SOC-SDS-1

SYSTEM DESIGN SPECIFICATIONS

SCIENCE OPERATIONS CENTER
TETHERED SATELLITE SYSTEM

Reviewed by: ______________________ Date: __________
D. Anderson

Reviewed by: ______________________ Date: __________
J.C. Biard

Reviewed by: ______________________ Date: __________
C. Gonzalez

Reviewed by: ______________________ Date: __________
C. Gurgiolo

Reviewed by: ______________________ Date: __________
R.L. Murphy

Reviewed by: ______________________ Date: __________
S.T. Nguyen

SOUTHWEST RESEARCH INSTITUTE
Instrumentation and Space Research Division
P.O. Drawer 28510, 6220 Culebra Road, San Antonio, Texas 78228-0510
(512) 684-5111 • FAX (512) 647-4325

1. Introduction

The Science Operations Center (SOC) is a group of computer systems and specialized software to:

- provide the TSS Principal Investigators with the real-time displays needed to assess the degree to which the mission functional objectives are achieved; and
- record an archive of the science data along with any associated orbital and engineering parameters necessary to aid in interpreting the data.

In order to achieve these high-level goals, it is necessary to:

- decommutate the STS PDI provided by the Johnson Space Center;
- divide this data into independent instrument streams;
- process it into scientific measurements; and
- pass it to a graphics subsystem.

It will also be necessary for the SOC to provide four displays of operational parameters normally provided by the POCC but unavailable for TSS-1. The SOC Configuration is illustrated in Appendix B of this document.

Due to the multi-mission capabilities of the TSS, two additional goals have been set to provide easy and flexible reconfiguration of the SOC:

- The basic design of the SOC should utilize commercial, "off-the-shelf" building blocks whenever possible and require no specialized proprietary technology.
- The system should be fairly portable and simple to install and operate.

2. Scope

2.1. Topics Described

- All the functional interfaces necessary within the SOC;
- All the software needed for both real-time and non-real-time processing and the hardware configuration of each SOC subsystem; and
- The function and structure of each module, data message, data file, and display to be developed.

2.2. Topics Not Described

- Configuration of the non-SOC software, including the operating system. This is described in the SOC Operations Plan, SOC-OP-1;
- The internal functions of the decommutation system, which are documented by the vendor; and
- Software or capabilities the SOC may possess when configured as independent computers or subsystems (these independent configurations may exist before and after the mission).

3. *Applicable Documents*

The following documents form a part of this specification to the extent specified herein.

3.1. *SwRI*

SOC-DR-1	Design Requirements
SOC-JDD-1	Joint Display Definition
SOC-UM-1	User Manual
IDFS-CAS-1	Calibrated Ancillary System (CAS) IDFS Definition
IDFS-DCORE-1	Deployer Core Equipment (DCORE) IDFS Definition
IDFS-DRB-1	Deployable/Retractable Boom (DRB) IDFS Definition
IDFS-DPLY-1	Deployer (DPLY) IDFS Definition
IDFS-EMP-1	Enhanced MDM Pallet (EMP) IDFS Definition
IDFS-RETE-1	Research on Electrodynamic Tether Effects (RETE) IDFS Definition
IDFS-ROPE-1	Research on Orbital Plasma Electrodynamics (ROPE) IDFS Definition
IDFS-SAHK-1	Satellite Housekeeping (SAHK) IDFS Definition
IDFS-SCORE-1	Satellite Core Equipment (SCORE) IDFS Definition
IDFS-SETS-1	Shuttle Electrodynamics Tether System (SETS) IDFS Definition
IDFS-SPREE-1	Shuttle Potential and Return Electron Experiment (SPREE) IDFS Definition
IDFS-TEMAG-1	Tether Magnetometer (TEMAG) IDFS Definition

3.2. *MMAG*

TSS-SRD-01	Software Requirements Document

3.3. *MDSSC*

MDC W5613	EMP/TSS-1 Integrated Instrumentation Program and Components List (IP & CL)

3.4. *NASA/JSC*

NSTS 18411 Annex 5	Tethered Satellite System Payload Operations Control Center Annex

4. *Definitions of Terminology*

The following definitions are provided to simplify and standardize the description of the SOC. Refer to Appendix B, Figure B-1 for a pictorial view of the referenced system and data levels.

- The software and hardware are divided into three levels of subsystems: Level I is the telemetry decommutation subsystem; Level II is the data handling subsystem; and Level III is the science and POCC display subsystem.
- The data levels are represented in a four-tiered fashion, but these levels are labeled A, B, C, and D to avoid confusion with the subsystem descriptions.
- Level A contains the raw telemetry and CAS data. The CAS data at this level are contained in NASCOM blocks. This level is an input-only interface to the SOC.
- Level B represents the minor frame-synchronized and blocked data messages sent to the archive subsystem. It is a unidirectional data stream produced and ingested within the SOC.
- Level C holds the processed data messages that are sent to the display subsystems. This is a bidirectional data stream accommodating archive playback requests coming from the display subsystem.
- Level D refers to data on LANs attached to the secondary Ethernet ports of workstations at Level III. (Within this document, only data originating from a Level III workstation is documented for Level D.)

It is necessary to distinguish between the engineering and scientific views of an instrument:

- Instruments usually are named (given an acronym) and assigned telemetry words according to an engineering viewpoint. The instrument has a monolithic appearance because it is built around a power subsystem, a control subsystem, and so forth. The engineering parameters in the telemetry usually refer to the status and health of these subsystems.
- On the other hand, the scientific viewpoint is concerned primarily with *sensors*. A *sensor* is defined as a logical data stream, such as a spectrum for a given pitch angle. One or more *detectors* make up a *sensor*. For example, two detectors may be necessary to cover a certain energy or frequency range, but their data is merged into a single stream. A collection of sensors measuring the same or related parameters is called a *logical instrument*.

The remaining terms and acronyms are found in the List of Acronyms in Appendix A.

5. *Data Level A*

5.1. *Introduction*

Tables 1 and 2 summarize the telemetry rates for TSS-1. They also provide the average data rates the decommutation subsystem must handle:

Table 1. TSS SFMDM Telemetry Rates by Experiment 8 mifs = 0.9984 sec; 16 mifs = MAF			
Experiment	*Bytes/8mifs*	*Bytes/Sec*	*Bits/Sec*
HK	640.00	641.03	5128.21
Deployer	512.00	512.82	4102.56
EMP			
DCORE	384.00	384.62	3076.92
SETS	1344.00	1346.15	10769.23
SPREE	1216.00	1217.95	9743.59
Fill	64.00	64.10	512.82
Total	4160.00	4166.67	33333.33

Table 2. TSS Satellite Telemetry Rates by Experiment 16 mifs = 1.024 sec; 256 mifs = MAF			
Experiment	*Bytes/16Mifs*	*Bytes/Sec*	*Bits/Sec*
SAHK	496.00	484.38	3875.00
DRB	19.00	18.55	148.44
RETE	440.00	429.69	3437.50
ROPE	464.00	453.13	3625.00
SCORE	197.00	192.38	1539.06
TEMAG	160.00	156.25	1250.00
Trans. wds	240.00	234.38	1875.00
Undef.	32.00	31.25	250.00
Total	2048.00	2000.00	16000.00

5.2. *Telemetry Formats*

Presently, five TLM formats are defined for TSS-1. They are defined, for the purposes of this document, in Table 3:

Table 3. TLM Formats Defined for TSS-1			
Acronym	*Description*	*Data Rate (kbps)*	*Format ID (hex)*
QF	Quiescent Mode	4.8	0A
DF	SFMDM	33.3	0F
SF	Satellite	16.0	TBD
DCF	SFMDM contingency	16.0	TBD
SCF	Satellite contingency	9.6	TBD

Quiescent mode is not processed by the SOC, although its format ID is recognized as legal and will not be counted as a format error in the decom subsystem. During the mission, the telemetry is processed by the Payload Data Interleaver (PDI), which appends the data words shown in Figure 1 to each satellite and SFMDM minor frame:

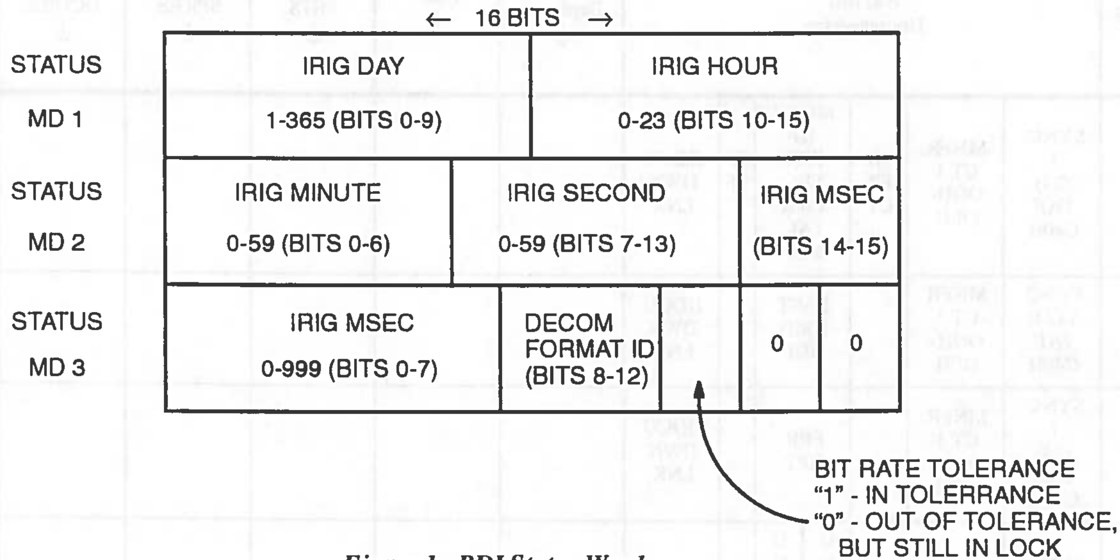


Figure 1. PDI Status Words

The PDI status words are not removed by the decommutation subsystem. The TLM format ID and PDI Decom Format ID are the same. This is important because the Decom Format ID can always be found at a fixed location prior to the next minor frame's synchronization word. The time in the PDI status words is in MTU format, which is also used for the GMT and MET times returned within the CAS.

