CCSDS Historical Document

This document's Historical status indicates that it is no longer current. It has either been replaced by a newer issue or withdrawn because it was deemed obsolete. Current CCSDS publications are maintained at the following location:

http://public.ccsds.org/publications/

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

SPACE LINK EXTENSION—RETURN CHANNEL FRAMES SERVICE SPECIFICATION

RECOMMENDED STANDARD

CCSDS 911.2-B-2

BLUE BOOK

January 2010

CCSDS HISTORICAL DOCUMENT

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

SPACE LINK EXTENSION—RETURN CHANNEL FRAMES SERVICE SPECIFICATION

RECOMMENDED STANDARD

CCSDS 911.2-B-2

BLUE BOOK

January 2010

AUTHORITY

Issue:	Recommended Standard, Issue 2
Date:	January 2010
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - -- The **standard** itself.
  - -- The anticipated date of initial operational capability.
  - -- The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Russian Federal Space Agency (RFSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 911.2-B-1	Space Link Extension—Return Channel Frames Service Specification	November 2004	Original issue, superseded
CCSDS 911.2-B-2	Space Link Extension—Return Channel Frames Service Specification, Recommended Standard, Issue 2	January 2010	Current issue: – corrects/clarifies/ updates text and adds the option of picosecond resolution to the earth-receive-time parameter.
EC1	Editorial Change 1	August 2010	Corrects editorial errors in A2.4.

NOTE – Substantive changes from the previous issue are indicated by change bars in the inside margin.

CONTENTS

CONTENTS (continued)

Section Page

Figure

Table

1 INTRODUCTION

1.1 PURPOSE OF THIS RECOMMENDED STANDARD

The purpose of this Recommended Standard is to define the Space Link Extension (SLE) Return Channel Frames (RCF) service in conformance with the SLE Reference Model (reference [1]). The RCF service is an SLE transfer service that delivers to a mission user all telemetry frames from one master channel or one virtual channel.

NOTE – The first issue of reference [1] defines the Return Master Channel Frames (Rtn MC Frames) service and the Return Virtual Channel Frames (Rtn VC Frames) service as two distinct services. Subsequent study has indicated that it is preferable to define one service that provides the functionality of both. The RCF service defined here does just that. It is anticipated that the next issue of reference [1] will take the same approach, deleting the Rtn MC Frames and Rtn VC Frames services and replacing them with the RCF service.

1.2 SCOPE

This Recommended Standard defines, in an abstract manner, the RCF service in terms of:

a) the operations necessary to provide the service;

b) the parameter data associated with each operation;

c) the behaviors that result from the invocation of each operation; and

d) the relationship between, and the valid sequence of, the operations and resulting behaviors.

It does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to acquire telemetry frames from signals received from a spacecraft;

d) the methods or technologies required to provide a suitable environment for communications; or

e) the management activities required to schedule, configure, and control the RCF service.

1.3 APPLICABILITY

1.3.1 APPLICABILITY OF THIS RECOMMENDED STANDARD

This Recommended Standard provides a basis for the development of real systems that implement the RCF service. Implementation of the RCF service in a real system additionally requires the availability of a communications service to convey invocations and returns of RCF service operations between RCF service users and providers. This Recommended Standard requires that such a communications service must ensure that invocations and returns of operations are transferred:

a) in sequence;

b) completely and with integrity;

c) without duplication;

d) with flow control that notifies the application layer in the event of congestion; and

e) with notification to the application layer in the event that communications between the RCF service user and the RCF service provider are disrupted, possibly resulting in a loss of data.

It is the specific intent of this Recommended Standard to define the RCF service in a manner that is independent of any particular communications services, protocols, or technologies.

1.3.2 LIMITS OF APPLICABILITY

This Recommended Standard specifies the RCF service that may be provided by an SLE Complex for inter-Agency cross support. It is neither a specification of, nor a design for, real systems that may be implemented for the control and monitoring of existing or future missions.

1.4 RATIONALE

The goal of this Recommended Standard is to create a standard for interoperability between the tracking stations or ground data handling systems of various Agencies and the consumers of spacecraft telemetry.

1.5 DOCUMENT STRUCTURE

1.5.1 ORGANIZATION

This document is organized as follows:

a) section 1 presents the purpose, scope, applicability and rationale of this Recommended Standard and lists the definitions, conventions, and references used throughout the Recommended Standard;

b) section 2 provides an overview of the RCF service including a functional description, the service management context, and protocol considerations;

c) section 3 specifies the operations of the RCF service;

d) section 4 specifies the dynamic behavior of the RCF service in terms of the state transitions of the RCF service provider;

e) annex A provides a formal specification of RCF service data types using Abstract Syntax Notation One (ASN.1);

f) annex B provides a conformance matrix that defines what capabilities must be provided for an implementation to be considered compliant with this Recommended Standard;

g) annex C lists all terms used in this Recommended Standard and identifies where they are defined;

h) annex D lists all acronyms used within this document;

i) annex E provides a list of informative references.

1.5.2 SLE SERVICES DOCUMENTATION TREE

This Recommended Standard is based on the cross support model defined in the SLE Reference Model (reference [1]). It expands upon the concept of an SLE transfer service as an interaction between an SLE Mission User Entity (MUE) and an SLE transfer service provider for the purpose of providing the RCF transfer service.

This Recommended Standard is part of a suite of documents specifying the SLE services. The SLE services constitute one of the three types of Cross Support Services:

a) Part 1: SLE Services;

b) Part 2: Ground Domain Services;

c) Part 3: Ground Communications Services.

The basic organization of the SLE services documentation is shown in figure 1-1. The various documents are described in the following subsections.

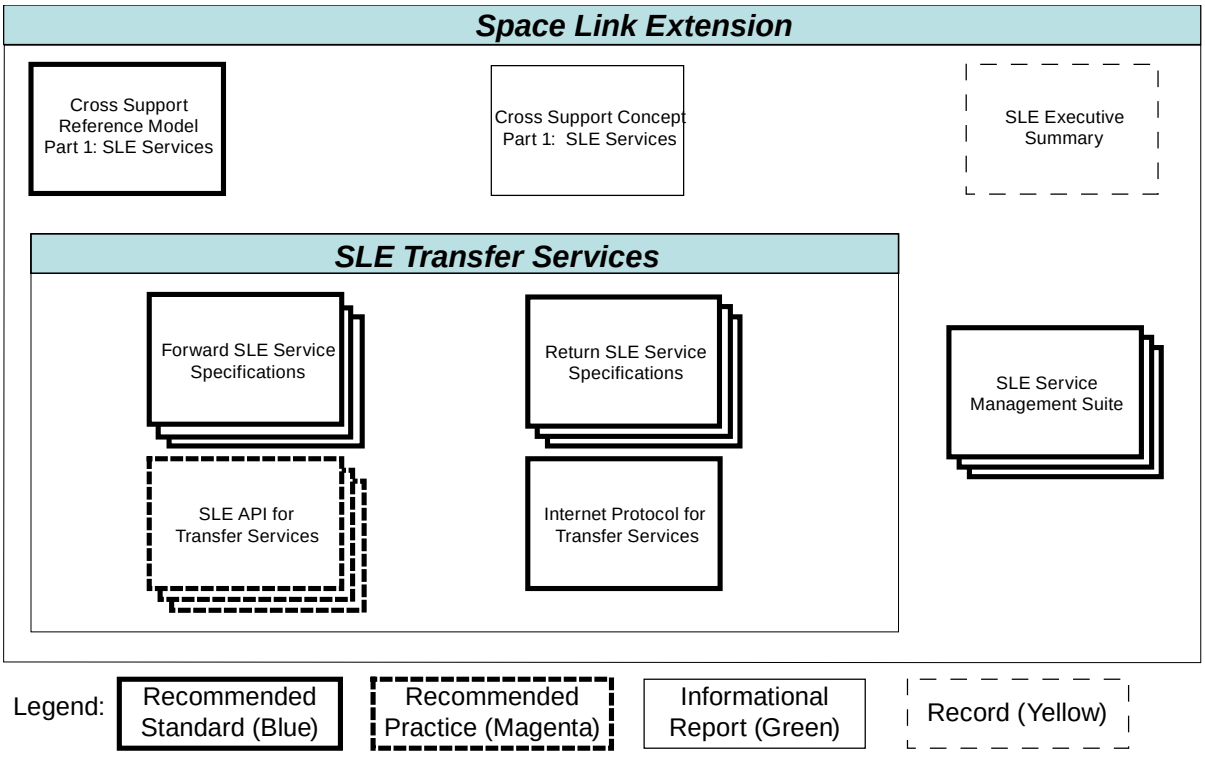

Figure 1-1: SLE Services Documentation

a) *Cross Support Concept—Part 1: Space Link Extension Services* (reference [E2]): a Report introducing the concepts of cross support and the SLE services;

b) *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [1]): a Recommended Standard that defines the framework and terminology for the specification of SLE services;

c) *SLE Return Service Specifications*: a set of Recommended Standards that will provide specification of all return link SLE services (this Recommended Standard is one of the specifications in that set);

d) *SLE Forward Service Specifications*: a set of Recommended Standards that will provide specification of all forward link SLE services;

e) SLE API for Transfer Services Specifications: a set of Recommended Practices that provide specifications of an Application Program Interface; a set of Recommended Standards that provide specifications of an Application Program Interface and a mapping to TCP/IP as underlying communications service for SLE services;

f) Internet Protocol for Transfer Services: defines a protocol for transfer of SLE Protocol Data Units using TCP/IP as underlying communications service for SLE services;

g) *SLE Service Management Specifications*: a set of Recommended Standards that establish the basis of SLE service management.

1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from Open Systems Interconnection (OSI) Basic Reference Model

This Recommended Standard makes use of a number of terms defined in reference [6]. The use of those terms in this Recommended Standard shall be understood in a generic sense, i.e., in the sense that those terms are generally applicable to technologies that provide for the exchange of information between real systems. Those terms are:

a) abstract syntax;

b) application entity;

c) application layer;

d) application process;

e) flow control;

f) Open Systems Interconnection (OSI);

g) real system;

h) Service Access Point (SAP).

1.6.1.2 Definitions from Abstract Syntax Notation One

This Recommended Standard makes use of the following terms defined in reference [7]:

a) Abstract Syntax Notation One (ASN.1);

b) object identifier;

c) (data) type;

d) (data) value.

NOTE – In annex A of this Recommended Standard, ASN.1 is used for specifying the abstract syntax of RCF service operation invocations and returns. The use of ASN.1 as a descriptive language is intended to support the specification of the abstract RCF service; it is not intended to constrain implementations. In particular, there is no requirement for implementations to employ ASN.1 encoding rules. ASN.1 is simply a convenient tool for formally describing the abstract syntax of RCF service operation invocations and returns.

1.6.1.3 Definitions from TM Synchronization and Channel Coding

This Recommended Standard makes use of the following terms defined in reference [2]:

a) Attached Sync Marker;

b) Reed-Solomon check symbols;

c) Reed-Solomon code.

1.6.1.4 Definitions from TM Space Data Link Protocol

This Recommended Standard makes use of the following term defined in reference [3]:

a) Frame Error Control Field (FECF);

b) TM Transfer Frame.

1.6.1.5 Definitions from AOS Space Data Link Protocol

This Recommended Standard makes use of the following terms defined in reference [4]:

a) AOS Transfer Frame;

b) Frame Error Control Field (FECF);

1.6.1.6 Definitions from SLE Reference Model

This Recommended Standard makes use of the following terms defined in reference [1]:

a) abstract binding;

b) abstract object;

c) abstract port;

d) abstract service;

e) invoker;

f) Mission Data Operation System (MDOS);

g) Mission User Entity (MUE);

h) offline delivery mode;

i) online delivery mode;

j) operation;

k) performer;

l) physical channel;

m) return data;

n) Return All Frames channel (RAF channel);

o) Return All Frames service (RAF service);

p) Return Master Channel Frame Service (MC service)

q) Return Virtual Channel Frame Service (VC Frame service)

r) service agreement;

s) service provider (provider);

t) service user (user);

u) SLE Complex;

v) SLE Complex Management;

w) SLE data channel;

x) SLE Functional Group (SLE-FG);

y) SLE Protocol Data Unit (SLE-PDU);

z) SLE Service Data Unit (SLE-SDU);

aa) SLE service package;

bb) SLE transfer service instance;

cc) SLE transfer service production;

dd) SLE transfer service provision;

ee) SLE Utilization Management;

ff) space link;

gg) space link data channel;

hh) Space Link Data Unit (SL-DU);

ii) space link session.

1.6.1.7 Additional Definitions

1.6.1.7.1 Association

An association is a cooperative relationship between an SLE service-providing application entity and an SLE service-using application entity. An association is formed by the

exchange of SLE protocol data units through the use of an underlying communications service.

1.6.1.7.2 Communications Service

A communications service is a capability that enables an SLE service-providing application entity and an SLE service-using application entity to exchange information.

NOTE – If an SLE service user and an SLE service provider are implemented using different communications services, then interoperability between them is possible only by means of a suitable gateway. Adherence to this Recommended Standard ensures, at least in principle, that it is possible to construct such a gateway.

1.6.1.7.3 Confirmed Operation

A confirmed operation is an operation that requires the performer to return a report of its outcome to the invoker.

1.6.1.7.4 Delivery Criteria

Delivery criteria are rules that determine whether a data unit acquired from the space link by an SLE service provider shall be delivered to a user.

NOTE – For RCF service, the delivery criteria are:

a) the Earth Receive Time (ERT) of the frame is within the period defined by the start and stop times specified in the RCF-START operation;

b) the spacecraft identifier (SCID) of the frame matches the SCID of the global VCID specified in the RCF-START operation; and

c) the Virtual Channel Identifier (VCID) of the frame matches the VCID of the global VCID specified in the RCF-START operation.

1.6.1.7.5 Frame Error Control Field

The Frame Error Control Field (FECF) of a frame is the FECF of a TM Transfer Frame (reference [3]) or the FECF of an AOS Transfer Frame (reference [4]), as applicable.

1.6.1.7.6 Frame Version Number

The frame version number is either the transfer frame version number (reference [3]) or the version number in the AOS transfer frame primary header (reference [4]).

NOTE – The definitions of frame version number given in references [3] and [4] are equivalent. If a CCSDS-compatible telemetry frame is known to contain no errors, the frame version number enables one to distinguish between a transfer frame and an AOS transfer frame.

1.6.1.7.7 Initiator

The initiator is the object that issues the request to bind to another object (the responder).

NOTE – In other words, the initiator is always the invoker of the request to bind to another object. Therefore, in the context of the request to bind, the terms 'initiator' and 'invoker' refer to the same object and are synonyms.

