# ORBCOMM Serial Interface Specification

**ORBCOMM, Inc.**

21700 Atlantic Boulevard
Dulles, Virginia 20166, U.S.A..

Issue Date: 1996

Revised August 2005

Revision G

**NOTE:** This document contains technical information and descriptions of the ORBCOMM Serial Interface. The contents of this document are subject to change without notice, and no warranty or representation, expressed or implied, is made with respect to its contents.

**ORBCOMM PROPRIETARY INFORMATION**

This document contains information that is confidential and proprietary to ORBCOMM Inc. its suppliers, and/or other third parties, and may not be disclosed except under obligations of confidentiality and subject to provisions of a Non-Disclosure Agreement between the proposed recipient and ORBCOMM Inc.
or one of its affiliates.

**NOTE:** ORBCOMM® is a registered service mark of ORBCOMM Inc. Other marks as used herein are the property of their respective owners.

Copyright © 2005 ORBCOMM Inc. All Rights Reserved.

| Change Record Sheet | | |
|---|---|---|
| **Date** | **Revision No.** | **Description** |
| April 1999 | F | This version of the Serial Interface Specification represents a change in direction for the Subscriber Communicators (SCs). In an effort to support lower SC prices, ORBCOMM intends to eliminate requirements for functionality that is not frequently used. In addition, ORBCOMM wishes to promote a new, more robust serial packet retry mechanism and a more consistent status and configuration mechanism.<br>Consequently, after March 1, 2000, ORBCOMM will no longer require SC's to implement Doppler-based position determination and byte Mode. NOTE: should an SC Manufacturer elect to implement Doppler-based position determination or byte Mode, the implementation shall adhere to the specifications described herein.<br>In addition, all SC's Type-Approved after August 1, 1999 shall implement the following new functionality:<br><br>Additional status_code values for the link level ack packet that indicate rejection of a packet (see Table 3.1);<br>A new serial packet retry mechanism to correct a deficiency in the original method (see Section 2.4.1); and<br>A consistent mechanism for evaluating the SC's status and specifying it's configuration:<br>Creation of a new get parameter packet (Table 3.20);<br>Creation of a new set parameter packet (Table 3.21);<br>Creation of a new parameter response packet (Table 3.22); and<br>Definition of additional, generally-useful, configuration parameters.<br>Three new packets that can transfer to the DTE information useful to Satellite Pass Prediction (see Tables 3.23 – 3.25).<br><br>Finally, in order to promote the standardization of Command Mode interfaces, this specification introduces an AT-style Command Mode whose implementation is optional. Note however, that if an SC manufacturer elects to implement any Command Mode interface, the AT-style format, as described in Section 2.3, shall be implemented. Additional manufacturer-specific Command Mode variations may also be implemented if desired. |

| Change Record Sheet | | |
|---|---|---|
| **Date** | **Revision No.** | **Description** |
| August 2005 | G | Spelling and typographical errors corrected.<br><br>Removed references to documents no longer supported or released by ORBCOMM.<br><br>Removed all references to byte Mode, Doppler-based position determination, and the AT Command set as no current SCs support this functionality.<br><br>Added notes to THE SC-ORIGINATED MESSAGE and SC-ORIGINATED DEFAULT MESSAGE packets specifying a maximum total packet length of 8192 bytes.<br><br>Updated Figure 4.1 to illustrate the response of the SC to an invalid GET or SET PARAMETER packet.<br><br>Updated Figure 4.7 to illustrate the handling of a recipient acknowledgment of an SC-ORIGINATED MESSAGE.<br><br>Updated Figures 4.3, 4.8 and 4.12 to show the complete Acquire-Communicate cycle for all interactions between the SC and the Satellite.<br><br>In Chapter 3, repeated the note regarding the Data Type byte found in the discussion of the SC-ORIGINATED MESSAGE packet in the relevant part of the SC-TERMINATED MESSAGE packet.<br><br>Removed SC-ORIGINATED ENHANCED GLOBALGRAM and SC-TERMINATED ENHANCED GLOBALGRAM packets from Chapter 3 since ORBCOMM has no plans to support them.<br><br>Revised diagrams and corresponding text in Note 11 at the end of Chapter 3.<br><br>Added type_code 30 to the COMMUNICATIONS COMMAND which signals the SC to prepare for the removal of external power. Documented the SC's response in Table 3-27.<br><br>Clarified format and order of PIN code bytes in CONFIGURATION COMMAND packet.<br><br>In Section 2.1, provided additional information about the order of the length and Fletcher checksum packet bytes.<br><br>Clarified order of latitude and longitude code bytes in POSITION STATUS and SC-ORIGINATED POSITION REPORT packets and removed reference to the bytes formerly associated with Doppler-based position determination from the POSITION STATUS packet.<br><br>Corrected latitude resolution in the POSITION REPORT packet.<br><br>Added header byte for new serial packet retry mechanism to description of the POSITION DETERMINATION and POSITION STATUS packets. |

| | | |
|---|---|---|
| | | In Table 3.5, adjusted names of bytes 5 & 6 of the STATUS packet (***sc_state*** and ***sc_diag_code***) to match corresponding configuration parameters. Added reference to Appendix A for definition of these parameters.<br><br>In Appendix A, changed the access type of the configuration parameters ***ser_max_retries*** (0x21) and ***max_pos_det_queue*** (0x78) from read-only to read-write.<br><br>In the PARAMETER RESPONSE packet description in Chapter 3, corrected numbering of packet bytes and a typographical error in the Note.<br><br>In Chapter 3, note 6, added reference associating IA5 data table with data type 5.<br><br>Corrected the mean motion multiplier in the SATELLITE ORBITAL ELEMENTS packet<br><br>In Appendix A, changed the default, minimum, or maximum values for the following parameters: *desired_gwy, queued_ob_msgs, queued_ib_msgs,* and *time_bytes.*<br><br>In Appendix A, deleted parameter 0x25, *ops_mod.*<br><br>Edited Figure 4.5 SC-Originated Globalgram to show delivery to GCC.<br><br>Globally replaced references to GlobalGram with Globalgram. |

# TABLE OF CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

# Chapter 1 Overview

## 1.1 Scope

This document describes the interface between ORBCOMM-certified SCs and Data Terminal Equipment (DTE). The intent is to provide an acceptable interface to every possible data communications application suitable for the ORBCOMM System, and to standardize the interface so that an SC can be used for many different applications.

The document is divided into five chapters and six appendices, which are summarized below:

*Chapter 1 OVERVIEW* defines the document scope, presents text conventions, provides terminology clarifications, and lists related documentation.

*Chapter 2 PROTOCOL DESCRIPTION* provides an introduction to the SC protocol, the SC's position determination functionality, and the internal parameters that affect the SC's operation.

*Chapter 3 PACKET DESCRIPTIONS & NOTES* defines the packets exchanged between the SC and the DTE that controls it.

*Chapter 4 EXAMPLES* presents ladder diagrams that detail the response of the SC and the ORBCOMM Network to each serial interface packet.

*Appendix A SC PARAMETERS* lists internal SC parameters accessible via GET and SET PARAMETER packets.

*Appendix B NETWORK CAPABILITIES* discusses the four basic formats of exchanging data over the ORBCOMM Network.

*Appendix C FLETCHER CHECKSUM* provides C source code implementing the Fletcher encoding and decoding functionality.

*Appendix D LATITUDE & LONGITUDE CONVERSION* illustrates the process of converting between ORBCOMM position format and Geodetic position format.

*Appendix E SATELLITE ELEMENT SET FORMAT DESCRIPTION* defines the format of the Satellite Element Set used in the ORBCOMM Network.

*Appendix F GLOSSARY OF TERMS & ACRONYMS* provides a list of all acronyms used in this document.

## 1.2 Text Conventions

This document uses the following text conventions:

Phrases that identify packets described in the *ORBCOMM Subscriber Communicator Air Interface Specification* (document A80TD0009 - Revision B) appear in SMALL CAPS.

THE bold **SMALL CAPS** font identifies ORBCOMM serial interface packets, as enumerated in Chapter 3 of this document.

Text appearing in *italicized* type indicates configuration parameters, which are described in Appendix A.

Parameters within Air Interface or Serial Interface packets appear in ***boldface italicized*** type.

## 1.3 Terminology Clarifications

This document uses the following terminology:

**"SC-Originated messages"**

This term refers to "inbound" transferred data—i.e., data that originates from SCs and travels to the Satellites and the ORBCOMM System.

**"SC-Terminated messages"**

This term refers to "outbound" transferred data—i.e., data that transfers from the ORBCOMM System and Satellites to SCs.

**"message"** (lower case "m")

This is a generic term that refers to any subscriber data that passes through the ORBCOMM System.

**"Message"** (upper case "M")

This term refers to a specific type of subscriber data that passes over the ORBCOMM System.

## 1.4 Related Documentation

The following documents provide background information on the ORBCOMM System and provide information on the SC's communication with Satellites:

*ORBCOMM System Overview* (document no. A80TD0008).

*ORBCOMM Messaging Services Description* (document no. A80MK0019).

## 1.5 Revision Control

Revision history is listed at the beginning of this document. Changes for this document start from Revision F.

# Chapter 2 PROTOCOL DESCRIPTION

In this document, the generic descriptions “data terminal” and “data terminal equipment” (DTE) refer to the device connected to the SC, and not to the SC itself. Also, the term “message handler agent” (MHA) refers to the process in the DTE which manages the sending and receiving of messages via the SC.

The role of the SC is to connect customer applications to the ORBCOMM Network. ORBCOMM requires general-purpose SCs to implement a serial interface for this purpose.

## 2.1 Application Layer

The packets exchanged by the SC and DTE are specified in Chapter 3. All packets begin with a header byte (byte 0) to enable the receiver to synchronize to the packet boundaries and to indicate the desired packet retry mechanism (see Section 2.4.1). Packets transmitted by the DTE shall begin with a header byte of 0x85 or 0x86; a header byte of 0x05 or 0x06 indicates the packet was transmitted by the SC. All packets also contain a unique packet type identifier (byte 1) and a two-byte length field. The least-significant byte of the length field is inserted into byte 2 of the packet. The protocol also mandates the use of two Fletcher checksum bytes to detect errors in the packet bytes. This checksum is calculated over all packet bytes. The least-significant byte of the checksum precedes the most significant in the packet.

The least-significant bit of the least-significant packet byte is transmitted first. The bytes of a packet are generally transmitted without an inter-character delay—the start bit of the next byte immediately follows the stop bit of the previous byte.

The protocol provides more functionality than may be required for some applications. In those cases, the DTE need only implement a subset of the described packets/protocol. It is the intention of ORBCOMM, however, to require that all SCs support the full functionality of the protocol, except in the case when the SC is not equipped with position determination capability or a serial port. As previously stated, this will enable an SC to be employed in many different applications.

The SC will queue messages, reports and Globalgrams from the DTE on a priority basis, to be sent either in a polled or SC-Originated mode. The DTE may track the acknowledgment of each Message, Report, or Globalgram it generates by specifying a unique reference number in each of these packets (the ***mha_ref_num***). Once the SC’s internal message buffer is full, additional Messages, Reports, or Globalgrams from the DTE are discarded and an **ACK** packet is sent with ***status_code*** = 1 (see Section 2.4 and Table 3.1). The message capacity of the SC is limited by its available message queue memory.

If the SC is not storing messages or reports received from the DTE, reception of the POLL COMMAND will result in transmission of a **SYSTEM ANNOUNCEMENT** packet to the DTE (see Figure 4.14). Otherwise, the first queued message or report from the DTE that satisfies the

requested response (indicated by the ***control_code***) in the POLL COMMAND will be removed from the queue and sent to the ORBCOMM Gateway. If no messages from the DTE in the queue satisfy the request, a **SYSTEM ANNOUNCEMENT** packet will be sent to the DTE.

The SC will also queue SC-Terminated messages, commands, and Globalgrams until the DTE is ready to accept them, as signaled by the activation of DTR or de-activation of RTS. If the SC's internal buffer is full, the SC transmits a SUBSCRIBER RECEIVER READY packet to the ORBCOMM Gateway–with the ***control_code*** = 1, NOT ready to receive–in response to SC-TERMINATED ASSIGNMENT packets.

Manufacturers may implement application-specific packets of their own design. The packet types must range from 30 to 255. Packet types 24 through 29 are reserved for future protocol use. The manufacturer-defined packets must adhere to the specification format for the header (bytes 0 - 4) and checksum bytes.

# 2.2 Link Layer

It is expected that the DTE will be co-located with the SC, or at least within 50 feet (15 meters), which is the typical RS-232 cable length specification. This distance can be extended to 200 feet (61 meters) with high-grade cable, or even farther with RS-232 line drivers. The transmission reliability provided by this high-quality, low-error-rate link enables simplification of the serial link protocol—long messages need not be broken into multiple packets before being transferred between the SC and DTE.

Consequently, the protocols between the SC and the DTE are half-duplex (i.e., a packet must be acknowledged by the recipient before another can be sent), thereby enabling simple software flow control. Furthermore, the protocol is designed exclusively for point-to-point links–only one DTE and one SC can be connected to the serial link.

The protocol requires only one control packet–the **LINK LEVEL ACKNOWLEDGMENT**, or **ACK** packet. It is used to acknowledge the reception of "information packets" (see Chapter 3.1, Packet Descriptions). Error-free packets are acknowledged by responding with an **ACK** packet with a ***status_code*** of 0. If, however, the received information packet is in error (e.g., bad checksum), the packet receiver will transmit an **ACK** packet with a non-zero ***status_code*** (see Table 3.1).

If the application does not require or cannot support link layer acknowledgment, the SC can be configured with *ser_max_retries = 0* and *abort_response = 0.* In this case, the SC will make only one attempt to send an information packet to the DTE, and the lack of an **ACK** from the DTE will not generate a serial port failure indication (abort report).

## 2.2.1 Invalid Packets

Though the serial link between the DTE and SC is expected to be very reliable, there are two mechanisms for handling scrambled packets. The first mechanism, which has been described in previous versions of this specification and is implemented in all SCs available as of the date of the release of this specification, defines the fifth byte of each packet as a ***retry_count*** field that indicates how many times the packet has been transmitted. When the packet originator receives an **ACK** packet with a non-zero ***status_code***, or fails to receive an

**ACK** packet at all, it transmits the corresponding information packet again, after incrementing the ***retry_count*** field of the re-transmitted packet. A packet with a non-zero ***retry_count*** indicates to the recipient that the information packet is a duplicate of a previously received information packet. The receiver then re-sends the appropriate **ACK** packet, and discards the second information packet if the packet has been processed previously. If the receiver observes the ***retry_count*** drop to zero or decrease, it should assume that a new packet has been received.

Unfortunately, there is a flaw in this mechanism that can result in lost messages when communications between the DTE and the SC are not reliable.

**EXAMPLE:** Initially the DTE sends a message to the SC (msg 1, retry = 0). The SC receives the message and acknowledges it. Next, the DTE sends a new message (msg 2). In the event the SC does not receive message 2 (e.g., electrical noise, transients, etc.), it will not respond to the DTE with an **ACK** packet. After the DTE times-out, it retransmits message 2 with a retry count of 1. The SC receives this packet and **ACK**s it, but discards the packet, incorrectly believing it to be a retransmission of message 1. Having received an **ACK** for message 2, the DTE believes the packet was accepted and therefore does not re-send it.

The fundamental cause of this problem is the inability of the SC to distinguish between the scenario in which a new message from the DTE is lost and the case in which the DTE fails to receive one of the SC's **ACK**s. To address this problem, ORBCOMM has developed an alternate mechanism for handling packet errors.

This revised error handling mechanism redefines the ***retry_count*** field of the serial interface packets (byte four) to represent a packet sequence number (***pkt_seq_num***), with the packet sender selecting a unique value for each distinct message. Retransmissions of packets rejected by the recipient or not acknowledged retain the ***pkt_seq_num*** of the original packet. Messages having the same ***pkt_seq_num*** are processed only once. A ***status_code*** in the **ACK** packet indicates that the packet being acknowledged was a duplicate and was therefore discarded.

To achieve backwards compatibility with the original protocol, packets whose fifth byte is to be interpreted as a ***pkt_seq_num*** instead of a ***retry_count*** shall begin with a packet header byte of 0x86 (for packets from the DTE to the SC) or 0x06 (packets transmitted from the SC to the DTE).

It is possible to write application software to use older SCs that do not support this enhanced retry processing mechanism yet adopt the more robust retry mechanism should the SCs be replaced with fully-compliant units. As part of its application software initialization, the DTE can send a packet to the SC with a header byte of 0x86. If the SC supports the enhanced retry handling scheme, this packet will be processed as expected. If the SC does not support the new retry mechanism, it will not recognize the packet header byte (0x86 instead 0x85) and will not respond to the DTE. If the SC repeatedly fails to acknowledge the packet, the DTE can assume that the SC does not support the enhanced retry handling mechanism.

Upon power-up, the SC will assume that the DTE software does not support the enhanced retry mechanism. Should the SC need to spontaneously send a packet to the DTE (e.g., **SYSTEM ANNOUNCEMENT**) it will use a packet header byte of 0x05 and a fifth byte that indicates the packet's retry count until it receives a packet from the DTE with a packet header byte of 0x86.

The SC shall respond to DTE packets using the packet header corresponding to that which started the DTE packet. In other words, DTE packets that begin with 0x85 will generate an SC response that begins with 0x05; packets from the DTE starting with 0x86 will cause the SC to respond with a packet beginning with 0x06.

### 2.2.2 DTE Failure Notification

To enable detection of DTE failure, the SC implements a response timeout mechanism. After transmitting a packet to the DTE, the SC will set a timer for *ser_pkt_timeout* seconds. As stated above, the DTE should always respond to packets transmitted by the SC with an **ACK** packet, regardless of the received packet type or presence of errors within the packet. Should the DTE fail to respond within *ser_pkt_timeout* seconds, the SC will re-transmit the packet, increment a DTE response failure count, and re-start the response timer. The SC can be configured to send an abort report to the customer's application hub after *ser_max_retry* DTE response timeouts. More information on these parameters is provided in Appendix A.

If communication with the DTE has failed, communication with the DTE should still be attempted. The only consequence of the failure is the transmission of the abort report and the removal of the failed packet from SC memory.

# 2.3 Physical Layer

While the protocol described in Section 2.1 is independent of the actual method used for communicating the data bits, all SCs built so far have implemented an asynchronous serial link, which is described below.

Although RS-232 signal levels are mandated by this specification, ORBCOMM acknowledges that the ability to remove the relatively inefficient RS-232 voltage regulator and line driver circuits may be required by some applications that require extremely low power and long battery life. ORBCOMM will consider waiving the requirement for RS-232 levels on a case-by-case basis.

While a three-wire (TxD, RxD, and ground) interface is likely to meet the requirements of nearly all applications, flow control may be used to prevent data overruns. This implies manipulation of the Data Terminal Ready (DTR) and Request-To-Send (RTS) lines by the DTE and the Data Set Ready (DSR) and Clear-To-Send (CTS) lines by DCE (SC).

