Title **Herschel EGSE Packet Router ICD**

Prepared by : Albrecht de Jonge   Date : July 30, 2001

Oct 23, 2001

Agreed by : HIFI EGSE manager: L. Dubbeldam   Date :

HIFI ICC manager: P. Roelfsema

PACS EGSE manager: E. Wiezorrek

PACS ICC manager: O. Bauer

SPIRE EGSE manager: J. Payne

SPIRE ICC manager: K. King

ESA HSC manager: J. Riedinger

ESA AIV manager: (TBS)

Distribution list

HIFI	L. Dubbeldam P. Roelfsema A. de Jonge
PACS	E. Wiezorrek O. Bauer
SPIRE	J. Payne K. King
HSC	J. Riedinger (AIV manager)

Document Change Record

Issue	Date	Changed Section	Description of Change
Draft 2	February 8, 2001		Rewrite to standard table of contents
Version 1.0	July 30, 2001	5.2	First official release Added recovery procedures
Version 1.1	October 23, 2001	1, 3.1	Refer to interface requirements

Table of contents

1 INTRODUCTION

1.1 Purpose

The Herschel EGSE packet router will distribute telemetry and telecommands, in the standard format given by **[AD 1]**, among the components of the EGSE set up during instrument level tests. The same interface will be used by the HCSS for ingestion of raw telemetry in later phases of the Herschel mission. The context of this is shown in **[AD 2]**.

1.2 Scope

This document describes the protocol used to connect to the routing system and exchange data.
It also describes the protocol by which the HCSS expects to receive raw telemetry.
This document corresponds to ICD #4 in **[RD 1]** and covers requirements given in **[AD 3]** section 4.2.1.

1.3 Applicable Documents

1	Packet Structure ICD	SCI-PT-ICD-7527	Issue 1.0	1 September 2000
2	Ground System Design Description	FIRST/FSC/DOC/0146	v1.0	3 November 2000
3	FIRST ground segment Interface Requirements Document	FIRST/FSC/DOC/0117	Issue 1 rev 3	3 November 2000

1.4 Reference Documents

1	Ground Segment List of ICDs	FIRST/FSC/DOC/0150	v1.0	23 November 2000

1.5 Acronyms List

AD	Applicable Document
APID	Application Program Identifier
CCE	Central Checkout Equipment
CDMS	Command and Data Management System
EGSE	Electrical Ground Support Equipment
HCSS	Herschel Common Science System
ICD	Interface Control Document
ILT	Instrument Level Test
IST	Integrated Satellite Test
LAN	Local Area Network
MOC	Mission Operations Centre
PS-ICD	Packet Structure Interface Control Document
RD	Referenced Document
SCOS	Spacecraft Operating System
TC	TeleCommand
TCP/IP	Transfer Control Protocol / Internet Protocol
TM	TeleMetry

2 OPERATIONAL ASSUMPTIONS AND CONSTRAINTS

During instrument level tests, it is intended to exchange telecommands and telemetry between a SCOS-2000 system, test equipment interface computers, the HCSS telemetry ingester and the CDMS simulator, connected together by a TCP/IP local area network. The router is a software package running on one of the computers in the network enabling this exchange. All clients will open a TCP connection to the router. The router will receive messages containing telemetry and telecommand packets, and forward them to the clients for which they are intended. Forwarding is based on the packet address, where packet address is defined as the packet APID and the distinction between telemetry/telecommand.

Clients can also exchange messages with the router containing control information in order to configure the forwarding process.

2.1 Communications

During instrument level tests, it is intended to exchange telecommands and telemetry between a SCOS-2000 system, test equipment interface computers, the HCSS telemetry ingester and the CDMS simulator, connected together by a TCP/IP local area network. The router is a software package running on one of the computers in the network enabling this exchange.
For each setup of the router, there has to be an agreement on the TCP port to be used for the router and the names to be used by the clients.

2.2 Hardware

The router will run on any computer supplying the required software environment.

2.3 Software

The router requires to be run on a platform supporting Java.

2.4 User

Upon starting of the router, a TCP port number will have to be specified.

2.5 Timing

N/A

3 REQUIREMENTS

3.1 Functional Requirements

This interface meets the requirements specified in **[AD 3]** section 4.2.1, with the following exceptions:

- Requirement FSG-IR-4.2-21 (association of telemetry with observation context) must be met by the test equipment generating the telemetry.
- The performance requirements FSG-IR-4.2-40 and FSG-IR-4.2-50 may not be met if the supporting platform and network have insufficient free capacity.