1.6.1.7.8 Invocation

The invocation of an operation is the making of a request by an object (the invoker) to another object (the performer) to carry out the operation.

1.6.1.7.9 Master Channel

The sequence of all telemetry frames with the same Transfer Frame Version Number (TFVN) and the same SCID on the same physical channel constitutes a master channel.

NOTE – Depending on the TFVN, the definition of SCID is as given in either reference [3] or reference [4].

1.6.1.7.10 Parameter

A parameter of an operation is data that may accompany the operation's invocation or return.

NOTE – The term parameter is also used to refer to mission-dependent configuration information used in the production or provision of the service.

1.6.1.7.11 Performance

The performance of an operation is the carrying out of the operation by an object (the performer).

1.6.1.7.12 Port Identifier

A port identifier identifies a source or a destination in a communications system.

NOTE – See 2.6.4.5 for more information.

1.6.1.7.13 Responder

The responder is the object that receives a request to bind and completes the binding (if possible) with the initiator in order for a service association to exist between the two objects.

NOTE – In other words, the responder is always the performer of the binding. Therefore, in the context of binding, the terms 'responder' and 'performer' refer to the same object and are synonyms.

1.6.1.7.14 Return

The return of an operation is a report, from the performer to the invoker, of the outcome of the performance of the operation.

1.6.1.7.15 Service Instance Provision Period

A service instance provision period is the time during which a service instance (i.e., the capability to transfer one or more SLE data channels of a given type) is scheduled to be provided.

NOTE – Reaching of the beginning of this period constitutes the event 'start of service instance provision period' (see 4.2.2).

1.6.1.7.16 Spacecraft Identifier

The spacecraft identifier (SCID) of a telemetry frame is as defined in reference [3] if the frame is a TM Transfer Frame or as defined in reference [4] if the frame is an AOS Transfer Frame.

1.6.1.7.17 Telemetry Frame

A telemetry frame is a TM Transfer Frame (as defined in reference [3]) or an AOS Transfer Frame (as defined in reference [4]). In case a distinction of the frame versions is necessary, the full term as per references [3] or [4] is used.

1.6.1.7.18 Transfer Frame Version Number

The Transfer Frame Version Number (TFVN) is either the TFVN as defined in reference [3] or the TFVN as defined in reference [4].

NOTE – The definitions of TFVN given in references [3] and [4] are equivalent. If a CCSDS-compatible telemetry frame is known to contain no errors, the TFVN enables one to distinguish between a TM Transfer Frame and an AOS Transfer Frame.

1.6.1.7.19 Unconfirmed Operation

An unconfirmed operation is an operation that does not require a report of its outcome to be returned to the invoker by the performer.

1.6.1.7.20 Virtual Channel

All telemetry frames with the same TFVN, the same SCID, and the same virtual channel identifier (VCID) on the same physical channel constitute a virtual channel.

1.6.1.7.21 Virtual Channel Identifier

The virtual channel identifier (VCID) of a telemetry frame is as defined in reference [3] if the telemetry frame is a TM Transfer Frame or as defined in reference [4] if the telemetry frame is an AOS Transfer Frame.

1.6.2 NOMENCLATURE

The following conventions apply throughout this Recommended Standard:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

1.6.3 CONVENTIONS

1.6.3.1 Specification of Operations

1.6.3.1.1 General

Section 3 of this Recommended Standard specifies the operations that constitute the RCF service. The specification of each operation is divided into subsections as described in 1.6.3.1.2 through 1.6.3.1.4.

1.6.3.1.2 Purpose Subsection

The Purpose subsection provides a brief description of the purpose of the operation. Additionally, it indicates whether the operation may be invoked by the user, provider, or both; whether the operation is confirmed or unconfirmed; and whether there are any constraints on when the operation may be invoked.

1.6.3.1.3 Invocation, Return, and Parameters Subsection

The Invocation, Return, and Parameters subsection describes the parameters associated with each operation, including their semantics. A accompanying the description of each operation lists all parameters associated with the operation and, for both the invocation and return, whether the parameter is always present, always absent, or conditionally present.

For parameters that are conditionally present, the parameter description specifies the conditions for the presence or absence of the parameter. The condition is generally based on the value of another parameter in the same invocation or return; for example, in the return of an operation, the `diagnostic` parameter is present if and only if the value of the `result` parameter is 'negative result'. For a conditional parameter in a return, the condition may be based on the value of a parameter in the corresponding invocation.

In the table, the following convention is used to indicate whether a parameter is always present, always absent, or conditionally present:

M	Always present
C	Conditionally present
Blank	Always absent

NOTE – Even though a parameter may be characterized as always present, its description may specify that its value is permitted to be 'null' or 'unused' or the like.

1.6.3.1.4 Effects Subsection

The Effects subsection describes the effects an operation has on the invoker, the performer, the association between them, or any combination thereof. The details of how those effects occur or the mechanisms used are outside the scope of this Recommended Standard.

1.6.3.2 Typographic Conventions

1.6.3.2.1 Operation Names

Names of RCF service operations appear in uppercase and begin with the characters 'RCF-' (e.g., RCF-TRANSFER-DATA).

1.6.3.2.2 Parameter Names

In the main text, names of parameters of RCF service operations generally appear in lowercase and are typeset in a fixed-width font (e.g., `responder-port-identifier`). In annex A, the corresponding name is formed by omitting any hyphens contained in the name and using mixed-case (e.g., `responderPortIdentifier`).

1.6.3.2.3 Value Names

The values of many parameters discussed in this Recommended Standard are represented by names. In the main text, those names are shown in quotation marks (e.g., 'no such service instance'). The corresponding name in annex A is formed by omitting any hyphens or white space contained in the name and using mixed-case (e.g., `noSuchServiceInstance`). The actual value associated with the name is constrained by the type of the parameter taking on that value. Parameter types are specified in annex A of this Recommended Standard.

NOTE – The name of a value does not imply anything about its type. For example, the value 'no such service instance' has the appearance of a character string but might be assigned to a parameter whose type is 'integer'.

1.6.3.2.4 State Names

This Recommended Standard specifies the states of RCF service providers. States may be referred to by number (e.g., state 2) or by name. State names are always shown in quotation marks (e.g., 'active').

1.6.3.2.5 SLE-PDU Names

The names of SLE-PDUs appear in mixed-case (e.g., rcfBindInvocation).

1.6.3.2.6 Data Type Definitions

Data type definitions for the RCF service are presented in annex A in the form of a set of ASN.1 modules. Regardless of the conventions used elsewhere in this Recommended Standard, the text of the ASN.1 modules is typeset entirely in a fixed-width font.

1.6.3.3 Other Conventions

This Recommended Standard uses the conventions specified in reference [1].

1.7 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommended Standards.

NOTES

1 A list of informative references is provided in annex E.

2 This document takes advantage of the harmonized terminology introduced by restructured documentation of the space link protocols (references [2], [3], and [4]). From an interoperability point of view, they do not introduce any incompatibilities with respect to the original set of space link protocol documents (references [E3], [E4], and [E5]).

[1] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Recommendation for Space Data System Standards, CCSDS 910.4-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, October 2005.

[2] *TM Synchronization and Channel Coding*. Recommendation for Space Data System Standards, CCSDS 131.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[3] *TM Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 132.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[4] *AOS Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 732.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, July 2006.

[5] *Time Code Formats*. Recommendation for Space Data System Standards, CCSDS 301.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, January 2002.

[6] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1:1994. 2nd ed. Geneva: ISO, 1994.

[7] *Information Technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)*. International Standard, ISO/IEC 8824-1:2002. 3rd ed. Geneva: ISO, 2002.

2 DESCRIPTION OF THE RETURN CHANNEL FRAMES SERVICE

2.1 OVERVIEW

The RCF service enables the user of the service to obtain all telemetry frames from one master channel or one virtual channel. A master channel consists of all telemetry frames with the same Transfer Frame Version Number (TFVN) and the same spacecraft identifier (SCID) on the same physical channel. A virtual channel consists of all telemetry frames with the same TFVN, the same SCID, and the same virtual channel identifier (VCID) on the same physical channel. A telemetry frame is a TM Transfer Frame or a AOS Transfer Frame. A space link physical channel carries one stream of telemetry frames separated by attached sync markers. A physical channel may be comprised of one or more master channels, each of which may be comprised of one or more virtual channels. A complete specification of these concepts is provided in references [2], [3], and [4].

For delivery to the user, each frame acquired from the space link is encapsulated in an SLE SDU that also carries annotation, i.e., additional information such as the Earth Receive Time (ERT) of the frame. In general, the RCF service delivers frames to the user in the order in which they were received from the space link.

The operations defined in section 3 of this Recommended Standard enable an RCF service user to interact with an RCF service provider to:

a) establish an association between the user and the provider;

b) receive annotated telemetry frames from one master channel or from one virtual channel;

c) obtain notifications and reports regarding the status, configuration and performance of the service;

d) temporarily suspend and later re-start the delivery of telemetry frames from the same master channel or any of the permitted virtual channels;

e) change the values of certain parameters that affect the behavior of the service; and

f) release an association.

In any given service instance, only one master channel, or only one VC, or a single master channel plus a set of VCs (where the set may have a single member), is permitted. Only the permitted master channel or one of the permitted virtual channels is delivered to the user at a time.

The provision of RCF service for one master channel or one virtual channel for access by one service user constitutes one instance of service. The provision of RCF service for one master channel or one virtual channel to multiple service users and the provision of RCF service for multiple master channels or multiple virtual channels concurrently to one or more service users are permitted but are specified to constitute multiple service instances.

2.2 SPACE LINK EXTENSION REFERENCE MODEL

2.2.1 INTRODUCTION

The RCF service is specified within the framework defined by the SLE Reference Model (reference [1]). The following subsections summarize selected concepts from the SLE Reference Model.

2.2.2 ABSTRACT OBJECT

An abstract object is a functional entity that interacts with other abstract objects. Objects are of different types, which determine their function and behavior. Objects are characterized by their interfaces, which are called abstract ports, and the operations that are made available through those interfaces. One object may provide multiple abstract ports.

2.2.3 ABSTRACT SERVICE

An abstract service is the capability provided by a set of operations that an abstract object exposes at one or more of its abstract ports.

NOTE – The concept of an abstract service is to be distinguished from the concept of an (N)-service as defined in the OSI Basic Reference Model (reference [6]). The definition of (N)-service is in terms of the capability provided by one layer in the OSI architecture to the layer above it. The definition of abstract service is in terms of the capability provided by one abstract object to another abstract object. In a cross support scenario where one Agency is providing an SLE service to another Agency, the object that provides the service typically is associated with one Agency, and the object that uses the service typically is associated with the other Agency.

2.2.4 ABSTRACT BINDING

When two abstract ports have an association established between them, they are said to be bound. The act of establishing such an association is called abstract binding. One object (the initiator) invokes a bind operation that is accepted (or rejected) by another object (the responder).

2.2.5 SERVICE USER/PROVIDER

An object that offers a service to another by means of one or more of its ports is called a service provider (provider). The other object is called a service user (user). An object may be a provider of some services and a user of others.

The terms user and provider are used to distinguish the roles of two interacting objects. In this Recommended Standard, when two objects are involved in provision of a service, the

object closer to the space link is considered to be the provider of the service, and the object further from the space link is considered to be the user.

2.2.6 OPERATION

An operation is a procedure or task that one object (the invoker) can request of another (the performer) through a bound port pair. The terms invoker and performer are used to describe the interaction between two objects as the operations that constitute the service occur. One object invokes an operation that is performed by the other. For most services, each object invokes some operations and performs others.

2.3 SERVICE MANAGEMENT

SLE service management determines the number and schedule of RCF service instances to be provided, the resources required to enable those service instances, and the initial configuration of all service instances and their supporting resources. SLE service management is the subject of separate CCSDS Recommended Standards.

The SLE Reference Model (reference [1]) distinguishes between service provision and service production:

a) service provision makes available to the user the operations necessary to obtain the service;

b) service production transforms a space link channel to an RCF channel, possibly using the service provision and production of another SLE provider or the equivalent capability.

Certain configuration parameters are associated with provision of RCF services, while others are associated with production. Changes to RCF provision configuration parameters (e.g., `requested-global-VCID`) affect only a single service instance; the values of such parameters are initialized by service management when the service instance is created, but may be modified subsequently by the user through RCF service operations specified in this Recommended Standard. Changes to RCF production configuration parameters (e.g., bit rate, frame length, coding type) potentially affect multiple service instances or potentially impact SLE Complex resources; consequently, those parameters may be modified only through service management.

RCF service may be user-initiated (i.e., the user invokes the bind operation) or provider-initiated (i.e., the provider invokes the bind operation). A particular instance of RCF service shall support either user initiation or provider initiation but not both. The form of initiation that applies to a particular service instance is set by service management.

The SLE Reference Model defines two delivery modes: online delivery mode and offline delivery mode. Online delivery mode indicates that the provision of service is generally coincident in time with the space link session, whereas offline delivery mode indicates that the telemetry frames acquired during a space link session are provided to the user some time

after the end of the space link session. Within the online delivery mode, the SLE Reference Model defines two quality factors: timeliness and completeness. Within this RCF service specification, the two variants of online delivery are regarded distinct delivery modes: online timely and online complete. Both assume the use of a reliable communications service. They differ in that the timely mode allows for the controlled discarding of telemetry frames at the application layer if it is not possible to deliver those telemetry frames within a certain amount of time after they are acquired from the space link (e.g., because of communications service backlog). While the RCF service is defined for the complete online delivery mode, the timely online delivery mode, or the offline delivery mode, any particular instance of RCF service shall support only one of those modes. The delivery mode applicable to a particular service instance is set by service management.

2.4 ARCHITECTURE MODEL—FUNCTIONAL VIEW

2.4.1 RETURN FRAME PROCESSING FUNCTIONAL GROUP

The Return Frame Processing Functional Group (RFP-FG, shown in figure 2-1) is the SLE functional group (SLE-FG) that produces the RCF service. As shown in the figure, the RFP-FG provides other services in addition to the RCF service. This Recommended Standard addresses only the RCF service; the other services are to be defined in companion Recommended Standards.