## 2.3.1 Signal Description

The following list identifies the signals of the SC serial interface. Some of the signals have non-standard definitions. Not all signals will be used by every application. To simplify selection of DTEs, ORBCOMM considers optional all signals but Ground, Transmitted Data, and Received Data.

**Data Terminal Ready (DTR)**

The DTR signal may be configured for differing responsibilities via the *ob_flow_ctrl* and *pwr_down_mode* configuration parameters (Appendix A). It may be used to enable the DTE to command a low-power mode within the SC–the DTE simply deactivates DTR and the SC will power-down non-essential components. The transition of DTR from inactive to active will result in the awakening of the SC. The SC can otherwise ignore DTR, or it can be used for flow control.

**Request-To-Send (RTS) & Clear-To-Send (CTS)**

Manipulation of RTS and CTS may be required for reliable full-duplex operation; however if half-duplex is configured, then implementation of these two signals is optional. CTS can be used for flow control by the SC by itself, however half-duplex operation should provide sufficient flow control when the messages are not large.

**Carrier Detect (CD)**

The SC activates CD whenever it is locked onto the Satellite downlink and is receiving GATEWAY INFORMATION, DOWNLINK CHANNEL INFORMATION and UPLINK CHANNEL INFORMATION packets.

**Data Set Ready (DSR)**

DSR is activated whenever the SC is powered. DSR can also be configured for flow control responsibilities, or to indicate queued SC-Terminated messages.

**Signal Ground (GD)**

Signal ground assures a common ground reference between the SC and the DTE.

**Transmitted Data (TxD)**

TxD conveys serial data from the DTE to the SC.

**Received Data (RxD)**

RxD conveys serial data from the SC to the DTE.

### 2.3.2 Timing

The SC must support baud rates of 300, 600, 1200, 2400, 4800, and 9600.

The duration of bits transmitted by the SC shall differ by not more than 1% from the nominal duration derived from the selected baud rate (i.e., 300 baud: 3.33 ms, 600 baud: 1.67 ms, 1200 baud: 83.3 μs, 2400 baud: 41.7 μs, 4800 baud: 20.8 μs, and 9600 baud: 10.4μs).

At all baud rates, the SC shall accept from the DTE without error serial transmissions in which the data is contiguous, i.e., with no inter-character delay.

## 2.4 Configuration

To support the widest variety of applications, the SC's operational parameters must be configurable. In previous versions of this Serial Interface specification, not all of the SC's parameters were accessible and there was no consistent mechanism for accessing those parameters that were. This specification addresses these issues by aggregating all of the DTE-accessible parameters and status into a table. DTE's can access the SC's operating parameters and internal status via the **GET PARAMETER**, **SET PARAMETER**, and **PARAMETER RESPONSE** packets (Tables 3.20, 3.21, and 3.22). These packets reference parameters and status information in the table via indexes that correspond to the parameters' locations within the table. Appendix A presents the parameters and status data that are accessible in this manner, their corresponding indexes, and their pre-set factory default values and ranges. (The factory default values listed in the table are recommended settings. Each SC manufacturer may assign different values.)

Two sets of these parameters exist within the SC–one fixed set which contains the pre-set factory defaults, and a second set which can be customized for the application. Both sets must be stored in non-volatile memory. Only one set is active at any time, and the active set must be re-selected automatically following a cycling of power. It may be necessary to restore the pre-set factory defaults if the configuration becomes unknown and the SC becomes inaccessible through the serial link. The implementation of this functionality is left to the SC manufacturers.

# 2.5 Position Determination Algorithm

The DTE can control the position determination capability of the SC using the packets described in Tables 3.15 and 3.16. Following the successful calculation of SC position, the SC automatically returns a **POSITION STATUS** packet to the DTE. A POSITION REPORT is not automatically sent to the ORBCOMM Gateway. The application can initiate this action, however, through the **COMMUNICATIONS COMMAND** packet (see Table 3.3) or can send an **SC-ORIGINATED POSITION REPORT** packet to the SC.

# Chapter 3 PACKET DESCRIPTIONS & NOTES

## 3.1 Packet Descriptions

### LINK LEVEL ACKNOWLEDGMENT

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x85 or 0x06/0x86 | packet header byte (0x05/0x06: SC ->DTE, 0x85/0x86: DTE -> SC) |
| 1 | 0x01 | packet type |
| 2 | 0x07 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | *status_code* | identifies reason for error if any (see below) |
| 5 | checksum byte 0 | Fletcher checksum |
| 6 | checksum byte 1 | |

*status_code* descriptions:

0 = no error

1 = buffer unavailable - wait 30 seconds then send again

2 = packet rejected, invalid checksum

3 = packet rejected, invalid parameter

4 = packet rejected, size exceeds queue capacity

5 = packet rejected, ill-formed

6 = packet rejected, unrecognized packet type

7 = packet rejected, duplicate packet sequence number

**Table 3.1 Link Level Acknowledgment**

## CONFIGURATION COMMAND

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x02 | packet type |
| 2 | 0x14 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *pin code byte 0* | personal identification number, used as a security measure, 4 BCD digits (min = 0, max = 9999), pin code byte 0 is least significant |
| 6 | *pin code byte 1* | |
| 7 | *pin code byte 2* | |
| 8 | *pin code byte 3* | |
| 9 | *desired_gwy_id* | instructs SC to acquire Satellite having this ORBCOMM Gateway |
| 10 | *def_polled* | SC-Originated: polled by ORBCOMM Gateway or initiated by SC (see Section 3.2, note 1) |
| 11 | *def_ack_level* | default acknowledgement level (see Section 3.2, note 3) |
| 12 | *def_rep_or_ind* | default OR indicator for Reports (see Section 3.2, note 4) |
| 13 | *def_msg_or_ind* | default OR indicator for Messages (see Section 3.2, note 4) |
| 14 | *def_priority* | default priority level (see Section 3.2, note 5) |
| 15 | *def_msg_body_type* | default body type for Messages (see Section 3.2, note 6) |
| 16 | *def_serv_type* | default service type for Reports (see Section 3.2, note 2) |
| 17 | *gwy_search_mode* | method for searching for downlinks (see Appendix A) |
| 18 | checksum byte 0 | Fletcher checksum |
| 19 | checksum byte 1 | |

**Table 3.2 Configuration Command**

## COMMUNICATIONS COMMAND

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x03 | packet type |
| 2 | 0x0D | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *type_code* | type of action requested (see below) |
| 6 | *value byte 0* | generic value pertinent to type of action requested (used with *type_code*: 4, 5, 7, 8, 10-15, 18, & 25) |
| 7 | *value byte 1* | |
| 8 | *value byte 2* | |
| 9 | *value byte 3* | |
| 10 | *gwy_id* | destination ORBCOMM Gateway (used with *type_code*: 0, 1, 3, 5-15, 23) |
| 11 | checksum byte 0 | Fletcher checksum |
| 12 | checksum byte 1 | |

*type_code* Definitions:

0 = request all SC-Terminated messages/commands queued in ORBCOMM Gateway

1 = request all SC-Terminated messages/commands queued in ORBCOMM Gateway of size no greater than 150 bytes

2 = request all Globalgrams queued in satellite

3 = request OR indicator addresses (Section 3.2, note 4)

4 = request status of SC-Originated message/report/Globalgram identified by MHA message reference number in value field

5 = request status of SC-Originated message/report identified by ORBCOMM Gateway message reference number in value field

6 = request list of subjects of SC-Terminated messages queued in ORBCOMM Gateway

7 = request a single SC-Terminated message, identified by index in list of subjects, indicated in the value field. (Index value is inferred from position of message subject relative to other message subjects, starting with value 1. List of subjects ob tained from previous transmission of SC communications command with *type_code* = 6)

8 = delete a single SC-Terminated message, identified by index in list of subjects, indicated in the value field (see type_code =7 above)

9 = request registration with ORBCOMM Gateway (not currently used)

10 - 15 = generic functions (see Section 3.2, note 15)

16 = request status packet

17 = clear active message being transferred (either SC-Originated or SC-Terminated) between SC and ORBCOMM Gateway

18 = clear SC-Originated message identified by *mha_ref_num* in *value* field

19 = clear all messages in SC-Originated queue

20 = clear all messages in SC-Terminated queue

21 = perform self-test (internal check of memory, modem, etc.)

22 = perform local loop-back test (Dummy packet sent internally through SC - nothing transmitted to Satellite)

23 = perform loop-back test with ORBCOMM Gateway (SC sends Report with OR indicator = 0, ORBCOMM Gateway sends Command with first 5 Report bytes)

24 = power down until DTR transitions from inactive to active

25 = power down for *value* seconds

26 = power down for at least *value* seconds, then awaken with the rise of the next satellite pass

27 = send position report to ORBCOMM Gateway

28 = send orbital elements as an SC-Terminated message (NASA 2-line format, Appendix A) to the DTE.

29 = force SC to next known downlink in system

30 = prepare for removal of external power

31-47 = reserved for future ORBCOMM use

48 and above = SC Manufacturer defined

## Table 3.3 Communications Command

## SYSTEM ANNOUNCEMENT

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x04 | packet type |
| 2 | 0x0A | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *announce_code* | system provided information |
| 6 | *gwy_id* | originating ORBCOMM Gateway (not applicable for announce code 2) |
| 7 | *dest_or_ind* | destination OR indicator (must be used in response to poll if report, see below) |
| 8 | checksum byte 0 | Fletcher checksum |
| 9 | checksum byte 1 | |

*announce_code* Definitions:

0 = calculate position estimate then send SC-Originated Position Report (for DTE equipped with position estimation capability, i.e. GPS)

1 = send one report

2 = not used

3 = not used

4 = send one SC-Originated message

5 = send one SC-Originated message or report

6 = send pre-defined OR addresses

7 = not used

8 = not used

9 - 15 generic functions (see Section 3.2, note 15)

16 and above = SC manufacturer defined (see Section 3.2, note 15):

- *dest_or_ind (used as destination for response of poll)*
- *dest_or_ind = 0 has no effect, should be ignored*
- *If a Report is polled and dest_or_ind is non-zero, the dest_or_ind must be used as the or_ind in the Report.*
- *If a message is polled and dest_or_ind is non-zero, this dest_or_ind must be included in the responding Message, if any.*

**Table 3.4 System Announcement**

## STATUS

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x05 | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/pkt sequence number |
| 5 | *sc_state* | state of SC, message transport processes (see App. A) |
| 6 | *sc_diag_code* | diagnostic code from self test, etc. (see Appendix A) |
| 7 | *active_mha_msg_ref* | MHA # of message being transferred (0xFF = no msg) |
| 8 | *sat_in_view* | current *satellite_#* (0 if no Satellite in view) |
| 9 | *gwy_quan* | quantity of ORBCOMM Gateways currently connected to this satellite |
| | *gwy_id_0* | list of ORBCOMM Gateway's and the minimum priority of a message acceptable to the respective ORBCOMM Gateway; Messages of lower priority must not be sent until the congestion clears |
| | *min_pri_gwy_0* | |
| | . | |
| | *gwy_id_i-1* | |
| | *min_pri_gwy_i-1* | |
| 10+2i | *queued_ob_msgs* | number of SC-Terminated messages in SC memory |
| 11+2i | *queued_ib_msgs* | number of SC-Originated messages in SC memory |
| 12+2i | *week_byte_0* | UTC time week, with week #0 starting January 6, 1980 (value corresponds to time of next synchronization segment) |
| 13+2i | *week_byte_1* | |
| 14+2i | *time_byte_0* | 24 bit integer representing the number of seconds since 00:00:00 UTC, resets midnight Sunday, time_byte_0 is LSB |
| 15+2i | *time_byte_1* | |
| 16+2i | *time_byte_2* | |
| 17+2i | *total_sats* | total number of satellites in system |
| 18+2i | *stored_sats* | number of stored satellite orbital elements |
| 19+2i | *check_errs* | number of downlink packets with bad checksums since last status packet |
| 20+2i | checksum byte 0 | Fletcher checksum |
| 21+2i | checksum byte 1 | |

**Table 3.5 Status**

## SC-ORIGINATED MESSAGE

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x06 | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum, **MUST NOT EXCEED 8192 bytes** |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | destination ORBCOMM Gateway ID |
| 6 | *polled* | SC-Originated: polled by ORBCOMM Gwy or initiated by SC (see Section 3.2, note 1) |
| 7 | *ack_level* | acknowledgment level (see Section 3.2, note 3) |
| 8 | *priority* | message priority (see Section 3.2, note 5) |
| 9 | *msg_body_type* | message body type (see Section 3.2, note 6) |
| 10 | *mha_ref_num* | DTE assigned, used to identify among multiple messages |
| 11 | *rcpnt_quan* | number of recipients for this message |
| 12 | *subject_ind* | 1=message contains a subject; 0=no subject (see Section 3.2 note 11) |
| | *rcpnt_addr_0 byte 0* | null-terminated recipient address strings and/or recipient indicators (see Section 3.2, notes 10,11) |
| | | |
| | *rcpnt_addr_6 byte i-1* | |
| | *subj byte 0* | null-terminated message subject (see Section 3.2, note 11) (max length = 80 characters) |
| | | |
| | *subj byte j-1* | |
| | *msg body byte 0* | message body (max length determined by SC message queue size and 8 Kb limit) NOTE: for message body types 0 (text) and 15 (externally-defined), a one-byte Data Type is inserted before the actual message body (see Section 3.2, notes 6 and 11) |
| | | |
| | *msg body byte k-1* | |
| | checksum byte 0 | Fletcher checksum |
| 14+i+j+k | checksum byte 1 | |

**NOTE: The total length of the packet must not exceed 8192 bytes**

**Table 3.6 SC-Originated Message**

## SC-ORIGINATED DEFAULT MESSAGE

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x07 | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum, |
| 3 | length byte 1 | **MUST NOT EXCEED 8192 bytes** |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | mha_ref_num | used to uniquely identify messages |
| | *msg body byte 0* | message body |
| | . | |
| | . | |
| | *msg body byte k-1* | |
| | checksum byte 0 | Fletcher checksum |
| 7+k | checksum byte 1 | |

NOTES:

1. The total length of the packet must not exceed 8192 bytes.

2. The active configuration values are used for destination ORBCOMM Gateway (desired_gwy_id), acknowledgment level (def_ack_level), OR indicator (def_or_ind), polled (def_polled), priority (def_priority), and body type (def_msg_body_type).

3. If the def_msg_body_type is 0 or 15, the SC must insert the appropriate single byte data type parameter before msg body byte 0. See the description of def_msg_body_type in the Appendix A.

**Table 3.7 SC-Originated Default Message**

## SC-ORIGINATED REPORT

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x08 | packet type |
| 2 | 0x12 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | destination ORBCOMM Gateway ID |
| 6 | *polled* | SC-Originated: polled by ORBCOMM Gateway or initiated by SC (see Section 3.2, note 1) |
| 7 | *serv_type* | service type (see Section 3.2, note 2) |
| 8 | *or_ind* | originator/recipient indicator, only values 0-3 (see Section 3.2, note 4) |
| 9 | *mha_ref_num* | DTE assigned, used to uniquely identify messages |
| 10 | *user data byte 0* | user data |
| 11 | *user data byte 1* | |
| 12 | *user data byte 2* | |
| 13 | *user data byte 3* | |
| 14 | *user data byte 4* | |
| 15 | *user data byte 5* | |
| 16 | checksum byte 0 | Fletcher checksum |
| 17 | checksum byte 1 | |

**Table 3.8 SC-Originated Report**

## SC-ORIGINATED DEFAULT REPORT

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x09 | packet type |
| 2 | 0x0E | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | mha_ref_num | DTE assigned, used to uniquely identify messages |
| 6 | *user data byte 0* | user data |
| 7 | *user data byte 1* | |
| 8 | *user data byte 2* | |
| 9 | *user data byte 3* | |
| 10 | *user data byte 4* | |
| 11 | *user data byte 5* | |
| 12 | checksum byte 0 | Fletcher checksum |
| 13 | checksum byte 1 | |

NOTE: The active values in the SC configuration are used for Polled (*def_polled*), Service Type (*def_serv_type*), and OR indicator (*def_or_ind*).

**Table 3.9 SC-Originated Default Report**

## SC-ORIGINATED GLOBALGRAM

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x0A | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | destination ORBCOMM Gateway ID |
| 6 | *mha_ref_num* | used to identify among multiple messages, returned in ACK |
| 7 | *or_ind* | originator/recipient indicator (see Section 3.2, note 4) |
| | *user data byte 0* | user data (max = 229 bytes) |
| | *user data byte 1* | |
| | *user data byte 2* | |
| | . | |
| | . | |
| | *user data byte i-1* | |
| 8+i | checksum byte 0 | Fletcher checksum |
| 9+i | checksum byte 1 | |

NOTE: SYSTEM RESPONSE packet is always sent to DTE to indicate success or failure of transmission to Satellite.