A subset of the full science and engineering measurements is returned when the deployer and satellite operate at less than their nominal data rates. MMAG Document TSS-SRD-1 and MDSSC Document MDC W5613 describe the telemetry stream in detail. For the purposes of this document, Tables 4 and 5 provide a high-level view of the telemetry allocation by source at nominal data rates:

Table 4. SFMDM Telemetry Allocation

33,280 Kbps/1/2 Major Frame Cycle at a 33.333 Kbps Rate
4160 Bits or 520 - Byte Words, where 1 Byte = 8 Bits

MIF	640 Bits Housekeeping							512 Bits Deployer ↓	EMP ↓	1344 Bits SETS ↓	1216 Bits SPREE ↓	384 Bits DCORE ↓	64 Bits Fill* ↓
0	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH	MJ FR CT	MODE ID ERR FEG PWR FAL STS		G M T	DDCU DWN LNK						0
1	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH		LAST CMD BUF			DDCU DWN LNK						0
2	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH		ERR RPT			DDCU DWN LNK						0
3	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH		DOC CHN	DOC CHN		DDCU DWN LNK						0
4	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH		DOC CHN	DOC CHN		DDCU DWN LNK						0
5	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH					DDCU DWN LNK						0
6	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH					DDCU DWN LNK						0
7	SYNC 1 (24) FAF C40H	MNFR CT 1 OOH-OFH					DDCU DWN LNK						0

* The PDI strips these bits, so the effective minor frame size at the CIP is 512 bits. When the PDI is not processing the TLM data, such as during the Level IV Testing at KSC, all 520 bytes are passed to the decom subsystem.

Table 5. TSS Satellite Telemetry Allocation					
Experiment Name	*Minor Frame Byte*	*Number in MAF*	*Initial Minor Frame*	*Length in Bytes*	*Subcom Depth*
Sync	0	256	0	2	1
Count	2	256	0	1	1
Trans	8	256	0	1	1
	16	256	0	1	1
	24	256	0	1	1
	32	256	0	1	1
	40	256	0	1	1
	48	256	0	1	1
	56	256	0	1	1
	64	256	0	1	1
	72	256	0	1	1
	80	256	0	1	1
	88	256	0	1	1
	96	256	0	1	1
	104	256	0	1	1
	112	256	0	1	1
	120	256	0	1	1
Housekeeping	49	256	0	7	1
	57	256	0	7	1
	113	256	0	7	1
	121	256	0	7	1
DRB	4	64	0	2	4
	46	64	0	2	4
	9	16	3	1	16
	9	16	7	1	16
	9	16	11	1	16
SCORE	6	256	0	2	1
	12	256	0	2	1
	20	256	0	2	1
	82	256	0	2	1
	3	256	0	1	1
	17	256	0	1	1
	25	256	0	1	1
	33	256	0	1	1
	9	64	0	1	4
	9	16	15	1	16

Table 5. TSS Satellite Telemetry Allocation (cont'd)					
Experiment Name	*Minor Frame Byte*	*Number in MAF*	*Initial Minor Frame*	*Length in Bytes*	*Subcom Depth*
RETE	10	256	0	2	1
	14	256	0	2	1
	18	256	0	2	1
	22	256	0	2	1
	26	256	0	2	1
	30	256	0	2	1
	34	256	0	2	1
	38	256	0	2	1
	42	256	0	2	1
	66	256	0	2	1
	70	256	0	2	1
	74	256	0	2	1
	78	256	0	2	1
	46	64	1	2	4
	46	64	2	2	4
	46	64	3	2	4
TEMAG	28	256	0	2	1
	36	256	0	2	1
	44	256	0	2	1
	68	256	0	2	1
	76	256	0	2	1
ROPE	41	256	0	1	1
	65	256	0	1	1
	73	256	0	1	1
	81	256	0	1	1
	84	256	0	4	1
	89	256	0	5	1
	94	256	0	2	1
	97	256	0	7	1
	105	256	0	7	1

The SOC design allows for two independent bit serial streams (SFMDM and Satellite) to be handled by the decommutation subsystem. Presently, the telemetry formats for the quiescent and contingency modes are undefined within the SOC. This design reflects the recognition of these modes by the DCS but not the processing of this data.

5.3. NASCOM Block Format

This format is documented in reference 3.1.a, page B-7. It is relevant only to the decommutation subsystem, which must extract the TSS-1 CAS data.

5.3.1. CAS Data Rate and Format

The TSS-1 CAS parameters are documented in NSTS 18411 Annex 5. The data formats described are those that exist before the CAS data is converted for SOC usage. All CAS floating point values are converted to IEEE floating point representation. CAS bit fields are placed right-justified within integers except for single bit monitors. Monitors, which can appear within both 8- and 16-bit values, are passed unchanged. These conversions are done within the DHS.

6. System Level I (Decommutation Subsystem)

The decommutation subsystem (DCS) is based on redundant microcomputers which use 80386 CPUs and specialized software provided by BGJA. This software is divided into discrete submodules called *framers* and *applications*. A special serial interface board that has four input channels is used to acquire the data from the CIP. See Appendix B, Figure B-1, for a schematic view of the critical interfaces to the decommutation system.

6.1. Framers

A framer module is responsible for handling a specific data stream from the CIP. There are two framers for the TSS-SOC, one for the NASA TLM Type-3 stream and the other for the CAS NASCOM block stream. They differ in their frame validation methods, as described later. The primary function of a framer is to synchronize the incoming data and, thus, recognize minor frames to be checked and passed to the application process.

In general, a framer will acquire and maintain synchronization using these rules:

a) The first sync after a LOD is obtained by the hardware, thus ensuring bit synchronization.

b) One minor frame is read, including the next sync pattern.

c) The leading pattern is checked; then the tailing pattern. A search for the sync pattern does not take place within the frame unless either the leading or tailing pattern did not check.

d) If the leading sync is missing, bytes are discarded up to the next sync pattern.

e) If the tailing sync is missing, or in the wrong location, the frame is passed on but marked "bad."

f) If the tailing sync is found within the frame, the framer recovers any bytes after the sync.

g) If a frame is "good" under the above rules, then the decomm format ID is checked for the two TSS TLM formats that are to be processed (see Table 3). A frame that does not match is discarded but counted for statistical purposes.

Framers only pass complete minor frames on for processing, so incomplete frames occuring at AOS/LOS boundaries are filtered out. The CAS framer checks the NASCOM block CRC field to validate the frame rather than checking the next sync pattern.

The framer modules log this information by input source for display during SOC operation:

- format ID, time this format was last received
- frames received (total)
- leading sync lost counter
- tailing sync lost counter (CRC bad for CAS)
- bad frame counter and time of last bad frame
- LOD indicator and counter, time of last LOD
- LAN messages sent counter
- number of rejected NASCOM blocks

This information is logged to file c:\syslog\TBD once per second.

6.2. *Application Layer Processes*

The application layer of the DCS provides these functions:

- display of system status parameters
- transmission of the minor frames on the LAN
- logging of how many frames are broadcast
- time extraction from the SFMDM and satellite data streams (time is inserted into LAN message control field). A time adjustment can be applied to the satellite time tag. MTU format is converted to binary integer milliseconds and seconds since the epoch.
- places the TLM minor frame counter into the *message counter* field.
- places the TLM format ID into the *data key* field.
- maintains a counter of CAS blocks which is inserted into the *message counter* field.
- CAS processing
  - Floating point types SPL and DPL will be converted to IEEE double precision floating point format
  - Parameters of Type DB are single bit quantities that can appear anywhere within 16- and 8-bit fields; these bits will be returned one per byte in the LSB of the byte.
  - The type BSS bit fields will be extracted and returned as signed 16-bit values (shorts), sign extended, and the most significant byte first.
  - Type HMD values are passed unchanged.

- Parent words containing thruster firing monitors are passed unmodified.
- extraction of the TSS-1 CAS parameters from the input data. CAS parameters requested by other payloads are discarded by this action.
- provides a small (1-10 ms) delay before sending messages to the Level B LAN.