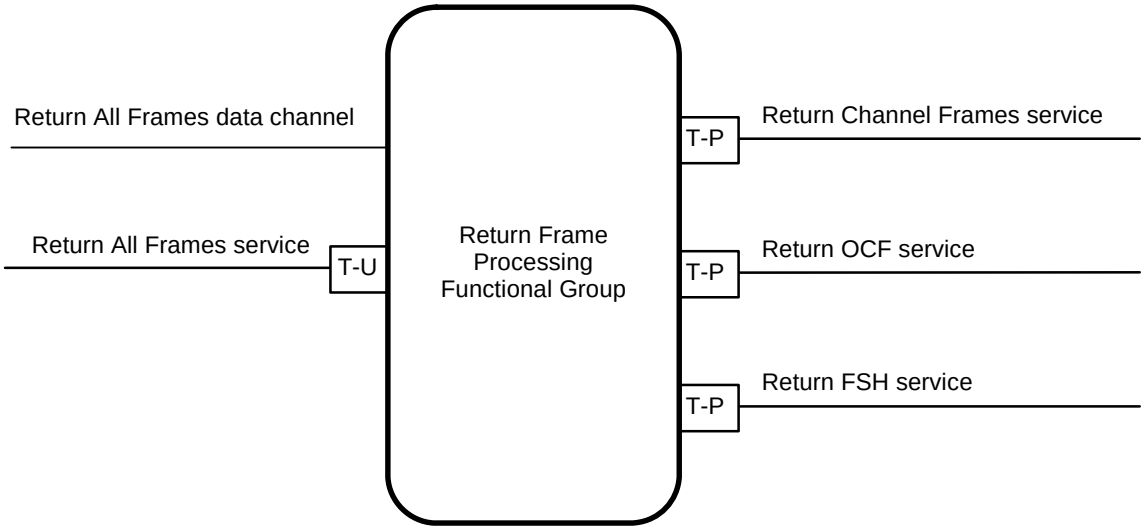

Figure 2-1: Return Frame Processing SLE-FG

Return Frame Processing Functional Group (RFP-FG, shown in figure 2-1), is the SLE functional group (SLE-FG) that produces the RCF service. As shown in the figure 2-1, the RFP-FG provides other services in addition to the RCF service. This Recommended Standard addresses only the RCF service; the other services are (to be) defined in companion Recommended Standards.

As described in reference [1], the RFP-FG consumes a Return All Frames (RAF) channel and provides RCF service. The RAF channel is provided to the RFP-FG either (a) directly from a Return Space Link Processing Functional Group (RSLP-FG) in the same SLE Complex, or (b) through an instance of RAF transfer service that is provided by a different SLE Complex. The RAF channel consists of a stream of SLE-SDUs that encapsulate the telemetry frames acquired from one space link physical channel. From this input, the RFP-FG produces one or more RCF channels. Each RCF channel consists of a stream of SLE-SDUs. Most of these SLE-SDUs encapsulate telemetry frames from the master channel or virtual channel specified by the user of the RCF service; such SLE-SDUs also carry annotation information associated with the frame (e.g., the ERT of the frame). Other SLE-SDUs in an RCF channel carry notifications of the occurrences of certain events that may pertain to the RCF service (e.g., loss of frame synchronization on the physical channel associated with this instance of RCF service).

An RCF channel produced by the RFP-FG is delivered to a user by means of the RCF service. More specifically, the RFP-FG performs the following functions with respect to RCF service:

a) consumes one RAF channel;

b) demultiplexes the RAF channel into its component master channels, demultiplexes the master channels into their component virtual channels, annotates each frame in each channel to form RCF SLE-SDUs, and injects the resulting RCF SLE-SDU into RCF channels;

c) optionally, stores (and subsequently retrieves) sufficient data to reconstruct the RCF channels for delivery through one or more offline RCF service instances;

d) makes RCF channels available to online and offline RCF service instances to effect the provision of RCF service.

The telemetry frames delivered by the RCF service are encapsulated in SLE-SDUs. RCF SLE-SDUs that encapsulate telemetry frames are annotated with information that pertains to that specific frame. The annotation consists of:

a) the ERT of the frame;

b) an identifier that indicates the antenna used to acquire the frame;

c) a parameter that characterizes the data link continuity of this frame with respect to the preceding frame on the same master or virtual channel;

d) an optional octet string that may be used to provide additional, non-standard annotations that are mutually agreed to by the SLE Complex providing the service and the Mission Data Operations System (MDOS) associated with the user of the service.

NOTES

1 The RCF service delivers only telemetry frames that are error-free. The determination that a frame is error-free is based on the frame quality annotation provided by the RAF service production: a frame is considered error-free if it was annotated by the RAF service production with a frame quality of 'good'. The RAF service production annotates a frame as 'good' if the frame contains only valid codewords of the Reed-Solomon code or—if the frame is not Reed-Solomon encoded—if the frame error control field (FECF) decodes successfully.

2 As defined by this Recommended Standard, telemetry frames do not include the attached sync marker but are delimited by them.

2.4.2 RCF SERVICE PRODUCTION AND PROVISION

One instance of RCF service production (or, one RFP-FG instance) may be associated with multiple RCF service instances. RCF production is concerned with the production of RCF channels independent of any particular instance of service. In contrast, RCF service provision is concerned with delivering an RCF channel to an RCF service user. Service provision addresses such matters as when service is provided (e.g., service start and stop times), how service is provided (e.g., user-initiated or provider-initiated), and delivery service (e.g., whether the delivery mode is timely online, complete online, or offline).

RCF service production receives the input telemetry frames encapsulated in RAF SLE-SDUs. If the complete production process, i.e., RAF and RCF production are performed within a single SLE Complex, then the RAF SLE-SDUs are possibly not exposed on an interface but exist only conceptually inside the return link production. If a separate SLE Complex hosts the RAF production process, then the RAF SLE-SDUs are made available by that complex by means of the RAF service exposed on the SLE Complex interface.

RCF service production receives the input telemetry frames encapsulated in RAF SLE-SDUs. The RCF service production performed by the RFP-FG separates the RAF SLE-SDUs by master channel and virtual channel to form RCF SLE-SDUs. The SLE-SDUs generated by RCF service production are delivered to the service user by means of the RCF service operations defined in section 3, which also provide additional functionality to facilitate the provision of RCF service. In turn, the RCF service operations are realized as SLE Protocol Data Units (SLE-PDUs) that are exchanged between the RCF service provider and the RCF service user by means of an underlying communications service. Typically, an SLE-PDU corresponds to the invocation or return of an RCF operation. (Because of the buffering mechanisms described in 3.1.9, there are certain exceptions; for example, multiple RCF-TRANSFER-DATA invocations may be mapped to a single SLE-PDU.) The general relationship between RAF SLE-SDUs, RCF SLE-SDUs, and RCF SLE-PDUs is illustrated in figure 2-2. This figure assumes that as in the example shown in figure 2-3 all return service production is implemented within a single SLE Complex. This may or may not be the case. For instance, one SLE Complex could host the RCF production process and use the RAF service provided by another SLE Complex. In that case, the latter SLE Complex would

consume the space link channel, i.e., receive the SL-DUs while the two SLE Complexes would exchange RAF SLE-PDUs.

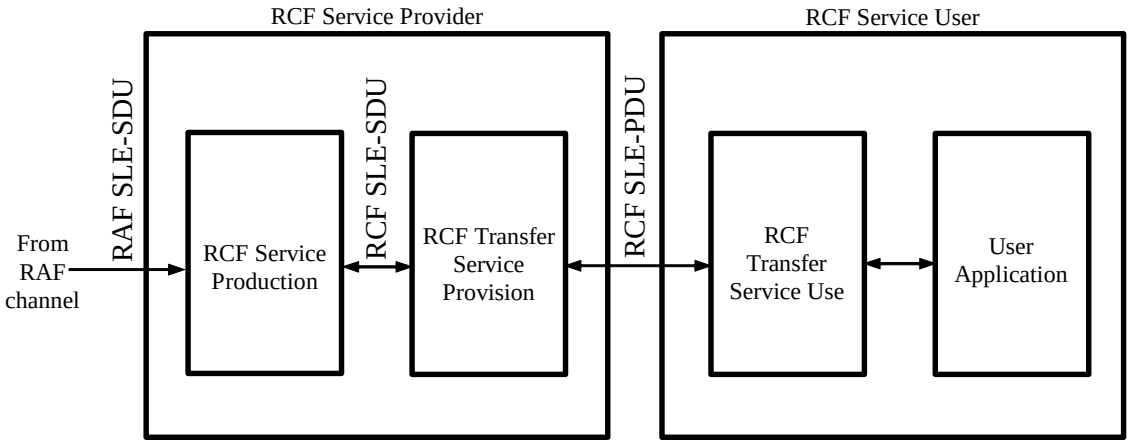

Figure 2-2: RCF Service Production and Provision

For the online delivery mode, production and provision of the RCF service by the provider occur, at least in part, concurrently with the space link session. For the offline delivery mode, service production and provision are detached, with service provision occurring some time after the end of the space link session. In the offline case, data acquired during the space link session are stored for later delivery by an offline service instance.

2.5 ARCHITECTURE MODEL—CROSS SUPPORT VIEW

The management and control of the production and provision of SLE transfer services is described in general terms in reference [1]. Figure 2-3 shows an example operational scenario and the related binding of RCF transfer service ports and SLE management ports. This scenario shows an SLE Complex with one Return Space Link Processing SLE-FG instance and one Return Frame Processing SLE-FG instance providing two instances of RCF service to an MDOS.

NOTE – Although not shown in this scenario, other combinations are possible. For example, it is also possible to have several RFP-FG instances, each consuming a different RAF channel and each providing one or more instances of service. It is also possible for the RSLP-FG and the RFP-FG to be located in different SLE Complexes. In such a case, the RAF channel would be provided to the RFP-FG via an RAF transfer service instance.

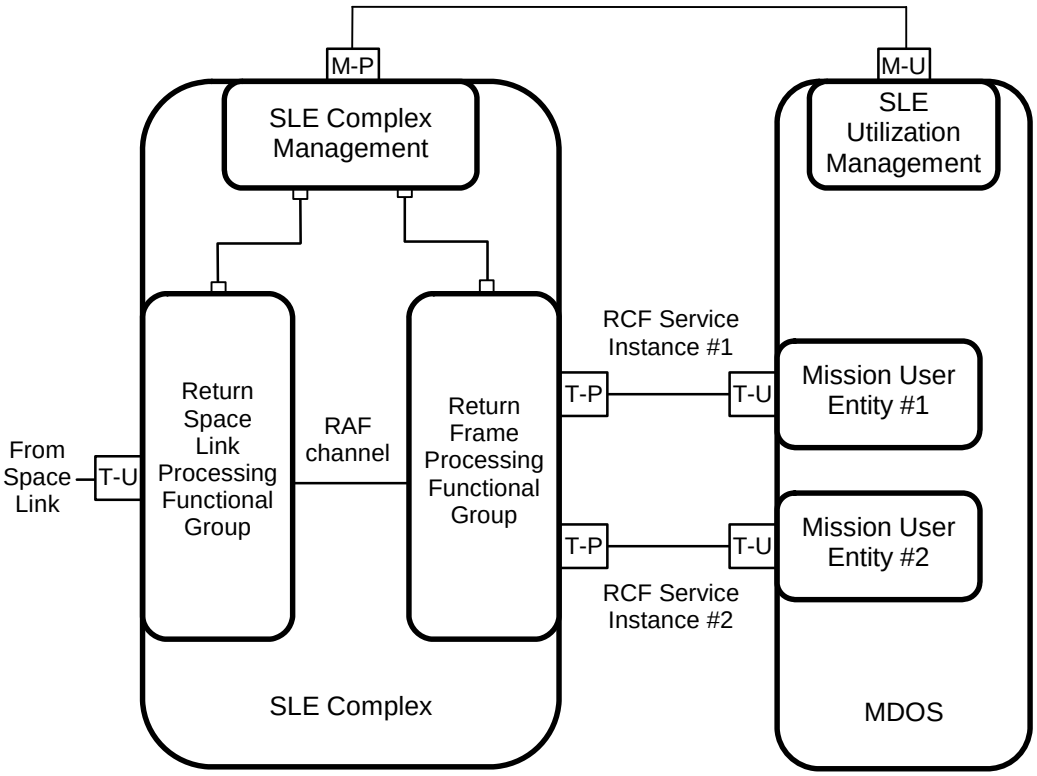

Figure 2-3: Example of the Management and Provision of RCF Service

2.6 FUNCTIONAL DESCRIPTION

2.6.1 GENERAL

Subsections 2.6.2 through 2.6.4 describe the RCF service with respect to scheduling, configuration, underlying services, provider states, and protocol considerations.

2.6.2 SCHEDULING AND CONFIGURATION

SLE Utilization Management negotiates with SLE Complex Management to establish mutually agreed upon SLE service packages. Among other things, SLE service packages specify what service instances are to be provided, when those services are to be provided, and what resources are needed to enable those services.

Service packages also specify the initial values of mission-dependent parameters required for service production and provision. RCF service provision parameters include such things as the scheduled start and stop times of the RCF service instance and the delivery mode.

Service production is guaranteed to occur only as needed to support service packages that have been scheduled and mutually agreed upon by SLE Complex Management and SLE

Utilization Management. Service provision occurs only within the bounds of the agreed upon schedule of service instances and only during those periods when there is an association between the service provider and the service user.

2.6.3 UNDERLYING SERVICES

The RCF service is based on the functionality provided by the SLE RAF service production (reference [1]) or an equivalent capability. The RAF service production (or its equivalent) may be provided by the same SLE Complex that provides the RCF service or by a different SLE Complex. Additionally, provision of RCF service depends on service management for scheduling, resources, and configuration, and on the availability of a suitable communications service to enable the exchange of information between the RCF service user and provider.

2.6.4 PROTOCOL DESCRIPTION

2.6.4.1 RCF Operations

The operations that constitute the RCF service are listed in table 2-1. Section 3 of this Recommended Standard provides the detailed specification of these operations.