**Table 3.10 SC-Originated Globalgram**

## SYSTEM RESPONSE

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x0B | packet type |
| 2 | 0x0F | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *origin* | type of originator of this acknowledgment (see Section 3.2, note 7) |
| 6 | *origin_id* | ID of originator, either *sat_id* or *gwy_id* |
| 7 | *status* | status of message transfer or message enquiry (see Section 3.2, notes 8,15) |
| 8 | *diag_code* | diagnostic code (see Section 3.2, notes 9,15) |
| 9 | *mha_ref_num* | DTE assigned, used to uniquely identify messages |
| 10 | *gwy_ref_num* | ORBCOMM Gateway assigned, used for delivery confirmation (see Section 3.2, note 15) (0xFFFF = not applicable), byte 10 is LSB |
| 11 | *gwy_ref_num* | |
| 12 | *ack_mask* | which recipients apply to this ACK (see Section 3.2, note 12) |
| 13 | checksum byte 0 | Fletcher checksum |
| 14 | checksum byte 1 | |

*origin code* definitions:

- if origin = 0 (satellite), *mha_ref_num* is valid, *gwy_ref_num* is not valid
- if origin = 1 (ORBCOMM Gateway), both *mha_ref_num* and *gwy_ref_num* are valid
- if origin = 2 (recipient), *mha_ref_num* is not valid, *gwy_ref_num* is valid
- if origin = 3 (SC), *mha_ref_num* is valid, *gwy_ref_num* is not valid

**Table 3.11 System Response**

## SC-TERMINATED MESSAGE

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x0C | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | originating ORBCOMM Gateway ID |
| 6 | *subject_ind* | 1 = message contains a subject;<br>0 = no subject (see Section 3.2, note 11) |
| 7 | *msg_body_type* | message body type (see Section 3.2, note 6) |
| 8 | *or_quan* | number of recipients + one originator (see Section 3.2, notes 10,11) |
| | *or_addr_0 byte 0* | originator address or indicator and list of recipient addresses and/or indicators (see Section 3.2, notes 10,11) |
| | . | |
| | . | |
| | *or_addr_7 byte i-1* | |
| | *subj byte 0* | null-terminated message subject<br>(see Section 3.2, note 11) (max length = 80 characters) |
| | . | |
| | . | |
| | *subj byte j-1* | |
| | *msg body byte 0* | message body, max length determined by SC message queue size<br>NOTE: for message body types 0 (text) and 15 (externally-defined), a one-byte Data Type is inserted before the actual message body (see Section 3.2, notes 6 and 11) |
| | . | |
| | . | |
| | *msg body byte k-1* | |
| | checksum byte 0 | Fletcher checksum |
| 10+i+j+k | checksum byte 1 | |

**Table 3.12 SC-Terminated Message**

## SC-TERMINATED USER COMMAND

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x0D | packet type |
| 2 | 0x0D | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | ORBCOMM Gateway ID |
| 6 | *user data byte 0* | user data |
| 7 | *user data byte 1* | |
| 8 | *user data byte 2* | |
| 9 | *user data byte 3* | |
| 10 | *user data byte 4* | |
| 11 | checksum byte 0 | Fletcher checksum |
| 12 | checksum byte 1 | |

**Table 3.13 SC-Terminated User Command**

## SC-TERMINATED GLOBALGRAM

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x0E | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *gwy_id* | ORBCOMM Gateway ID |
| 6 | *gwy_dgram_ref_num* | used to identify among multiple Globalgrams (assigned by ORBCOMM Gateway) |
| 7 | *or_ind* | originator/recipient Indicator (see Section 3.2, note 4) |
| | *user data byte 0* | user data (max length = 182) |
| | *user data byte 1* | |
| | *user data byte 2* | |
| | . | |
| | . | |
| | *user data byte i-1* | |
| 8+i | checksum byte 0 | Fletcher checksum |
| 9+i | checksum byte 1 | |

**Table 3.14 SC-Terminated Globalgram**

## POSITION DETERMINATION COMMAND

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x0F | packet type |
| 2 | 0x08 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/pkt sequence number |
| 5 | *type_code* | type of action requested |
| 6 | checksum byte 0 | Fletcher checksum |
| 7 | checksum byte 1 | |

type_code Definitions:

0 = send SC position status

1 = start process using no initial position information

2 = start process using current position as starting point

3 = stop position determination process

**Table 3.15 Position Determination Command**

## POSITION STATUS

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x10 | packet type |
| 2 | 0x12 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/pkt sequence number |
| 5 | *pos_calc_active* | state of position determination process (1=active, 0=inactive) |
| 6 | *Not specified* | The contents of these bytes is not specified – ignore them. |
| 7 | *Not specified* | |
| 8 | *Not specified* | |
| 9 | *Not specified* | |
| 10 | *lat_code byte 0* | coded geodetic latitude - 0: North Pole, 0x0FFFFFF: South Pole; resolution: 1.2 meters; byte 10 is LSB |
| 11 | *lat_code byte 1* | |
| 12 | *lat_code byte 2* | |
| 13 | *lon_code byte 0* | coded geodetic longitude - 0: Greenwich Meridian, increasing in eastern direction; resolution: 2.4 meters; byte 13 is LSB |
| 14 | *lon_code byte 1* | |
| 15 | *lon_code byte 2* | |
| 16 | checksum byte 0 | Fletcher checksum |
| 17 | checksum byte 1 | |

pos*_calc_active:*

0 = the position determination facility is idle

1 = the position determination facility is currently attempting to estimate SC position

**Table 3.16 Position Status**

## SC-ORIGINATED POSITION REPORT

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x11 | packet type |
| 2 | 0x0e | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *mha_ref_num* | DTE assigned, used to identify among multiple messages |
| 6 | *lat_code byte 0* | coded geodetic latitude - 0: North Pole, 0x0FFFFFF: South Pole; resolution: 1.2 meters, byte 6 is LSB |
| 7 | *lat_code byte 1* | |
| 8 | *lat_code byte 2* | |
| 9 | *lon_code byte 0* | coded geodetic longitude - 0: Greenwich Meridian; increasing in eastern direction; resolution: 2.4 meters, byte 9 is LSB |
| 10 | *lon_code byte 1* | |
| 11 | *lon_code byte 2* | |
| 12 | checksum byte 0 | Fletcher checksum |
| 13 | checksum byte 1 | |

NOTE: The active values in the SC configuration are used for Service Type (*def_serv_type*), and OR indicator (*def_or_ind*). The polling parameter (*def_poll*) is ignored for this report. The report is always sent immediately.

**Table 3.17 SC-Originated Position Report**

## GET PARAMETER

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x12 | packet type |
| 2 | 0x08 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | **r**etry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *parameter_num* | number of parameter whose value should be returned, see Appendix A |
| 6 | checksum byte 0 | Fletcher checksum |
| 7 | checksum byte 1 | |

**Table 3.18 Get Parameter**

## SET PARAMETER

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x85/0x86 | packet header byte |
| 1 | 0x13 | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *parameter_num* | number of parameter whose value should be returned, see Appendix A |
| 6 | *parameter value byte count* | number of parameter value bytes specified (n) |
| 7 | *parameter value byte 0* | one or more hexadecimal bytes indicating the desired value of the specified parameter, parameter value byte 0 is LSB |
| | *parameter value byte 1* | |
| | | |
| 7+n | *parameter value byte n-1* | |
| 8+n | checksum byte 0 | Fletcher checksum |
| 9+n | checksum byte 1 | |

**Table 3.19 Set Parameter**

## PARAMETER RESPONSE

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x14 | packet type |
| 2 | length byte 0 | packet length, starting with byte 0, includes checksum |
| 3 | length byte 1 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *status* | status of get or set parameter request to which this refers (see below) |
| 6 | *parameter_num* | number of parameter referenced, see Appendix A |
| 7 | *parameter value byte count* | number of parameter value bytes returned (n) |
| 8 | *parameter value byte 0* | one or more hexadecimal bytes indicating the value of the specified parameter, parameter value byte 0 is LSB |
| | *parameter value byte 1* | |
| | | |
| 8+n | *parameter value byte n-1* | |
| 9+n | checksum byte 0 | Fletcher checksum |
| 10+n | checksum byte 1 | |

**Table 3.20 Parameter Response**

NOTE: the PARAMETER RESPONSE packet is returned in response to GET and SET PARAMETER packets. If *status* = 0 and byte 7 ≠ 0, (responding to a GET PARAMETER command) the value of byte 7 specifies the number of parameter value bytes returned between byte 7 and the checksum.

*status:*

0: parameter returned (GET PARAMETER) or parameter value changed (SET PARAMETER)

1: specified parameter number not supported

2: access denied (SET PARAMETER)

3: specified parameter number out of range

4: specified value out of bounds (SET PARAMETER)

## SATELLITE STATE VECTOR

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x15 | packet type |
| 2 | 0x17 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *sat_id* | ID of satellite to which this data pertains |
| 6 | *X nibbles 1,0* | 20 bit integer, LSB =8 meters for a 1000 km orbital altitude<br>Coordinates are ECEF. Decoding:<br>tmp = (X nibbles 1,0) + 256*(X nibbles 3,2) + 256*256* (X nibble 4)<br>X = [(2*tmp*MAX_R_SAT)/VAL_20_BITS-MAX_R_SAT]/1000<br>where:MAX_R_SAT = 8378155<br>VAL_20_BITS = 1048576 |
| 7 | *X nibbles 3,2* | |
| 8 | *Y nibble 0 \| X nibble 4* | |
| 9 | *Y nibbles 2,1* | |
| 10 | *Y nibbles 4,3* | |
| 11 | *Z nibbles 1, 0* | |
| 12 | *Z nibbles 3,2* | |
| 13 | *X_DOT nibble 0 \| Z nibble 4* | 20 bit integer, LSB =0.0143 meters/sec for a 750 km orbital altitude<br>Decoding:<br>tmp = (X_DOT nibbles 1,0) + 256*(X_DOT nibbles 3,2) + 256*256* (X_DOT nibble 4)<br>X_DOT = [(2*tmp*MAX_V_SAT)/VAL_20_BITS-MAX_R_SAT]/1000<br>where:MAX_V_SAT = 7700 |
| 14 | *X_DOT nibbles 2,1* | |
| 15 | *X_DOT nibbles 4,3* | |
| 16 | *Y_DOT nibbles 1,0* | |
| 17 | *Y_DOT nibbles 3,2* | |
| 18 | *Z_DOT nib 0 \| Y_DOT nib 4* | |
| 19 | *Z_DOT nibbles 2,1* | |
| 20 | *Z_DOT nibbles 4,3* | |
| 21 | checksum byte 0 | Fletcher checksum |
| 22 | checksum byte 1 | |

Sent to DTE when *send_pass_predict* = 1

**Table 3.21 Satellite State Vector**

## SATELLITE ORBITAL ELEMENTS

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x16 | packet type |
| 2 | 0x10 | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | retry_count/pkt_seq_num | number of times this packet has been re-sent/packet sequence number |
| 5 | *plane_id* | Identifies the satellite plane |
| 6 | *sat_id* | ID of satellite to which this data pertains |
| 7 | *mean_anom 0* | Mean Anomaly for this satellite<br>(other elements taken from Plane Orbital Elements set)<br>0 = 0 degrees, 0xFFFFFF = 360 degrees<br>Mean Anomaly = {[mean_anom0 + mean_anom1*0xFF + mean_anom2*0xFFFF + mean_anom3*0xFFFFFF]/0xFFFFFFFF*2*pi} |
| 8 | *mean_anom 1* | |
| 9 | *mean_anom 2* | |
| 10 | *mean_motion 0* | (mean_motion - MIN_MOTION)/<br>MAX_MOTION *0xFFFFFF |
| 11 | *mean_motion 1* | |
| 12 | *mean_motion 2* | |
| 13 | *mean_motion 3* | |
| 14 | checksum byte 0 | Fletcher checksum |
| 15 | checksum byte 1 | |

Sent to DTE when *send_pass_predict* = 1

**Table 3.22 Satellite Orbital Elements**

## SATELLITE PLANE ORBITAL ELEMENTS

| | 7 6 5 4 3 2 1 0 | |
|---|---|---|
| 0 | 0x05/0x06 | packet header byte |
| 1 | 0x17 | packet type |
| 2 | 0x1C | packet length, starting with byte 0, includes checksum |
| 3 | 0x00 | |
| 4 | *retry_count/pkt_seq_num* | number of times this packet has been re-sent/packet sequence number |
| 5 | *plane_id* | Identifies satellite plane |
| 6 | *epoch byte 0* | epoch time (milliseconds since 1/1/93, good for 35 years) |
| 7 | *epoch byte 1* | |
| 8 | *epoch byte 2* | |
| 9 | *epoch byte 3* | |
| 10 | *epoch byte 4* | |
| 11 | *inclin byte 0* | resolution: 1/26 second<br>0->0x7FFFFF: prograde, 0x7FFFFF->0xFFFFFF, retrograde, increasing towards -90 (polar = 0x7FFFFF)<br>inclination = {[inclin 0 + 0xFF*inclin 1 + 0xFFFFFF*inclin 2)]/ 0xFFFFFF}* pi |
| 12 | *inclin byte 1* | |
| 13 | *inclin byte 2* | |
| 14 | *raan byte 0* | Right ascension of ascending node, res = 1/13 second<br>0= 0 degrees, 0xFFFFFF = 360 degrees<br>raan ={[raan0 + 0xFF*raan1 + 0xFFFF*raan2]/0xFFFFF}*pi |
| 15 | *raan byte 1* | |
| 16 | *raan byte 2* | |
| 17 | *eccent byte 0* | Eccentricity = [eccen 0 + eccen1*0xFF)*0xFFFF]*0.1 |
| 18 | *eccent byte 1* | |
| 19 | *perigee byte 0* | Argument of Perigee |
| 20 | *perigee byte 1* | |
| 21 | *sat_quan* | Total number of satellites in the system |
| 22 | *decay byte 0* | decay = (decay0 + 0xFF*decay1)/0xFFFF revs/day |
| 23 | *decay byte 1* | |
| 24 | *orbit_num byte 0* | Number of complete orbits since epoch |
| 25 | *orbit_num byte 1* | |
| 26 | checksum byte 0 | Fletcher checksum |
| 27 | checksum byte 1 | |

Sent to DTE when *send_pass_predict* = 1

**Table 3.23 Satellite Plane Orbital Elements**

# 3.2 Packet Description Notes

**1. Polling of SC-Originated Messages/Reports**

0 = SC initiates transfer immediately

1 = Messages/reports are queued in SC, awaiting poll from ORBCOMM Gateway

**2. Service Type (a combination of ACK level and priority for space efficiency)**

NOTE**:** While the service types preceded by an asterisk are not currently implemented by the ORBCOMM Network, they are reserved for future use and must be supported by the SC.

0 = Normal priority, no acknowledgment expected

1 = Normal priority, only non-delivery to ORBCOMM Gateway acknowledged - acknowledgement generated by SC based on lack of communication with satellite or ORBCOMM Message Switch (OMS)

2 = Normal priority, delivery to ORBCOMM Gateway acknowledged

3 = Normal priority, only non-delivery to recipient acknowledged (from X.400 MTA, via Internet, acknowledgment comes from Simple Mail Transfer Protocol (SMTP) Gateway)

4 = Normal priority, delivery to recipient acknowledged

5 = Same as 0, but indicates report sent in response to poll from ORBCOMM Gateway

6 = Same as 1, but indicates report sent in response to poll from ORBCOMM Gateway

7 = Same as 2, but indicates report sent in response to poll from ORBCOMM Gateway

*8 = Same as 3, but indicates report sent in response to poll from ORBCOMM Gateway

9 = Same as 4, but indicates report sent in response to poll from ORBCOMM Gateway

10 = Special delivery priority, no acknowledgment expected

*11 = Special delivery priority, only non-delivery to ORBCOMM Gateway acknowledged

*12 = Special delivery priority, delivery to ORBCOMM Gateway acknowledged

*13 = Special delivery priority, only non-delivery to recipient acknowledged

*14 = Special delivery priority, delivery to recipient acknowledged

*15 = Cancel previous special delivery report (acknowledgment of delivery or non-delivery to recipient)

**3. Acknowledgment level**

0 = No acknowledgment expected

1 = Only non-delivery to ORBCOMM Gateway acknowledged

2 = Delivery to ORBCOMM Gateway acknowledged

3 = Only non-delivery to recipient acknowledged

4 = Delivery to recipient acknowledged

4. **Valid Originator/Recipient Indicator Values**

0 = For SC-Terminated: Null address; For SC-Originated: self (loop back test).

1-8 = User pre-defined in ORBCOMM Gateway database

9-15 = System pre-defined in ORBCOMM Gateway database

NOTE: Reports support only OR Indicator values 0-3.

5. **Priority Level**

NOTE**:** Due to the low level of loading of the network, there is currently no effective difference in delivery as a function of priority.

0 = Non-urgent (lowest priority)

1 = Normal

2 = Urgent

3 = Special delivery (SC-Originated only, highest priority)

6. **Message body type (1988 X.400 specification)**

Most of the X.400 defined body types defined on the next page have two components: data type parameters and the data itself. The data type parameters comprise a sequence of information items that describe what the information object in the data component represents. They are typically format and control descriptors. The data component is the information object itself. The data type parameters precede the data within the user data field of the message packet (bytes 13 through the Fletcher checksum), and generally consist of a single byte each.

The Externally Defined message body type enables X.400 service providers to carry unique and unusual message types. This category contains the body types unique to the ORBCOMM System.

NOTE: While the message body types preceded by an asterisk are not currently implemented by the ORBCOMM network, they are reserved for future use and must be supported by the SC.

0 = IA5TEXT (one single byte data type parameter: 5 = ia5; 2 = ita2 (telex) character set in message body.

*1 = not used (formerly telex)

*2 = Voice (under study by CCITT)

*3 = G3 Facsimile

*4 = G4 Class 1 Facsimile

*5 = Teletex

*6 = Videotex

*7 = Nationally Defined

*8 = Encrypted (under study by CCITT)

*9 = Message (a message inside of a message)

*10 = not used (formerly "simple formattable text")

*11 = Mixed Mode

*12 = not used

*13 = not used

14 = Bilaterally Defined (binary data)

15 = Externally Defined (one single byte data type parameter) as defined below:

**Data Type Parameter byte:**

0 = Text-to-Facsimile (SC-Originated Only). The initial portion of the body type contains a null-terminated telephone number. The next portion contains a null-terminated character string which is presented on the facsimile cover page immediately following the "ATTENTION:" header. (If nothing is desired in this field, then only a null must be used as a place holder). The remaining portion of the message body becomes the body of the facsimile.

1 = Text-to-Voice (SC-Originated Only). The initial portion of the body type contains a null-terminated telephone number. The next portion contains a null-terminated character string which is the name of the individual to be contacted. The remaining portion of the message body is read over the phone by customer service operators.

2 = Pre-defined OR Addresses (For the SC-Originated, the addresses for OR indicators 1-8 are placed in order, null-terminated, into an SC-Originated message. For the SC-

Terminated, the addresses for OR indicators 1-15 are placed in order, null- terminated, into an SC-Terminated message).

3 = Date, Originator, and Subject for each SC-Terminated message queued in the ORBCOMM Gateway for this SC. Each attribute (originator, date, and subject) of the message is null-terminated. Non-existent attributes are identified by a single null.

4 = Sets of orbital elements in NASA 2-line format (as specified in Appendix H).

5 = IA5 data (see below)

| | --Control Codes-- | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | Null | TC7 (DLE) | Space | 0 | @ | P | ‘ | p |
| 1 | TC1 (SOH) | DC1 | ! | 1 | A | Q | a | q |
| 2 | TC2 | DC2 | “ | 2 | B | R | b | r |
| 3 | TC3 (ETX) | DC3 | # | 3 | C | S | c | s |
| 4 | TC4 (EOT) | DC4 | ($) | 4 | D | T | d | t |
| 5 | TC5 (ENQ) | TC8 (NAK) | % | 5 | E | U | e | u |
| 6 | TC6 (ACK) | TC9 (SYN) | & | 6 | F | V | f | v |
| 7 | BELL | TC10 (ETB) | ‘ | 7 | G | W | g | w |
| 8 | FE0 (BS) | CAN | ( | 8 | H | X | h | x |
| 9 | FE1 (HT) | EM | ) | 9 | I | Y | i | y |
| A | FE2 (LF) | SUB | * | : | J | Z | j | z |
| B | FE3 (VT) | ESC | + | ; | K | [ | k | { |
| C | FE4 (FF) | IS4 (FS) | , | < | L | \ | l | \| |
| D | FE5 (CR) | IS3 (GS) | - | = | M | ] | m | } |
| E | SO | IS2 (RS) | . | > | N | ^ | n | ~ |
| F | SI | IS1 (US) | / | ? | O | _ | o | DEL |

**Table 3.24 International Alphabet Number 5 (IA5)**

Notes:

IA5 is defined for values 0x00 through 0x7F. If a message or Globalgram is sent as IA5, each character value must be in this range. Behavior outside this range (i.e. 0x7F to 0xFF) must be determined. The names of some of the control characters between IA5 and ASCII are different. The ASCII name is shown in parentheses for these cases.

Hex 0x42 is the international currency symbol. The ASCII equivalent is the "$".

