CGMS-XXIX USA-WP-15
Prepared by USA
Agenda Item: G.1
To be discussed in Plenary

LRIT RECEIVER SPECIFICATION

This document describes the H/W components and S/W processing for a user terminal receiving digital transmissions from the GOES-E or GOES-W satellites. This paper is intended to become a design specification for developers of USA LRIT receive stations.

LRIT RECEIVER SPECIFICATION

1.0 Introduction

This document describes the technical specifications for operating a LRIT receive station to capture the GOES digital broadcast. The USA LRIT receive station is designed to be interoperable with the JMA and EUMETSAT systems.

1.2 LRIT Service

The mission shall be named Low-Rate Information Transmission (LRIT) because the communications link provides for a relatively low data rate below 256kbps.

1.3 Design Application

The design of the LRIT user station will be consistent with the design of the receivers for using the recommended CCSDS standard for Packet Telemetry. It will be limited to reception LRIT transmissions.

1.4 System Overview

The user station will consist of four main components as illustrated in Figure 1.

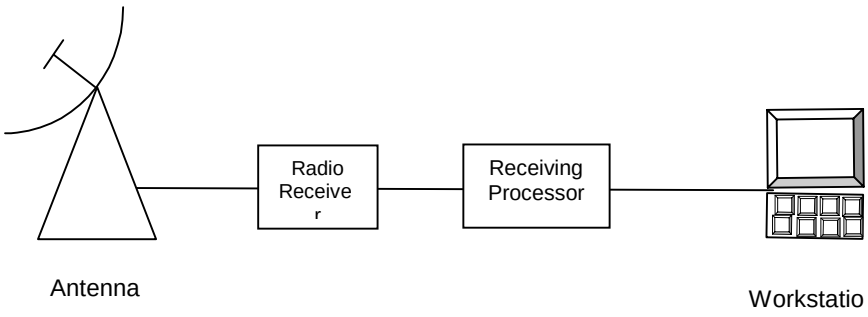

Figure 1. LRIT User Station System Components

The antenna is a parabolic dish antenna with no auto tracking. The downlink signal is received at 1691 megahertz (MHz). The signal may be filtered to reduce adjacent channel interference and/or amplified by a low-noise amplifier. Then it is down-converted to the receiver IF frequency. The IF amplifiers have an IF bandwidth capable of receiving a 293 kbps symbol stream. The IF signal is then demodulated in a BPSK demodulator and the baseband output to the receiving processor is a serial bit stream. Table 1 shows significant parameters of the RF system.

The IOC will operate at 64 kbps and use the 1-meter antenna that was used for WEFAX reception. The FOC will operate at 128 kbps and use a 1.8-meter antenna.

Table 1. LRIT Downlink Characteristics

Parameter	Value
Satellite EIRP	48.2 dBm
Center Frequency	1691.0 MHz
Useful bandwidth (@ -1 dB)	Sufficient for 293,000 symbols/sec BPSK
Packetized data rate	64 kbps during IOC 128 kbps during FOC
Total transmitted symbol rate	293 ksymbols/s at FOC
Modulation	PCM/NRZ-M/BPSK
Receiver Gain/Temperature (G/T)	-0.3 dB during IOC +3.2 dB during FOC
ber	$1x\ 10^{-8}$

The receiving processor decodes the bit stream, disassembles the LRIT packets, removes filler packets, removes the header information, reassembles and decompresses the original files, and sends the files to the workstation. The workstation contains the S/W to produce the images, lists, and text messages.

2.0 Introduction to the GOES-Specific OSI Reference Model

Table 2 given below presents the OSI layers from top to bottom and the equivalent functionality included in the LRIT communication model from the view of the transmission service.

Table 2. LRIT OSI Layer Functionality

OSI Layer	Layer Functionality
Physical Layer	- Convolutional coding - Demodulation
Data Link Layer	- Disassembly of source packets - Demultiplexing - Acquisition of VCDUs - Reed-Solomon decoding - Derandomizing
Network Layer	- (none)
Transport Layer	- Final assembly
Session Layer	- Decryption - Decompression
Presentation Layer	- Retrieval of User Data from Files
Application Layer	- Processing of application data

3.0 Physical Layer

The physical layer on the LRIT service performs demodulation of the incoming signal into a serialized data stream. The serialized data stream is decoded with a Viterbi soft-decision (a.k.a. maximum likelihood) decoding algorithm.