6.3. General Functions

The redundant decommutation subsystems operate in parallel to provide the highest reliability. No action is needed if a hardware failure occurs; data messages will still be sent from the unaffected system. Figure 2 illustrates information that will be displayed on the DCS monitor:

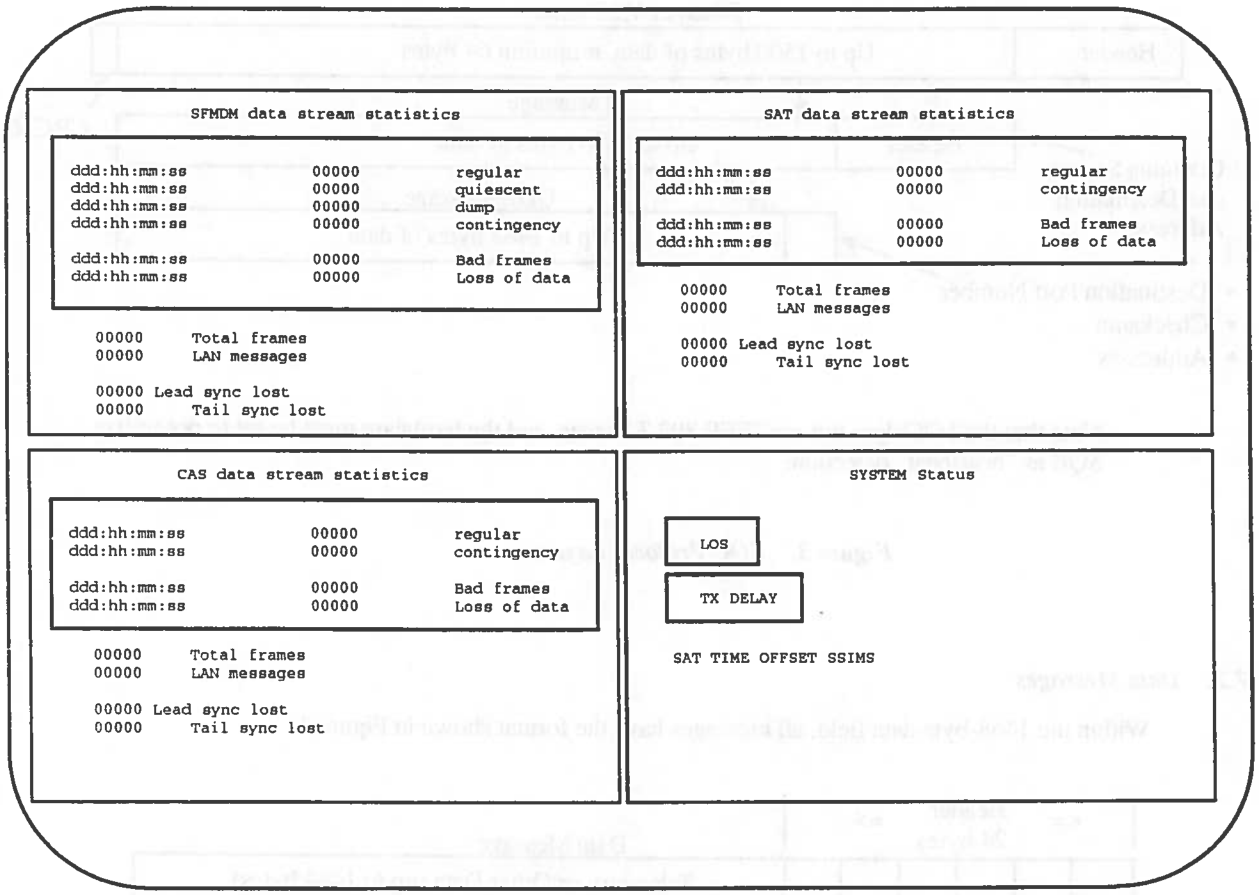


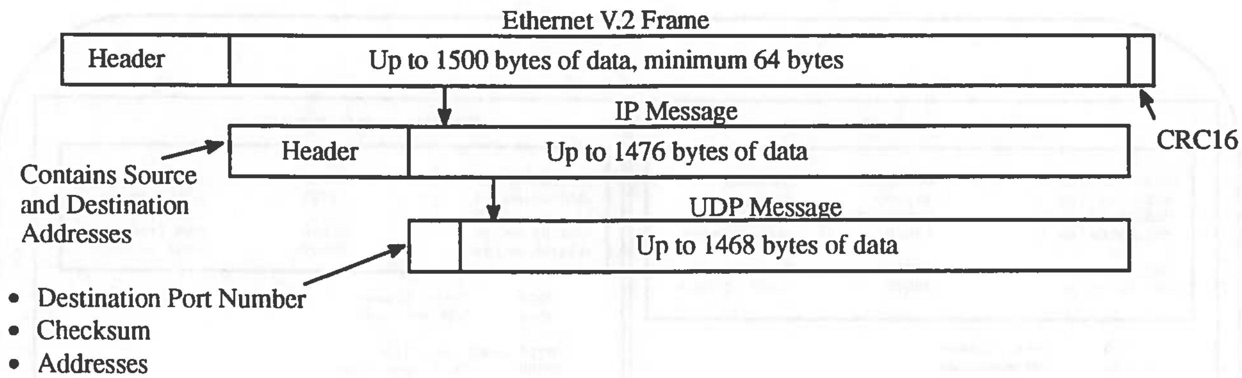
Figure 2. DCS Status Display

7. Data Level B

7.1. Introduction

The physical medium for this data level is Ethernet V.2 10 Base 2 standard. The data messages transmitted on the Level B LAN are contained in User Datagram Protocol packets (UDP). The rationale for using UDP rather than a lower level protocol or a virtual connection is dictated by the real-time nature of the SOC. UDP messages are simple, involve little overhead, and carry information that simplifies the job of the recipient system, something lower level protocols lack. Also, the software task must run with superuser (called "root" in UNIX) privileges to access either an IP or an Ethernet level message. Clearly, this might enable a misbehaving process to affect the integrity of the entire system. The use of broadcast packets allows for each block of data to be placed on the LAN once, regardless of the number of recipients. Virtual connections are dependent on the exact SOC configuration, which may change due to workstation failure or other circumstances.

The SOC network protocol hierarchy is illustrated in Figure 3:



Note that the SOC does not use IEEE 802.3 frames, and the hardware must be set to not utilize SQE as "heartbeat" detection.

Figure 3. SOC Protocol Layers

7.2. Data Messages

Within the 1468-byte data field, all messages have the format shown in Figure 4:

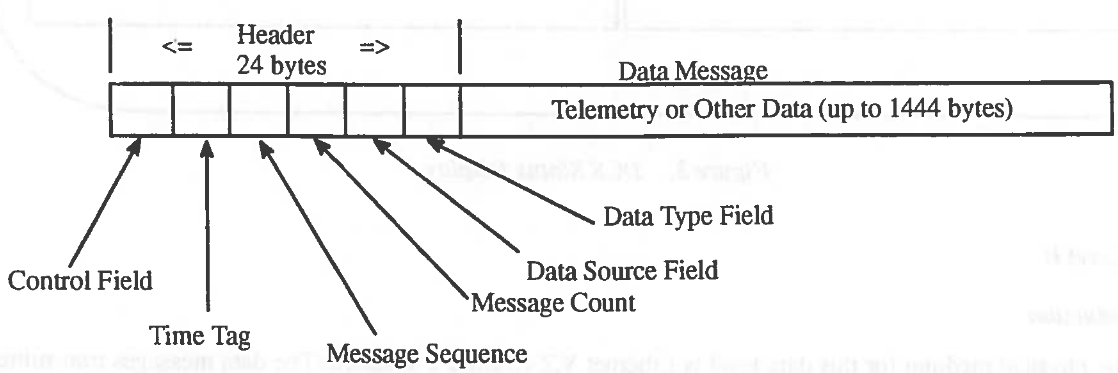


Figure 4. SOC Data Message Description

The format and contents of the control fields are defined as follows:

Control Field	:	byte 0	:	Bit 7 Reserved
				Bit 5 Playback Response
				Bit 4 Redelivery Request
				Bit 3 Playback Request
				Bit 2 EOF Flag
				Bit 1 LOD Flag
				Bit 0 LOS Flag
Msec	:	byte 0-1	:	Time tag
Sec	:	byte 0-3	:	Time tag
Message Sequence	:	byte 0-3	:	Number of related messages following
Message Counter	:	byte 0-3	:	Sequential count of messages of a given type, modulo 65536
Data Source	:	byte 0	:	TLM source number
		byte 1	:	Logical instrument number
Data Type	:	byte 0		Message type code

- within the *control field:*
  - The playback response bit labels the packet as containing status information for a playback request.
  - The *Redelivery request bit* specifies that the packet is a request for retransmission of Level C lost data for a specific logical instrument.
  - The *Playback request bit* specifies that the packet is a request for a Level C archive playback for a specific logical instrument.
  - The *EOF flag bit* indicates that the packet contains an End Of File record, which is defined to be a record in which all bits are set.
  - The *LOD flag bit* indicates that a Loss Of Data occurred between the previous packet contents and the present packet contents.
  - The *LOS flag bit* indicates that a Loss Of Signal condition has been entered. When this bit is set, the EOF flag bit should also be set, and the contents of the packet should be an EOF record.
- The *data type* field is used to distinguish between messages containing header data and regular data on the Level C LAN. A header data block will have a 0 in this field and a data block will have a 1. Blocks of TLM minor frames will have a 2 in this field.
- The *message sequence* field is used to block data that has been sent as several messages. If a message is independent of the subsequent messages, this field contains zero.
- The *message counter* field is utilized by the receiving process to determine if a message has been lost or received out of sequence.

- The *time tag* refers to the time associated with the first data item in the data field; it consists of the milliseconds and seconds since the epoch.

- The *data source* values are listed in Table 6.