3.2 On-Line Delivery Requirements

N/A

3.3 Off-Line Delivery Requirements

N/A

4 INTERFACE CHARACTERISTICS

4.1 Interface Location and Medium

Clients and router are to be connected on a TCP/IP based local area network with a capacity of 5 Mbit/s (TBC) freely available for router traffic.

4.2 Hardware Characteristics and Limitations

Other data traffic using the same LAN will limit the performance of the router. The capacity calculation must be analyzed taking into account that packets are routed from source A to the router and then to destination B. The total traffic can be calculated as

Total network bitrate =
Sum over (all clients) of sending bitrate
+ Sum over (all clients) of receiving bitrate
– Sum over (all clients running on the router machine) of sending bitrate
– Sum over (all clients running on the router machine) of receiving bitrate

This means that it can be advantageous to run the router on a major source or sink of data.

4.3 Data Source, Destination and Transfer Mechanism

N/A

4.4 Node and Device Addressing

The router will listen on a TCP port to be agreed upon for the particular setup.

4.5 Relationship with other interfaces

N/A

5 ACCESS

5.1 Programs generating or using the Interface Data

N/A

5.2 Failure Protection, Detection and Recovery Procedures

The client can break the connection at any time. The router will break the connection for a client that does not adhere to the protocol.
The following crash recovery procedures are available:

5.2.1 Router crash recovery

In case the router crashes or is killed, all clients connected will normally notice this by losing the network connection. Recovery is by simple re-establishing the connection. This should be done by the same procedure as the original connection, re-using the same client name, and repeating the necessary requests for packet forwarding. Any packets in transit during the crash are lost.

5.2.2 Client crash recovery

In case the client crashes or is killed, the router handles this as (normal) disconnect by the client. Upon restart the client can reconnect in the same way as originally. Packets received by the router for a dead client are lost.

5.2.3 Network failure recovery

In case the network connection fails, the router will handle this as if the client failed, and the client can handle this as if the router failed.

5.3 File Naming Conventions

N/A

5.4 Storage and File Deletion Requirements

N/A

5.5 Security Requirements

The router does not provide any security. Unauthorized access must be prohibited by external means, e.g. using a disconnected network, a private network, a firewall, or similar means.

5.6 Data Integrity Checks

There is no explicit handshake, error checking, or flow control foreseen, as this is provided by the TCP/IP protocol.

5.7 Backup Requirements

N/A

5.8 Input / Output Protocols, Calling Sequences

An example showing the basic use of the router for to exchange packets between two client applications is shown in appendix A.

5.9 Synchronisation Requirements

N/A

5.10 Error Handling

See section 5.2 and 5.6

6 DETAILED INTERFACE SPECIFICATIONS

6.1 Data Structure

The interface components connect by opening a TCP/IP connection with the router. After opening the connection they can send messages or expect to receive messages. All message share the same overall structure shown in table 1.

6.1.1 Overall message structure

Table 1

Start octet	Field length (octets)	Field name	Format
0	1	messageType	Integer
1	4	contentLength	Integer
5	ContentLength	messageContent	Octet array

- The actual number of octets in a message is 5 more than the value of the contentLength field.
- The value of contentLength is limited to 1100 (octets) (TBC).
- When integer values are included in a multi-octet field, they are stored most-significant octet first.

This ICD defines 12 different types of message, which will use three different structures, as shown in Table 2.

Table 2

Type value	Mnemonic	Content	Source	Structure
1	USER_DATA	PS-ICD packet	Router, client	Data-packet
2	ADD_CLIENT	Ask to receive certain packet addresss	Client	Client-info
3	DEL_CLIENT	Ask not to receive certain	Client	Client-info

		packet addresss		
4	ASK_CLIENT	Ask to list clients	Client	Client-info
5	SHOW_CLIENT	Reply listing a client	Router	Client-info
6	NAME_CLIENT	Ask to name this interface	Client	Client-info
7	ADD_BLOCK	Ask to block a traffic route	Client	Route-info
8	DEL_BLOCK	Ask to remove a block	Client	Route-info
9	ASK_BLOCK	Ask to list all blocks	Client	Route-info
10	SHOW_BLOCK	Reply listing a blocked route	Router	Route-info
11	ASK_TRAFFIC	Ask to list traffic counts	Client	Route-info
12	SHOW_TRAFFIC	Reply giving traffic count	Router	Route-info