7. **Message/Report/Globalgram Acknowledgment Origin**

   0 = Satellite (Globalgram only).

   1 = ORBCOMM Gateway (SC-Originated message and report only).

   2 = recipient (SC-Originated message and report only).

   3 = SC (abnormal conditions)

8. **Status of message transfer (1988 X.400 specification + ORBCOMM)**

   0 = transfer failure

   1 = unable to transfer

   2 = conversion not performed

   3 = reserved for future use

   4 = reserved for future use

   5 = reserved for future use

   6 = reserved for future use

   7 = reserved for future use

   8 = reserved for future use

   9 = reserved for future use

   10 = transfer attempt failed but message has been re-queued and will be attempted again

   11 = response to COMMUNICATIONS COMMAND

   12 = status unknown

   13 = no acknowledgment from recipient at this time - please wait

   14 = message aborted by user

   15 = message was received by indicated acknowledgment originator

9. **Message transfer diagnostic codes (1988 X.400 specification + ORBCOMM)**

0 = unrecognized originator/recipient name

1 = ambiguous originator/recipient name

2 = MTA congestion

3 = loop detected

4 = recipient unavailable

5 = transfer time-out

6 = body type not supported

7 = content too long

8 = convert impractical

9 = convert prohibited

10 = convert not registered

11 = invalid parameter

12-99 = reserved for future use

100 = Message-level checksum failure

101 = SC ID not registered

102 = PIN code not valid

103 = requested ORBCOMM Gateway could not be found in any Satellite downlink

104 = insufficient message priority (ORBCOMM Gateway may be congested)

105 = satellite not responding (uplink may be congested)

106 = SC access restriction

107 = SC registration has expired

108 = SC-Originated message already exists in ORBCOMM Gateway

109 = no active session, or SC-Originated message number error

110 = error occurred in ORBCOMM Gateway while saving message to non-volatile memory

111 = database error occurred in ORBCOMM Gateway

112 = no additional diagnostic information indicated

113 = maximum retries attempted

114 = Globalgram is not permitted at this time

115 = no Satellite in view at this time

116 = position report currently not available, but starting calculation

117 = no position determination capability

118 = Globalgram size exceeded

119 = no SC-Terminated messages/commands queued in ORBCOMM Gateway

120 = no SC-Terminated messages/commands queued in ORBCOMM Gateway of size less than 150 bytes

121 = no Globalgrams queued in current satellite

122 = requested message deleted

123 = no stored Satellite orbital elements

124 = registration request received, please wait

125 = registration granted by ORBCOMM Gateway

126 = registration rejected by ORBCOMM Gateway

127 = maximum number of Globalgrams (16) stored in current satellite

128 = invalid range of one or more fields

129 = ready for power removal

130-140 = reserved for future use

141 and above = reserved for generic function diagnostic codes (see note 15)

10. **Parsing and Filling of Originator/Recipient Addresses**

The recipients of an SC-Originated message, and the originator and recipients of an SC-Terminated message are carried by the packet according to the following rules:

a. An OR address must be an ASCII character string which adheres to the X.400 standard syntax. A 0h, 80h, or 90h follows the last character of the string to indicate that the O/ R address identifies a primary, blind, or copied recipient, respectively.

b. An OR indicator must be a number between 0 and 15, occupying one byte in the packet. The upper nibble is set to zero, 8, or 9 to indicate that the OR address identifies a primary, blind, or copied recipient, respectively.

c. OR addresses and OR indicators are placed sequentially in the packet, without the use of length fields.

d. For the SC-Terminated message, the first OR address or indicator identifies the originator. (An OR indicator value of 0 indicates that the originator address is not being sent. This case occurs when the originator cannot be matched to any of the 14 originator addresses, but the SC user has specified that he/she does not want the X.400 address of the originator or the copied recipients transmitted with any message.

The remaining addresses or indicators are the copied recipients (up to a total of 6 in this case) Note that there is never a blind recipient filled into an SC-Terminated message.

e. The maximum OR address length is 128 characters (null included).

f. The maximum quantity of OR addresses and/or OR indicators per message is seven.

g. An SC-Originated message must have at least one OR address or OR indicator, however an SC-Terminated message may have none.

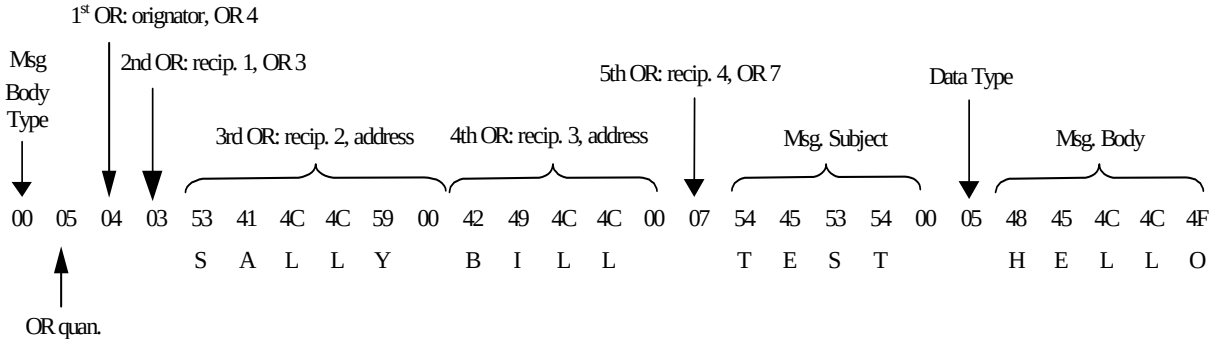
h. For the SC-Originated message, an OR indicator of 0 indicates a request for a message loopback; the ORBCOMM Gateway sends the received message back to the SC. If the SC-Originated message is a report, the first five bytes of the report are placed into a User Command (with no acknowledgment expected).

11. **Additional Contents within an SC-Originated or SC-Terminated Message**

SC-Terminated and SC-Originated Messages may contain a body type parameter, a subject and up to seven OR indicators and/or addresses, in addition to the normal message body. If specified, the body type parameter is inserted before any other field, followed by the OR indicators and/or addresses, followed by the subject (a null-terminated ASCII string), followed by the actual message body.

In the following example of an SC-Terminated Message, the text message (message body type = 0 and data type parameter = 5) has a subject "TEST", a message body "HELLO", an originator revealed by OR indicator 4, and four primary recipients identified by OR indicator 3, null-terminated address "SALLY", null-terminated address "BILL", and OR indicator 7. These bytes would be present in the SC-Terminated Message, carried in one or more packets.

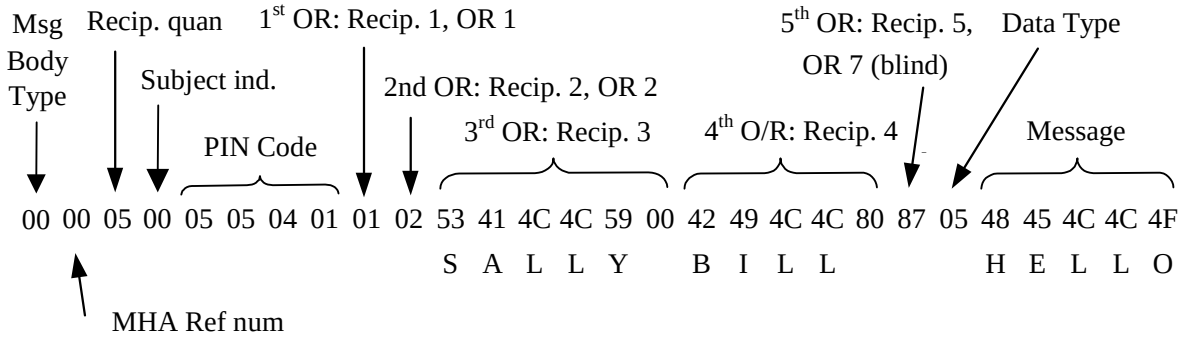
Hex and Text representation:



An SC-Originated Message may contain a body type parameter, a PIN code, a subject and up to seven OR indicators and/or addresses, in addition to the normal message body. The body type parameter is inserted first, followed by the four-byte PIN code, and the OR indicators and/or the OR addresses. This string is followed by the subject (a null- terminated ASCII string), which in turn, is followed by the actual message body. The ORBCOMM Gateway can request that the SC supply its PIN code via the control field of the SC-ORIGINATED ASSIGNMENT packet (see Note 13). The presence of a subject and the quantity of OR indicators and/or addresses is revealed by the REQUEST FOR SC-ORIGINATED MESSAGE TRANSMISSION contents field. The message body type also is revealed in the REQUEST FOR SC-ORIGINATED MESSAGE TRANSMISSION packet. If the type specified in byte 12 implies a data type parameter within the message itself, the ORBCOMM Gateway will expect this value as the first byte of the user data (bytes 13 through the Fletcher checksum) of the SC-ORIGINATED MESSAGE packet.

In the following example of an SC-Originated message, the message includes a PIN code, a message body "HELLO", five recipients identified by OR indicator 1 (primary), OR indicator 2 (primary), null-terminated OR address "SALLY" (primary), null-terminated OR address "BILL" (blind), and OR indicator 7 (blind). These bytes would be present in the SC-ORIGINATED MESSAGE packet user data fields (bytes 13 through the Fletcher checksum), carried in one or more packets.

Hex and text representation:




12. **Acknowledgment of SC-Originated Messages**

The SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet is transmitted to the SC after the ORBCOMM Gateway has received indication of a successful or unsuccessful transfer for each recipient. It is used for both SC-Originated messages and SC-Originated reports, but not for Globalgrams. The ORBCOMM Gateway sets bits in the ***ack_mask*** field to indicate to which of the original message recipients (up to seven) this acknowledgment applies. Each bit corresponds to one of the OR indicators and/or OR address of the original message, starting with bit 0 (SC-ORIGINATED REPORT will always use bit 0). Multiple SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packets may be sent to an SC. For example, if a SUBSCRIBER MESSAGE ENQUIRY is received at the ORBCOMM Gateway for a message that had

seven recipients, but only four acknowledgments have been received, the ***ack_mask*** will have the appropriate four bits set, and ***status_code*** = 15. The remaining three acknowledgments will be indicated later by subsequent SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet(s), whenever the acknowledgment or negative acknowledgment is received from the recipients at the ORBCOMM Gateway.

Since notification of message delivery failure may take up to five days from the Internet, the SC should not wait for the SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet after sending the message to the ORBCOMM Gateway. If it is not received after some unacceptable period, the user (or MHA) may generate a **COMMUNICATIONS COMMAND**, with ***type_code*** = 5. The SC will then generate a SUBSCRIBER MESSAGE ENQUIRY packet with ***control_code*** = 5. If the ***status_code*** of the SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet is not equal to 15, then the SC must generate a SUBSCRIBER ACKNOWLEDGMENT packet. The SC must not send a SUBSCRIBER ACKNOWLEDGMENT if the ***status_code*** = 15.

The SESSION CLEAR and REPORT ACKNOWLEDGMENT packets only indicate successful transmission to the ORBCOMM Gateway, for SC-Originated messages and SC-Originated reports, respectively. They do not imply successful transmission to the recipient(s). The Satellite transmits an SC-ORIGINATED GLOBALGRAM ACKNOWLEDGMENT packet upon receiving a Globalgram from the SC. There is no packet that explicitly indicates that the Globalgram has reached the ORBCOMM Gateway. The recipient of the Globalgram should generate an SC-Terminated Globalgram to indicate successful reception of the SC-Originated Globalgram, if that is desired.

In detail, if the SC-ORIGINATED REPORT is sent using ***serv_type*** = 3, 4, 8, 9, 13, or 14, then the ORBCOMM Gateway will return a two-byte reference number assigned by the ORBCOMM Gateway to the SC using the REPORT ACKNOWLEDGMENT packet. The SC then sends a **SYSTEM RESPONSE** packet to the DTE, with the ***originator_code*** = 1 (ORBCOMM Gateway), and the ***gwy_ref_num*** specifying the reference number assigned by the ORBCOMM Gateway assigned reference number. The DTE then knows that the message has reached the ORBCOMM Gateway and is now on its way to the recipient.

Once the ORBCOMM Gateway receives indication of the recipient's reception of the report, it sends an SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet to the SC, which in turn sends another **SYSTEM RESPONSE** packet to the DTE (with ***originator_code*** = 2). If the recipient did not get the report, the ORBCOMM Gateway sends the SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet with the appropriate status and diagnostic code. The ORBCOMM Gateway sends the first SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet when indications of reception or failed reception have been received from all recipients. If the SC receive it, the DTE user can choose to send a **COMMUNICATIONS COMMAND**, with ***type_code*** = 5, and with the ***gwy_ref_num*** valid (***mha_ref_num*** is no longer used once the message has been acknowledged by the ORBCOMM Gateway). The SC then sends a SUBSCRIBER MESSAGE ENQUIRY to the ORBCOMM Gateway, which then sends the SC-

ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT packet. (The state of all pending and failed message transfers is recorded in the ORBCOMM Gateway.)

If an SC-Originated message is sent using ***ack_level*** = 3 or 4, the same sequence of packet transmissions occurs as described above for the report. However, if multiple recipients are included in the message, then the ***ack_mask*** field of the SC-ORIGINATED MESSAGE RECIPIENT ACKNOWLEDGMENT identifies the recipients in which the status (and possibly diagnostic, if there is a transfer failure) codes apply.

13. **PIN Code Handling**

The PIN code is an extra measure of protection against theft of services using a stolen or cloned SC. It is not normally required for message transmission. The following paragraphs describe its use in the SC-Originated and SC-Terminated directions.

In the SC-Terminated direction, the SUBSCRIBER RECEIVER READY packet sent to the ORBCOMM Gateway from the SC must contain the valid PIN code. The PIN code is stored in the SC but can be updated from the DTE or the application task. If the PIN code does not match the value stored in the ORBCOMM Gateway, the SC receives a SESSION CLEAR packet with the appropriate diagnostic and status codes.

In the SC-Originated direction, the ***control_code*** of the SC-ORIGINATED ASSIGNMENT packet specifies the inclusion of the PIN code in the message, if so provisioned. The SC must then insert the four PIN code bytes according to the details provided in Note 11.

14. **Serial Interface/API Communications Command Handling**

The response to the COMMUNICATIONS COMMAND varies depending on the requested action. Each response is described in the table starting on the next page. (When referring to a particular value of a diagnostic code, 'X' represents any of the values described in Note 9).

| Type Code | Desired Response<br>[serial pkt(s) sent to DTE, other than **LINK LEVEL ACK** which is always sent immediately] | Abnormal Response<br>[serial pkt(s) sent to DTE, other than **LINK LEVEL ACK** which is always sent immediately] |
|---|---|---|
| 0 | **SC-TERMINATED MESSAGE(S)** or **SYSTEM RESPONSE**<br>**(*origin* = 1, *status* = 11, *diagnostic* = 119)** | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** < > 119) |
| 1 | **SC-TERMINATED MESSAGE**(S) or **SYSTEM RESPONSE**<br>(***origin*** = 1, ***status*** = 11, ***diagnostic*** = 120) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** < > 120) |
| 2 | **SC-TERMINATED GLOBALGRAM**(s) or **SYSTEM RESPONSE**<br>(***origin*** = 0, ***status*** = 11, ***diagnostic*** = 121) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** < > 121) |
| 3 | **SC-TERMINATED MESSAGE**<br>containing OR addresses (see note 6) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = X) |
| 4 | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 15, ***diagnostic*** = 112) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 12, ***diagnostic*** = 112 or 109) |
| 5 | **SYSTEM RESPONSE**<br>(***origin*** = 2, ***status*** = 15, ***diagnostic*** = 112)<br>(also see note 12) | **SYSTEM RESPONSE**<br>(***origin*** = 2, ***status*** = 11, ***diagnostic*** = X) |
| 6 | **SC-TERMINATED MESSAGE**<br>containing list of originator, time, and subject for each message (see note 6) or<br>**SYSTEM RESPONSE**<br>(***origin*** = 1, ***status*** = 11, ***diagnostic*** = 119) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** < > 119) |
| 7 | one **SC-TERMINATED MESSAGE** | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = X) |
| 8 | **SYSTEM RESPONSE**<br>(***origin*** = 1, ***status*** = 11, ***diagnostic*** = 122) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** < > 122) |
| 9 | **SYSTEM RESPONSE**<br>(***origin*** = 1, ***status*** = 11, ***diagnostic*** = 124 or 125 or 126) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = X) |
| 10-15 | **SYSTEM RESPONSE**<br>(***origin*** = 1, see note 15 for other fields) | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = X) |
| 16 | **STATUS** | Not Applicable |

| Type Code | Desired Response<br>[serial pkt(s) sent to DTE, other than **LINK LEVEL ACK** which is always sent immediately] | Abnormal Response<br>[serial pkt(s) sent to DTE, other than **LINK LEVEL ACK** which is always sent immediately] |
|---|---|---|
| 17 | No packet sent.<br>DTE can request **STATUS** if desired. | If **STATUS** requested, state = 0 means no additional message being transmitted or received. Otherwise, new message is being transmitted or received |
| 18 | | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = 109) |
| 19 | | Not Applicable |
| 20 | | Not Applicable |
| 21 | **STATUS** (***state*** = 0, ***st_diag_code*** = 0) | **STATUS** with ***st_diag_code*** reveals failure |
| 22 | **STATUS** (***state*** = 0, ***st_diag_code*** = 0) | **STATUS** with ***st_diag_code*** reveals failure |
| 23 | **STATUS** (***state*** = 0, ***st_diag_code*** = 0) | **STATUS** with ***st_diag_code*** reveals failure |
| 24 | No packet sent. SC powers down immediately. | Not Applicable |
| 25 | | |
| 26 | | |
| 27 | If ***def_serv_type*** is not 0 or 10, SC sends **SYSTEM RESPONSE** indicating success according to description in note 12 | If ***def_serv_type*** is not 0 or 10, SC sends **SYSTEM RESPONSE** indicating failure according to description in note 12 |
| 28 | **SC-TERMINATED MESSAGE**<br>with Orbital Elements, message body type 15,4 | **SYSTEM RESPONSE**<br>(***origin*** = 1, ***status*** = 11, ***diagnostic*** = 123) |
| 29 | No packet sent. Change to the next down link channel | Not Applicable |
| 30 | **SYSTEM RESPONSE**<br>(***origin*** = 3, ***status*** = 11, ***diagnostic*** = 129) | Not Applicable |

**Table 3.25 Communication Command Responses to Type Code Inputs**

15. **Preservation of Future System Capabilities using Generic Functions**

Unused values of some fields of the **COMMUNICATIONS COMMAND**, **SYSTEM ANNOUNCEMENT**, and **SYSTEM RESPONSE** packets have been specified as generic functions in order to enable the SC to support future system capabilities without code modification. This will hopefully enable the support of applications not envisioned at this time. The packets involved include the following:

POLLING COMMAND (ORBCOMM Gateway to SC)

**SYSTEM ANNOUNCEMENT** (SC to DTE)

SUBSCRIBER MESSAGE ENQUIRY (SC to ORBCOMM Gateway)