The convolutional coding has the following characteristics:

- Nomenclature: Convolutional code with maximum-likelihood (Viterbi) decoding
- Code rate: 1/2 bit per symbol
- Constraint length: 7 bits
- Connection vectors: G1 = 1111001; G2 = 1011011
- Phase relationship: G1 is associated with the first symbol
- Symbol inversion: On the output path of G2

4.0 Data Link Layer

This section gives a general overview and discusses input to data link layer, VCA sublayer processing, as well as VCA sublayer processing.

4.1 Input to Data Link Layer

The physical layer provides a decoded serial data stream to the data link that contains LRIT packets.

4.2 General

This layer consists of two sublayers for VCLC processing and VCA processing. It receives a bit stream from the physical layer that must be decomposed into the individual packets. Fill packets are identified and discarded. Data packets are further processed and sent to the session layer.

4.3 VCA Sublayer Processing

The VCDU structure is shown in Figure 2.

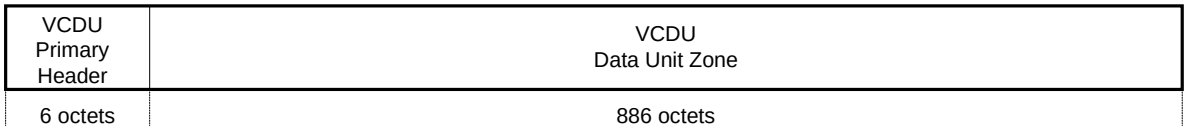

Figure 2. VCDU Structure

The decomposition of the VCDU header is given in Figure 3.

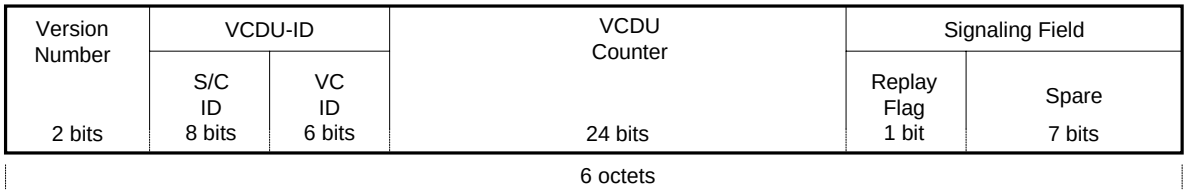

Figure 3. VCDU Primary Header

Mission-specific use:

Version Number	'01'b
VCDU ID	Spacecraft (S/C) ID representing on the disseminating spacecraft VC ID '63'd ('all ones')
VCDU Counter	Sequential count (modulo 16777216) of VCDUs on each virtual channel.
Signaling Field	'all zeros'

The VCA sublayer receives a data stream from the physical layer in a sequence of coded VCDU packets (C_VCDU) (Figure 2). The incoming serial datastream is synchronized into discrete CADUs. After this frame synchronization process, one randomized coded virtual channel unit CVCU is extracted from each CADU by means of stripping the synchronization markets off. Multiplying all 8160 bits of the randomized C_VCDU with a statically defined psuedonoise pattern performs derandomization. The packet structure now looks like Figure 4. After derandomization, each clear C_VCDU undergoes a forward error correction based on the Reed-Solomon check symbols included in the packet.

After FEC, fill VCDUs, with a VC = 63 are discarded. The VCA-SDU is extracted from the data unit zone of the VCDU; the VCU_ID is defined in the primary header (Figure 4).