7.2.1. *Structures for the Data Message Control Fields*

The structures and defined constants to be used to access the control fields of data messages are:

```
/*
Include the contents of this file if it has not yet been included.
*/
#ifndef _loc_message_h

#ifndef __sys_types_h
#include <sys/types.h>
#endif

typedef struct mtu_s
{
        u_short d100                    : 2;
        u_short d10                     : 4;
        u_short d1                      : 4;
        u_short h10                     : 2;
        u_short h1                      : 4;
        u_short m10                     : 3;
        u_short m1                      : 4;
        u_short s10                     : 3;
        u_short s1                      : 4;
        u_short mshi                    : 2;
        u_short mslo                    : 8;
        u_short stat                    : 8;
} mtu_t;

typedef struct hdr_s
{
        struct
        {
                u_short reserved_bits:          2;
                u_short pback_response_bit:     1;
                u_short pback_request_bit:      1;
                u_short watchdog_bit:           1;
                u_short los_bit:                1;
                u_short eof_bit:                1;
                u_short lod_bit:                1;
                u_short unused_bits:            8;
        }           control;
        u_short     msec;
        u_int       sec;
        u_int       sequence;
        u_int       count;
        u_int       datakey;
        u_char      source;
        u_char      stream;
        u_char      rec_type;
```

```
    u_char     unused;
} hdr_t;

#define     PLAYBACK_RESPONSE control.pback_response_bit
#define     PLAYBACK_REQUEST control.pback_request_bit
#define     WATCHDOG control.watchdog_bit
#define     LOS control.los_bit
#define     EOFD control.eof_bit
#define     LOD control.lod_bit

/*
    Define _loc_message_h to mark this file as included.
*/
#define _loc_message_h

#endif _loc_message_h
```

7.2.2. Message Transport Between Subsystems

7.2.2.1. Data Messages

All data messages are broadcast to specific UDP ports, thus enabling any station to listen for any type of data. Table 6 defines the data source tags for both data source fields:

Table 6. Data Source Tags

Basic Data Source Name / Number	Logical Instruments Name / Number			
RETE/1	RTLA/1 RTLB/2 RTMC/3 RTMD/4 RTME/5 RTMF/6 RTHF/7	RTCA/8 RTCB/9 RTCC/10 RTCD/11 RTLP/12 RTAC/13 RTIP/14	RTDP/15 RTEF/16 RTFP/17 RTBW/18 RTBT/19 RTKA/20 RTKB/21	RTKC/22 RTKD/23 RTKE/24 RTCE/25 RTCF/26 RTCH/27
ROPE/2	RPBB/1 RPEA/2 RPIA/3 RPEB/4 RPIB/5	RPDP/6 RPEM/7 RPHA/8 RPHB/9 RPHC/10	RPHD/11 RPHE/12 RPHF/13 RPSA/14 RPSB/15	RPHG/16 RPSC/17 RPDM/18 RPSD/19
SETS/3	STTA/1 STTB/2 STAA/3 STCA/4 STCB/5 STCC/6	STCD/7 STFA/8 STFB/9 STTC/10 STTD/11 STSA/12	STSB/13 STSC/14 STMA/15 STMB/16 STDA/17 STDB/18	STDC/19 STBA/20 STBB/21 STBC/22 STBD/23 STBE/24
SPREE/4	SPIA/1 SPIB/2 SPIC/3 SPID/4 SPIE/5 SPIF/6 SPEA/7	SPEB/8 SPEC/9 SPED/10 SPHA/11 SPHB/12 SPHC/13 SPHD/14	SPHE/15 SPMF/16 SPOP/17 SPSA/18 SPLA/19 SPLB/20 SPLC/21	SPLD/22 SPFA/23 SPFB/24 SPTLM/25
TEMAG/5	TMMI/1 TMMO/2	TMHA/3 TMTA/4	TMHB/5	

DRAFT

Table 6. Data Source Tags (cont'd)

Basic Data Source Name / Number	*Logical Instruments Name / Number*			
DCORE/6	DCGA/1	DCSB/5	DCMB/9	DCTLM/13
	DCGB/2	DCTA/6	DCMC/10	
	DCGC/3	DCDV/7	DCMD/11	
	DCSA/4	DCMA/8	DCME/12	
SCORE/7	SCAF/1	SCSA/3	SCMA/5	SCMB/7
	SCAC/2	SCSB/4	SCMB/6	
CAS/8	CA41/1	CA72/5	CA79/9	CA92/13
	CA45/2	CA74/6	CA90/10	
	CA62/3	CA75/7	CA92/11	
	CA63/4	CA76/8		
DPLY/9	DPMA/1	DPTB/9	DPSH/17	DPMI/25
	DPMB/2	DPSB/10	DPSI/18	DPHD/26
	DPMC/3	DPMF/11	DPHA/19	DPHE/27
	DPMD/4	DPSC/12	DPHB/20	DPMK/28
	DPME/5	DPSD/13	DPHC/21	DPTC/29
	DPSA/6	DPSE/14	DPSJ/22	DPTLM/30
	DPMJ/7	DPSF/15	DPMG/23	
	DPTA/8	DPSG/16	DPMH/24	
SAHK/10	SHSA/1	SHAE/9	SHSG/17	SHSI/25
	SHSB/2	SHAF/10	SHSH/18	SHHB/26
	SHSC/3	SHSD/11	SHMA/19	SHHC/27
	SHTA/4	SHTB/12	SHMB/20	SHHD/28
	SHAA/5	SHTC/13	SHMC/21	SHHE/29
	SHAB/6	SHSE/14	SHMD/22	SHHF/30
	SHAC/7	SHPA/15	SHME/23	SHSJ/31
	SHAD/8	SHSF/16	SHHA/24	SHAG/32
DRB/11	DBTA/1	DBSB/3	DBPO/5	
	DBSA/2	DBSC/4		
EMP/12	EPMA/1	EPSC/5	EPMF/9	EPMH/13
	EPMB/2	EPMC/6	EPSD/10	
	EPSA/3	EPMD/7	EPSE/11	
	EPSB/4	EPME/8	EPMG/12	
SFMDM PDI/20				
SATELLITE PDI/21				
CAS DATA/22	N/A	N/A	N/A	N/A

The ports used to send and receive data messages are defined as:

Port number = 5001 + basic data source number

The base port number is selected as 5001 to insure that there is no possibility a port may already be in use by some system process. All logical instruments for a basic data source use the same port.

These port assignments are used to send data onto the secondary (Data Level D) Ethernet by default. However, the output port can be selected by the operator.

7.2.2.2. *Control Messages*

The UDP ports used by the various types of control messages are listed in Table 7.

Table 7. UDP Ports

Function	*Source*	*Destination*	*Port*
TLM Database Service	DHS	DHS	6020
IDFS Database Service	DHS	DHS	6010
IDFS Database Service	W/S	W/S	6010
TLM Archive Service	DHS	DHI	6021
IDFS Archive Service	DHS	DHI	6011
IDFS Archive Service	W/S	W/S	? 6011
Playback Request	W/S	Bcast	6012
Playback Response	DHS	W/S	6113
Heartbeat	DHS	Bcast	6001
Redelivery Request	W/S	Bcase	6003

These messages are used only at Data Level C. While both data handling systems will receive broadcast requests, only the non-muted one is able to reply; there will never be duplicate responses.

7.3. *Message Rates*

Table 8 shows the data message rates for the Level B LAN:

Table 8. PC Decom Output

Source	*Size*		*Message Rate*
	Minor Frames	Bytes	
SFMDM	1	518	8/.9984 sec.
Satellite	1	128	16/1.024 sec.
CAS	N/A	1200	2/sec. (maximum)

8. *System Level II (Data Handling Subsystem)*

The purpose of the data handling subsystem (DHS) is to ingest the telemetry and CAS messages and apply further processing. There are three main objectives of this processing: archiving the data, IDFS building, and transferring data to the display subsystem. This level must also respond to data playback requests from the display level during LOS periods. Utilities are provided to generate data tapes from the archive and perform other post-mission data formatting tasks. The software and hardware at this level are duplicated for increased SOC reliability. Only one system, selected by the operator, will be able to send data on to the Level C LAN or respond to requests coming from

Level III workstations. In the event that the active system fails, an operator command will immediately place the second system in active mode. When one of the systems is acting as a backup, it is running in *muted* mode.

8.1. Data Handling Subsystem Processes

The DHS software is composed of three layers. The first layer is the telemetry ingest layer, which obtains telemetry packets from the Level B LAN and writes the data to a file. The second layer is the data product-generation layer, in which the IDFS records are constructed and written to disk, and in which the TLM "pages" for SETS, SPREE, and DCORE are built. The third layer is the network distribution layer, in which data product records are broadcast on the Level C LAN.

8.1.1. Telemetry Ingest

The telemetry ingest process layer is composed of one process for each separate telemetry stream being received on the Level B LAN. Each telemetry stream arrives in broadcast fashion on a different port. Each packet received on a port contains a single minor frame of telemetry, along with a packet message header as described in section 7.2. There is one telemetry ingest program that is configurable for each of the different telemetry streams received. In operation, the telemetry ingest process first receives a TLM packet from the Level B LAN and checks it for sequentiality. Under normal conditions there is a doubling of each packet sent, since there are two different DCS systems broadcasting the same information. This is to insure data integrity, so that there is no loss of information if one of the DCS systems were to fail. As a result, the ingest process has to keep track of which data packet it is expecting next, since when two packets containing the same minor frame arrive, one is accepted and the other discarded. The DCS systems will place the minor frame number of each packet in a given data stream in the message counter field of the packet. The ingest process will read each packet and check that it has the expected length; any improper packets will be discarded. The ingest process will then compare the time stamps and minor frame numbers of the last packet received and a newly received packet. If the time stamps and minor frame numbers are equal, then the new packet is discarded. If the time stampes and minor frame numbers are not equal, then the new packet is accepted. The minor frame number is also checked for sequentiality.

When sequentiality has been confirmed, the data packet is written to a shared memory segment circular buffer. This makes the data available to any special computations processes that require it. When N packets have been written to the circular buffer, where N is specified at run-time, the process copies the data to disk. If a non-sequential packet is received, the ingest process sets the loss of data (LOD) bit in the packet header to mark the presence of a gap before writing the data to the circular buffer, and indicates an error condition to the system manager process. If a non-sequential packet is received in which the LOD flag is set by the DCS, the ingest process takes no special action. If a LOS condition is signaled to the process, then a file closure is performed on the archive file associated with the circular buffer, a special LOS record is written to the circular buffer, and a new archive file is opened. Once this is done, the process enters initialization mode and waits for new data packets. In the case that no closure has occurred for the last N writes to the circular buffer, where N is specified to the process at run–time, the process performs a file closure on the archive file associated with the circular buffer, a special EOF record is written to the circular buffer, and a new archive file is opened. The process does not enter initialization mode, since there is no loss of sequentiality implied by a simple file closure.