**COMMUNICATIONS COMMAND** (DTE to SC)

SYSTEM COMMAND (ORBCOMM Gateway to SC)

**SYSTEM RESPONSE** (SC to DTE)

a. POLLING COMMAND & **SYSTEM ANNOUNCEMENT**

b. POLLING COMMAND *control_codes* and **SYSTEM ANNOUNCEMENT** codes not yet specified include codes 9-15 (future system use) and 16 and above (SC manufacturer defined). If the SC receives a **POLLING COMMAND** with these values, it performs the following functions:

   i. Copies the POLLING COMMAND control field into the **SYSTEM ANNOUNCEMENT** *announce_code* field.
   ii. Copies the POLLING COMMAND *gwy_id* field into the **SYSTEM ANNOUNCEMENT** *gwy_id* field.
   iii. Copies the POLLING COMMAND *dest_or_ind* field into the **SYSTEM ANNOUNCEMENT** *dest_or_ind* field.
   iv. Sends the **SYSTEM ANNOUNCEMENT** to the DTE/MHA.
   v. Sends a SUBSCRIBER ACKNOWLEDGMENT packet to the ORBCOMM Gateway

c. SUBSCRIBER MESSAGE ENQUIRY & **COMMUNICATIONS COMMAND**

SUBSCRIBER MESSAGE ENQUIRY *control_codes* and **COMMUNICATIONS COMMAND** *type_codes* not yet specified include codes 10-15. If the SC receives a **COMMUNICATIONS COMMAND** with these values, it performs the following functions:

   i. Copies the **COMMUNICATIONS COMMAND** *type_code* field into the SUBSCRIBER MESSAGE ENQUIRY *control_code* field.
   ii. Copies the **COMMUNICATIONS COMMAND** *gwy_id* field into the SUBSCRIBER MESSAGE ENQUIRY *gwy_id* field.
   iii. Copies the **COMMUNICATIONS COMMAND** *value* field (byte 0 and byte 1) into the SUBSCRIBER MESSAGE ENQUIRY message reference field.
   iv. Copies the PIN value into the SUBSCRIBER MESSAGE ENQUIRY PIN field.
   v. Repeatedly sends the SUBSCRIBER MESSAGE ENQUIRY packet to the designated ORBCOMM Gateway until maximum retries are exceeded (determined from the value of *ul_max_retries* in the Read/Write Configuration Parameters) or until a SYSTEM COMMAND is received from the designated ORBCOMM Gateway having a *control_code* value of 32. The SC then sends a **SYSTEM RESPONSE** to the DTE/ MHA according to the procedure listed below under SYSTEM COMMAND and **SYSTEM RESPONSE**.

d. SYSTEM COMMAND & **SYSTEM RESPONSE**

SYSTEM COMMAND *control_code* = 32 indicates a Generic Function. If the SC receives a SYSTEM COMMAND with this value, it performs the following functions:

i. Copies byte 6 of the SYSTEM COMMAND into the **SYSTEM RESPONSE** status field.

ii. Copies byte 7 of the SYSTEM COMMAND into the **SYSTEM RESPONSE** *diag_code* field.

iii. Copies bytes 8 and 9 of the SYSTEM COMMAND into the **SYSTEM RESPONSE** *gwy_ref_num* field (bytes 0 and 1 respectively).

iv. Sets **SYSTEM RESPONSE** *origin_field* = 1 (ORBCOMM Gateway).

v. Copies the SYSTEM COMMAND *gwy_id* field into the **SYSTEM RESPONSE** *origin_id* field.

vi. Sets **SYSTEM RESPONSE** *mha_ref_num* = 0xFF and *ack_mask* = 0.

vii. Sends the **SYSTEM RESPONSE** packet to the DTE/MHA.

viii. Sends SUBSCRIBER ACKNOWLEDGMENT PACKET to the ORBCOMM Gateway.

# Chapter 4 SERIAL INTERFACE PACKET EXAMPLES

Consistent with the expression that “one picture is worth a thousand words,” this Chapter presents ladder diagrams illustrating how each of the serial interface packets is used. While the resulting interaction between the SC and the Satellite, and between the Satellite and the GCC are arguably outside the scope of this SC Serial Interface Specification, the context this information provides is invaluable, particularly to the reader newly introduced to the ORBCOMM system. Consequently, the ladder diagrams in this Chapter include packets from the Air Interface Specification and also illustrate the Satellite’s interaction with the GCC.

NOTE: In the following ladder diagrams, the vertical axis (time) is not to scale.