6.1.2 Data-packet structure

The content of this message is a complete packet conforming to the PS-ICD **[AD-1]**.
Not that the message content length field contentLength is then related to the packet header length field defined in the PS-ICD as

contentLength = packet_length_field + 7

The packet address is an integer value, used to determine the destination of the telemetry/telecommand packets. For telemetry source packets the packet address is equal to the 11-bit APID value from the packet header. For telecommand source packets it is 4096 + the 11-bit APID value. Note that this is equivalent to masking out the Type and APID fields of the packet header.
Packet address value 8192 is reserved for special purposes.

6.1.3 Client-info structure

The structure of the content of these messages is shown in table 4

Table 3

Start octet	Field length (octets)	Field name	Format	Valid for message types
0	4	packetAddress	Integer	ADD_CLIENT, DEL_CLIENT, SHOW_CLIENT
4	4	clientAddress	Octet array	SHOW_CLIENT
8	4	clientPortNumber	Integer	SHOW_CLIENT
12	4	messageSequenceNumber	Integer	SHOW_CLIENT
16	contentLength – 16	clientName	ASCII char array	NAME_CLIENT, SHOW_CLIENT

Notes:

1. The start octet is with respect to the message content, excluding the message header
2. These messages all share the common structure, but the value of the field must be ignored if is not shown as valid for the current message type. This means that the contentLength is always at least 16 octets.

6.1.4 Route-info structure

Table 4

Start octet	Field length	Field name	Format	Valid for message types

0	4	packetAddress	Integer	ADD_BLOCK, DEL_BLOCK, SHOW_BLOCK, SHOW_TRAFFIC
4	4	sourceNameLength	Integer	ADD_BLOCK, DEL_BLOCK, SHOW_BLOCK, SHOW_TRAFFIC
8	4	destinationNameLength	Integer	ADD_BLOCK, DEL_BLOCK, SHOW_BLOCK, SHOW_TRAFFIC
12	4	messageSequenceNumber	Integer	SHOW_BLOCK, SHOW_TRAFFIC
16	4	packetCount	Integer	SHOW_TRAFFIC
20	source NameLength	sourceName	ASCII char array	ADD_BLOCK, DEL_BLOCK, SHOW_BLOCK, SHOW_TRAFFIC
20 + source NameLength	destination NameLength	destinationName	ASCII char array	ADD_BLOCK, DEL_BLOCK, SHOW_BLOCK, SHOW_TRAFFIC

Notes:
1. The start octet is with respect to the message content, excluding the message header
2. These messages share a common structure, but the value of the field must be ignored if is not shown as valid for the current message type.
3. The following shall hold to ensure message integrity: sourceNameLength + destinationNameLength + 20 = contentLength
4. The value of destinationNameLength and/or sourceNameLength can be zero, with a special meaning
5. The value of packetAddress can be the special value 8192.

6.2 Generation Method

6.2.1 USER_DATA message

These messages may always be sent from any client to the router. The packet address is computed from the packet ID field in the packet header.
The router will not check the packet content in any way.
Upon reception of this message, the router will send (forward) a copy of this message to all clients satisfying:
1. The client has sent an ADD_CLIENT message specifying this packet address.
2. The router has not received an ADD_BLOCK message specifying to block the forwarding of these packet addresss from the sending client to the receiving client.

Note that the effect of ADD_CLIENT and ADD_BLOCK messages to the router can be revoked by corresponding DEL_CLIENT and DEL_BLOCK messages

6.2.2 ADD_CLIENT message

These messages may always be sent from any client to the router. The only valid field in the message content is the packetAddress field.
Upon reception of this message, the router will update its internal state so that the client sending this message will be forwarded a copy of all USER_DATA message with packet address equal to packetAddress field, unless this route is explicitly blocked.

6.2.3 DEL_CLIENT message

These messages may always be sent from any client to the router. The only valid field in the message content is the packetAddress field. The packetAddress value shall not be the special value 8192.
Upon reception of this message, the router will update its internal state so that the effect of any preceding ADD_CLIENT message from the same client with the same packetAddress field is undone.

6.2.4 ASK_CLIENT message

These messages may always be sent from any client to the router. There are no valid fields in the message content.
Upon reception of this message, the router will send back to the client one or more SHOW_CLIENT messages.