Table 2-1: RCF Operations

Operation	Invoked By	Purpose	Confirmed
RCF-BIND	User or provider	To establish an association with the peer	Yes
RCF-UNBIND	User or provider	To release an association previously established by an RCF-BIND operation	Yes
RCF-START	User	To request that the SLE service provider start the delivery of telemetry frames	Yes
RCF-STOP	User	To request that the SLE service provider stop the delivery of telemetry frames	Yes
RCF-TRANSFER-DATA	Provider	To transfer a telemetry frame to the SLE service user	No
RCF-SYNC-NOTIFY	Provider	To notify the user of an event affecting production or provision of the RCF service	No

Operation	Invoked By	Purpose	Confirmed
RCF-SCHEDULE-STATUS-REPORT	User	To request that the provider send a status report immediately or periodically or to stop such reporting	Yes
RCF-STATUS-REPORT	Provider	To send a status report to the user	No
RCF-GET-PARAMETER	User	To ascertain the value of an SLE service parameter (see 3.10.2.5.1)	Yes
RCF-PEER-ABORT	User or Provider	To notify the peer that the local SLE application detected an error that requires the association to be terminated	No

2.6.4.2 States of the Service Provider

Once an RCF service instance is created, the RCF service provider is in one of three states, as follows:

a) State 1 (‘unbound’): In state 1, all resources required to enable the provision of the RCF service have been allocated, and all objects required to provide the service have been instantiated. However, no association yet exists between the user and the provider (i.e., the RCF transfer service provider port is not bound).

b) State 2 (‘ready’): In state 2, an association has been established between the user and the provider, and they may interact by means of the operations described in section 3 of this Recommended Standard. However, the delivery of telemetry frames (by means of the RCF-TRANSFER-DATA operation) is not permitted. The user may enable the delivery of telemetry frames by means of the appropriate service operation (RCF-START); that, in turn, will cause the provider to transition to state 3 (‘active’) and enable frame delivery.

c) State 3 (‘active’): State 3 is identical to state 2 (‘ready’) except that telemetry frames that meet the delivery criteria specified by the user by means of the RCF-START operation are delivered to the user as they become available. The service continues in this state until the user invokes the RCF-STOP operation to suspend frame delivery and transition back to state 2 (e.g., in response to an ‘end of data’ notification from the service provider signaling that the space link session has ended and all available frames have been delivered) or all RCFs meeting the user selected delivery criteria (see 1.6.1.7.4) have been sent).

A simplified RCF service provider state transition diagram is shown in figure 2-4. A detailed state transition matrix is provided in section 4.

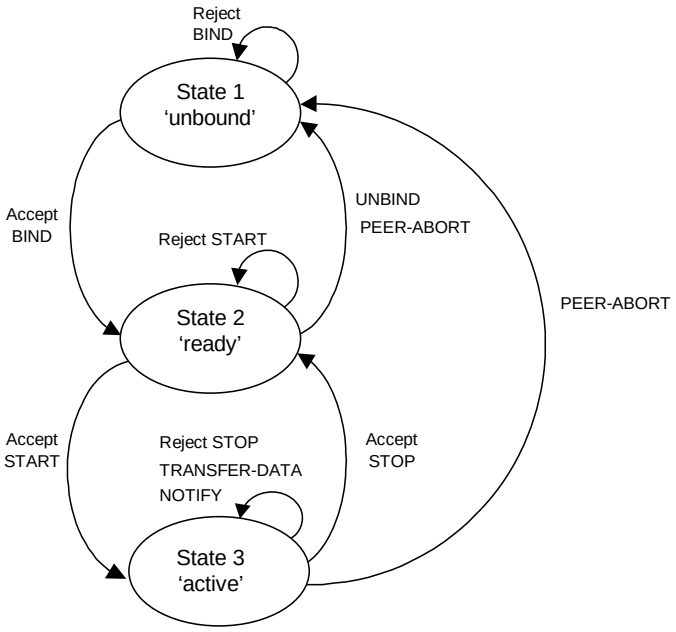

Figure 2-4: Simplified RCF Service Provider State Transition Diagram

2.6.4.3 Termination of Association

An association is released normally when an RCF-UNBIND operation is invoked by the initiator of the association and performed by the responder. An association may be aborted by either the user or the provider by means of the RCF-PEER-ABORT operation. An association also may be aborted due to certain failures of the communications service; such failures are signaled to the local application by the 'protocol abort' event described in 4.1.5.

2.6.4.4 Effects of Termination

When an association is released or aborted, no further operations shall be invoked by the user or the provider. As a consequence, the delivery of frames stops immediately. The user and provider may re-establish an association via a new RCF-BIND operation if that is consistent with the schedule for the provision of service. However, status information from the prior association is not preserved and is not available to the new association except that:

a) if the delivery mode is complete online, the contents of the online frame buffer (see 3.1.9) shall be persistent;

b) if the delivery mode is offline, the contents of the offline frame buffer (see 3.1.9) shall be persistent;

c) statistics reported by means of the RCF-STATUS-REPORT operation (see 3.9) shall be accumulated for the entire service instance provision period.

2.6.4.5 Technology-specific Aspects

This Recommended Standard defines the RCF service. Provision of the RCF service in a real system also requires a specification of how the RCF service defined here is mapped to a communications service such that all invocations and returns of RCF service operations can be conveyed between the user and the provider. In order not to restrict the applicability of this Recommended Standard to a specific communications technology, as few assumptions as possible have been made about the characteristics of the underlying communications service (see 1.3.1).

The RCF service interface between the user and the provider is specified in this Recommended Standard in terms of the operations that the service provides. Those operations are realized by mapping the operation invocations and returns to protocol data units that can be exchanged by means of the underlying communications service. This Recommended Standard conceptualizes such a mapping in two parts. First, RCF service operation invocations and returns (defined in section 3) are mapped to SLE-PDUs (defined in annex A). Second, SLE-PDUs are mapped to protocol data units that can be exchanged by means of the underlying communications service. The mapping of RCF service operation invocations and returns to SLE-PDUs is specified by this Recommended Standard. The mapping of SLE-PDUs to an underlying communications service is intentionally outside the scope of this Recommended Standard (e.g., so that the RCF service may be mapped to more than one communications technology). In order to achieve interoperability, the user and provider must conform not only to this Recommended Standard but also to an agreed upon specification of the mapping of the RCF service to the underlying communications service. Figure 2-5 illustrates a communications realization of the RCF service that results from such a mapping. The specification of such mappings is the subject of separate CCSDS Recommended Standards.

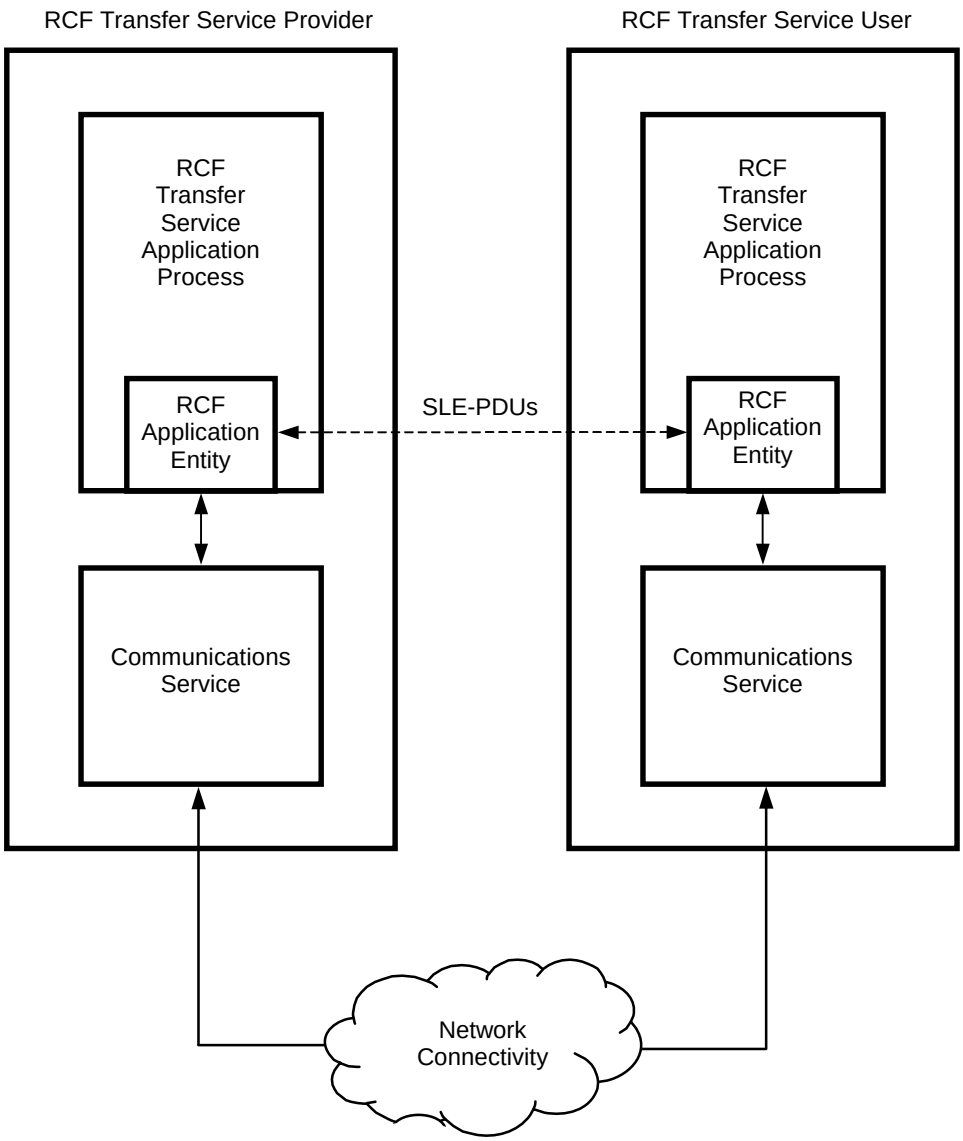

Figure 2-5: Communications Realization of RCF Service

The specification of a mapping of the RCF service onto a particular communications service must address such things as:

a) selection of communication networks to ensure connectivity;

b) compatible configuration of protocol stacks;

c) specification of port-identifiers and their translation onto the underlying communications service; and

d) specification of security related information.

Because the operations of the RCF service are relatively simple, once an association is in place between the service user and the service provider, the technology-specific elements

involved in the exchange of SLE-PDUs are generally minor. However, the way an association is established (i.e., the binding) tends to vary significantly depending on the communications technology in use. Nonetheless, the RCF-BIND and RCF-UNBIND operations as specified in this document are intended to be 'technology neutral'. This neutrality is achieved as described in the following subsections.

For purposes of the communications mapping, the endpoints of an SLE association are identified by port identifiers, namely, an 'initiator port identifier' and a 'responder port identifier'. The port identifiers represent all the technology-specific addressing information needed to establish communications between the user and provider and to route SLE-PDUs between them. The initiator port identifier identifies the endpoint that will invoke the RCF-BIND operation (initiator). The responder port identifier identifies the endpoint that will perform the RCF-BIND operation (responder). Generally speaking, the information represented by a port identifier consists of:

a) information needed in order to route data between two real systems over a communications channel or network; and

b) information needed in order to route data within a real system to a particular application entity.

For example, the information represented by a port identifier might be the combination of an Internet Protocol (IP) network address and a Transmission Control Protocol (TCP) port number or the combination of an OSI network address and an associated set of Service Access Points (SAPs).

The exact relationship between SLE port identifiers and communications ports provided by the underlying communications service must be specified by the mapping of the RCF service to the underlying communications service. If the underlying communications service is connection-oriented, then the mapping may specify a one-to-one relationship between SLE associations and communications connections; however, that is not required. For example, two SLE associations involving the same pair of SLE endpoints may share a single connection. In that case, it is the responsibility of the mapping of the RCF service to the underlying communications service to specify how the SLE-PDUs of one association are distinguished from the SLE-PDUs of the other association.

One possible mapping of the SLE transfer service to the TCP/IP communications service is specified in [E6]. As part of this mapping, also issues such as sizing of TCP buffers in accordance with the bandwidth-delay product of the communication link and ways to manage relative priority of transfer services concurrently using the same connectivity are to be addressed.

In order for an SLE association to be established, SLE Complex Management and SLE Utilization Management must agree beforehand on the responder port identifier for the association. The responder needs the information represented by the responder port identifier to ensure that resources are allocated to recognize and respond to an RCF-BIND

invocation for that association. The initiator needs the information to ensure that the RCF-BIND invocation will be communicated to the appropriate responder.

In general, it is not necessary for SLE Complex Management and SLE Utilization Management to agree beforehand on the initiator port identifier for the association. Rather, the initiator should communicate that information to the responder in conjunction with the RCF-BIND invocation. The exact means by which the initiator port identifier is provided to the responder is technology-specific and must be specified by the mapping of the RCF service to the underlying communications service.

The responder port identifier is included as a parameter of the RCF-BIND operation. Generally speaking, that is unnecessary; it is only necessary that SLE application communicate the information represented by the port identifiers to the underlying communications service. The responder port identifier is provided as a parameter of the RCF-BIND operation to allow for the possibility that the implementation of a gateway might be simplified by the inclusion of this parameter in the RCF-BIND operation.

The information represented by the responder port identifier is technology-specific. In order to define the RCF-BIND operation in a way that is not technology-specific, the `responder-port-identifier` parameter of the RCF-BIND operation is defined to be a logical name. A logical name is an arbitrary identifier that has an appropriately chosen and agreed upon translation to technology-specific information. Prior to the start time of a service instance, SLE Complex Management and SLE Utilization must mutually agree upon the value of the responder port identifier (and its translation) applicable to that service instance. The actual process of translating logical names to technology-specific information is considered a local matter. The translation methodology may rely on simple techniques such as look-up tables or may use more elaborate mechanisms such as naming or directory services.

The above discussion describes the case that both the user and provider applications are implemented using the same communications service. It is possible to achieve interoperability even if the user and provider use different communications services. However, in that case interoperability requires the use of an appropriate gateway.

2.6.4.6 Buffering

2.6.4.6.1 General

Buffering mechanisms used by the RCF protocol are described in subsections 2.6.4.6.2 and 2.6.4.6.3. They are formally specified by the requirements in 3.1.9 and the state transition matrix in section 4.

2.6.4.6.2 Transfer Buffer

As described in 2.6.4.5, RCF operations (specified in section 3) are mapped to SLE-PDUs (specified in annex A) that are conveyed to the peer SLE entity by means of the underlying

communications service. In general, there is a one-to-one mapping between SLE-PDUs and the invocations or returns of RCF operations. However, that is not always the case. In particular, the RCF protocol provides that multiple RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY operations may be mapped to a single SLE-PDU (viz., the SLE-PDU named RcfTransferBuffer in annex A). In terms of the RCF service, the release of the RcfTransferBuffer to the communications service provider is equivalent to the near-simultaneous invocation of multiple RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY operations. These operations are invoked in the order in which the original annotated telemetry frames and synchronous notifications occur. However, the RCF service provider concatenates them in a buffer, the content of which forms a single RcfTransferBuffer SLE-PDU. This SLE-PDU is the service data unit passed to the communications provider.

The primary rationale for this approach is as follows: when the data rate on the space link exceeds the available communications bandwidth or when the ground communication link is congested or unavailable for a period of time, use of the complete online delivery mode may lead to the accumulation of a large backlog of undelivered data, resulting in the delivery of data past the point of usefulness. The timely online delivery mode is an alternative that limits the size of the backlog that is allowed to accumulate by discarding data that cannot be delivered within a certain time. Furthermore, when data is discarded, it is discarded 'in chunks', i.e., as a sufficiently large block of contiguous frames rather than as random frames here and there. In general, this approach maximizes the usefulness of the data that is delivered.