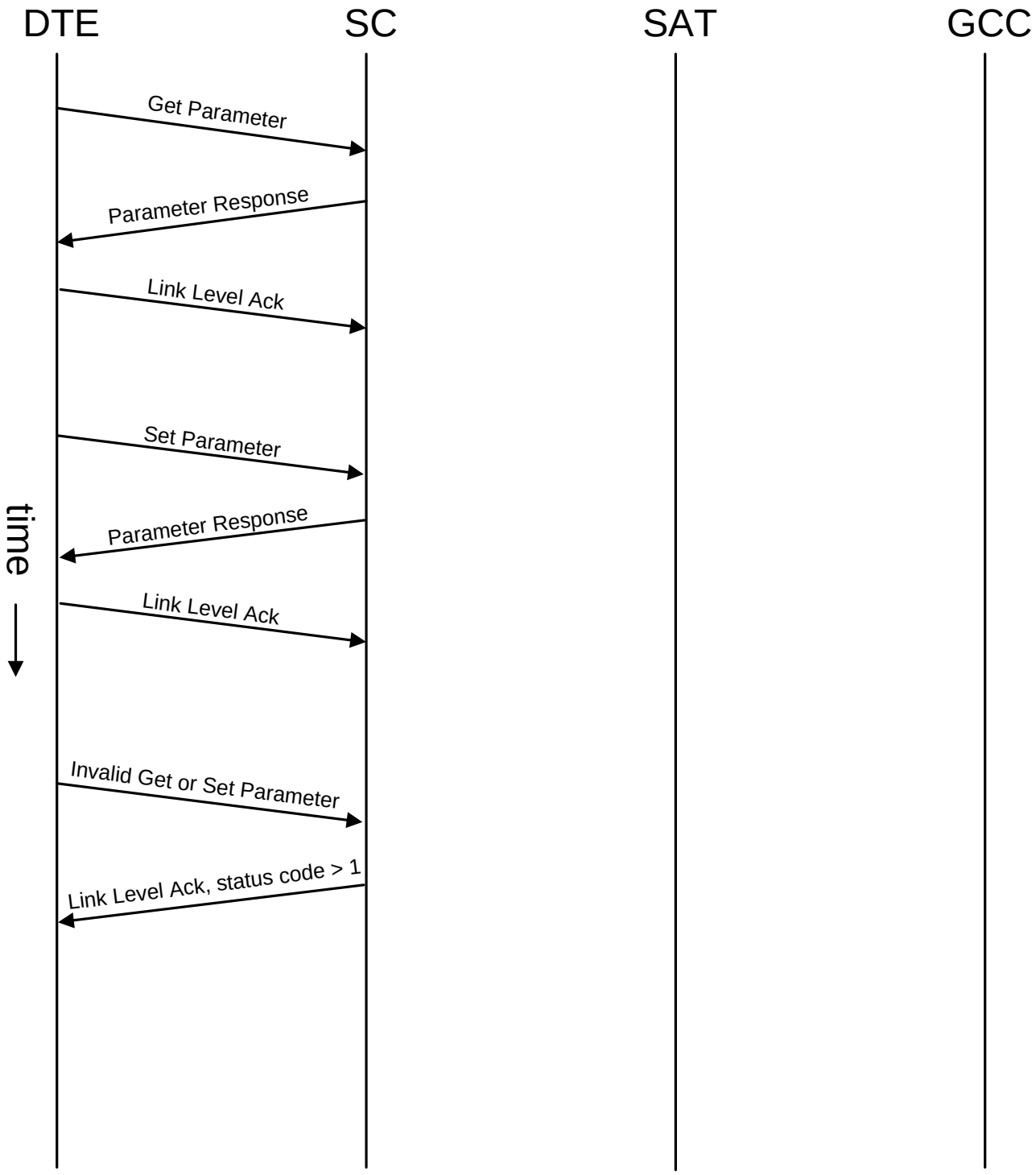



**Figure 4.1 Get and Set Parameter**

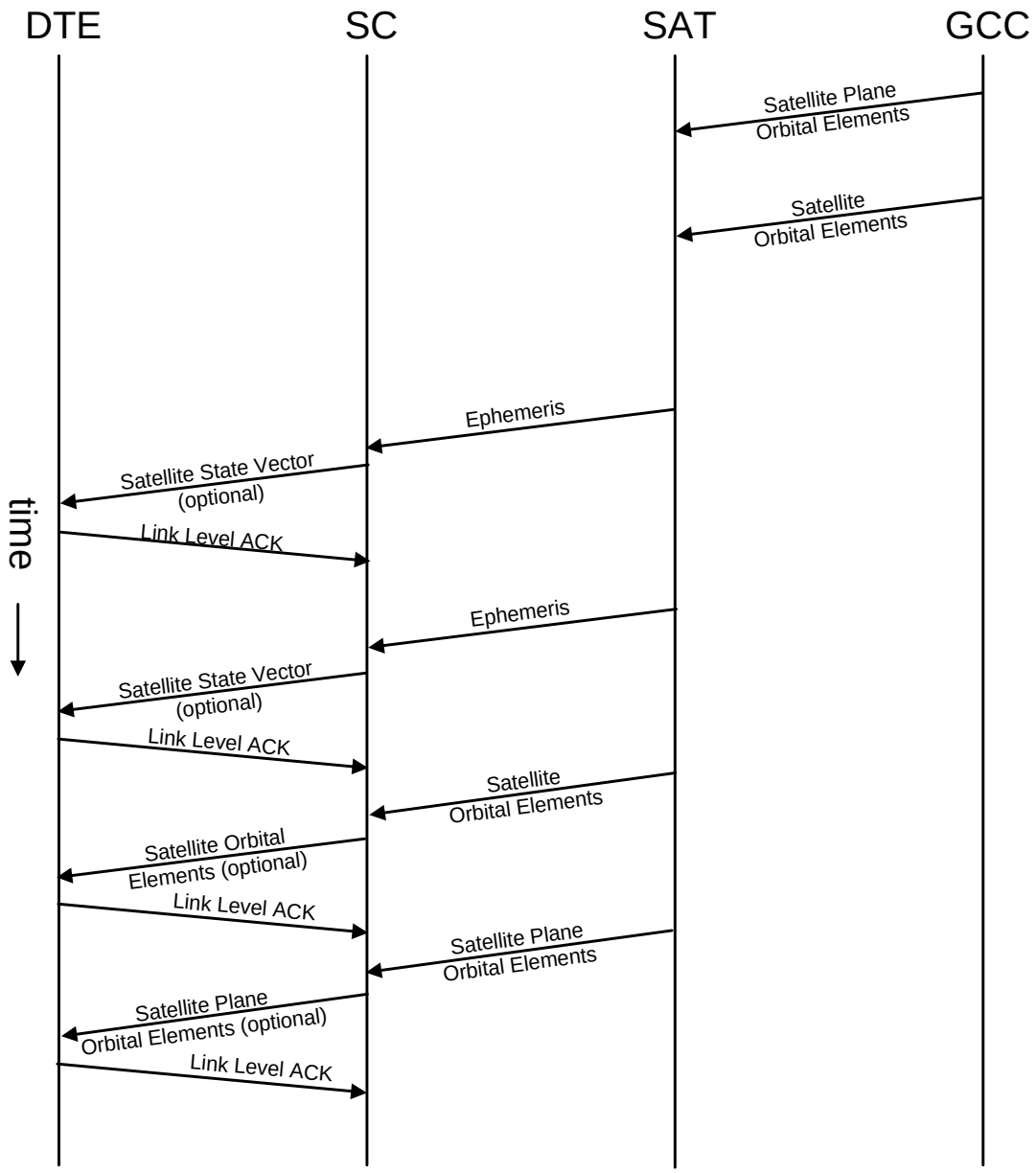



**Figure 4.2 Satellite Pass Prediction Data**

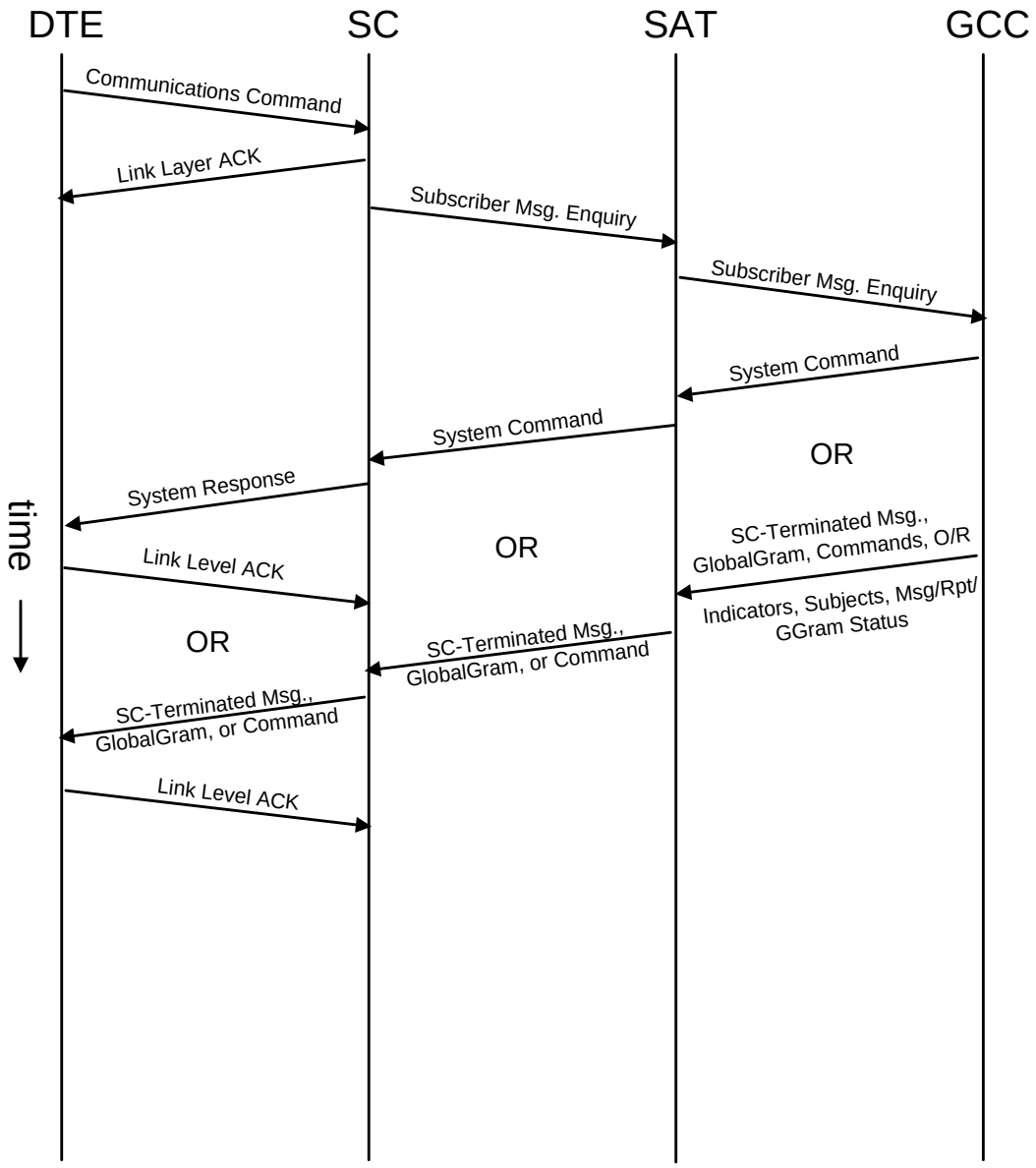



**Figure 4.3 Communications Command**

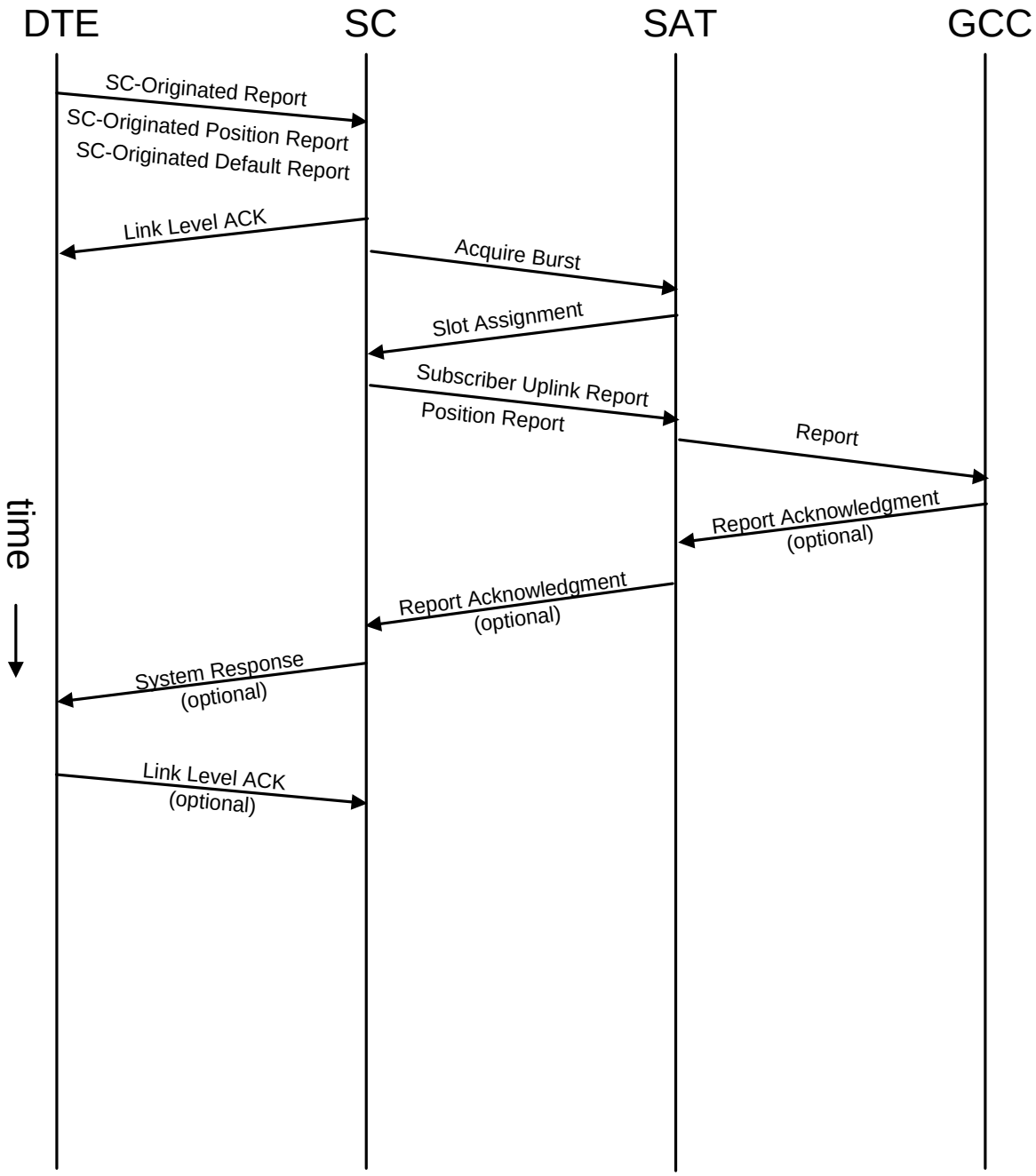



**Figure 4.4 SC-Originated Report**

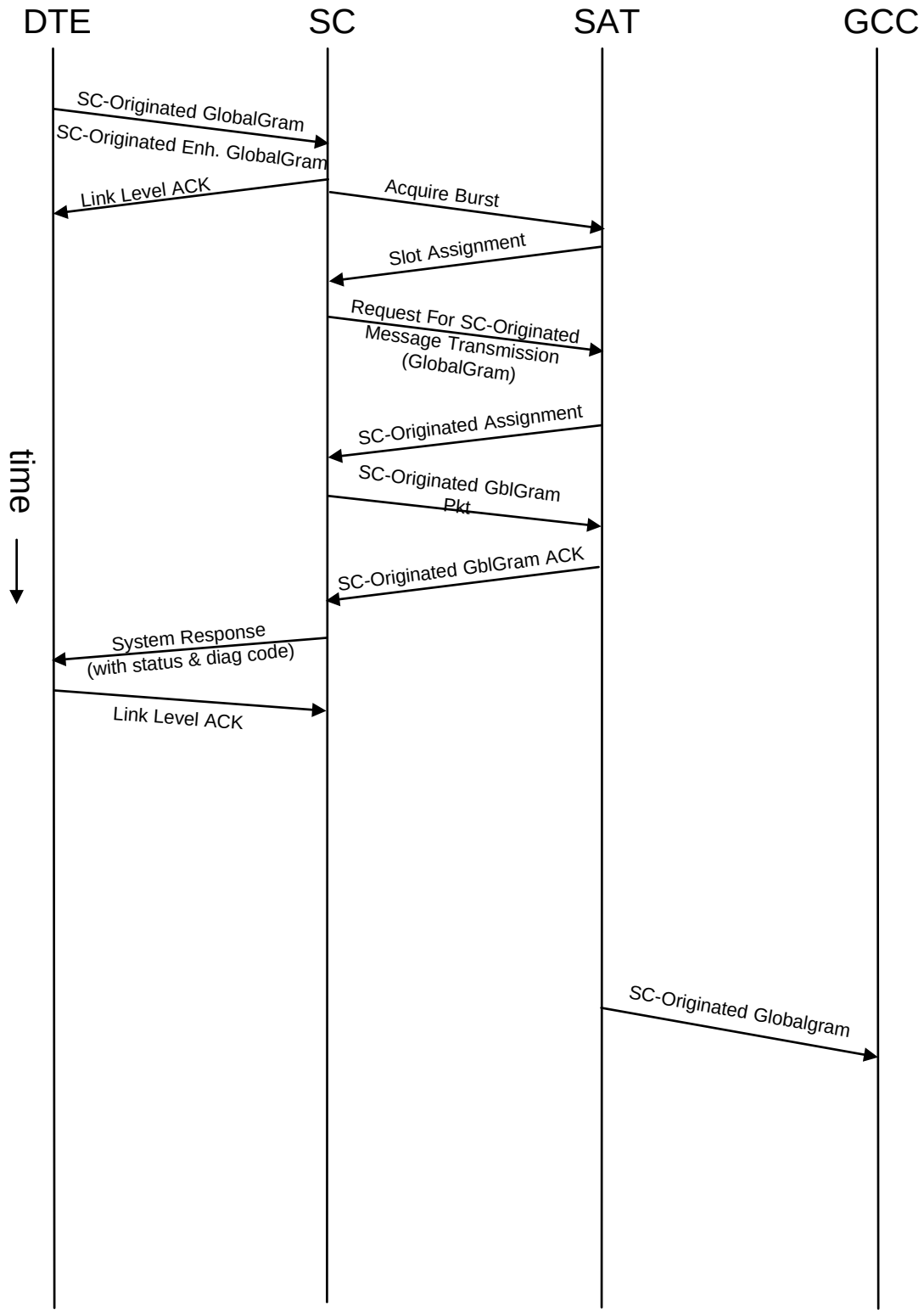



**Figure 4.5 SC-Originated Globalgram**

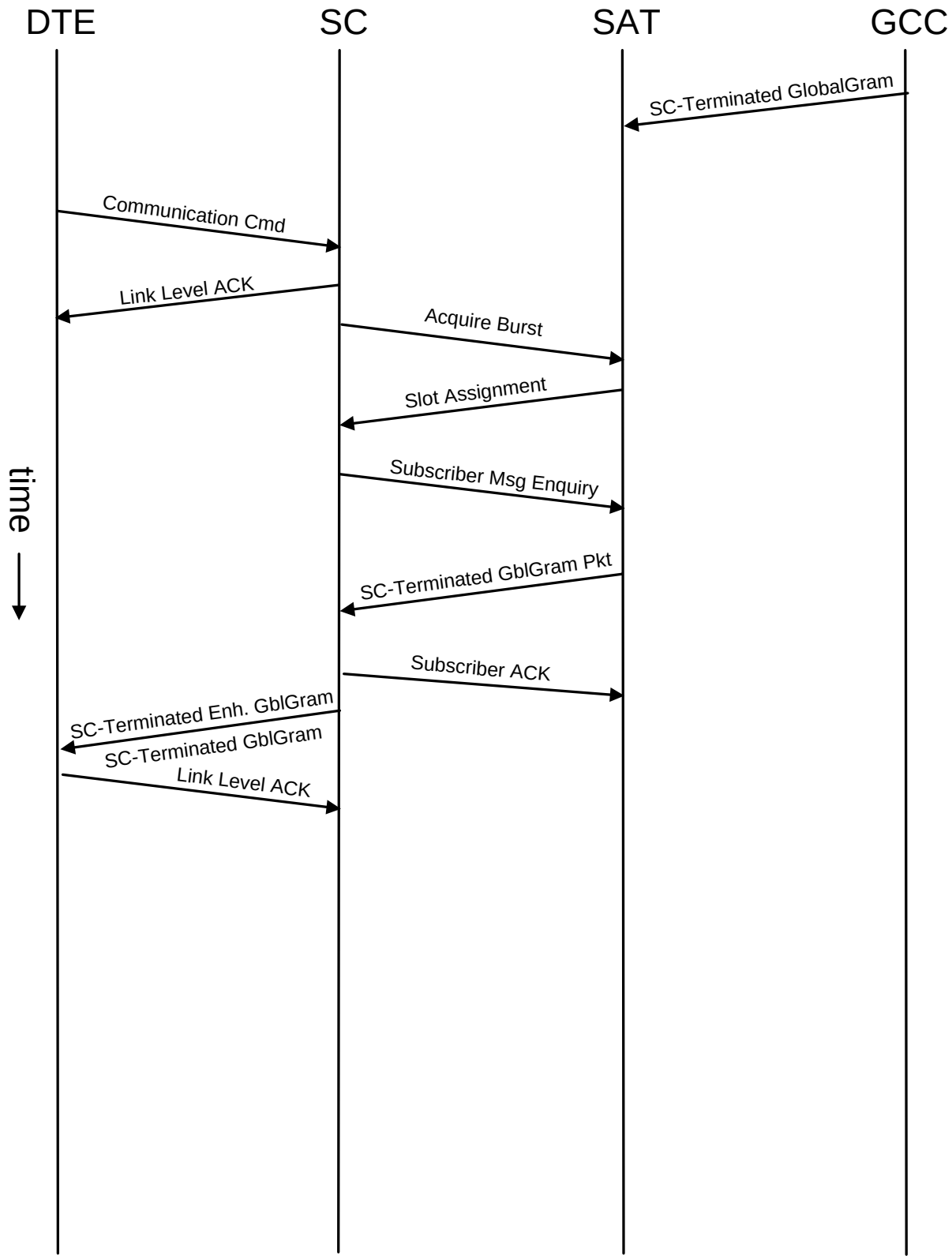



**Figure 4.6 SC-Terminated Globalgram**

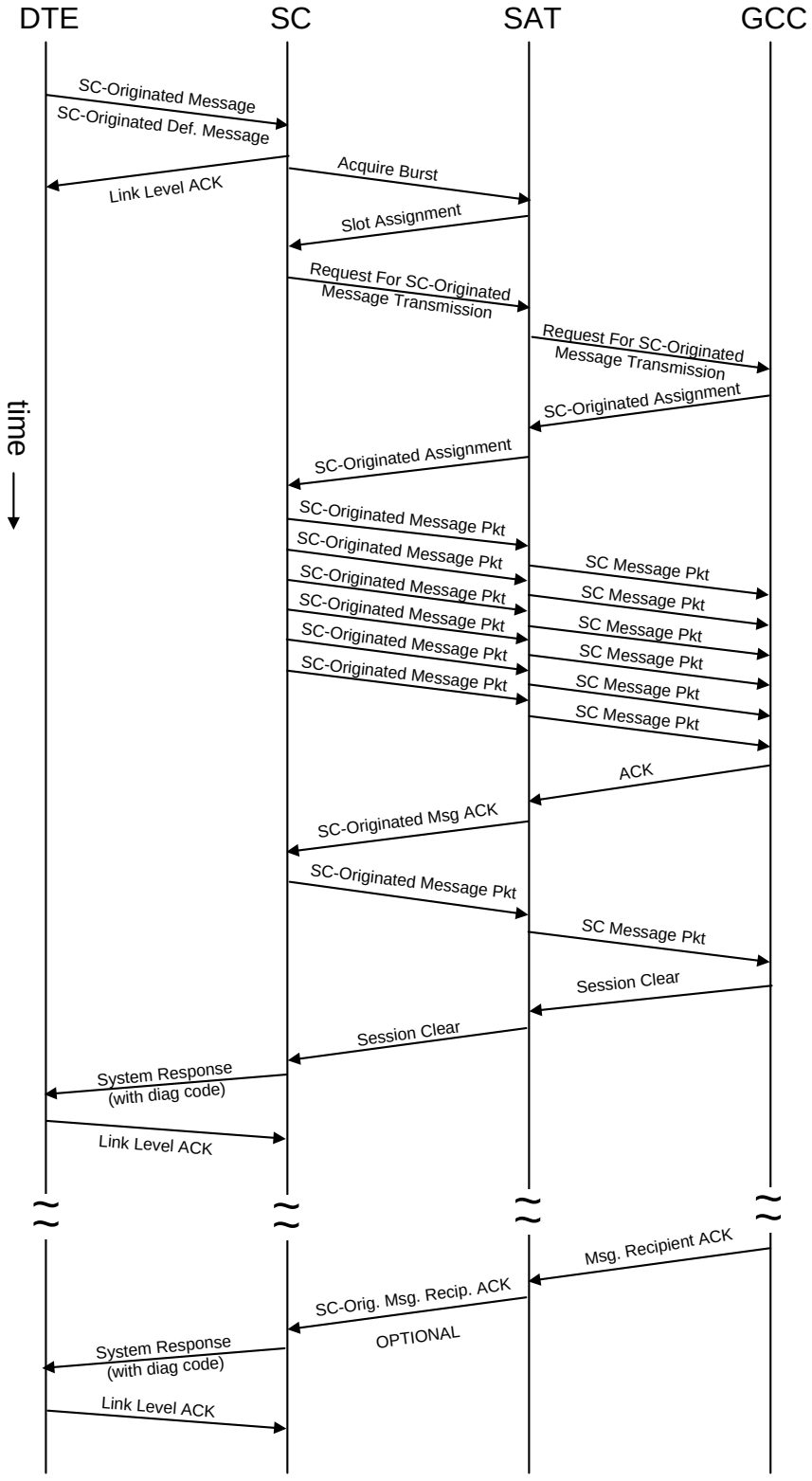



**Figure 4.7 SC-Originated Message**

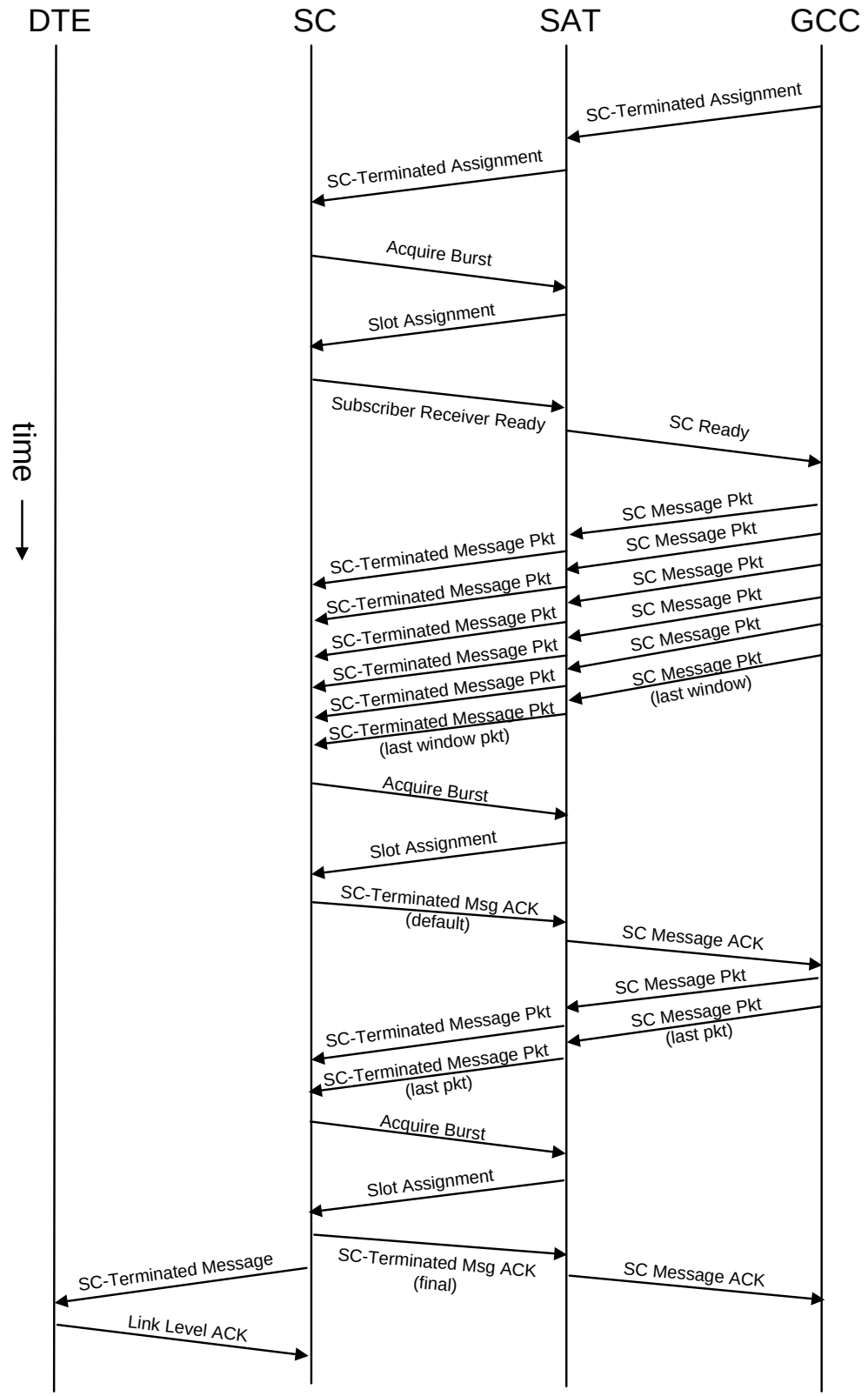



**Figure 4.8 SC-Terminated Message**

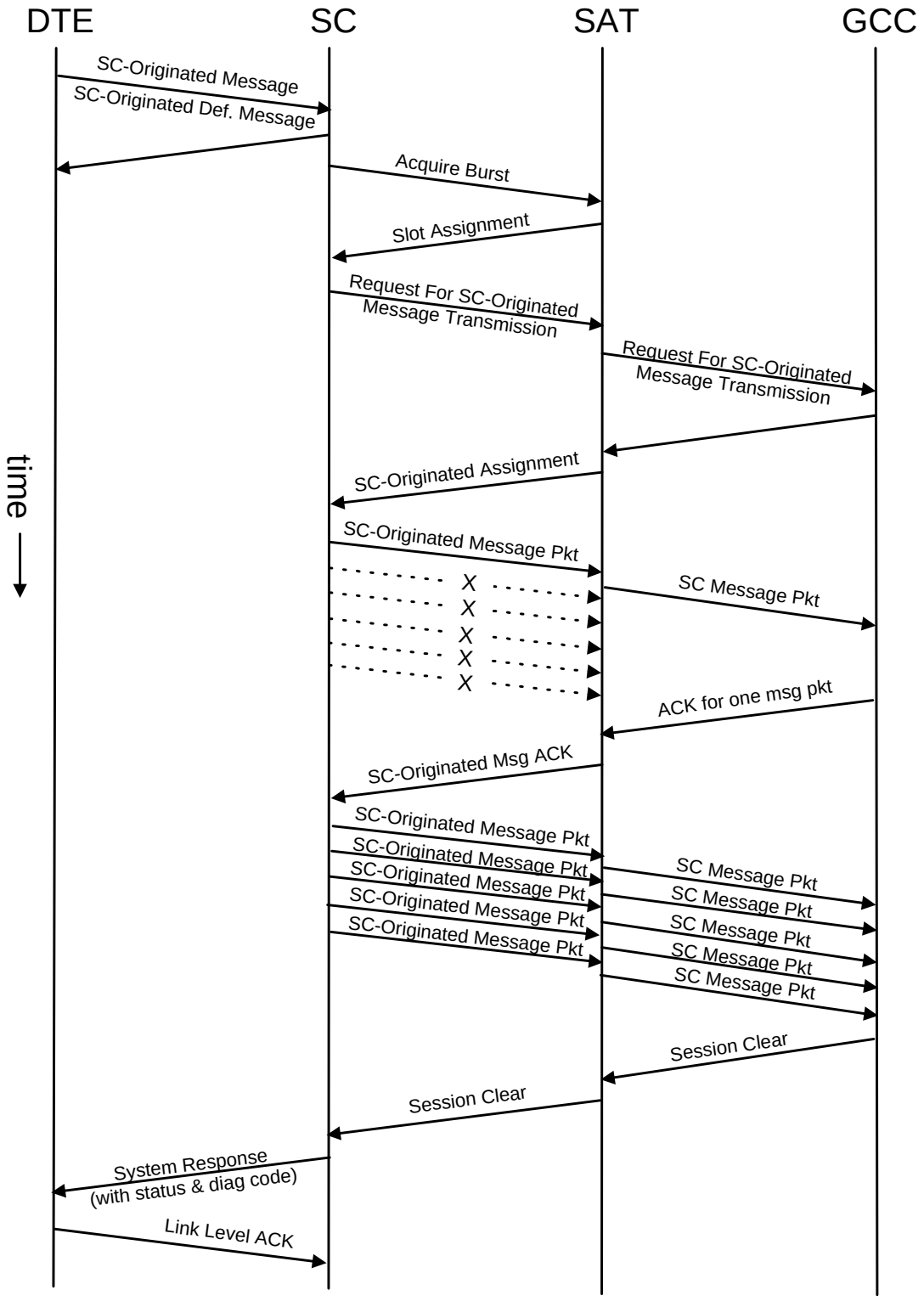



**Figure 4.9 SC-Originated Message, SC Transmission Error**

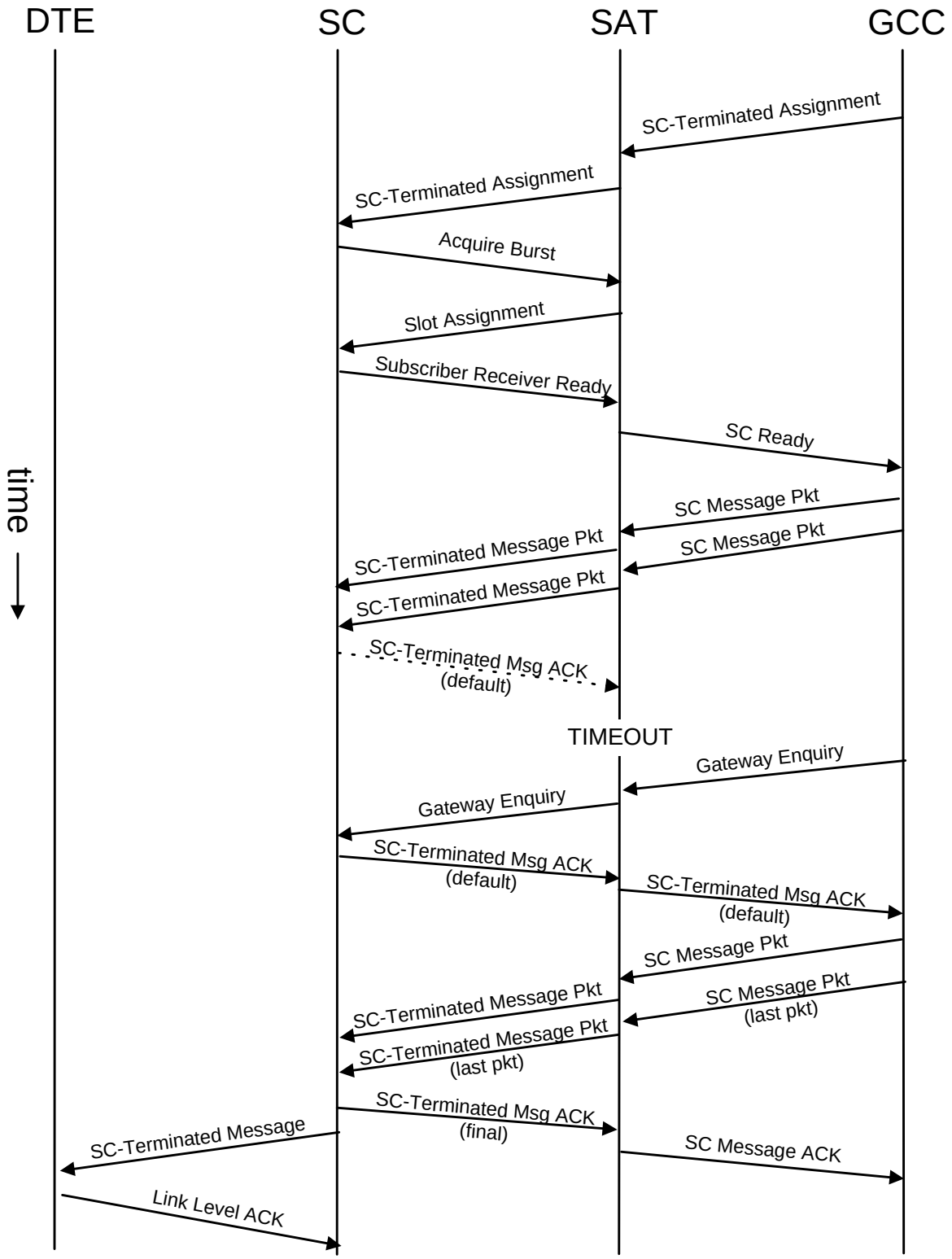