6.2.5 SHOW_CLIENT message

These messages will be sent by the router in response to an ASK_CLIENT message. All fields in the message content are valid.
Each message shows a possible destination for USER_DATA packets, as identified by the packetAddress and clientName fields. For debugging purposes, the clientAddress and clientPortNumber fields will show the IP address and port number of the client connection are also reported. Clients not having sent any ADD_CLIENT (or having revoked all ADD_CLIENT messages) will be reported with a single SHOW_CLIENT message with the packet address value set to the special value 8192.
The messageSequenceNumber field will be set to the number of SHOW_CLIENT messages that are to follow in response to the same ASK_CLIENT message. So the last SHOW_CLIENT message of a response will have the messageSequenceNumber field set to zero, and the first of a sequence of ten will have it set to 9.

6.2.6 NAME_CLIENT message

This message must be the first sent by a client after establishing contact with the router. The only valid field in the message is the clientName field. The clientName field value must be unique among all clients of the router. A client re-using the same name as an earlier disconnected client of the same router will be considered as the same client in the context of route-info messages.

6.2.7 ADD_BLOCK message

These messages may always be sent from any client to the router. All fields except messageSequenceNumber and packetCount are valid. The combination of zero destinationNameLength, zero sourceNameLength and special packetAddress value 8192 is disallowed.
Upon reception of this message, the router will add the contents to its internal blocking table.
Before forwarding a USER_DATA message to its destination the router will check the blocking table. If a matching entry is found, the USER_DATA message will not be sent.
A matching blocking table entry is found if all three of the following conditions are true:

1. The name of the sending client equals sourceName or sourceNameLength is zero
2. The name of the potential destination client equals destinationName or destinationNameLength is zero
3. The packet address matches packetAddress or packetAddress has the special value 8192

6.2.8 DEL_BLOCK message

These messages may always be sent from any client to the router. All fields except messageSequenceNumber and packetCount are valid.
Upon reception of this message, the router will update its internal state so that the effect of the preceding ADD_BLOCK message with the same message content is undone.

6.2.9 ASK_BLOCK message

These messages may always be sent from any client to the router. There are no valid fields in the message content.
Upon reception of this message, the router will send back to the client one or more SHOW_BLOCK messages

6.2.10 SHOW_BLOCK message

These messages will be sent by the router in response to an ASK_BLOCK message. All fields in the message content except packetCount are valid.
Each message shows one entry of the routers blocking table.
If the routers blocking table is empty, a special SHOW_BLOCK message is sent with messageSequenceNumber, destinationNameLength and sourceNameLength set to zero and packetAddress set to the special value 8192.
The messageSequenceNumber field will be set to the number of SHOW_BLOCK messages that are to follow in response to the same ASK_BLOCK message. So the last SHOW_BLOCK message of a response will have the messageSequenceNumber field set to zero, and the first of a sequence of ten will have it set to 9.

6.2.11 ASK_TRAFFIC message

These messages may always be sent from any client to the router. There are no valid fields in the message content.
Upon reception of this message, the router will send back to the client one or more SHOW_TRAFFIC messages

6.2.12 SHOW_TRAFFIC message

These messages will be sent by the router in response to an ASK_TRAFFIC message. All fields in the message content are valid.
Each message shows the total amount of packets forwarded since the router started, for the given combination of packet address, source and destination.
If the routers statistics table is empty, a special SHOW_TRAFFIC message is sent with messageSequenceNumber, destinationNameLength and sourceNameLength set to zero and packetAddress set to the special value 8192.
The messageSequenceNumber field will be set to the number of SHOW_TRAFFIC messages that are to follow in response to the same ASK_TRAFFIC message. So the last SHOW_TRAFFIC message of a response will have the messageSequenceNumber field set to zero, and the first of a sequence of ten will have it set to 9.

6.3 Data passed across the interface - direction of transfer

See Table 2.

6.4 Size and Frequency of Transfers

The data rate per client shall not exceed 500 kbit per second, averaged over one second. (TBC)

7 DATA DEFINITION

N/A

8 FIGURES/TABLES

APPENDIX A: EXAMPLE CALL SEQUENCE

Figure 1 shows as an example the calls sequences for two application clients. Client A only sends telemetry packets of APID 77, client B wants to inspect them.
Using a (TBS) interface client A opens the connection and identifies itself. Then it sends packets with the proper APID at regular intervals.
Client B opens the connection, identifies itself, and expresses an interest in packet address 77 (TM of APID 77). When the router has received the corresponding messages from the interface, it starts forwarding the packets received from A to B. After a while Client B stops, after first nicely revoking its request for packet address 77. The router then stops forwarding.
Note that this is only an example. The activities of Client A and Client B do not have to be synchronized in any way.