This result is achieved as follows: as the RCF service provider acquires telemetry frames and as events that must be synchronously notified occur, that information is stored in a buffer named the transfer buffer. The size of this buffer is set by service management to achieve the appropriate level of 'chunking' of data; this size corresponds to the maximum-sized RcfTransferBuffer SLE-PDU that will be passed to the communications service provider. (This size must also be compatible with the size of the service data units that can be handled by the underlying communications service.) When data is inserted into the transfer buffer, if the transfer buffer was previously empty, a timer, named the release timer, is started. The release timer counts down from an initial time value, named the latency limit, that is set by service management. If the transfer buffer becomes full or if the release timer expires, the entire transfer buffer, in the form of one RcfTransferBuffer SLE-PDU, is passed to the communications service provider as one service data unit. If new data needs to be inserted into the transfer buffer, but the transfer buffer is full and cannot be passed to the communications service provider because of congestion of the communications service, then the entire transfer buffer is discarded as one unit. When this happens, the new data is inserted into the newly emptied transfer buffer, along with a synchronous notification that some data was intentionally discarded due to timeliness considerations.

Strictly speaking, the transfer buffer is only required in the case of timely online delivery mode. However, the transfer buffer mechanism has a secondary benefit: it allows for 'tuning' of the size of the service data units that are normally passed to the communications service provider. In some cases, this may contribute to enabling the communications service to operate more efficiently. Thus, the transfer buffer is used in all delivery modes. However,

in the case of complete online delivery mode and offline delivery mode, data in the transfer buffer is never discarded. Rather, the contents of the transfer buffer are held until they can be passed to the communications service provider. Any backlog that may accumulate is handled by means of the online frame buffer or offline frame buffer (see 2.6.4.6.3).

2.6.4.6.3 Online Frame Buffer and Offline Frame Buffer

In the case of the timely online delivery mode, the only buffering that occurs is that provided by the transfer buffer. For complete online delivery mode and for offline delivery mode, additional buffering is needed because of the nature of the service. Since the complete online delivery mode is supposed to deliver all data even in the case of extended communications service outages or congestion, and since the offline delivery mode is supposed to deliver all data even several days after the space link session, more extensive buffering is required for these modes. This buffering is provided by means of the online frame buffer or the offline frame buffer, respectively. These buffers are relatively large and intended to hold all data (i.e., RCF-TRANSFER-DATA and, in the case of online delivery mode, RCF-SYNC-NOTIFY) for significant periods of time. In the case of complete online delivery mode, the online frame buffer is intended to overcome limitations of the communications service: bandwidth limitations, outages, and congestion. In the case of offline delivery mode, the offline frame buffer is intended to enable data to be delivered hours or days after the completion of the space link session. The exact size of these buffers is set by service management. It is normally expected that the online frame buffer is sufficiently large to hold all data that might be accumulated during one space link session and that the offline frame buffer is sufficiently large to hold all data that might be accumulated during several space link sessions.

NOTE – Synchronous notifications are generally not provided in offline mode, so the data associated with such notifications is not stored in the offline frame buffer.

Figure 2-6 illustrates the differences between timely online delivery mode, complete online delivery mode, and offline delivery mode with respect to the buffers that are used.

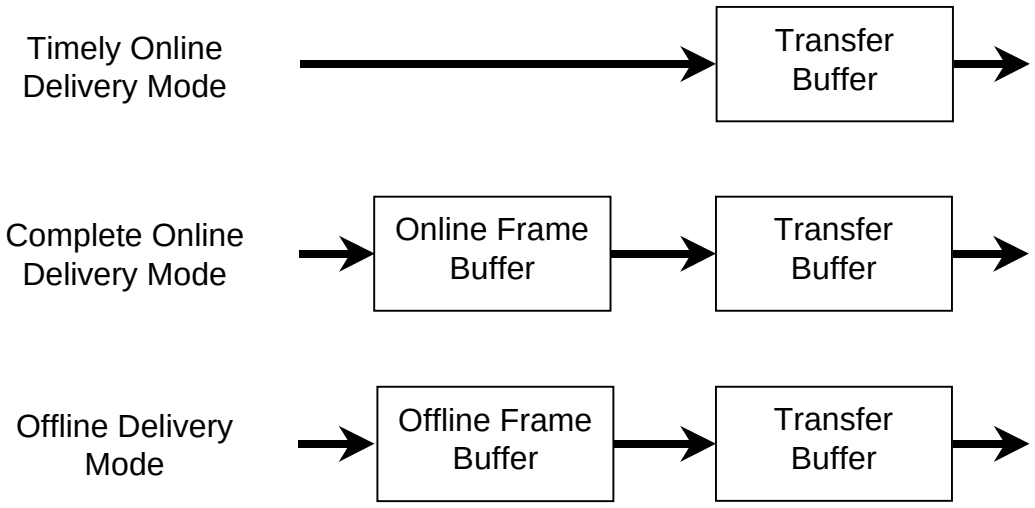

Figure 2-6: Buffers and Delivery Modes

2.7 OPERATIONAL SCENARIO

The following subsections illustrate a typical operational scenario for a user-initiated, online delivery mode RCF service instance:

a) Prior to the actual provision of service, start and stop times for both the space link session and the associated RCF service instance are mutually agreed upon by SLE Complex Management and SLE Utilization Management. Configuration and other information needed to enable the service are also agreed upon. Included in the configuration information for the planned RCF service instance is a list of global VCIDs that identifies the master channel and virtual channels in this master channel that are permitted to be selected by the user of RCF service by means of the RCF-START invocation.

b) Some time before the scheduled start time of the RCF service instance, the service instance is created by SLE Complex Management. Initially, the service provider is in state 1 (‘unbound’). At the scheduled start time of the space link session, the SLE Complex acquires the signal from the spacecraft and initiates the production of RCF service. Typically (but not necessarily) the start time of the service instance will precede by a small margin the start time of the space link session to allow the user to bind to the service before the start of the space link session.

c) The user invokes the RCF-BIND operation to establish an association.

d) The provider transitions from state 1 to state 2 (‘ready’) and returns a report of the successful completion of the bind operation to the user.

e) The user may now check parameters that control service provision by means of the RCF-GET-PARAMETER operation.

f) If the user is interested in obtaining periodic status reports, the RCF-SCHEDULE-STATUS-REPORT operation may be invoked to configure status reporting.

g) The user invokes the RCF-START operation to enable data flow. The RCF-START identifies the master channel or virtual channel to be transferred by the service instance. The selected master channel or virtual channel must be contained in the set of permitted channels (see item a) above).

h) The provider transitions from state 2 to state 3 (‘active’) and confirms the RCF-START operation to the user. As telemetry frames that meet the delivery criteria established by the user become available, they are delivered to the user by means of RCF-TRANSFER-DATA operations. In addition, notifications may be delivered by means of RCF-SYNC-NOTIFY operations, and requested status reports are delivered by means of RCF-STATUS-REPORT operations.

i) When all available frames are delivered, the provider will invoke the RCF-SYNC-NOTIFY operation to provide an ‘end of data’ notification. The ‘end of data’ notification may be triggered, for example, because the space link session ended and all frames have been delivered, or because the user specified a value for the `stop-time` parameter when invoking the RCF-START operation.

j) By invoking the RCF-STOP operation the user will cause the provider to transition to state 2 (‘ready’). The user may then ask for another master channel or virtual channel by invoking another RCF-START operation or may terminate the association by invoking RCF-UNBIND.

2.8 SECURITY ASPECTS OF THE SLE RCF TRANSFER SERVICE

2.8.1 SECURITY BACKGROUND/INTRODUCTION

The SLE transfer services explicitly provide authentication and access control. Additional security capabilities, if required, are levied on the underlying communication services that support the SLE transfer services. The SLE transfer services are defined as layered application services operating over underlying communication services that must meet certain requirements but which are otherwise unspecified. Selection of the underlying communication services over which real SLE implementations connect is based on the requirements of the communicating parties and/or the availability of CCSDS-standard communication technology profiles and proxy specifications. Different underlying communication technology profiles are intended to address not only different performance requirements but also different security requirements. Missions and service providers are expected to select from these technology profiles to acquire the performance and security capabilities appropriate to the mission. Specification of the various underlying communication technologies, and in particular their associated security provisions, are outside the scope of this Recommended Standard.

The SLE RCF transfer service transfers data that originates on a mission spacecraft. As such, the SLE RCF transfer service has custody of the data for only a portion of the end-to-end data path between mission spacecraft and MDOS. Consequently the ability of an SLE transfer service to secure the transfer of mission spacecraft data is limited to that portion of the end-to-end path that is provided by the SLE transfer service (i.e., the terrestrial link between the MDOS and the ground termination of the space-ground link to the mission spacecraft). End-to-end security must also involve securing the data as it crosses the space-ground link, which can be provided by some combination of securing the mission data itself (e.g., encryption of the mission data within CCSDS space packets) and securing the space-ground link (e.g., encryption of the physical space-ground link). Thus while the SLE RCF transfer service plays a role in the end-to-end security of the data path, it does not control and cannot ensure that end-to-end security. This component perspective is reflected in the security provisions of the SLE transfer services.

2.8.2 STATEMENTS OF SECURITY CONCERNS

This section identifies RCF transfer service support for capabilities that responds to these security concerns in the areas of data privacy, data integrity, authentication, access control, availability of resources, and auditing.

2.8.2.1 Data Privacy (also known as Confidentiality)

This SLE RCF transfer service specification does not define explicit data privacy requirements or capabilities to ensure data privacy. Data privacy is expected to be ensured outside of the SLE transfer service layer, by the mission application processes that communicate over the SLE transfer service, in the underlying communication service that lies under the SLE transfer service, or some combination of both. For example, mission application processes might apply end-to-end encryption to the contents of the CCSDS space link data units carried as data by the SLE transfer service. Alternatively or in addition, the network connection between the SLE entities might be encrypted to provide data privacy in the underlying communication network.

2.8.2.2 Data Integrity

The SLE RCF transfer service defines and enforces a strict sequence of operations that constrain the ability of a third party to inject operation invocations or returns into the transfer service association between a service user and provider (see 4.2.2). This constrains the ability of a third party to seize control of an active RCF transfer service instance without detection.

The SLE RCF transfer service requires that the underlying communication service transfer data in sequence, completely and with integrity, without duplication, with flow control that notifies the application layer in the event of congestion, and with notification to the application layer in the event that communication between the service user and the service

provider is disrupted (see 1.3.1). No specific mechanisms are identified, as they will be an integral part of the underlying communication service.

2.8.2.3 Authentication

This SLE RCF transfer service specification defines authentication requirements (see 3.1.5), and defines `initiator-identifier`, `responder-identifier`, `invoker-credentials`, and `performer-credentials` parameters of the service operation invocations and returns that are used to perform SLE transfer service authentication. The procedure by which SLE transfer service operation invocations and returns are authenticated is described in annex F of the Cross Support Service Green Book (reference [E2]). The SLE transfer service authentication capability can be selectively set to authenticate at one of three levels: authenticate every invocation and return, authenticate only the BIND operation invocation and return, or perform no authentication. Depending upon the inherent authentication available from the underlying communication network, the security environment in which the SLE service user and provider are operating, and the security requirements of the spaceflight mission, the SLE transfer service authentication level can be adapted by choosing the SLE operation invocation and returns that shall be authenticated. Furthermore the mechanism used for generating and checking the credentials and thus the level of protection against masquerading (simple or strong authentication) can be selected in accordance with the results of a threat analysis.

2.8.2.4 Access Control

This SLE RCF transfer service specification defines access control requirements (see 3.1.4), and defines `initiator-identifier` and `responder-identifier` parameters of the service operation invocations and returns that are used to perform SLE transfer service access control. The procedure by which access to SLE transfer services is controlled is described in annex F of the Cross Support Service Green Book (reference [E2]).

2.8.2.5 Availability of Resources

The SLE transfer services are provided via communication networks that have some limit to the resources available to support those SLE transfer services. If these resources can be diverted from their support of the SLE transfer services (in what is commonly known as ‘denial of service’) then the performance of the SLE transfer services may be curtailed or inhibited. This SLE RCF transfer service specification does not define explicit capabilities to prevent denial of service. Resource availability is expected to be ensured by appropriate capabilities in the underlying communication service. The specific capabilities will be dependent upon the technologies used in the underlying communication service and the security environment in which the transfer service user and provider operate.

2.8.2.6 Auditing

This SLE RCF transfer service specification does not define explicit security auditing requirements or capabilities. Security auditing is expected to be negotiated and implemented bilaterally between the spaceflight mission and the service provider.

2.8.3 POTENTIAL THREATS AND ATTACK SCENARIOS

The SLE RCF transfer service depends on unspecified mechanisms operating above the SLE transfer service (between a mission spacecraft application process and its peer application process on the ground), underneath the SLE transfer service in the underlying communication service, or some combination of both, to ensure data privacy (confidentiality). If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could read the spacecraft telemetry data contained in the RCF protocol data units as they traverse the WAN between service user and service provider.

The SLE RCF transfer service constrains the ability of a third party to seize control of an active SLE transfer service instance, but it does not specify mechanisms that would prevent an attacker from intercepting the protocol data units and replacing the contents of the `data` parameter. The prevention of such a replacement attack depends on unspecified mechanisms operating above the SLE transfer service (between a mission spacecraft application process and its peer application process on the ground), underneath the SLE transfer service in the underlying communication service, in bilaterally agreed extra capabilities applied to the SLE transfer service (e.g., encryption of the `data` parameter) or some combination of the three. If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could substitute telemetry data without detection.

If the SLE transfer service authentication capability is not used and if authentication is not ensured by the underlying communication service, attackers may somehow obtain valid `initiator-identifier` values and use them to initiate SLE transfer service instances by which they could gain access to spacecraft telemetry data.

The SLE RCF transfer service depends on unspecified mechanisms operating in the underlying communication service to ensure that the supporting network has sufficient resources to provide sufficient support to legitimate users. If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could prevent legitimate users from receiving telemetry from their spacecraft.

If the provider of SLE RCF transfers service provides no security auditing capabilities, or if a user chooses not to employ auditing capabilities that do exist, then attackers may delay or escape detection while stealing or altering telemetry data.

2.8.4 CONSEQUENCES OF NOT APPLYING SECURITY

The consequences of not applying security to the SLE RCF transfer service are possible degradation and loss of ability to receive telemetry from the spacecraft, or the substitution of altered telemetry data.