**Figure 4.10 SC-Terminated Message, SC Transmission Error**

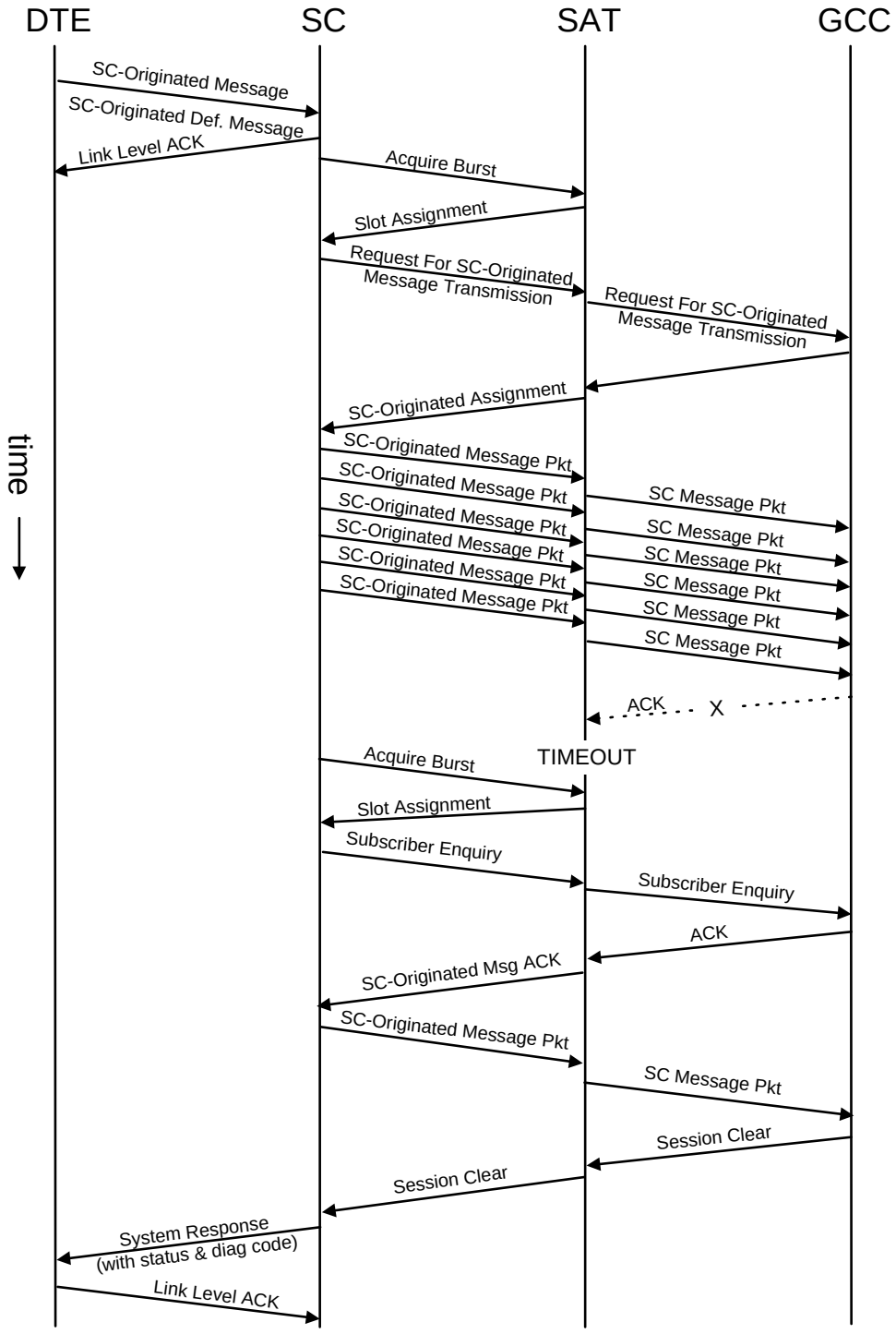



**Figure 4.11 SC-Originated Message, Satellite/GCC Error**

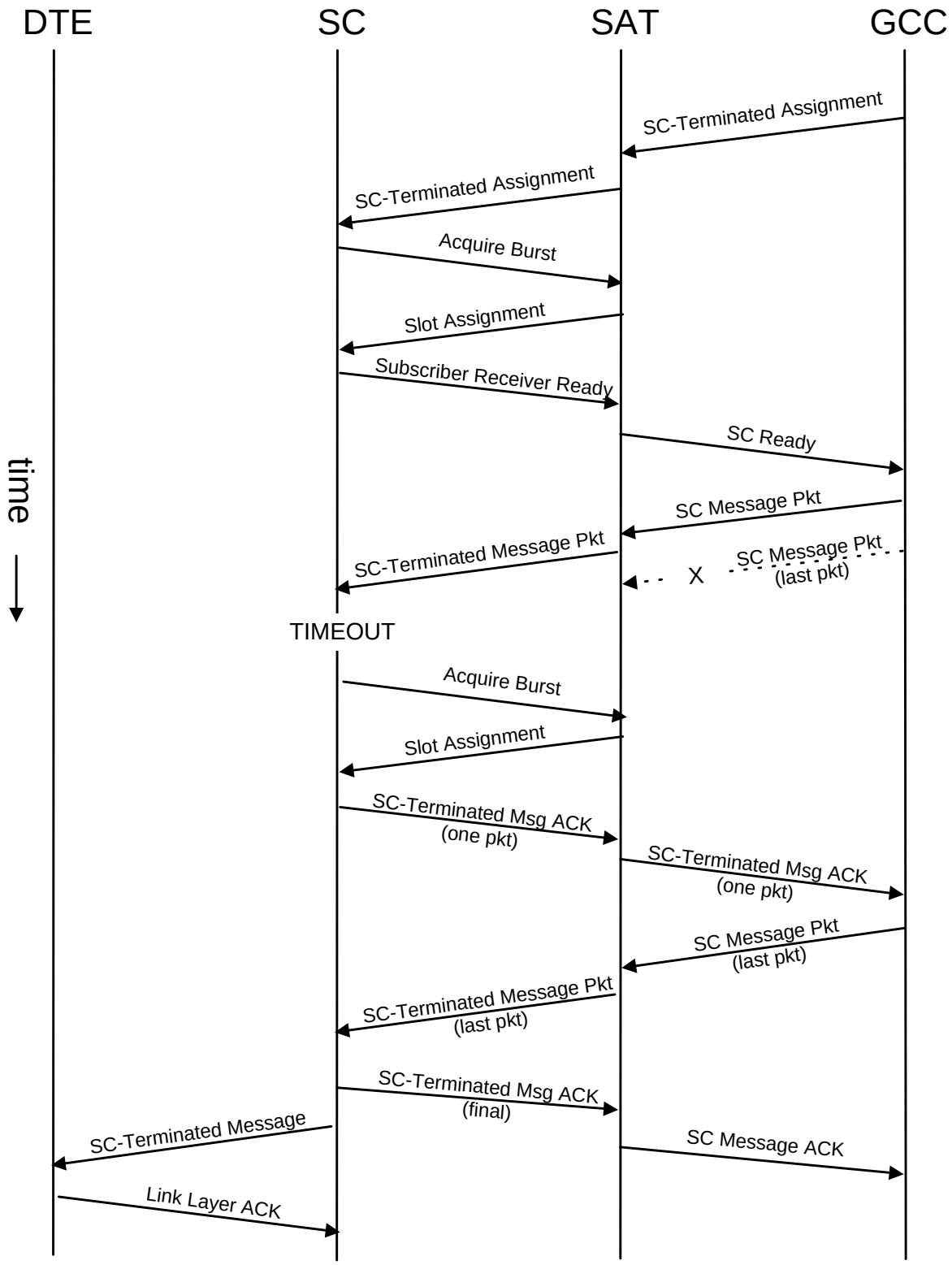
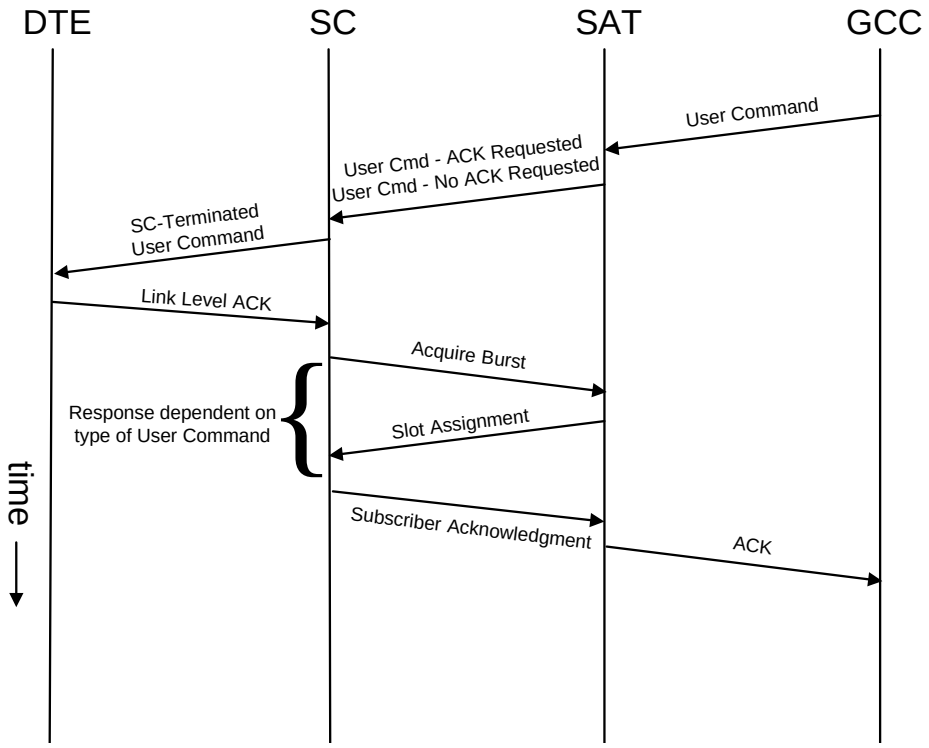



**Figure 4.12 SC-Terminated Message, Satellite/GCC Transmission Error**




User Command:

May elicit a response from the application within the SC or DTE

An ACK may be requested. If not, the elicited response will signify the reception of the command.

**Figure 4.13 SC-Terminated User Command**

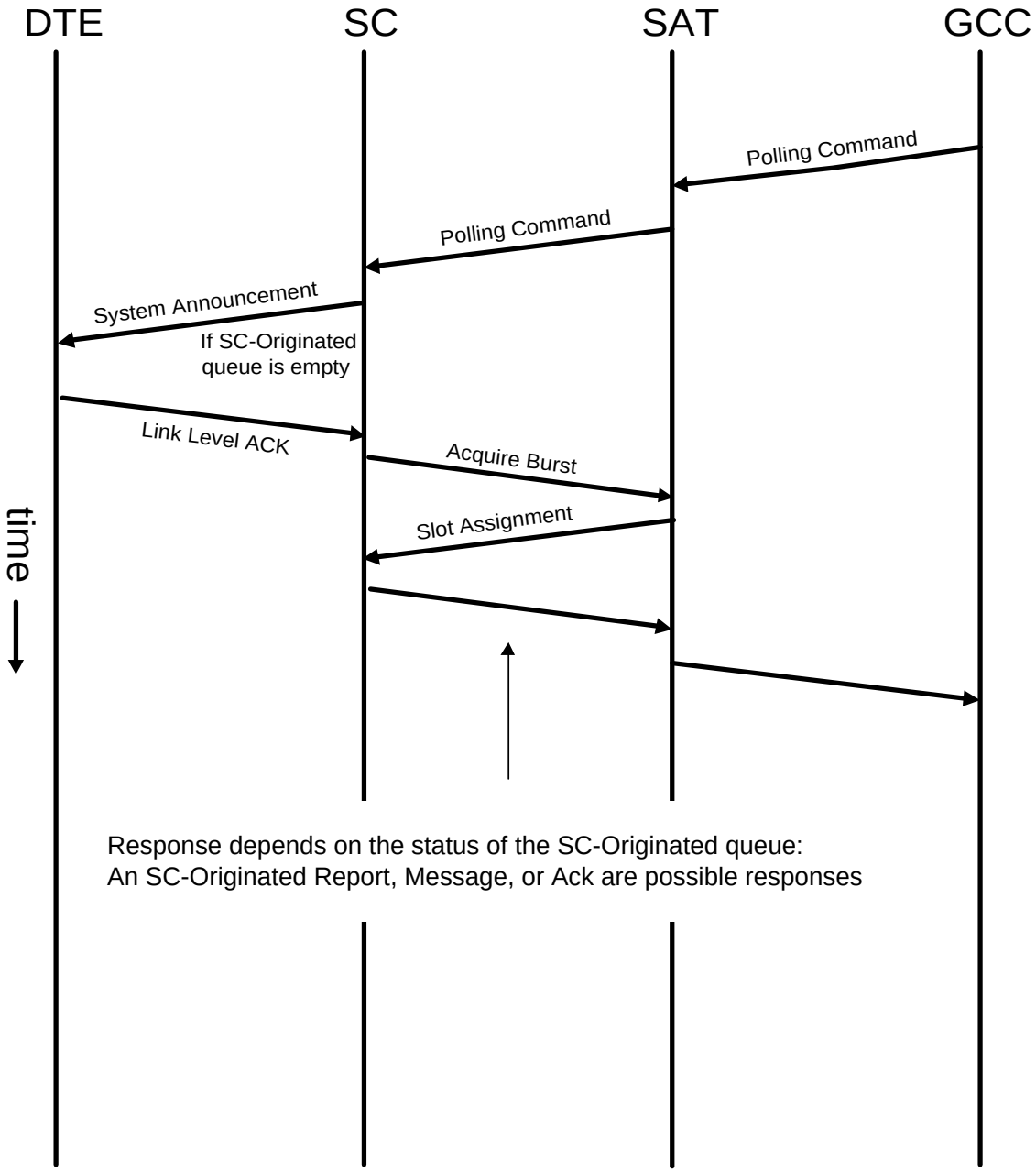



**Figure 4.14 Polling by GCC**

# Appendix A SC Parameters

The SC Parameters listed in the following table are parameters that are available by using the **SET** and **GET PARAMETER** commands.