In all cases in this and other process layers, a special EOF or LOS record is a 12-byte object that contains a 4-byte identifier, which specifies the existence of an EOF or LOS condition, followed by two 4-byte integers containing the start and end time for the archive file in seconds since the UNIX epoch. An archive file closure is generally composed of five steps. Any unarchived data records in the circular buffer are flushed to the archive file; the archive file is renamed to indicate the start time of the file; the file is closed; a message is sent to the database server to indicate the name, start time, stop time, and size of the file; and a new file with the same name as the old file previous to renaming is opened. The initialization mode for the process is a state in which it is assumed that no packets have yet been received. The message count of the first packet received is used to initialize the sequentiality checking operation.

If the process receives a reinitialization signal from the DHS manager process, it will perform the same sort of processing as specified for LOS, with the exception that an EOF special record will be written to the circular buffer instead of a LOS special record. The telemetry ingest process maintains a count of the number of packets received, the number of LOD events seen, the number of packet loss events detected, and the number of packets lost. This information is maintained in shared memory in a status data structure. Each process also maintains a telemetry reception status, which indicates whether or not telemetry reception has resumed since the last LOS event. This amounts to a flag for the acquisition of signal (AOS) event. When an AOS event occurs, the ingest process will signal the DHS manager process to allow it to perform necessary AOS activities.

8.1.2. Data Product Generation

The data product generation process layer consists of a number of processes that read the data found in telemetry ingest circular buffers in shared memory and generate IDFS header and data records and TLM "page" products. These processes can also be directed to read data from files. There is one data product process for each experiment group substream found in the telemetry. The internal actions of each special computations process include synchronizing to the data substream, identifying experiment modes, packaging the data in a fashion appropriate for the experiment mode, performing specialized calculations on the data to convert or reorder the data or to generate additional values needed for the interpretation of the data, and writing completed records, such as IDFS header and data records, to shared memory output circular buffers for reading by the network distribution processes. The data in each output circular buffer will also be periodically copied to files on the disk when the run-time specified number of records for a given data product have been written into the corresponding circular buffer.

Apart from data mode specific changes in the operation of the data product processes, each process recognizes the flag that indicates an LOD condition in the message header found at the beginning of each telemetry data record. Each process also recognizes EOF and LOS conditions indicated by reading an EOF or LOS special record from a circular buffer. On recognition of an LOD condition in an input telemetry stream, a data product generation process will perform output data record padding and input resynchronization operations as needed. On recognition of an EOF condition in an input telemetry stream, a process takes no special action, since there ought to be continuity between the data before and after an EOF special record in a circular buffer. On recognition of a LOS condition in an input telemetry stream, a process will pad any incomplete output records as needed, and will perform archive file closure and database notification actions as described for the ingest processes on each output circular buffer. LOS special records as described in section 8.1.1 will be written to all output circular buffers. If there has been no LOS condition found in the last N input telemetry record reads, where N is specified to a process at run–time, a process will perform the same actions as specified for LOS, with the exceptions that an EOF special record is written to each output circular buffer instead of a LOS special record; and the process does not enter synchronization mode, since there has been no data gap. Each data product process maintains a count of the number of records read and the synchronization state (synchronized or not). This information is maintained in shared memory in a status data structure.

8.1.2.1. Main Module

The main module of each data product process is responsible for buffering telemetry data for an experiment and calling the IDFS builders. It reads the raw telemetry from the circular buffer created by the Telemetry Ingest Process and extracts data pertaining only to that experiment. When "enough" data has been buffered, the IDFS builder for each logical instrument is called. "Enough" varies from experiment to experiment and from telemetry stream to telemetry stream. For Satellite Telemetry, "enough" could be a quarter of a major frame, half of a major frame, or one major frame. For Smart Flex, one major frame is sufficient. The following is the basic algorithm for buffering the experiment data and calling the IDFS Builder.

```
Definitions:
                exp_mif_cnt= expected minor frame count
                mif_cnt = current minor frame count
                maf = major frame buffer
                mif = minor frame buffer
Algorithm:
                exp_mif_cnt = 0;
                Loop:
```

```
Read mif from Ingest circular buffer.
If (mif_cnt=exp_mif_cnt)
    Fill maf w/ experiment data from mif.
Else
    If (mif_cnt>exp_mif_cnt)
        Pad missing mif's from exp_mif_cnt to mif_cnt-1.
        Set data quality flag to partially filled.
        Fill maf w/ experiment data from mif.
    Else
        Pad remaining maf starting at exp_mif_cnt.
        Set data quality flag to partially filled.
        Call IDFS Builders to process maf.
        Pad maf from 0 to mif_cnt-1.
        Set data quality flag to partially filled.
        Fill maf w/ experiment data from mif.
    Endif
Endif
exp_mif_cnt = (mif_cnt + 1) mod (number of mifs per maf)
If (buffer is filled "enough")
    Call IDFS Builder.
```

8.1.2.2. IDFS Builder

Each Logical Instrument has its own IDFS builder module. When the builder is called for the first time, it creates two circular buffers in shared memory; one for the IDFS header records and the other for the IDFS data records. Using the data buffered by the main module, the IDFS builder extracts the necessary information and builds the IDFS data and header records as outlined in the IDFS Definition Document. If the telemetry buffer is padded by the main module or the IDFS builder, the data quality flag in the header record is set to indicate partially filled data. If the telemetry buffer has not been padded by either the main module or the builder and the data is in a mode not recognized by the builder, then the builder will assume the default mode and the the data quality flag is set to indicate questionable data. When an LOS signal is received or when n records have been written, the builder writes an EOF record to the data circular buffer and closes the circular buffers. The EOF record is to indicate that the file is closed but there is another file coming. When a QUIT signal is received, the builder writes an LOS record to the data circular buffer before closing the buffers. The LOS record is to indicate that the file is closed and no more files are coming. and closes the circular buffers.

8.1.2.3. Telemetry Page Builder

In addition to the IDFS builders, DCORE, SETS, and SPREE also have a Telemetry Page builder. This module is called by the main module immediately after every read from the SFMDM circular buffer. The TLM builder buffers the experiment data from eight SFMDM minor frames to build an instrument "page". These "pages" are passed immediately to the network distribution process for each experiment stream. This results in the broadcast of this data on the Level C LAN, but it bypasses the IDFS database and archive processes.

8.1.2.4. DCORE Synchronization

The DCORE experiment does not synchronize its minor frame 0 with the SFMDM minor frame 0. Thus, data product process buffers a whole SFMDM major frame and scans this data for DCORE minor frame 0. The frame offset into the major frame is checked for each SFMDM major frame.

8.1.2.5. SETS RPA Synchronization

The SETS Langmuir Probe and RPA have two problems which affect the data processing. The first is that the sweep voltages for these instruments are temperature dependent, which results in a variable sweep length. The second is that the sweep voltage is not returned along with each sample but only at selected steps within the sweep. To compound this problem, no sweep start indicator is returned. Thus, the IDFS builder determines the starting point for each sweep by using this method:

- A series of unambiguously descending or ascending sweep voltages is identified.
- A linear fit is made to these voltages.
- The sweep stop and start positions are determined by comparing the fitted voltages to predetermined absolute upper and lower limits.

There is no attempt to compute a voltage for each element in the data array.

8.1.3. Network Distribution

The record distribution process layer consists of one process per experiment substream. Each distribution process is identical, with the particular information required for the different substreams provided at run time. The network distribution process reads records from all of the data products produced by a single experiment group special computations process and broadcasts them on a single port on the Level C LAN. The data products are found in a set of circular buffers in shared memory. Each data record is packed into an output packet of the form described in section 7.2. The packet header fields are set appropriately and the packet is transmitted. This action is repeated for each data product in the experiment substream set. For a record from a particular data product, the packet header source and stream fields are set as described in section 7.2.2.1 according to which data product is being packaged. The type field will be set according to the data product subclass. There are presently three subclasses specified; IDFS data, IDFS header, and other. The count field is set to the number of records of this type that have been read from the associated circular buffer since the last EOF or LOS special record. At run–time it is possible to specify linkages between data products. If one data product is linked to another, the datakey field in the packet header for each record of the data product will be set to the number of records of the linked data product that have been read since the last EOF or LOS for the linked data product. This is to allow an IDFS data subclass data product to specify a link to an IDFS header subclass data product. This information is used by the listener process on each user workstation at Level III to ensure that no IDFS header data records are lost.If an EOF or LOS condition is read for a given record type, a special data packet is written in which the EOF or LOS flag is set. The packet data contents will be 4-byte start and 4-byte stop time for the associated archive file. The start and stop times are in seconds since the UNIX epoch. The count field will be set to the total number of data records read for that data product. This special packet is broadcast in triplicate to guard against loss by the receivers.

An option in the network distribution process can be used to prevent the final transmission of data packets on the Level C LAN. The process can be set to active or quiet mode by operator action through the system manager process. Each network distribution process maintains a count of packets transmitted. This information is maintained in shared memory in a status data structure.