3 RCF SERVICE OPERATIONS

NOTE – This section (3) specifies the processing of valid SLE-PDUs (i.e., those that are recognized as the invocation or return of an RCF service operation). Subsection 3.1 specifies behaviors that are generally applicable to all operations. Subsections 3.2 through 3.11 specify individual operations. Handling of invalid SLE-PDUs is specified in 4.1.

3.1 GENERAL CONSIDERATIONS

3.1.1 RESULT OF OPERATIONS

3.1.1.1 All confirmed operations shall report on the outcome of the operation in a return, except as specified otherwise in section 4.

3.1.1.2 All returns shall include a `result` parameter that indicates whether the outcome of the operation was successful ('positive result') or unsuccessful ('negative result').

3.1.1.3 In the event of a 'negative result', the return shall also include a `diagnostic` parameter, the value of which is descriptive of the reason for the 'negative result'.

NOTE – Possible values of the `diagnostic` parameter are listed in the description of each operation.

3.1.1.4 A `diagnostic` parameter value of 'other reason' shall be returned only if no other value in the list adequately describes the reason for the 'negative result'.

3.1.2 PARAMETER TYPES

The types of all parameters shall conform to the abstract syntax specified in annex A.

NOTE – Some parameter types in annex A are chosen such that possible future extensions of the range of allowed values of a parameter will not cause a type mismatch. For example, parameters that logically are of the 'enumerated' type may be specified as being of the 'named integer' type.

3.1.3 PARAMETER CHECKING

3.1.3.1 Validity checks shall be performed on the values of parameters associated with an operation.

NOTE – Rules governing the validity of parameter values are included in the specification of individual operations. General reasons for regarding a parameter value as invalid are specified in the following subsections.

3.1.3.2 A parameter value shall be treated as invalid if it is outside the range or not in the set of values currently permitted by service management for the given parameter.

NOTE – A conforming implementation shall be capable of supporting the full range or set of values as specified in annex A.

3.1.3.3 A parameter value shall be treated as invalid if it is in conflict with the value of another parameter in the same invocation.

NOTE – For example, the value of the `start-time` parameter in an invocation of RCF-START is invalid if it is later than the value of the `stop-time` parameter.

3.1.3.4 If a parameter value is not valid, the operation shall not be performed and, for confirmed operations, a report of 'negative result' shall be returned to the invoker.

3.1.3.5 Except as noted in 3.2.2.11, checks for invalid parameters or for other conditions that can cause a report of 'negative result' should be performed in the order in which diagnostics are listed in the descriptions of the operations, and the `diagnostic` parameter should be set to the value defined for the first problem found.

3.1.3.6 In the case that an implementation does not adhere to the sequence of checks as specified by the sequence of diagnostics values, such implementation shall specify the sequence in which the checks are actually performed.

3.1.4 ACCESS CONTROL

3.1.4.1 The RCF service shall implement access control based on the identity of the initiator and responder. Access control is performed at two levels:

a) the initiator must be registered at the responder and the responder must be registered at the initiator;

b) the initiator and responder must be authorized for the given service instance.

3.1.4.2 The initiator shall have access to a registry of authorized responders and the responder shall have access to a registry of authorized initiators. These registries shall be maintained by SLE Complex Management and SLE Utilization Management, respectively.

3.1.4.3 Service management shall specify the authorized initiator and responder for each service instance.

3.1.4.4 The initiator and responder shall indicate their identity by setting the parameters `initiator-identifier` and `responder-identifier` in the RCF-BIND operation to the values assigned by service management.

3.1.5 AUTHENTICATION

NOTE – Requirements for security depend on the application and the environment of the SLE Complexes and the MDOS (e.g., whether closed or public networks are used or if access is only from physically restricted areas). In many environments, security may be provided by the communications service, transparently to the SLE application. This Recommended Standard does not preclude the use of security features that are provided by the communications service or the local environment, nor does it assume the availability of such features.

3.1.5.1 The RCF service shall provide the following options with respect to the level of authentication of invocations and returns of operations:

a) 'all': all RCF invocations and returns, except the invocation of RCF-PEER-ABORT, shall be authenticated;

b) 'bind': only the RCF-BIND invocation and return shall be authenticated;

c) 'none': no RCF invocations or returns shall be authenticated.

3.1.5.2 SLE Complex Management and SLE Utilization Management shall agree on the level of authentication to be required for an association between a service user and a service provider and shall configure both entities accordingly.

3.1.5.3 SLE Complex Management and SLE Utilization Management shall agree on the algorithm used to generate and check credentials parameters and shall make this algorithm known to the service user and service provider together with associated parameters such as passwords or keys as necessary for the selected algorithm.

NOTES

1 The specification of the algorithms themselves is outside the scope of this Recommended Standard.

2 The `initiator-identifier` and `responder-identifier` parameters of the RCF-BIND operation identify the user and provider and therefore the applicable authentication level and algorithm necessary to generate and check credentials.

3.1.5.4 For operations for which authentication is required by the terms of the agreement between SLE Complex Management and SLE Utilization Management:

a) invocations shall include an `invoker-credentials` parameter to permit the performer to authenticate the invocation;

b) returns shall include a `performer-credentials` parameter to permit the invoker to authenticate the return.

3.1.5.5 For operations for which authentication is not required, the `invoker-credentials` or `performer-credentials` parameter should be set to the value 'unused' to signify that the invocation or return does not carry credentials.

3.1.6 BLOCKING AND NON-BLOCKING OPERATIONS

3.1.6.1 To support applications that may need to invoke several operations concurrently, the parameter `invoke-ID` is specified for all confirmed operations except RCF-BIND and RCF-UNBIND.

NOTES

1 The `invoke-ID` parameter allows the invoker to correlate a particular return to the invocation that prompted it.

2 Confirmed operations that include the `invoke-ID` parameter are non-blocking operations; those that do not are blocking operations. Unconfirmed operations are always non-blocking.

3.1.6.2 After invoking a blocking operation, the invoker shall not invoke another operation for the same service instance until the return from the blocking operation is received; except that, if the return is not received in a timely manner, the invoker may invoke RCF-PEER-ABORT to terminate the association.

3.1.6.3 After invoking a non-blocking operation, the invoker may invoke another operation without waiting for the return from the first invocation.

3.1.6.4 The value of the `invoke-ID` parameter shall be an invoker-supplied arbitrary integer value that shall be returned, unchanged, by the performer.

3.1.6.5 An error condition shall exist if an invocation includes an `invoke-ID` whose value is the same as that of another invocation that is awaiting confirmation within the context of the same service instance.

3.1.6.6 To ensure that the RCF service behaves in a predictable manner, the effects of operations shall be as though the operations were performed in the order that their invocations were received by the performer.

3.1.6.7 The invoker may choose not to exploit the non-blocking capability and always wait for the return from a non-blocking operation before invoking another operation.

NOTE – An invoker wishing to operate in blocking mode (i.e., to invoke a new operation only when the return from the previous operation has been received) may use a constant value for the `invoke-ID` parameter. As long as a return is still outstanding, the performer will reject any further invocations.

3.1.6.8 Compliance with this Recommended Standard does not require the performer to process invocations concurrently; however, the performer must accept invocations from a non-blocking invoker and buffer and serialize them by local means not visible externally.

3.1.7 TIME

3.1.7.1 The time reference for all parameters containing a time value shall be based on Coordinated Universal Time (UTC).

NOTE – The type of all time parameters is specified in annex A.

3.1.7.2 The `earth-receive-time` parameter (see 3.6.2.3) shall be expressed using the CCSDS Day Segmented (CDS) time code (reference [5]) with an epoch of 1958-01-01 and a 16-bit day segment. Depending on the RCF service provider capabilities and/or the supported mission requirements, the time tag may have either a resolution of microseconds or a resolution of picoseconds.

3.1.7.3 The `earth-receive-time` parameter shall have a precision of one millisecond or better.

3.1.7.4 The `earth-receive-time` parameter shall be accurate to within one millisecond or better.

3.1.8 SETTING OF PARAMETERS

3.1.8.1 An RCF provider shall permit setting of the service configuration parameters as specified in table 3-1.

3.1.8.2 The range or set of values a parameter may assume is constrained by specification of its data type (see annex A).

3.1.8.3 Service management may further constrain the allowed values for a given service instance.

Table 3-1: Setting of RCF Service Configuration Parameters

Parameter	Service Management	RCF-START Operation	RCF-SCHEDULE-STATUS-REPORT Operation
delivery mode	X		
latency-limit	X		
maximum-delivery-rate	X		
maximum-reporting-cycle	X		
minimum-reporting-cycle	X		
permitted-global-VCID-set	X		
reporting-cycle			X
requested-global-VCID		X	
return-timeout-period	X		
service-instance-provision-period	X		
service-version-number	X		
transfer-buffer-size	X		

NOTES

1 The user can ascertain the current value of the parameters presented in table 3-11 by means of the RCF-GET-PARAMETER operation.

2 This Recommended Standard also refers to parameters that are set by service management, but are not listed in table 3-1. These parameters cannot be ascertained by means of the RCF-GET-PARAMETER operation.

3 The methods used by service management to control service provision and service production parameters are outside the scope of this Recommended Standard.

3.1.9 DELIVERY MODES

3.1.9.1 Timely Online Delivery Mode

3.1.9.1.1 For timely online delivery mode, the RCF service provider shall store frames acquired from the space link and certain information associated with those frames (as per 3.6.2) in a buffer called the transfer buffer. The stored information shall be an RCF-TRANSFER-DATA invocation or the equivalent thereof.

3.1.9.1.2 The extraction of an RCF from the RAF channel that matches the delivery criteria and thus the possibility to build an RCF-TRANSFER-DATA invocation constitutes the 'data available' event (see 4.2.2) in timely online delivery mode.

NOTE – For convenience, the following subsections are written as if the contents of the transfer buffer consist of an ordered list of records of type RcfTransferData-Invocation (see A2.7) or type RcfSyncNotifyInvocation (see 3.1.9.1.3 and A2.7). However, that is not intended to constrain how the transfer buffer is implemented in a real system. It is sufficient that a real system provide the externally visible behaviors that are specified herein.

3.1.9.1.3 Upon the occurrence of any of several events that cause a change to or disruption of the data being provided to the service instance, the RCF service provider shall store a synchronous notification record of the event in the transfer buffer. The notification record shall be an RCF-SYNC-NOTIFY invocation or the equivalent thereof. The notification record shall be stored in the transfer buffer after the last annotated frame acquired before the event and before the first annotated frame acquired following the event. The events and associated information that are stored shall be as defined in 3.7; in particular, at the end of the space link session, an 'end of data' RcfSyncNotifyInvocation shall be stored following the last RcfTransferDataInvocation.

3.1.9.1.4 If the transfer buffer was empty before the service provider inserted an RcfTransferDataInvocation or RcfSyncNotifyInvocation record into the transfer buffer, the service provider shall start a timer for the transfer buffer. This timer shall be named the release timer.

3.1.9.1.5 The duration from the time that the release timer is started until it expires is given by the parameter `latency-limit`, the value of which shall be set by service management.

3.1.9.1.6 For a given instance of RCF service, the transfer buffer shall accommodate a set number of RcfTransferDataInvocation and/or RcfSyncNotifyInvocation records. That number, given by the parameter `transfer-buffer-size`, shall be set by service management.

3.1.9.1.7 The contents of the transfer buffer shall be passed to the communications service (in the form of one RcfTransferBuffer SLE-PDU) and the transfer buffer shall be cleared as soon as one of the following conditions is met:

a) the buffer is full (i.e., the number of RcfTransferDataInvocation and/or RcfSyncNotifyInvocation records contained in the buffer is equal to the value of the `transfer-buffer-size` parameter); or

b) the release timer expires; or

c) an RcfSyncNotifyInvocation 'end of data' record was inserted into the transfer buffer.

3.1.9.1.8 The RcfTransferBuffer SLE-PDU shall contain the records in the same sequence as they were inserted into the transfer buffer.

3.1.9.1.9 If the underlying communications service generates backpressure because of congestion (e.g., it does not accept the RcfTransferBuffer SLE-PDU or would block the RCF service provider from continuing), the RCF service provider shall discard this RcfTransferBuffer SLE-PDU and clear the transfer buffer. It shall then insert a 'data discarded due to excessive backlog' RcfSyncNotifyInvocation record into the transfer buffer and restart the release timer.

3.1.9.1.10 When the 'data discarded due to excessive backlog' RcfSyncNotifyInvocation record is inserted into the transfer buffer, the size of the transfer buffer shall be incremented by one. That new size shall remain in effect until the contents of the transfer buffer are passed to the communications service, after which the transfer buffer size shall be decremented by one.

NOTE – The temporary increment in the transfer buffer size ensures a minimum of telemetry flow in case of congestion. Otherwise, only 'data discarded due to excessive backlog' notifications might be sent in case a buffer size of one was specified.

3.1.9.1.11 When the RCF service provider accepts an RCF-STOP invocation from the user, it shall immediately build from the transfer buffer contents an RcfTransferBuffer SLE-PDU and shall immediately pass that to the communications service, subject to the provisions in 3.1.9.1.9.

3.1.9.1.12 The transfer buffer shall be cleared whenever the association is aborted.

3.1.9.1.13 Only RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY invocations shall be buffered through the transfer buffer. The invocations or returns of all other operations shall be asynchronous (in the sense that they are not required to be invoked or returned in sequence with respect to the delivery of the contents of the transfer buffer). Therefore, they shall be invoked or returned as soon as possible without regard to the contents of the transfer buffer.

3.1.9.2 Complete Online Delivery Mode

3.1.9.2.1 For complete online delivery mode, the RCF service provider shall store frames acquired from the space link and certain information associated with those frames (as per 3.6.2) in a buffer called the online frame buffer. The stored information shall be an RCF-TRANSFER-DATA invocation or the equivalent thereof.

3.1.9.2.2 The availability of such RCF-TRANSFER-DATA invocation or the equivalent thereof at the output of the online RCF buffer for insertion into the transfer buffer constitutes the 'data available' event (see 4.2.2) in complete online delivery mode.

NOTES

1 Complete online delivery mode attempts to deliver all acquired frames, in order, with minimum delay consistent with the available ground communications bandwidth. Complete online delivery requires that the online frame buffer be sufficiently large to deal with communications service delays, outages, and bandwidth limitations.

2 For convenience, the following subsections are written as if the contents of the online frame buffer and the transfer buffer consist of an ordered list of records of type RcfTransferDataInvocation (see A2.7) or type RcfSyncNotifyInvocation (see 3.1.9.2.3 and A2.7). However, that is not intended to constrain how the online frame buffer or the transfer buffer is implemented in a real system. It is sufficient that a real system provide the externally visible behaviors that are specified herein.