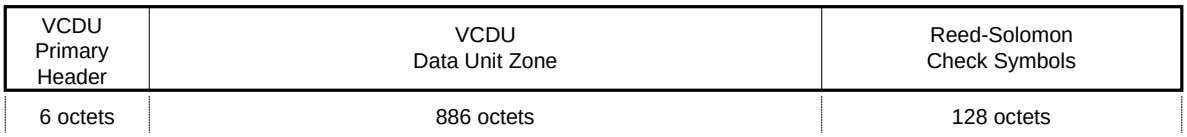

Figure 4. C_VCDU Structure

4.3.1 Reed-Solomon Coding

The LRIT dissemination service is a Grade-2 service; therefore, the transmission of user data will be error controlled using Reed-Solomon coding as an outer code.

The used Reed-Solomon code is (255,223) with an interleaving of I = 4.

The Reed-Solomon check symbols are extracted from the last 128 octets of the C_VCDU packets forming VCDU packets.

4.3.2 Derandomization

Randomization was applied to all LRIT CVDUs. It is a process in which a pseudo-random sequence is bitwise exclusive-ORed to all 8160 bits of the CVDU to ensure sufficient data transitions.

The de-randomization process will generate the same pseudo-random sequence, synchronize with the incoming bit stream, and exclusive-OR it to extract the original data stream.

The pseudo-random sequence was be generated using the following polynomial:

$$h(x) = x^8 + x^7 + x^3 + 1$$

This randomizing sequence began at the first bit of the CVCDU and repeated after 255 bits, continuing repeatedly until the end of the CVCDU. The sequence generator was then reinitialized to all-ones for the processing of the next CVCDU.

4.4 VCLC Sublayer Processing

Upon acquisition of an M_PDU, the layer demultiplexes zero or more source packets from the acquired M_PDU and eventually the available data belonging to the same virtual channel. Whenever a source packet is complete, it is checked for being a fill packet: if the APID equals 2047, the packet is assumed to be a fill packet. Fill packets are discarded, whereas other source packets are forwarded to the network layer.

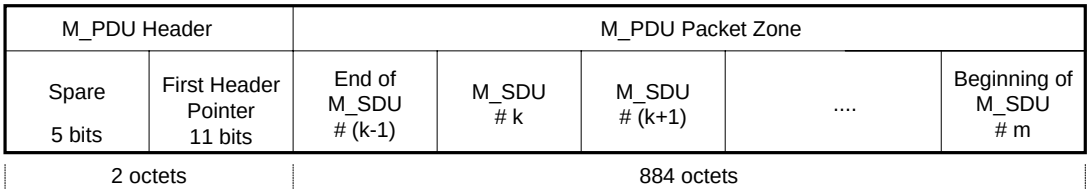

Figure 5. M_PDU Structure

5.0 Network Layer

Upon acquisition of a source packet, the network layer forwards the source packet to the transport layer. There is no other processing in this layer.

6.0 Transport Layer

This layer reassembles the LRIT files that were subdivided before transmission into transmission packets.

Upon acquisition of source packets, the packets are sorted by their APIDs. The contents of the data fields, except the last two octets are concatenated under control of the sequence flags in the packet headers, resulting in a transport file. As soon as a transport file is complete, the TP_SDU is extracted and routed to the session layer.

The CRC field included in each segment is checked to verify the integrity of the received data. Failures of this test are reported to the user application processor with a warning statement. Errors in the packet headers, which are recognized by the presence of unexpected information, may be corrected by means of redundancy (e.g., implied by sequence) and semantics.

6.1 Source Transport Service Data Unit

The TP_SDU packet structure is defined in detail in Section 6.2.1 (Source Packet Structure) of the transmit specification.

Source Packet Header (48 bits)							Packet Data Field (variable)	
Packet Identification				Packet Sequence Control		Packet Length	User Data Field	
Version No.	Type	Secondary Header Flag	APID	Sequence Flags	Packet Sequence Count		Application Data Field	Packet Error Control (CRC)
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits	16 bits	Variable	16 bits
2 octets				2 octets		2 octets	Max. 8190 octets	2 octets

Figure 6. Source Packet Structure (TP_PDU)

6.2 Data Field Integrity Check

The CRC was computed over the entire application data field, using the following generator polynomial:

$$g(x) = x^{16} + x^{12} + x^5 + 1$$

Section C.7 Session Layer

The protocol data unit (S_PDU) may be encrypted or compressed. The P_SDU is scanned for a key header record first. If such a record is found then the data field of the file is decrypted.