8.2. Ancillary Processes

There are a number of ancillary processes that work in concert with the main flow of the data handling system. The manager process starts all processes and constitutes the user interface with the process group. The database server process handles the maintenance of the file system and initiates archive tape writing and file deletion activities. The record redelivery server handles lost header record requests that come from user workstations on the Level C LAN during AOS periods. The data playback server handles data transfer requests that come from user workstations on the Level C LAN during LOS periods. The tape archiver process creates *tar* (IEEE 1003.1-1988) compatible cartridge tapes of data files. The ping server obtains information about the health of the user workstations on the Level C LAN. The heartbear server has the sole function of broadcasting the LOS/AOS state on the Level C LAN. These processes communicate with the main DHS process layers and each other through ports, shared memory, and signals.

8.2.1. Manager

The data handling system manager process has many tasks. This process spawns all other processes in the system and communicates any operator originated messages to those processes. The process also provides the display interface for messages and status information originating with the children. This process provides system status displays of various sorts, such as disk activity and CPU activity. When this process is started, a shared memory segment is obtained for use by it and its children to maintain a status structure region for communicating status variables. A shared memory segment is obtained for use as a master name table for use in accessing shared memory circular buffers. The shared memory segments needed for the circular buffers are also obtained. The manager process maintains a port for receiving ping server LAN status messages. LOS events are signaled to the system through this process. AOS events are also communicated to the relevant processes, such as the playback server, from this process. This process provides the ability to take "snapshots" of the data being written by the ingest and special computations process layers and display that data in a hexadecimal dump format by byte columns.

8.2.2. Database Server

The database server process performs a number of tasks. The process maintains a port for receiving messages about newly closed files generated by the ingest and special computations process layers. The data about the newly closed files (name, start time, stop time, and size) is placed in the database file maintained by the process. The process also sends the database file record number for each file to an archiver process that will write the file to tape. The process keeps track of the amount of data present in each of the experiment directories, and begins deleting the oldest archived files when the number of bytes in a given directory increases beyond the allowable limit. The deleting of files continues until the directory size falls below a low-water mark. The data quantity variables are kept in shared memory for display by the manager process.

8.2.3. Redelivery Server

The redelivery server process monitors a port on the Level C LAN for redelivery request messages from user workstations and sends the requested data to the user workstation via the experiment port for which it was originally sent. The received message is stored into a FIFO queue, one for each possible requesting workstation. The workstation queues are polled in a round-robin fashion for pending requests. Upon finding a pending request, the server process accesses the current archive file to acquire the record requested. This record is then sent to the user workstation through the port related to the experiment requested. To prevent doubling of messages sent from both archive systems, this process responds to the active/mute capability of the manager process. If an archive system is muted, this server process does not send the requested message on the Level C LAN. This server process also responds to the AOS/LOS state of the archive system; if the current state is AOS, then the number of requests currently being processed is kept in shared memory for display by the manager process. During the LOS state, the server process clears all pending requests from all workstation queues; and the number of requests being processed is reset to zero.

8.2.4. Playback Server

The playback listens on a port on the Level C LAN for playback requests from user workstations and spawns processes which have the task of sending the requested data to the user workstations. When a request occurs, the port is read, and the playback request is put into a FIFO queue. If, however, the playback server has been notified of an AOS by the data handling system manager process, the playback request is discarded. This prevents any conflicts over system resources during normal telemetry processing. The playback server is only allowed to queue playback requests during an LOS.

When there are pending requests, the server queries the database for each entry in the request queue and copies the database entries that satisfy the playback request into workstation queues, with a maximum of ___ workstation FIFOs. The requesting workstation is then sent acknowledgement of data being sent through a special port. If no data is available on the DHS, a "no data" message is sent to the client.

The workstation queues are polled in a round-robin fashion for pending entries that need copying to the requesting workstation. When a pending request is found in any of the workstation queues, a process is spawned and the queued entry is passed to the process. The RCP facility is then used to copy the data associated with the database entry to the requesting workstation. After the RCP has completed, the database entry is sent to the requesting workstation on another Level C LAN port to be entered in the workstation local database. If the manager process notifies the playback server of an AOS, all queues are cleared. This server process has mute and active options that may be selected by the manager process. The playback server process maintains a count of requests made and the number of requests currently being processed in the status structure in shared memory for display by the manager process.

8.2.5. Archiver

The archiver process writes files in a sequential fashion to a tape in tar format. The process listens on a named pipe for database file record numbers. When a record number read on the pipe, the process places the information in an archive queue. In a background fashion, the process writes each file specified in the queue to a tape, encapsulating it in tar format. If the archiver determines that the tape is full, the process signals the manager process, dismounts the tape and waits for a new tape to be mounted before continuing. The manager process will notify the operator of the need for a new tape and ask for a name to label the tape with. The archiver process is signaled and the name is communicated through shared memory when the new tape is in place. The process will write a first file to the tape containing the label name and the date before writing data files. When a file has been written to tape the process alters the entry in the database file to indicate that the file has been archived and the name of the tape it was written to. The tape name, number of tapes written, and the amount of data written to the current tape are kept in shared memory in a status structure for display by and communication with the manager process.

8.2.6. Ping Server

The ping server process periodically sends messages over a range of addresses on the Level C LAN and listens for responses from the user workstations. This utility allows the status of each network interface to be monitored, as well as the configuration of the network software on each workstation. The process queries the network every 10 seconds with UDP packets directed to the *inetd* echo port and constructs a list of the workstations that respond. Changes in the number of workstations responding are reported by writing messages to a named pipe read by the manager process. Changes include stations that don't respond, stations that resume responding after a period of not responding, and ICMP messages that indicate errors, such as excess network loading. Corrupted packets will also be reported. These messages have the following syntax:

```
GAIN: <system name>

or

LOSS: <system name>
```

The ping server may be signalled to re-initialize the response list by the manager process.

8.3. System Configuration

The system is configured in a way that distributes the data storage burden across the disks in an even fashion. There is a first level directory which contains all the files used by the DHS. In this directory there are directories for each telemetry data stream and for each experiment grouping. These directories may be soft links to different disks, which allow the data burden to be shared equitably while maintaining a logical hierarchy that is easy to deal with. All database files are maintained in a directory at this same level. There is also a first level directory containing all configuration files needed for the various processes. When the manager process starts the various processes, it sets the working directory for each process to be the one in which its dataset resides. In this way, the various processes need only minimal information about the file system hierarchy. The only processes that need more global information are the database server, the redelivery server, and the playback server.

8.4. The Logical Instrument Data File Set Paradigm

The key data structure to be used by the special computations software is held by a hierarchical set of files. The data for each logical instrument will be contained in a set of three files: the Instrument Description File (IDF), the Header File (HF), and the Data File (DF). This set of files is called an Instrument Data File Set or IDFS. A one-to-one correspondence exists between the instrument description, header, and data files.

The reasons for utilizing this scheme are varied:

- data that changes infrequently does not have to be stored more often than necessary;
- searching for data from specific time periods does not require the use of special file types, indexing methods, or database software;
- the original telemetry words are preserved, thus providing compactness in storage;
- canonical data structures isolate applications software from instrument particularities;
- explicit metadata needed to utilize the data samples is packaged with the samples; and
- using this paradigm forces one to completely understand each data source.

The IDFS structure and fields are defined in SwRI Document "IDFS File System Definition." A listing of the IDFS definition documents for the TSS-1 missionis given in Table 9. The instrument data files are built record-by-record within this special computation process of the DHS.

Table 9. TSS-1 IDFS Definitions

	Title
IDFS-CAS-1	Calibrated Ancillary System (CAS) IDFS Definition
IDFS-DCORE-1	Deployer Core Equipment (DCORE) IDFS Definition
IDFS-DRB-1	Deployable/Retractable Boom (DRB) IDFS Definition
IDFS-DPLY-1	Deployer (DPLY) IDFS Definition
IDFS-EMP-1	
IDFS-RETE-1	Research on Electrodynamic Tether Effects (RETE) IDFS Definition
IDFS-ROPE-1	Research on Orbital Plasma Electrodynamics (ROPE) IDFS Definition
IDFS-SAHK-1	Satellite Housekeeping (SAHK) IDFS Definition
IDFS-SCORE-1	Satellite Core Equipment (SCORE) IDFS Definition
IDFS-SETS-1	Shuttle Electrodynamics Tether System (SETS) IDFS Definition
IDFS-SPREE-1	Shuttle Potential and Return Electron Experiment (SPREE) IDFS Definition
IDFS-TEMAG-1	Tether Magnetometer (TEMAG) IDFS Definition

8.5. Standard Format Data Units

A library of routines is provided to allow the DHS to encapsulate any data product with the CCSDS Standard Format Data Unit (SFDU) structure. This processing is optional and is not part of the SOC real-time activity. The library is called **/usr/lib/libsfdu.a** and contains the routines listed below.

TBA

9. Data Level C

The message format shown in Section 7.2 is used at this level. also. The primary differences between Levels B and C are:

- Level C contains data in the form of IDFS data and header file records.

- Level C contains messages directed towards the DHS level as well as those coming from the DHS.
- Level C will have activity during LOS periods. This playback activity will involve the remote copy utility (RCP) moving entire IDFS files to the requesting workstations. It may also involve RCP in copying Instrument Description Files that need to reside on a certain workstation.