3.1.9.2.3 Upon the occurrence of any of several events that cause a change to or disruption of the data being provided to the service instance, the RCF service provider shall store a synchronous notification record of the event in the online frame buffer. The notification record shall be an RCF-SYNC-NOTIFY invocation or the equivalent thereof. The notification record shall be stored following the last annotated frame acquired before the event and before the first annotated frame acquired following the event. The events and associated information that are stored shall be as defined in 3.7; in particular, at the end of the space link session, an 'end of data' RcfSyncNotifyInvocation shall be stored following the last RcfTransferDataInvocation.

3.1.9.2.4 The RCF service provider shall start to fill the online frame buffer as soon as both the service instance provision period and the space link session have started and frames are being acquired from the space link.

3.1.9.2.5 While the RCF service provider is in state 3 ('active') and the transfer buffer is not full, the provider shall remove RcfTransferDataInvocation and RcfSyncNotifyInvocation records from the online frame buffer and insert them, in the same sequence, into the transfer buffer.

3.1.9.2.6 If the transfer buffer was empty before the service provider inserted an RcfTransferDataInvocation or RcfSyncNotifyInvocation record into the transfer buffer, the service provider shall start a timer for the transfer buffer. This timer shall be named the release timer.

3.1.9.2.7 The duration from the time that the release timer is started until it expires is given by the parameter `latency-limit`, the value of which shall be set by service management.

3.1.9.2.8 For a given instance of RCF service, the transfer buffer shall accommodate a set number of RcfTransferDataInvocation and/or RcfSyncNotifyInvocation records. That number, given by the parameter `transfer-buffer-size`, shall be set by service management.

3.1.9.2.9 As soon as the transfer buffer is full or an 'end of data' RcfSyncNotifyInvocation record is inserted into the transfer buffer or the provider has accepted an RCF-STOP invocation or the 'release timer expired' event was generated, the service provider shall stop

extracting RcfTransferDataInvocation or RcfSyncNotifyInvocation records from the online frame buffer and build an RcfTransferBuffer SLE-PDU from the transfer buffer contents. The provider shall attempt to pass this SLE-PDU to the communications service until it is accepted.

3.1.9.2.10 When the communications service provider has accepted the RcfTransferBuffer SLE-PDU, the RCF service provider shall clear the transfer buffer and resume removing frames and synchronous notifications from the online frame buffer as described above.

3.1.9.2.11 The RCF service provider shall continue to remove RcfTransferDataInvocation and RcfSyncNotifyInvocation records from the online frame buffer, insert them into the transfer buffer, and pass the contents of the transfer buffer to the communications service as long as the service instance remains in state 3 (‘active’).

3.1.9.2.12 In the complete online delivery mode, the transfer buffer shall be cleared and removal of frames and synchronous notifications from the online frame buffer shall stop whenever the association is aborted.

NOTE – The requirement 3.1.9.2.12 implies that a truly complete delivery can only be achieved within a given association. Recovery from data loss caused by an association termination (release or abort) can only be accomplished by using the offline delivery mode. Such data loss could be avoided by means of an application-to-application acknowledgement mechanism, but at the expense of a serious throughput performance degradation. Considering that the resulting gap can be filled by means of the offline delivery mode, the option offering better performance and simpler implementation was chosen.

3.1.9.2.13 The RCF service provider shall continue to store acquired frames and notification records in the online frame buffer until the end of the service instance provision period, regardless of the state of the service instance and regardless of whether an association with the service user is established.

3.1.9.2.14 In the case that the user invokes the RCF-STOP operation or the association becomes unbound, the user may, after re-binding if necessary, invoke a new RCF-START operation, with a start time in the past, to effect delivery of the data buffered in the online frame buffer. Any frames with an ERT older than the start time specified in the RCF-START operation and any notifications falling into the same interval shall be removed from the online frame buffer.

3.1.9.2.15 If the online frame buffer becomes full (e.g., because an extended communications outage prevents it from being emptied), the provider shall discard Rcf-TransferDataInvocation and RcfSyncNotifyInvocation records from the online frame buffer in oldest-first order. The number of frames to be discarded in such event is set by service management. The RCF service provider shall also insert an RcfSyncNotifyInvocation record indicating a ‘data discarded’ event into the transfer buffer as soon as this is possible. Extraction of RcfTransferDataInvocation and RcfSyncNotifyInvocation records from the online frame buffer shall then resume as before.

NOTE – For the complete online delivery mode, it is intended that the size of the online frame buffer be selected such that overflow of the buffer is a very rare event.

3.1.9.2.16 If the service user, in the RCF-START invocation, requests a start time earlier than any frame still held in the online frame buffer, the provider shall deliver the earliest data available.

3.1.9.2.17 At the end of the scheduled service instance provision period, the contents of the online frame buffer shall be discarded.

3.1.9.2.18 Only RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY invocations shall be buffered through the online frame buffer and the transfer buffer. The invocations or returns of all other operations shall be asynchronous (in the sense that they are not required to be invoked or returned in sequence with respect to the delivery of the contents of the online frame buffer). Therefore, they shall be invoked or returned as soon as possible without regard to the contents of the online frame buffer or the transfer buffer.

3.1.9.2.19 During complete online service provision, the RCF service provider shall extract RcfTransferDataInvocation and RcfSyncNotifyInvocation records from the online frame buffer, insert them into the transfer buffer, and pass RcfTransferBuffer SLE-PDUs to the communications service without undue delay, subject only to limitations imposed by the underlying communications service, or to any maximum data rate limitation (‘metering’) that may be imposed through service management.

3.1.9.2.20 For complete online delivery mode, the size of the online frame buffer, the transfer buffer and the release timer shall be determined by arrangement between SLE Complex Management and SLE Utilization Management and shall be set by service management.

3.1.9.3 Offline Delivery Mode

3.1.9.3.1 Any frame acquired from the space link that may need to be provided through an offline delivery mode service instance, as well as certain information associated with the frame (as per 3.6.2), shall be stored in a buffer called the offline frame buffer. The stored information shall be an RCF-TRANSFER-DATA invocation or the equivalent thereof. There should be one offline frame buffer for all service instances associated with a particular service agreement. This implies that any deletion of telemetry in the offline frame buffer affects all offline SIs that exist for the associated service agreement and any deletion of telemetry in the offline fame buffer does not affect any SI that exists under a different service agreement.

3.1.9.3.2 The availability of an RCF-TRANSFER-DATA invocation or the equivalent thereof at the output of the offline RCF buffer for insertion into the transfer buffer constitutes the ‘data available’ event (see 4.2.2) in offline delivery mode.

NOTE – For convenience, the following subsections are written as if the contents of the offline frame buffer and the transfer buffer consist of an ordered list of records of type RcfTransferDataInvocation (see A2.7). However, that is not intended to constrain how the offline frame buffer or the transfer buffer is implemented in a real system. It is sufficient that a real system provide the externally visible behaviors that are specified herein.

3.1.9.3.3 There may be a significant delay from the time when a frame is acquired from the space link until that frame is available for delivery through an offline delivery mode service instance. Every service provider shall document the characteristics of their service with respect to that delay.

3.1.9.3.4 When an RCF-START operation is invoked in the context of an offline delivery mode RCF service instance, the RCF service provider shall extract RcfTransferData-Invocation records from the offline frame buffer and insert them into the transfer buffer. Such extraction shall begin with the RcfTransferDataInvocation record in the offline frame buffer with the earliest ERT that is equal to or later than the start time designated in the RCF-START invocation. Subsequent RcfTransferDataInvocation records shall be extracted from the offline frame buffer and inserted into the transfer buffer in the same order in which they were originally received from the space link.

3.1.9.3.5 Extraction of RcfTransferDataInvocation records from the offline frame buffer and their insertion into the transfer buffer shall continue until:

a) the transfer buffer is full;

b) a frame is retrieved with an ERT that is later than the stop time in the RCF-START invocation (in which case an ‘end of data’ notification shall be generated and inserted into the transfer buffer);

c) the user invokes the RCF-STOP operation; or

d) the association is aborted.

3.1.9.3.6 As soon as the transfer buffer is full or an ‘end of data’ RCFSyncNotifyInvocation record is inserted into the transfer buffer, or the provider has accepted an RCF-STOP invocation, the service provider shall stop extracting RcfTransferData records from the offline RCF buffer, build an RcfTransferBuffer SLE-PDU from the transfer buffer contents. The provider shall attempt to pass this SLE-PDU to the communications service until it is accepted. The RcfTransferBuffer SLE-PDU shall contain the RcfTransferDataInovcation records in the same order as they were originally received from the space link.,

3.1.9.3.7 Once the RcfTransferBuffer SLE-PDU has been accepted by the communications service, the RCF service provider shall clear the transfer buffer. It shall also resume extracting frames from the offline frame buffer unless the ‘end of data’ notification was generated.

3.1.9.3.8 In the offline delivery mode, the transfer buffer shall be cleared and extraction of frames from the offline frame buffer shall stop whenever:

a) the user stops data flow by invoking the RCF-STOP operation; or

b) the association is aborted.

3.1.9.3.9 In the case that the user invokes the RCF-STOP operation or the association becomes unbound, the user may, after re-binding if necessary, invoke a new RCF-START operation, specifying a new ERT interval for which frames shall be delivered from the offline frame buffer. The start and stop times of such an RCF-START invocation may be earlier, later, or the same as the start and stop times of any previous RCF-START invocation, provided that they are valid start and stop times as specified in 3.4.

3.1.9.3.10 If the user, in the RCF-START invocation, requests a start time earlier than any frame still held in the offline frame buffer, the provider shall deliver frames beginning with the earliest data available. If there are no frames with an ERT in the interval specified by the start and stop times of the RCF-START invocation, then only the 'end of data' notification shall be delivered.

3.1.9.3.11 Except for 'end of data' notifications as described above, synchronous notifications shall not be provided in the offline delivery mode.

3.1.9.3.12 Only RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY invocations shall be buffered through the offline frame buffer. Except for the RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY invocations, the invocations or returns of all other operations shall be asynchronous (in the sense that they are not required to be invoked or returned in sequence with respect to the delivery of the contents of the offline frame buffer). Therefore, they shall be invoked or returned as soon as possible without regard to the contents of the offline frame buffer or the transfer buffer.

3.1.9.3.13 During offline service provision, the RCF service provider shall extract Rcf-TransferDataInvocation records from the offline frame buffer, insert them into the transfer buffer, and pass RcfTransferBuffer SLE-PDUs to the communications service without undue delay, subject only to limitations imposed by the underlying communications service, or to any maximum data rate limitation ('metering') that may be imposed through service management.

3.1.9.3.14 The size of the offline frame buffer and the transfer buffer shall be determined by arrangement between SLE Complex Management and SLE Utilization Management and shall be set by service management.

3.1.9.3.15 Every service provider shall document its policy regarding when, or under what circumstances, records in the offline frame buffer are deleted.

3.2 RCF-BIND

3.2.1 PURPOSE

3.2.1.1 The RCF-BIND operation shall be used to establish an association between the initiator and the responder.

3.2.1.2 For every instance of RCF service, service management shall establish whether that instance of service is to be user-initiated or provider-initiated:

a) for a user-initiated service instance, only the service user is permitted to invoke the RCF-BIND operation;

b) for a provider-initiated service instance, only the service provider is permitted to invoke the RCF-BIND operation.

3.2.1.3 The responder shall return a report of the outcome of the performance of the RCF-BIND operation to the initiator.

3.2.1.4 Except as provided in 3.2.1.5, the invoker of RCF-BIND shall not invoke any further operations for this service instance until the return from the responder is received.

3.2.1.5 If the return from RCF-BIND is not received after a sufficiently long time, the initiator may attempt to recover by invoking the RCF-PEER-ABORT operation followed by another RCF-BIND.

NOTE – The length of the duration that constitutes ‘a sufficiently long time’ is determined by service management.

3.2.1.6 The RCF-BIND operation is valid only in state 1 (‘unbound’).

3.2.2 INVOCATION, RETURN, AND PARAMETERS

3.2.2.1 General

The parameters of the RCF-BIND operation shall be present in the invocation and return as specified in table 3-2.

3.2.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-BIND invocation (see 3.1.4.1).

Table 3-2: RCF-BIND Parameters

Parameter	Invocation	Return
invoker-credentials	M	
performer-credentials		M
initiator-identifier	M	
responder-identifier		M
responder-port-identifier	M	
service-type	M	
version-number	M	C
service-instance-identifier	M	
result		M
diagnostic		C

3.2.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-BIND (see 3.1.4.1).

3.2.2.4 initiator-identifier

The **initiator-identifier** parameter shall identify the authority on whose behalf the initiating SLE application is initiating the association.

NOTES

1 The initiator-identifier parameter permits the responder to determine if the RCF-BIND operation is being invoked by the authorized initiator for this service instance.

2 Each value of the initiator-identifier parameter is associated with exactly one authentication level and exactly one authentication scheme.

3 If authentication based on credentials is used, this parameter may be redundant since the initiator-identifier value may be one constituent of the invoker-credentials parameter. However, the encoding may differ, and it may be convenient to have this parameter available in ‘clear text’ form.

3.2.2.5 `responder-identifier`

The `responder-identifier` parameter shall identify the authority on whose behalf the responding SLE application is acting.

NOTES

1 The `responder-identifier` parameter permits the initiator to determine if the RCF-BIND return is from the authorized responder for this service instance.

2 Each value of the `responder-identifier` parameter is associated with exactly one authentication level and exactly one authentication scheme.

3 If authentication based on credentials is used, this parameter may be redundant since the `responder-identifier` value may be one constituent of the `performer-credentials` parameter. However, the encoding may differ, and it may be convenient to have this parameter available in 'clear text' form.

3.2.2.6 `responder-port-identifier`

The **`responder-port-identifier`** parameter shall specify the port identifier of the responding SLE application entity with which the initiator seeks to establish an association.

NOTES

1 The value of the `responder-port-identifier` parameter is a logical name that can be translated into the technology-specific addressing information required to establish a connection with the responder using the agreed upon communications service. See 2.6.4.5 for more information.

2 SLE Complex Management and SLE Utilization Management must have previously agreed on the `responder-port-identifier` and its translation that is applicable to a particular instance of service.

3 The `responder-port-identifier` parameter is included in the RCF-BIND invocation to support its possible use by particular kinds of gateways.

3.2.2.7 `service-type`

3.2.2.7.1 The **`service-type`** parameter shall specify the type of service that will be provided if the bind operation succeeds.

3.2.2.7.2 For RCF service, the value of `service-type` shall be 'Rtn Ch Frames'.[1]

3.2.2.8 `version-number`

3.2.2.8.1 The **`version-number`** parameter shall identify the version number of the RCF service specification that is to govern this association if RCF-BIND succeeds.