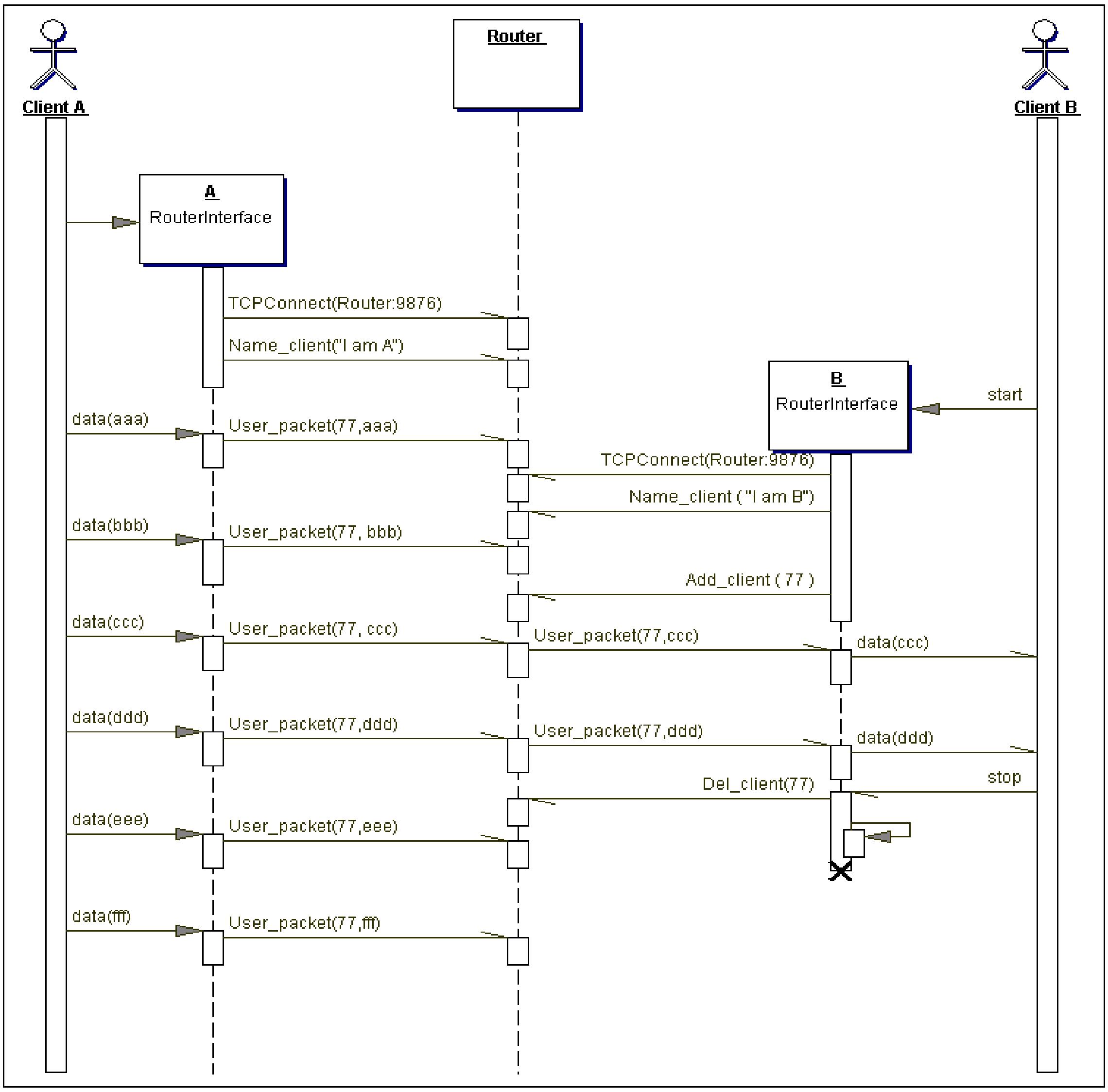

Figure 1 Example call sequence