After that, if the file is an image file, it is scanned for an image structure record. If the compression flag is non zero, than the data field is decompressed and the primary header corrected accordingly (data field length).

The resulting file is the service data unit (S_SDU) forwarded to the presentation layer.

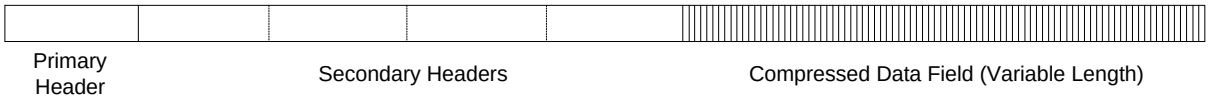

Figure 7. LRIT File Structure with Compressed Data Field

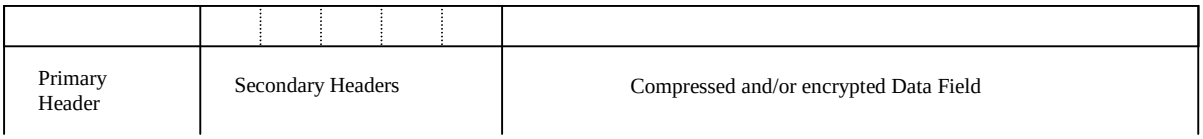

Figure 8. LRIT Session Protocol Data Unit (S_PDU)

7.1 Decompression

Lossless compression will be used in the GOES LRIT implementation. The exact algorithm is to be determined.

7.2 Decryption

The GOES LRIT service does not presently require a mechanism to control the access to LRIT. If capabilities for data control are anticipated, the means for encryption and the necessary fields on the file structures must be provided. Therefore the system design must provide a means to decode packets and to extract the key header record. Encrypted files intended for a specific recipient will have the recipient designated in header #129. The processor will check for header #129 and then determine if the message is intended for it before attempting to decrypt the message.

8.0 Presentation Layer

The presentation layer converts the decompressed LRIT files (packets) into user data files. Several LRIT packets may contain the data for a single application file. Packets and their sequence will be determined from the header file and the files reassembled in the correct order.

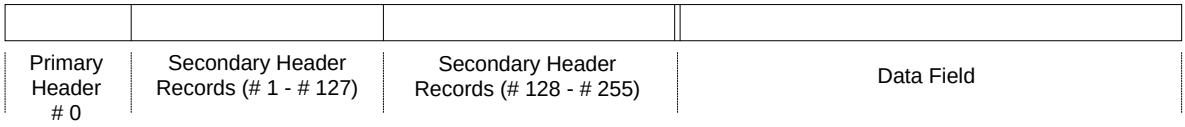

Figure 9. LRIT File Structure

The application files will be saved on a mass storage medium available to the LRIT processor and the application processor. The file name will be extracted from the secondary header record Type 4, the annotation record.

Image files that have been partitioned in the transmit application layer will have the partition number and number of partitions recorded in header type #128. The partitions are based upon a number of scan lines of the image. The partition files should be kept separate and identified as being part of an image.

The information contained in the headers should be extracted and placed in a metadata file with a name indicating its correspondence to the data file. The user-specific S/W may use this file to process and display the data. Header types are shown in Table 3.

Table 3. Header Types in the LRIT Files

Code	Header Record Type	Structure
Headers as Defined in LRIT Global Specification		
0	Primary header	
1	Image structure	
2	Image navigation	
3	Image data function	
4	Annotation	
5	Time stamp	
6	Ancillary text	
7	Key header	Optional
8…127	Reserved	
Mission-Specific Headers		
128	Segment identification	
129	Encryption key message header	
130…255	Reserved	

The headers present in the LRIT file will depend upon the file type. Table 24 below shows the headers that could come with each file type.

