00: Complete Packet
- 01: First Fragment of a Packet
- 10: Middle Fragment of a Packet
- 11: Last Fragment of a Packet

If a packet fragmentation is required, the first, middle, and last fragments are transmitted in the original sequence; they are not mixed with other fragmented packets – only LLPs can be inserted in between by using the LLP encapsulation mechanism.

Fragmentation is necessary if the information to be transmitted is larger than or equal to 64 kilobytes; however, its usage is allowed on smaller packets as well. When a packet is fragmented, any content header only exists in the complete packet or first fragment of a packet.

- Length: Packet Length. (bits 15..0). Packet length contains the number of bytes of the packet data part. The length of the packet header is not included in the packet length. The packet length is limited to 0xFFFF bytes. If longer information shall be telemetered, the fragmentation method shall be used.

- b. Packet Data 1..N. The data part contains the data bytes of the given type of packets defined by the Content field. A detailed description of the data bytes can be found in Subsection [7.4.1](#) through [7.4.7](#).

#### 7.4.1 Fill Packet

If no data is available to be transferred, fill packets shall be inserted. The fill packet size is arbitrary within the allowed sizes. The fill packets shall be filled by 0xAA data bytes.

#### 7.4.2 Application-Specific Packet

The format of these packets is not defined by this standard. Application-specific packets are allowed; however, they shall not be employed to carry data that conforms to another defined format. Specifically, application-specific packets shall not be used to carry Chapter 10, iNET TmNS message data, or Ethernet data.

#### 7.4.3 Test Counter Packet

The test counter packet is defined as a free-running 12-bit counter encoded by Golay coding on 24-bit. Its usage is optional, and the transmission rate is not specified by this standard.

7.4.4 Chapter 10 Packet

The Chapter 10 packet contains the Chapter 10 header, secondary header (if one exists), channel-specific data word, data, filler, and checksum. Structure-critical header information is protected by Golay code as described in below. The final packet structure can be seen in [Figure 7-10](#).

Protected Part of the Header
Unprotected Part of the Header
Secondary Header (optional)
Channel Specific Data Word
Data
Packet Trailer (optional fill and checksum)

Figure 7-10. Chapter 10 Packet with Protected Header

When a Chapter 10 packet is transmitted by using the packet fragmentation method described in Section 7.4 item a, only the first fragment will start with the Chapter 10 header; it will not be repeated in every fragment.

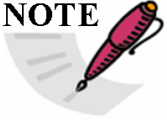
Chapter 10 packet headers are protected as follows. The first 12 bytes of the header shall be the Channel ID, Data Length, and Packet Length fields from the Chapter 10 header. These important fields will be called the protected part of the Chapter 10 header.

This 4 x 12 bit structure is Golay coded and transmitted as 4 x 24 bits (12 bytes) that replace the first 12 bytes of the standard Chapter 10 header (Packet Sync Word, Channel Id, Packet Length, and Data Length fields).

The structure of the protected part of the Chapter 10 header before the Golay coding can be seen in [Figure 7-11](#).

11	10	9	8	7	6	5	4	3	2	1	0
Reserved (0)								Channel ID (15..12)			
Channel ID (11..0)											
Packet Trailer Bytes (4..0)				Data Length (18..12)							
Data Length (11..0)											

Figure 7-11. Protected Part of the Chapter 10 Header

	<p><b>NOTE</b> The 19 bits of data length size is sufficient for all Chapter 10 packet sizes, except Computer-Generated Data Packet, Format 1 setup record. If the size of this packet exceeds the 19-bit limit, the setup record shall use the fragmentation method according to Section 7.4 item a</p>
---	--

The packet trailer bytes is the sum of the length of the secondary header, fill bytes, and Chapter 10 packet checksum. The number of fill bytes inserted in the Chapter 10 packets is restricted to maximum 3 bytes. If the original Chapter 10 packet contains more than 3 bytes, it shall be compressed before transmission.