9.1. Message Formats

Table 10 lists the data contents for each logical instrument message:

Table 10. IDFS LAN Messages

Primary Source	*Logical Instrument*	*Accumulation Period (secs)*	*Data Length*	*Header Length*
SPREE	SPIA	1	52	100
	SPIB	1	52	100
	SPIC	1	52	100
	SPID	1	52	100
	SPIE	1	180	108
	SPIF	1	180	108
	SPEA	1	52	100
	SPEB	1	52	100
	SPEC	1	180	108
	SPED	1	180	108
	SPHA	8	52	64
	SPHB	4	32	44
	SPHC	4	36	36
	SPHD	4	28	36
	SPHE	4	28	36
	SPMF	4	32	40
	SPOP	4	60	52
	SPSA	4	28	56
	SPLA	1–2	408	124
	SPLB	1–2	408	124
	SPLC	1–2	408	124
	SPLD	1–2	408	124
	SPFA	1–2	408	124
	SPFB	1–2	408	124

DRAFT

Primary Source	Logical Instrument	Accumulation Period (secs)	Data Length	Header Length
ROPE	RPBB	4	148	36
	RPEA	8	404	296
	RPIA	8	148	292
	RPEB	8	276	292
	RPIB	8	276	292
	RPDP	4	472	492
	RPEM	4	928	36
	RPHA	4	48	48
	RPHB	4	84	64
	RPHC	4	32	52
	RPHD	8	32	52
	RPHE	16	36	64
	RPHF	16	148	288
	RPSA	8	24	52
	RPSB	64	52	64
	RPHG	4	24	36
	RPSC	4	24	48
	RPDM	4	60	48
	RPSD	4	24	48
SCORE	SCAF	4	404	40
	SCAC	4	68	40
	SCSA	4	24	44
	SCSB	4	28	40
	SCMA	4	84	40
	SCMB	4	148	36
	SCMC	4	784	44
DRB	DBTA	4	32	40
	DBSA	4	28	52
	DBSB	4	48	60
	DBSC	4	36	36
	DBPO	4	84	40

Table 10. IDFS LAN Messages

DRAFT

Table 10. IDFS LAN Messages				
Primary Source	*Logical Instrument*	*Accumulation Period (secs)*	*Data Length*	*Header Length*
DCORE	DCGA	4	340	52
	DCGB	4	28	36
	DCGC	4	148	40
	DCSA	4	64	56
	DCSB	4	40	104
	DCTA	4	92	68
	DCDV	4	464	44
	DCMA	4	60	52
	DCMB	4	180	112
	DCMC	4	36	48
	DCMD	4	76	48
	DCME	4	116	44
	DCTLM	1	384	
TEMAG	TMMI	4	96	40
	TMMO	4	96	40
	TMHA	4	36	48
	TMTA	4	148	36
	TMHB	4	36	36

DRAFT

Table 10. IDFS LAN Messages				
Primary Source	*Logical Instrument*	*Accumulation Period (secs)*	*Data Length*	*Header Length*
DEPLOYER	DPMA	4	212	40
	DPMB	4	44	56
	DPMC	4	28	36
	DPMD	4	148	36
	DPME	4	212	80
	DPSA	4	24	56
	DPMJ	4	24	36
	DPTA	4	64	56
	DPTB	4	152	100
	DPSB	4	28	52
	DPMF	4	124	84
	DPSC	4	24	56
	DPSD	4	24	64
	DPSE	4	24	52
	DPSF	4	24	56
	DPSG	4	32	108
	DPSH	4	56	52
	DPSI	4	24	44
	DPHA	4	84	36
	DPHB	4	140	92
	DPHC	4	96	72
	DPSJ	4	24	64
	DPMG	4	36	36
	DPMH	4	36	36
	DPMI	4	148	36
	DPHD	4	28	40
	DPHE	4	212	40
	DPMK	4	24	36
	DPSK	4	28	36

DRAFT

Table 10. IDFS LAN Messages				
Primary Source	*Logical Instrument*	*Accumulation Period (secs)*	*Data Length*	*Header Length*
RETE	RTLA	32	84	68
	RTLB	32	60	68
	RTMC	32	776	72
	RTMD	32	776	72
	RTME	32	320	72
	RTMF	32	776	72
	RTHF	32	540	544
	RTCA	32		
	RTCB	32		
	RTCC	32		
	RTCD	32		
	RTLP	32	796	292
	RTAC	32	752	36
	RTIP	32	84	36
	RTDP	32	1044	36
	RTEF	32	80	36
	RTFP	32	80	36
	RTBW	16	1196	36
	RTBT	16	1196	36
	RTKA	32	88	68
	RTKB	32	28	48
	RTKC	32	24	52
	RTKD	32	24	36
	RTKE	32	276	36
	RTCE	32		
	RTCF	32		
	RTCH	32		

Table 10. IDFS LAN Messages

Primary Source	*Logical Instrument*	*Accumulation Period (secs)*	*Data Length*	*Header Length*
SAHK	SHSA	16	28	100
	SHSB	4	68	56
	SHSC	4	24	36
	SHTA	4	44	44
	SHAA	4	340	44
	SHAB	4	116	44
	SHAC	4	56	48
	SHAD	4	400	44
	SHAE	16	64	56
	SHAF	16	24	36
	SHSD	16	28	136
	SHTB	4	180	112
	SHTC	4	100	72
	SHSE	4	28	64
	SHPA	4	104	76
	SHSF	4	52	64
	SHSG	16	24	92
	SHSH	4	68	44
	SHMA	4	24	48
	SHMB	4	36	40
	SHMC	16	80	92
	SHMD	16	24	80
	SHME	16	28	44
	SHHA	16	36	60
	SHSI	4	28	52
	SHHB	16	40	44
	SHHC	16	32	44
	SHHD	4	132	48
	SHHE	16	24	36
	SHHF	4	40	40
	SHSJ	4	40	52
	SHAG	4	276	48

9.1.1. *Message Rates*

Messages are not buffered once they are complete; hence, the message rate depends upon the time to build a particular logical instrument record. Rates will be essentially the same as the accumulation period.

10. *System Level III (Joint Display and User Workstations)*

The workstations at this level capture and display various graphs of selected parameters. While all workstations at this level are loaded with the same software, two are dedicated to running predefined displays. These are called the Joint Science and Joint POCC displays. Besides various configuration files, discussed below, there are three main software subsystems: the listener processes, the database client, and the display clients. The database and display client subsystems are controlled through a central menu interface as explained in Sections 1-3 of the SOC User Manual, SOC-UM-1.

10.1. *Listener Processes*

Figure 5 illustrates both the processes and the data flow involved in recording data on a Level III workstation:

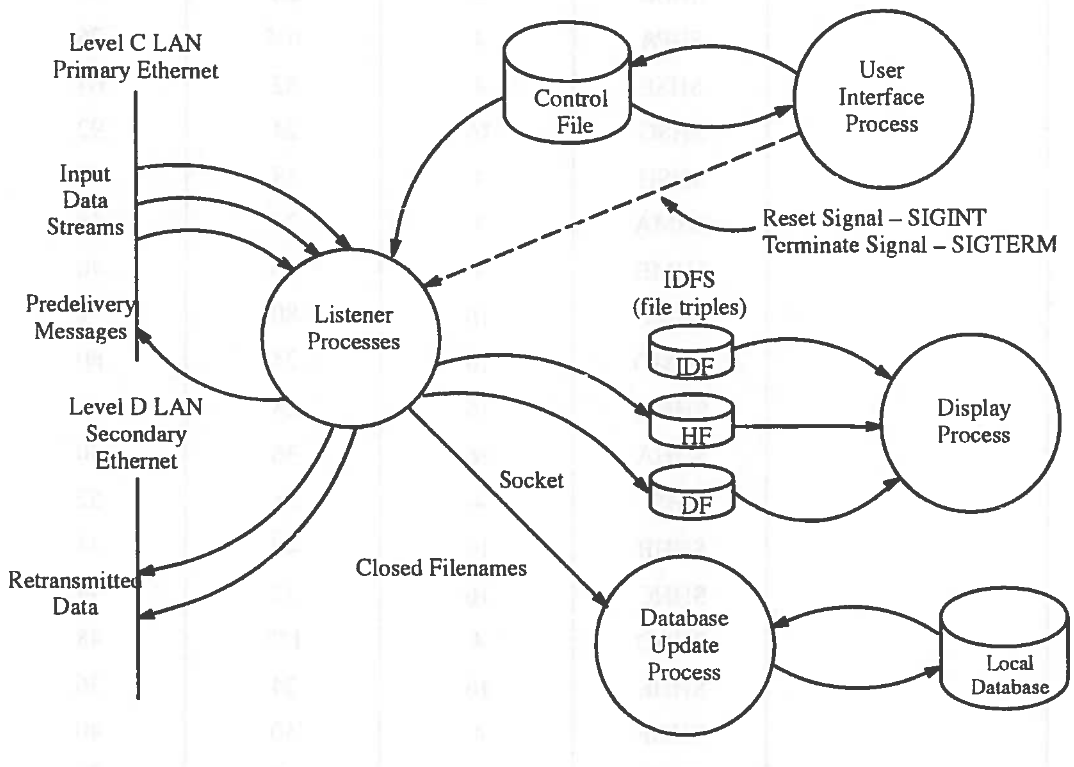


Figure 5. Workstation Data Capture

The listener has two main functions: a) the recording of data onto local disk storage; and b) the retransmission of data messages on a secondary LAN interface. The latter function is also called "echoing" the data. The These functions can be combined in several ways, depending upon the data sources selected. This process creates a child process to handle each experiment data stream to be processed. These processes are controlled by a configuration file that uses the following kekywords and syntax:

- Input is from a text file with one command per line.
- Sources and logicals can be enabled and have their echo turned on/off in any order.
- Data source configuration

```
source ECHO                -- echoes source to 2nd Ethernet
source NOECHO              -- stops source from echoing
source ON                  -- allows source to be captured
                           -- all active logicals capture
                           -- does NOT turn logicals on or off
source OFF                 -- stops all capture from source
                           -- does NOT turn logicals on or off
source INPUT port          -- input for source from port
source OUTPUT port         -- output (echo) source on port
source anystring           -- where anystring is not listed above
                           -- sets source for config of logicals
PRIMARIES ECHO             -- echo all data sources
PRIMARIES NOECHO           -- do not echo any sources
PRIMARIES ON               -- capture from all sources
                           -- all active logicals capture
                           -- does NOT turn logicals on or off
PRIMARIES OFF              -- stop capture from all sources
                           -- does NOT turn logicals on or off
```

- Logical instrument configuration

These commands refer to last source mentioned in the configuration file. Initially or after PRIMARIES commands, no source is current. Use a line like "ROPE LOGICALS" or "CAS INSTRUMENTS" to make the desired source active.

```
logical ON                 -- activates logical
                           -- will capture when source ON
logical OFF                -- disables logical
                           -- will not capture even if source ON
logical ARCHIVE            -- will archive header & data file
```

```
                                   -- only applies if logical is capturing
    logical ARCHIVE OFF            -- will not archive header & data file
                                   -- only applies if logical is capturing
    LOGICALS ON                    -- activates all logicals for source
                                   -- will capture when source ON
    LOGICALS OFF                   -- disables all logicals for source
                                   -- will not capture even if source ON
```

In this description, "source" names a primary data source (e.g. RETE), "logical" names a logical instrument (e.g. RPSE), and "port" is a numeric port address greater than 5000. Sidebar comments preceded by "—" are NOT part of a command line; such comments are part of this description only. Also, lines that start with a "#" are comment lines.