Table 4. Use of Header Records versus File Type

File Types	Header Record Types									
	0	1	2	3	4	5	6	7	128	129
Image Data	•	•	♦	♦	•	♦	♦	♦	♦	♦
Service Messages	•						♦			♦
Alphanumeric Text	•				•	♦	♦	♦		♦
Encryption key Message	•					•	♦			♦
Meteorological Data	•				•	♦	♦	♦		♦

• = Mandatory
♦ = Optional

The LRIT file structure, defined in Section C.4, will make the identification of the headers and their presence easily identifiable. If headers 7 and 129 are not present, the file is not encrypted. Header 5 contains a time stamp indicating the time that the file was transmitted, and is not specifically needed to interpret the data. Header 4 will contain the file name used by the GOES LRIT transmitting domain. It can be extracted and used as the name of the file and corresponding metadata file. Header type 0, which is present on all LRIT files will contain the length of the header and data fields for parsing of the LRIT files into application files and metadata files.

Service messages and encryption messages should be used as they arrive. The service message should be displayed for the attention of the operator and the encryption key used to set up the system for decryption of an incoming message.

The metadata files should be structured in such a way that the application programs can identify and use the data. The file will be a delimited character string as shown in Figure 10.

;0,16,25,32758,32741;1,9,64,1200,1400,1;…;4,16,DATAFILENAME.txt,

Figure 10. Partial Sample Delimited File Structure

9.0 Application Layer

The application layer receives the LRIT files from the session layer. The files are transmitted from the receiving processor to the workstation.

In the workstation, the files are stored and identified for their applicability to the specific requirements of the user facility.

LRIT files that are administrative messages should be either printed or displayed to the user automatically. Images are sent to the application program for interpretion and display. List files are printed or displayed under autonomous control based upon a preselected setup option of the workstation operator.

9.1 File Handling

The applications processor must contain an autonomous file handling system because:

- Data in the files requires immediate attention of the user
- Mass storage is limited
- The image files are large
- The data is time sensitive and becomes obsolete after a short period of time
- Data files will be received as a continuous input

Each file will be identified to determine what further processing is necessary. The file handling processor will keep a record of the activity of each file. The file transmission time will be extracted from header 5 and the reception time from the system clock. The file handler will pass the file to the appropriate applications processor and know when the processing is complete. The file will be purged from the mass storage according to a user-defined storage time or passed to another processor if required by the user's specific installation. The timing of the process should be fast enough so that a new image with the same file name will not coexist in the applications processor with the older image. The files

should be processed fast enough so that a file is not overwritten by the new image before it has completed processing.

9.2 Data Processing

The types of files that will be received are:

- Image data files
- Service messages
- Alphanumeric files
- Encryption Key files

9.2.1 Processing Image Files

Image files will contain images in a series of 4-bit or 8-bit pixels. Each pixel represents a 128-level or 256-level gray scale (no color in the WEFAX images). GOES and POES images will have 1395 pixels per line and a number of lines that varies with the image. Images from other sources may have different dimensions and gray scale. Header type 1 will identify the dimensions of the image and the number of bits per pixel. The applications processor should have the capability to produce the image on the display screen and to print it. The information from header type 1 will be in the metadata file.

Large images such as the full disk file will be stored in smaller files containing segments with only a certain number of lines of an image. In order to decrease the latency of the data image, the processor should start producing the image without waiting for reception of all the files. The data from each file should be used to build the image as soon as it is received.

If an image file is an overlay for another image, the processor should be able to overlay the image on the original figure. Overlays can be grids, isobars, or text.

9.2.2 Processing Alphanumeric Files

The files should contain data in American Standard Code for Information Interchange (ASCII) alphanumeric format. Often these files are tables or lists. The application processor should display or print these files in their original format.

9.2.3 Processing Service Messages

It will be assumed that these messages have high priority and the user terminal operator should be alerted to them as soon as possible. When these messages are received, some method of alerting the operator should be used such as an audible alarm or flashing screen. The message should be displayed on the monitor after the image processing is complete. As a user option the message can be printed.

9.2.4 Encryption Key

This file should be used directly to update the decryption key. It does not need to be displayed. The operator should be alerted that an encryption key has been received and processed.