The rest of the Chapter 10 header will be transmitted in its original form, as shown in [Figure 7-12](#)

31	24	23	16	15	8	7	0
Data Type		Packet Flags		Sequence Nr.		Data Type Ver.	
Relative Time Counter (low)							
Header Checksum				Relative Time Counter (high)			

Figure 7-12. Unprotected Part of the Chapter 10 Header

7.4.5 Raw Ethernet Media Access Control Frame Packet

The raw Ethernet MAC frame packet contains one physical layer MAC frame, starting with the MAC destination address and ending with the frame check-sum (FCS) bytes inclusive. The raw Ethernet MAC frame packet can contain any kind of message data, IPv4, IPv6, and Jumbo messages included. No extra protection is used for the raw Ethernet MAC frame packet.

7.4.6 Ethernet Internet Protocol Packet

The Ethernet IP packet contains one Ethernet message starting with the IP header (inclusive) and ending with the last byte before the FCS bytes. The FCS is excluded. No extra protection is used for the Ethernet IP packet.

7.4.7 iNET TmNS Packet

The iNET TmNS packet contains one TmNS data message structure captured by User Datagram Protocol/IP or Transmission Control Protocol/IP protocol.

The iNET TmNS packet contains only the TmNSDataMessageHeader and the TmNSDataMessagePayload. The Ethernet protocol headers are removed.

In general, the structure follows the TmNS message definition, with Golay code protected fields for the structure-critical fields in [Figure 7-13](#).

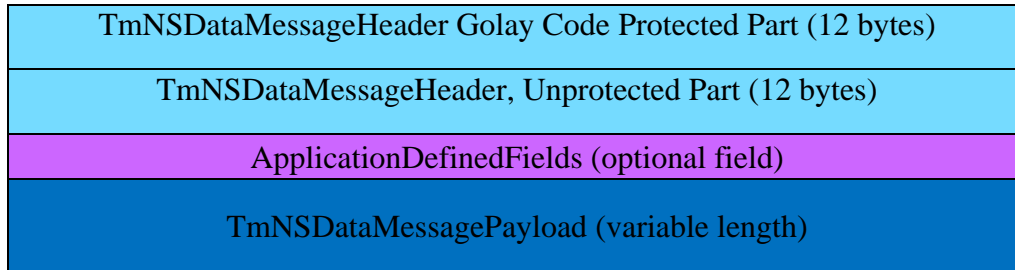


Figure 7-13. iNET TmNS Packet Structure

The TmNSDataMessageHeader Golay code protected part consists of 8 x 12 bit Golay coded words, which are coded and transmitted as 16 x 12 bits (24 bytes). [Figure 7-14](#) shows the Golay coded part, as well as the TmNSDataMessageHeader unprotected part.

11	10	9	8	7	6	5	4	3	2	1	0
Message Version				OptionWordCount				MessageFlags(3..0)			
MessageFlags (15..4)											
Reserved(0)				MessageDefinitionID(31..24)							
MessageDefinitionID(23..12)											
MessageDefinitionID(11..0)											
Reserved (0)				MessageLength (31..24)							
MessageLength (23..12)											
MessageLength (11..0)											

Figure 7-14. TmNSDataMessageHeader Golay Coded Part

The TmNSDataMessageHeader unprotected part consists of 12 bytes as shown in [Figure 7-15](#).

31	0
MessageDefinitionSequenceNumber (32 bits)	
MessageTimestamp (64 bits)	

Figure 7-15. TmNSDataMessageHeader Unprotected Part

All the fields in the TmNSDataMessageHeader Golay code protected part and in the TmNSDataMessageHeader unprotected part are identical with the ones defined in the iNET TmNS message definition.

### 7.5 Randomization, Encryption, and Error Correction.

With telemetry transmission in most of the cases, randomization (R-NRZ, CCSDS), encryption, and error correction (LDPC, Reed-Solomon) techniques are used. The usage of these techniques is outside of the scope of this chapter.

**\*\*\* END OF CHAPTER 7 \*\*\***