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x00 | pin_code | 0 | 0 | 9999 | R/W | Personal identification number, used as a security measure; four BCD digits |
| 0x01 | desired_gwy_id | 1 (U.S.) | 1 | 255 | R/W | Instructs SC to acquire satellite connected to this Gateway |
| 0x02 | def_polled | 0 | 0 | 1 | R/W | SC-Originated Messages polled by Gateway or initiated by SC (see Section 3.2, note 1) |
| 0x03 | def_ack_level | 1 | 0 | 4 | R/W | Default acknowledgement level for Messages (see Section 3.2, note 3) |
| 0x04 | def_rep_or_ind | 1 | 0 | 3 | R/W | Default Report OR indicator (see Section 3.2, note 4) |
| 0x05 | def_msg_or_ind | 1 | 0 | 15 | R/W | Default Message OR indicator (see Section 3.2, note 4) |
| 0x06 | def_priority | 0 | 0 | 3 | R/W | Default priority level (see Section 3.2, note 5) |
| 0x07 | def_msg_body_type | 14 | 0 | 15 | R/W | Default Message body type (see Section 3.2, note 6) |
| 0x08 | def_serv_type | 2 | 0 | 15 | R/W | Default service type for Reports (see Section 3.2, note 2) |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x09 | gwy_search_mode | 0 | 0 | 4 | R/W | 0: continuously search downlink band for desired Gateway;<br>1: search band once for desired Gateway, if not found then maintain lock with first discovered downlink;<br>2: maintain lock with first discovered downlink;<br>3: search band once for desired Gateway, if not found, open search to include any Gateway, if none found, maintain lock with first discovered downlink;<br>4: search band once for desired Gateway, if not found, continuously search band for downlink having no Gateway or desired Gateway. |
| 0x0A | ob_route | 1 | 0 | 2 | R/W | Route SC-Terminated Messages/commands to:<br>0: local application task (MHA)<br>1: serial port<br>2: both |
| 0x0B | inactive_interval | 0 | 0 | 86,400 | R/W | Number of seconds to power down after unsuccessfully searching for all channels in *dl_channel* list.<br>(0 = SC stays powered) |
| 0x0C | sc_state | 0 | 0 | 9 | R | State of SC message transport processes<br>(see *sc_state* on A-7) |
| 0x0D | sc_diag_code | 0 | --- | --- | R | SC diagnostics result code<br>(see *sc_diag_code* on A-7) |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x0E | active_mha_ref_num | 0 | 0 | 255 | R | Active MHA message reference number (255 = no message) |
| 0x0F | sat_in_view | 0 | 0 | 255 | R | Number (ID) of current Satellite in view (0 if no satellite in view) |
| 0x10 | gwy_id_list | --- | --- | --- | R | List of IDs of Gateways connected to the current satellite |
| 0x11 | min_gwy_pri_list | --- | --- | --- | R | List of minimum acceptable message priorities for each Gateway identified in parameter 0x10, in the same order as the Gateways identified in parameter 0x10 |
| 0x12 | msg_queue_size | --- | --- | --- | R | Total size of SC message queues, in kB |
| 0x13 | sco_msg_queue_size | --- | --- | --- | R/W | Size of SC-Originated message queue, in kB |
| 0x14 | sct_msg_queue_size | --- | --- | --- | R/W | Size of SC-Terminated message queue, in kB |
| 0x15 | queued_ob_msgs | 0 | 0 | --- | R | Number of SC-Terminated messages in queue |
| 0x16 | queued_ib_msgs | 0 | 0 | --- | R | Number of SC-Originated messages in queue |
| 0x17 | week_bytes | --- | --- | --- | R/W | UTC time week, week 0 = January 6, 1980 |
| 0x18 | time_bytes | --- | 0 | 604,800 | R/W | 24-bit integer representing the number of seconds 00:00:00 UTC last Sunday (resets 12:00 Saturday night/ Sunday morning) |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x19 | total_sats | --- | 0 | --- | R | Total number of Satellites in system |
| 0x1A | stored_sats | 0 | 0 | --- | R | Number of stored satellite orbital elements |
| 0x1B | pos_calc_active | 0 | 0 | 1 | R/W | State of position determination process (if supported),<br>0: off, 1: active |
| 0x1C | Reserved for Future Use | --- | --- | --- | --- | |
| 0x1D | lat_code | --- | 0 | 0xFFFFFF | R/W | Coded geodetic latitude 0x000000: North Pole, 0xFFFFFF: South Pole |
| 0x1E | lon_code | --- | 0 | 0xFFFFFF | R/W | Coded geodetic longitude: 0 = Greenwich Median, increasing in eastern direction |
| 0x1F | msg_requeue_opt | 1 | 0 | 1 | R/W | SC-Originated Message/ Report/Globalgram requeue options<br>1: requeue when transmission fails<br>0: do not requeue, notify DTE using **SYSTEM RESPONSE** packet |
| 0x20 | poll_response_timeout | 5 | 2 | 30 | R/W | Poll response timeout: number of seconds to wait for response from MHA/DTE before holding SC-Originated Messages (with *polled* set) for next poll |
| 0x21 | ser_max_retries | 0 | 0 | 255 | R/W | Number of successive packet retries without receiving valid ACK before discarding the packets |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x22 | ser_pkt_timeout | 5 | 1 | 30 | R/W | Serial packet timeout: number of seconds SC waits for **LINK LEVEL ACK** after sending last byte of a packet before resending |
| 0x23 | abort_response | 0 | 0 | 1 | R/W | Abort response<br>0: do nothing,<br>1: send abort Report |
| 0x24 | abort_report | --- | --- | --- | W | Abort Report: values of *gcc_id*, *polled*, *serv_type*, *or_ind*, and info bytes 0-5 for abort report |
| 0x25 | Reserved for Future Use | --- | --- | --- | | |
| 0x26 | ob_flow_ctrl | 3 | 0 | 3 | W | SC sending packets/bytes to DTE:<br>0: deactivated DTR stops it<br>1: activated RTS stops it<br>2: either<br>3: no flow control |
| 0x27 | ib_flow_ctrl | 3 | 0 | 3 | W | To stop DTE from sending to SC:<br>0: deactivate CTS<br>1: deactivate DSR<br>2: both<br>3: no flow control |
| 0x28 | DSR_treatment | 0 | 0 | 1 | W | 1: DSR activated if one or more SC-Terminated messages queued in SC |
| 0x29 | baud_rate | 4 | 0 | 5 | R/W | DTE baud rate<br>0: 300<br>1: 600<br>2: 1200<br>3: 2400<br>4: 4800<br>5: 9600 |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x2A | parity_bits | 0 | 0 | 2 | R/W | DTE Parity 0: none, 1: odd, 2: even |
| 0x2B | stop_bits | 1 | 1 | 2 | R/W | DTE stop bits |
| 0x2C | data_bits | 8 | 7 | 8 | R/W | DTE data bits |
| 0x2D | duplex | 1 | 0 | 2 | R/W | DTE duplex setting<br>0: half, 1: full, 2: receive only |
| 0x2E | test_mode | 0 | 0 | 3 | W | Test Mode<br>0: normal operation,<br>1: echo downlink bit error rate test to DTE,<br>2: echo uplink to DTE<br>3: echo both to DTE,<br>4: echo DTE to uplink as reservation burst |
| 0x2F | pwr_dwn_mode | 1 | 0 | 1 | W | 0: receiver power controlled by DTE DTR signal<br>1: receiver power controlled by *rvcr_power* parameter |
| 0x30 | active_set_id | 0 | 0 | 1 | R/W | Active configuration set ID<br>0: preset factory defaults<br>1: custom |
| 0x31 | serial_num | --- | --- | --- | R | SC serial number |
| 0x32 | sw_version | --- | --- | --- | R | SC software version |
| 0x33 | hw_version | --- | --- | --- | R | SC hardware version |
| 0x34 | ser_spec_rev | ‘G’ | ‘G’ | --- | R | SC serial interface specification revision supported |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x35 | manu_id | --- | 00 | 99 | R | SC Manufacturer ID<br>00: Panasonic<br>01: Scientific Atlanta<br>02: Stellar<br>03: Magellan<br>04: Quake<br>05: Delphi<br>06: Mobiapps |
| 0x36 | pos_det_supported | 0 | 0 | 3 | R | Onboard position determination capability:<br>0: not supported<br>1: GPS<br>2: Doppler<br>3: Glonass |
| 0x37 | most_recent_dl | --- | 50 | 349 | R | Most recent downlink channel |
| 0x38 | dl_chan_list | --- | 50 | 349 | R | Downlink channel list |
| 0x39 | debug_lvl | 0 | 0 | 5 | R/W | Specifies level of detail of messages transmitted from debug serial port (TBD) |
| 0x3A | rcvr_power | 2 | 0 | 2 | R/W | Receiver power state:<br>0: off<br>1: listen to DL 6/16 frames<br>2: on continuously |
| 0x3B | send_pass_predict | 0 | 1 | 0 | W | When 1, passes **EPHEMERIS**, **SATELLITE PLANE ORBITAL ELEMENTS** packets to the DTE |

| Number | Name | Def. Value | Min Value | Max Value | DTE Access | Description |
|---|---|---|---|---|---|---|
| 0x3C - 0x7F | RESERVED FOR FUTURE ORBCOMM USE | --- | --- | --- | --- | --- |
| 0x80 - 0xFF | RESERVED FOR SC MANUFACTURER-SPECIFIC PARAMETERS | --- | --- | --- | --- | |

**Table A.1 SC Parameters**

NOTE: Parameter numbers 0 – 127 (0x00 – 0x7F) are reserved for use by ORBCOMM. Parameters 128 – 255 (0x80 – 0xFF) are reserved for manufacturers' use.

SC State: (See page A-2)

0 = idle

1 = sending SC-Originated message

2 = sending SC-Originated report

3 = sending SC-Originated Globalgram

4 = receiving SC-Terminated message

5 = receiving SC-Terminated command

6 = receiving SC-Terminated Globalgram

7 = performing self-test

8 = performing local loop-back

9 = performing ORBCOMM Gateway loop-back test (may require a minute or two, depending on Satellite availability)

## SC Diag Code: (See page A-2)

0 = no indication

1 = self-test failed, PROM CHECKSUM failed

2 = self-test failed, bad memory location

3 = local loop-back failed, modem not responding

4 = remote loop-back failed, no Satellite visible

5 = remote loop-back failed, transmitter probably bad

6 = remote loop-back failed, ORBCOMM Gateway not responding

7 = remote loop-back failed, TX/RX data does not match

8-19 = reserved for ORBCOMM use

20 = SC detected general failure

21 and above: reserved for SC manufacturers

# Appendix B NETWORK CAPABILITIES

1. **Data Report**

   SCs generate short reports consisting of one packet that are transmitted via random access. Data Reports can be originated by an SC or network requested (polled). A Data Report may be acknowledged and, if so, a re-transmission protocol is used for failed packets.

2. **Message Transfer**

   A Message refers to a longer sequence of data to be transferred to or from the SC, typically between 10 and 1000 bytes. Transfer of messages involves several more steps for reliable transfer. Messages are transferred via short packets over the satellite links. All packets are acknowledged or re-transmitted. Messages are accepted/delivered via public or private data networks. These Messages can be transferred to or from SCs. Messages from SCs can be originated at the request of the SC (random access), or at the request of the network (polled). In either case the transfer of Message packets is in a reservation (“polled”) fashion.

3. **Globalgram**

   A Globalgram is a single, self-contained packet sent or received by an SC when no Network Control Center can be accessed by the current satellite. Globalgrams allow up to 229 user-defined bytes per packet to be sent by the SC and 182 user-defined bytes per packet to be received. This allows remote and oceanic areas to be served in a "store-and- forward" mode. A SC is only permitted to send or receive Globalgrams when com-munication is impossible with any ORBCOMM Gateway. The Globalgram is acknowl-edged by the Satellite and then archived in Satellite memory until the addressed ORBCOMM Gateway establishes contact with that Satellite. The Satellite will queue no more than 16 SC-Originated and 16 SC-Terminated Globalgrams per SC at a time.

4. **Commands**

   Short commands consisting of one packet are transmitted to an SC terminal. Commands could be signals to initiate communications via other links or commands from a billing system to enable another receiver of a different service. Acknowledgment may or may not be required.

# Appendix C Fletcher checksum

```
/*
 * operator fletcher_encode
 */
fletcher_encode( buffer, count )
unsigned char* buffer;
long      count;
{
  int i;
  unsigned char c0 = 0;
  unsigned char c1 = 0;
  *( buffer + count - 1 ) = 0;
  *( buffer + count - 2 ) = 0;
  for( i = 0; i < count; i++)
  {
       c0 = c0 + *( buffer + i );
       c1 = c1 + c0;
  }
  *( buffer + count - 2 ) = c0 - c1;
  *( buffer + count - 1 ) = c1 - 2*c0;
}
/*
 * operator fletcher_decode
 */
long fletcher_decode( buffer, count )
unsigned char* buffer;
long     count;
{
  long  result = 0;
  int i;
  unsigned char c0 = 0;
  unsigned char c1 = 0;
  for( i = 0; i < count; i++)
  {
       c0 = c0 + *( buffer + i );
       c1 = c1 + c0;
  }
  return( (long)(c0 + c1) );
```

# Appendix D LATITUDE & LONGITUDE CONVERSION

**From ORBCOMM Codes to Geodetic:**

```
long double lat, lon, lon_code, lat_code;

lat = 90.0 - ((unsigned long)lat_code/((double)0XFFFFFF))*180.0;
lon = ((unsigned long)lon_code/((double)0XFFFFFF))*360.0;
if(lon > 180.0)
     lon = lon - 360.0;
```

**From Geodetic to ORBCOMM Codes:**

```
long double lat, lon, lon_code, lat_code, tmp1;

if(lon < 0.0 )
{
     /* west */
     tmp1 = (lon + 360.0)/360.0;
}
else
{
     /* east */
     tmp1 = lon/360.0;
}
lon_code = (unsigned long) (tmp1 * 0x0FFFFFF) & 0x0FFFFFF;
tmp1 = -(lat - 90.0)/180.0;

lat_code = (unsigned long) (tmp1 * 0x0FFFFFF) & 0x0FFFFFF;
```

# Appendix E SATELLITE ELEMENT SET FORMAT DESCRIPTION

The satellite ephemeris data consists of two lines in the following format (not including the two-line column "ruler"). Multiple element sets should repeat the formatting above— i.e., a two-line element set for the first satellite, followed by a two-line element set for the second satellite, etc. All element sets should be sent from the SC in the form of an SC-Originated Message. This Message should be sent with a blank ID, no ORIGINATOR and no SUBJECT. The Message body field should contain the two-lined element sets with no additional characters.

```
Column                1         2         3         4         5         6
Number:      123456789012345678901234567890123456789012345678901234567890123456789
             1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
Element      2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNN
Set Format:
```

**EXAMPLE**: Actual F2 element set as received from USSPACECOM.

```
          1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890123456789
1 23546U 95017B   97072.53586170  .00000056  00000-0  32865-4 0 02534
2 23546 069.9776 265.2990 0013272 169.5560 190.5848 14.45508865102583
```

**Format Description**:

Line 1

| Column | Description |
|---|---|
| 01 | Line Number of Element Data |
| 02 | Blank |
| 03-07 | Satellite Number (corresponds to the *satellite_#* when sent by SC. Zero fill all leading characters.) |
| 08 | Classification of Element Set (U = unclassified) |
| 09 | Blank |
| 10-11 | International Designator (last two digits of launch year)-**N/A** |
| 12-14 | International Designator (Launch number of the year)-**N/A** |
| 15-17 | International Designator (Piece of launch)-**N/A** |
| 18 | Blank |

| | |
|---|---|
| 19-20 | Epoch Year (last two digits of year)— calculated from data in SATELLITE PLANE ORBITAL ELEMENTS |
| 21-32 | Epoch (Julian Day and fractional portion of the day)— calculated from data in SATELLITE PLANE ORBITAL ELEMENTS |
| 33 | Blank |
| 34-43 | Decay from SATELLITE PLANE ORBITAL ELEMENTS. |
| 44 | Blank |
| 45-52 | Second Time Derivative of Mean Motion— **N/A** |
| 53 | Blank |
| 54-61 | BSTAR drag term if GP4 general perturbation theory was used— **N/A** |
| 62 | Blank |
| 63 | Ephemeris type— **N/A** |
| 64 | Blank |
| 65-68 | Element set number— **N/A** |
| 69 | checksum (Modulo 10) |

Line2

| Column | Description |
|---|---|
| 01 | Line Number of Element Data |
| 02 | Blank |
| 03-07 | Satellite Number (same as above) |
| 08 | Blank |
| 09-16 | Inclination (degrees)— from SATELLITE PLANE ORBITAL ELEMENTS |
| 17 | Blank |
| 18-25 | Right Ascension of the Ascending Node (degrees)— from SATELLITE PLANE ORBITAL ELEMENTS |
| 26 | Blank |
| 27-33 | Eccentricity (decimal point assumed)— from SATELLITE PLANE ORBITAL ELEMENTS |
| 34 | Blank |
| 35-42 | Argument of Perigee (degrees)— from SATELLITE PLANE ORBITAL ELEMENTS |

Line 2 (continued)

| Column | Description |
|---|---|
| 43 | Blank |
| 44-51 | Mean Anomaly (degrees) — from SATELLITE ORBITAL ELEMENTS |
| 52 | Blank |
| 53-63 | Mean Motion (revs per day) — from SATELLITE ORBITAL ELEMENTS |
| 64-68 | Orbit number at epoch (revs) — from SATELLITE PLANE ORBITAL ELEMENTS |
| 69 | checksum (Modulo 10) |

**NOTES:**

1. To calculate checksum: letters, blanks, periods = 0; minus sign =1; plus sign = 2.
2. The International Designator and other complete data fields not required by the tracking algorithm are sent as blank spaces or zeros.
3. All epochs are Coordinated Universal Time (UTC).
4. In fields where there may be leading zeros, blanks cannot be substituted for those leading zeros.
5. Fields designated as **N/A** (not applicable) are not required and should be zero filled.
6. Plus signs (+) are optional.
7. Minus signs (-) are required.
8. A carriage return (CR) and a line feed (LF) must be appended to the end of each line.

# Appendix F GLOSSARY OF TERMS & ACRONYMS

| | |
|---|---|
| ACK | Acknowledgment |
| CMOS | Complementary Metal-Oxide Semiconductor |
| CR | carriage return |
| CTS | Clear-To-Send |
| DCE | Data Communications Equipment |
| DSR | Data Set Ready |
| DTE | Data Terminal Equipment |
| DTR | Data Terminal Ready |
| ECL | Emitter Coupled Logic |
| GCC | Gateway Control Center |
| GWY | ORBCOMM Gateway |
| Inbound Subscriber originated (SC-Originated) direction | The direction of a message originating from an SC and sent to an ORBCOMM Gateway/Satellite LF line feed |
| MHA | Message Handler Agent |
| MTA | Message Transfer Agent |
| OMS | ORBCOMM Message Switch |
| Outbound Subscriber terminated (SC-Terminated) direction | The direction of a message sent from an ORBCOMM Gateway/Satellite and terminating at an SC |
| RTS | Request-To-Send |
| SC | Subscriber Communicators |
| SMTP | Simple Mail Transfer Protocol |
| TTL | Transistor-Transistor Logic |
| UTC | Coordinated Universal Time |