The master listener process determines the number of network interfaces installed and the addresses broadcast address and network associated with each interface. If a workstation has only a single interface, the echo data functions will be ignored. A command line augment to the master process specifies whether the first interface is for input or output. Other augments define the directory path for captured data files and the log file pathname. If more than two network interfaces are installed, only the first two will be used.

The listener processes are controlled by signals. The signal SIGINT causes the configuration file to be reread the next time an LOS condition exists. It also causes all the child processes to terminate, because the master listener process will create the appropriate child processes, depending upon the size of the configuration file. The signal SIGKILL causes an orderly termination of the program. These processes are to be run in the background and stdem is attached to a log file.

Several data message header fields are interpreted by these processes. The LOS/LOD flag bits are used to close the specified output files or sockets. The message type field is used to determine if the name of a closed file must be passed to the local catalog process. This is usually the case with IDFS files, but is not done with raw telemetry data. The record sequence field contains a counter that indicates the next record to write. In the case of a data record, the data is written to the file regardless of the state of any previous records. Header records receive specialized consideration. If a record position is skipped, the listener will request that that record be redelivered.

These processes have been designed to be used to both send and receive data on a secondary LAN. However, redelivery requests are not available to workstations that are attached only to the secondary LAN.

10.2. Database Client

TBA

10.3. Joint DisplaySoftware

The displays defined in detail in SOC-JDD-1 are called the Joint Science and POCC Displays. They are created on two designated Level III workstations by providing the display clients with prebuilt files to select the appropriate logical instruments to capture onto disk storage.

10.4. Tabular Display Software

Figure 6 gives a high-level view of this software subsystem:

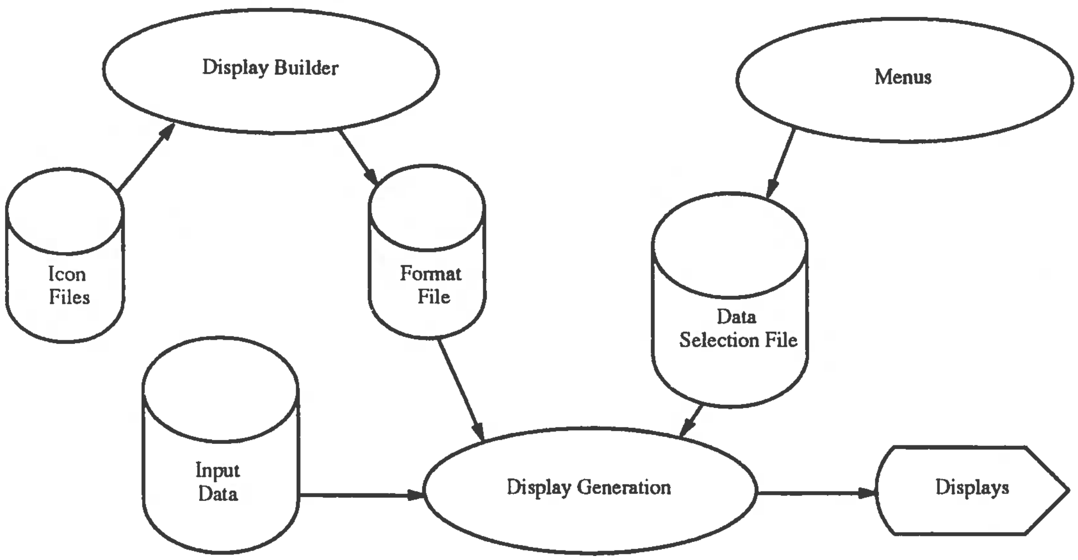


Figure 6. Tabular Display Subsystem

11. Data Level D

This level contains data rebroadcast on the secondary Ethernet port of a Level III workstation. Because non-SOC equipment may be attached to the Level D LAN, it is not possible to characterize the network load. SOC workstations are not configured to accept or respond to network requests originating from Level D.

12. Color Printer

Five Tektronix 4693RGB color printers with Option 02 four-channel multiplexers and Option 4L video adapters will be provided for the SOC. SwRI will provide three color printers, and ASI and MSFC will provide the other two. See Appendix B, Figure B-1, for a typical color printer station setup.

Color prints may be made at any time by screen dump.

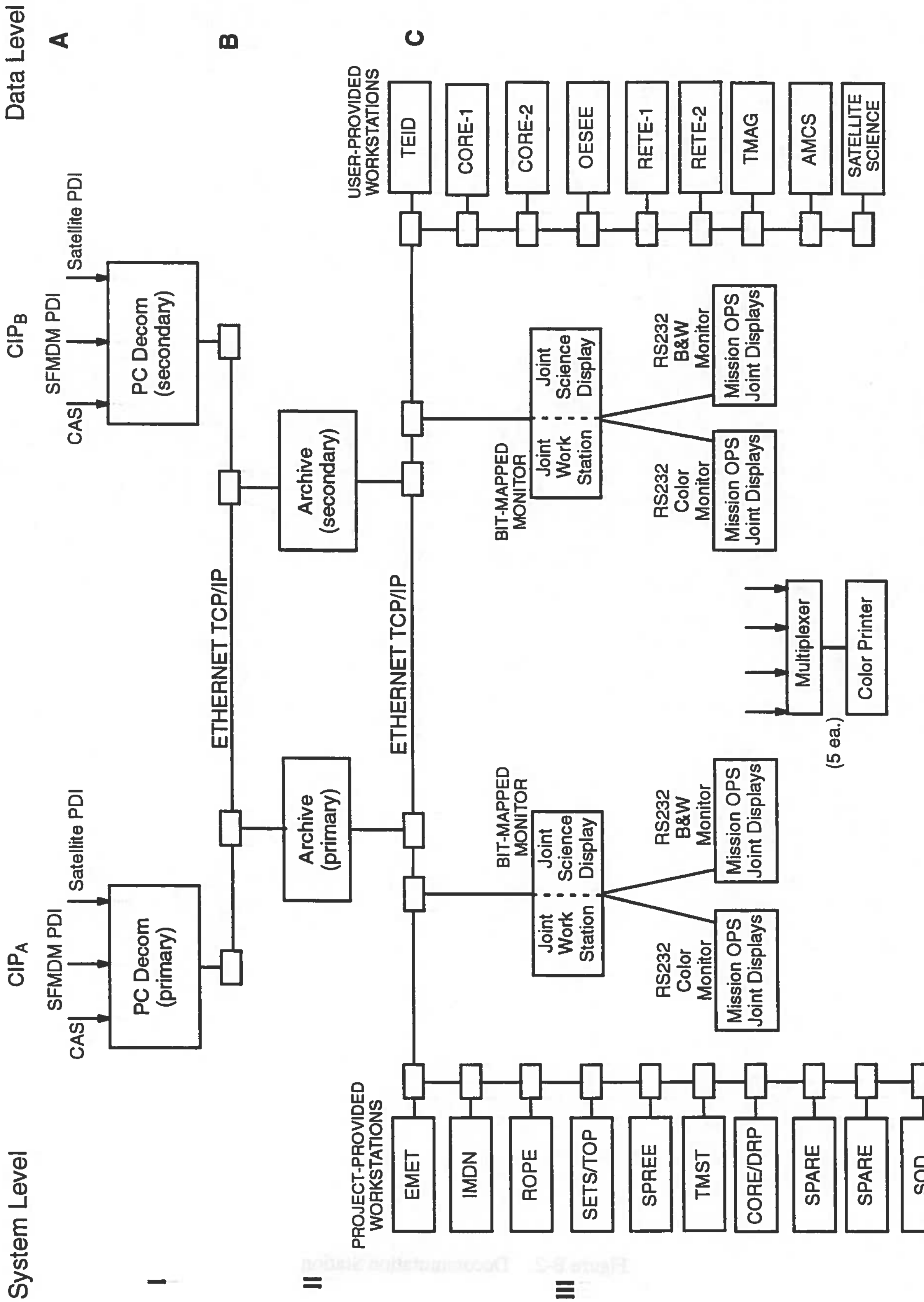


Figure 1. SOC Configuration

Figure B-2. Decommutation Station

Figure B-3. Joint Display Station

Figure B-4. Individual Data Handling System

Figure B-4. User Worksstation (with Video Adapter)

Figure B-5. User Workstation (without Video Adapter)

Figure B-7. Color Printer Multiplexor