3.2.2.8.2 `version-number` is conditionally present in the return based on the `result` parameter:

a) if the value of `result` is 'positive result', `version-number` shall be present in the return;

b) if the value of `result` is 'negative result', `version-number` shall not be present in the return.

3.2.2.8.3 If a provider does not support version negotiation, the `version-number` value it will accept during the BIND operation is configured by means of the managed parameter `service-version-number` (see table 3-1).

3.2.2.8.4 If the value of the `result` parameter is 'positive result', the responder shall either:

a) accept the version proposed by the initiator by putting the same version number into the return; or,

b) if the responder supports version negotiation, propose a lower (earlier) version number by putting the lower number into the return.

3.2.2.8.5 If the responder implementation does not support the requested version and does not support a lower version (or does not support version negotiation), the responder shall reject the bind with the `diagnostic` parameter set to 'version not supported'.

3.2.2.8.6 If the responder proposes a lower version and the initiator implementation does not support version negotiation or does not support the version proposed by the responder, the initiator shall unbind the association.

[1] For the RCF-BIND operation, the `service-type` parameter is redundant, because the only valid value of `service-type` is 'Rtn Ch Frames'. However, it is anticipated that future work by CCSDS may result in RCF-BIND being superseded by a generic SLE-BIND operation that is invoked with any one of several SLE service types. The RCF-BIND `service-type` parameter is provided in an attempt to facilitate such a change.

3.2.2.8.7 The value of the `version-number` parameter for the RCF service defined by this issue of this Recommended Standard shall be '4'.

NOTE – The version negotiation process as outlined above is feasible only as long as future versions of the RCF service do not modify the specification of the RCF-BIND operation.

3.2.2.9 `service-instance-identifier`

The **`service-instance-identifier`** parameter shall uniquely identify this service instance within the scope of the service-providing SLE Complex.

3.2.2.10 `result`

The **`result`** parameter shall specify the result of the RCF-BIND operation and shall contain one of the following values:

a) 'positive result'—the RCF-BIND invocation is accepted by the responder, and the association is established;

b) 'negative result'—the RCF-BIND invocation is rejected by the responder for the reason specified in the `diagnostic` parameter, and the association is not established.

3.2.2.11 `diagnostic`

3.2.2.11.1 If `result` is 'negative result', the `diagnostic` parameter shall be present in the return, and its value shall be one the following:

a) 'access denied'—the value of the `initiator-identifier` parameter is not recognized by the responder (e.g., the value does not identify the authorized initiator of any service instance known to the responder);

b) 'service type not supported'—the value of the `service-type` parameter of the RCF-BIND invocation does not identify a service type supported by the responder;

c) 'version not supported'—the responder does not support the requested version, and the responder implementation does not permit version negotiation or does not support any version of the service lower than the one requested by the initiator;

d) 'no such service instance'—the requested service instance is not defined in any agreed upon service package known to the responder;

e) 'already bound'—the service instance is already bound via a different association;

f) ‘service instance not accessible to this initiator’—the authorized initiator for the service instance identified by the `service-instance-identifier` parameter does not match the initiator identified by the `initiator-identifier` parameter of the RCF-BIND invocation;

g) ‘inconsistent service type’—the value of the `service-type` parameter of the RCF-BIND invocation is not ‘Rtn Ch Frames’, or the value of the `service-type` parameter does not match the service type of the service instance identified by the `service-instance-identifier` parameter;

h) ‘invalid time’—the RCF-BIND operation was invoked outside the service instance provision period of the service instance identified by the `service-instance-identifier` parameter;

i) ‘out of service’—the responder has been taken out of service for an indefinite period by management action (i.e., RCF production status is ‘halted’, see 3.7.2.4);

j) ‘other reason’—the reason for the negative result will have to be found by other means.

NOTES

1 In some implementations, under some circumstances, it may not be possible for the intended performer to provide a return in the event of the conditions indicated by diagnostics (d)), (h)), or (i)).

2 Implementations should consider that, under some conditions, RCF-BIND may fail with no return (e.g., if the value of the `responder-port-identifier` parameter is incorrect).

3.2.2.11.2 If `result` is ‘positive result’, the `diagnostic` parameter shall not be present in the return.

3.2.3 EFFECTS

3.2.3.1 If `result` is ‘positive result’, the RCF-BIND operation shall have the following effects:

a) An association between the user and the provider shall be established.

b) The provider shall transition from state 1 (‘unbound’) to state 2 (‘ready’).

c) All service parameters shall be set to the initial values determined by service management.

d) Upon receipt of the positive return, the user may proceed to invoke other RCF service operations (e.g., to configure the service and begin data transfer).

3.2.3.2 If `result` is 'negative result', the RCF-BIND operation shall have the following effects:

a) An association between the user and the provider shall not be established.

b) The provider shall remain in state 1 ('unbound').

c) Upon receipt of the negative return:

1) The initiator should examine the `diagnostic` parameter for the cause.

2) The initiator may attempt to re-invoke the RCF-BIND.

3.3 RCF-UNBIND

3.3.1 PURPOSE

3.3.1.1 The initiator shall invoke the RCF-UNBIND operation to release an association previously established by RCF-BIND.

3.3.1.2 The responder shall return a report of the outcome of the performance of the RCF-UNBIND operation to the initiator.

3.3.1.3 Except as provided in 3.3.1.4, the initiator shall not invoke any further operations for this service instance until the return from RCF-UNBIND is received; nor shall it perform any further operations invoked by the responder; nor shall it return to the responder any further reports of the outcome of operations invoked by the responder.

NOTE – The initiator may invoke the RCF-UNBIND operation even if it did not yet receive all returns from previously invoked operations. The initiator should be aware that the responder may choose not to send any further returns as soon as it has received the RCF-UNBIND invocation. It may then happen that the RCF-UNBIND return is not received before one of the missing returns causes a 'missing return' timeout (see section 4).

3.3.1.4 If the return from RCF-UNBIND is not received after a sufficiently long time, the initiator should invoke the RCF-PEER-ABORT operation to abort the association.

NOTES

1 The length of the duration that constitutes 'a sufficiently long time' is determined by service management.

2 Following receipt of the return from RCF-UNBIND or following the invocation of RCF-PEER-ABORT, the initiator may issue another RCF-BIND if otherwise permitted (e.g., if the end of the service instance provision period has not been reached).

3.3.1.5 The RCF-UNBIND operation is valid only in state 2 ('ready').

3.3.1.6 The RCF-UNBIND operation shall be invoked only by the initiator (i.e., the invoker of the RCF-BIND operation that established this association).

3.3.2 INVOCATION, RETURN, AND PARAMETERS

3.3.2.1 General

The parameters of the RCF-UNBIND operation shall be present in the invocation and return as specified in table 3-3.

Table 3-3: RCF-UNBIND Parameters

Parameter	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`unbind-reason`	M	
`result`		M

3.3.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-UNBIND invocation (see 3.1.4.1).

3.3.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-UNBIND (see 3.1.4.1).

3.3.2.4 `unbind-reason`

3.3.2.4.1 The **`unbind-reason`** parameter shall indicate the reason the RCF-UNBIND operation is being invoked.

3.3.2.4.2 If the RCF-UNBIND operation is invoked by the user, the `unbind-reason` parameter shall take one of the following values:

a) ‘end’—the user has obtained all frames that are needed or expected and is releasing the association normally; the provider may delete the service instance and release all resources associated with it;

NOTE – If `unbind-reason` is 'end', any subsequent attempt to invoke RCF-BIND will fail even if the service instance provision period has not expired, since the service provider may release the resources allocated to that service instance.

b) 'suspend'—the user is suspending usage of this service instance for an unspecified period of time; the user may or may not re-bind to the provider to continue data transfer at some time prior to the end of the service instance provision period;

c) 'version not supported'—the user does not support the version of the RCF service proposed by the provider in the return from RCF-BIND; this value of unbind-reason shall be used only if the RCF-UNBIND is the first operation invoked following the RCF-BIND;

d) 'other'—the reason for the release will have to be found by other means.

3.3.2.4.3 If the RCF-UNBIND operation is invoked by the provider, the `unbind-reason` parameter shall take one of the following values:

a) 'end'—the provider has transferred all available frames to the user and is releasing the association normally; the provider shall not attempt to re-bind to the user in the context of this service instance;

b) 'suspend'—the provider is suspending service provision for an unspecified period of time; the provider may attempt to re-bind to the user to continue data transfer at some time prior to the end of the service instance provision period;

c) 'version not supported'—the provider does not support the version of the RCF service proposed by the user in the return from RCF-BIND; this value of `unbind-reason` shall be used only if the RCF-UNBIND is the first operation invoked following the RCF-BIND;

d) 'other'—the reason for the release will have to be found by other means.

3.3.2.5 `result`

The **`result`** parameter shall specify the result of the RCF-UNBIND operation and shall always contain the following value:

'positive result'—the RCF-UNBIND operation has been performed by the responder, and the association is released.

NOTES

1 If RCF-UNBIND is invoked while the provider is not in the 'ready' state, the responder invokes RCF-PEER-ABORT; if the authentication level is 'all' and the invocation of RCF-UNBIND cannot be authenticated, the responder ignores it; but there is no situation in which the response to the invocation of RCF-UNBIND is to return 'negative result'.

2 The `result` parameter is returned for the RCF-UNBIND operation, even though the only permitted value is 'positive result', for consistency with other confirmed operations.

3 RCF-UNBIND is a confirmed operation in order to provide a definite indication to the initiator that the responder has performed the operation and the association is released.

3.3.3 EFFECTS

The RCF-UNBIND operation shall have the following effects:

a) The association between the initiator and the responder shall be released, and the initiator and the responder shall cease to communicate with each other.

b) The provider shall transition to state 1 ('unbound').

c) If `unbind-reason` is 'end', the provider may delete the service instance and release its resources.

d) If `unbind-reason` is not 'end', the initiator may attempt to re-bind at any time prior to the end of the service instance provision period.

NOTE – The performance of RCF-UNBIND for a particular service instance does not necessarily terminate the associated RCF production process (e.g., if `unbind-reason` is not 'end' and the delivery mode is complete online or if another service instance is dependent on the production).

3.4 RCF-START

3.4.1 PURPOSE

3.4.1.1 The user shall invoke the RCF-START operation to request that the provider begin the delivery of telemetry frames.

3.4.1.2 The provider shall return a report of the outcome of the performance of the RCF-START operation to the user.

3.4.1.3 Following a successful RCF-START, the provider shall deliver telemetry frames acquired from the space link to the user as quickly as those frames are available.

NOTES

1 Frames are delivered to the user by means of the RCF-TRANSFER-DATA operation (see 3.6).

2 Communications service delays may affect the rate at which available frames are delivered.

3.4.1.4 All frames delivered following the RCF-START but prior to the next RCF-STOP (see 3.5) shall be delivered in the order in which they were received from the space link.

3.4.1.5 The user may specify, as parameters of the RCF-START invocation, the ERTs of the first and last telemetry frames that are to be delivered by the provider.

3.4.1.6 The time parameters may be changed during an association by invoking an RCF-STOP followed by an RCF-START with new time parameters.

NOTE – This capability is intended primarily to support the offline delivery mode.

3.4.1.7 RCF-START is valid only in state 2 (‘ready’) and shall be invoked only by the user.

3.4.2 INVOCATION, RETURN, AND PARAMETERS

3.4.2.1 General

The parameters of the RCF-START operation shall be present in the invocation and return as specified in table 3-4.

3.4.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-START invocation (see 3.1.4.1).

Table 3-4: RCF-START Parameters

Parameter	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invoke-ID	M	M
start-time	M	
stop-time	M	
requested-global-VCID	M	
result		M
diagnostic		C

3.4.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-START (see 3.1.4.1).

3.4.2.4 invoke-ID

The RCF service provider shall return unchanged the user-supplied value of the invoke-ID parameter (see 3.1.6).

3.4.2.5 start-time

3.4.2.5.1 The value of the **start-time** parameter shall be 'null', or it shall be a time value that indicates that only frames with an ERT equal to or later than start-time shall be delivered.

3.4.2.5.2 For the online delivery mode, only frames acquired during the space link session associated with this service instance shall be delivered, regardless of the value of start-time.

3.4.2.5.3 For the offline delivery mode, the provider shall deliver all available frames that meet the delivery criteria regardless of the space link session in which they were acquired.

3.4.2.5.4 For the online delivery mode, if start-time is 'null', the data transfer shall begin with the next frame that is acquired from the space link.

3.4.2.5.5 For the offline delivery mode, start-time must not be 'null'.

3.4.2.5.6 To be valid, start-time must satisfy the following criteria:

a) for the online delivery mode, `start-time` must be equal to or later than the start time of the service instance provision period for this service instance;

b) for the online delivery mode, `start-time` must be earlier than the end time of the service instance provision period for this service instance;

c) `start-time` must be earlier than `stop-time` (see 3.4.2.6).

NOTE – The provider is able to deliver only frames that have been acquired from the space link. For example, in an online service instance, if `start-time` is earlier than the start time of the space link session, the first frame that is delivered will be the first frame acquired after the start of the space link session.

3.4.2.6 `stop-time`

3.4.2.6.1 The value of the **`stop-time`** parameter shall be 'null', or it shall be a time value that indicates that delivery of frames should cease when the next frame that would be delivered has an ERT later than `stop-time`.

3.4.2.6.2 For the online delivery mode, only frames acquired during the space link session associated with this service instance shall be delivered, regardless of `stop-time`.

3.4.2.6.3 For the offline delivery mode, the provider shall deliver all available frames that meet the delivery criteria regardless of the space link session in which they were acquired.

3.4.2.6.4 For the online delivery mode, if `stop-time` is 'null', the provider shall continue to transfer all frames that are acquired from the space link and satisfy the delivery criteria until either the user invokes an RCF-STOP operation or the association is released or aborted.

3.4.2.6.5 For the offline delivery mode, `stop-time` must not be 'null'.

3.4.2.6.6 To be valid, `stop-time` must satisfy the following criteria:

a) `stop-time` must be later than the `start-time` (see 3.4.2.5);

b) for the online delivery mode, `stop-time`, if not 'null', must be earlier than or equal to the end time of the service instance provision period for this service instance;

c) for the offline delivery mode, `stop-time` plus the offline processing latency must be earlier than the current time.

NOTES

1 Offline processing latency is the length of time after a frame is acquired from the space link before the frame or any fields contained in the frame is available for retrieval using the offline delivery mode. The actual value of offline processing

latency is negotiated between SLE Complex Management and SLE Utilization Management.

2 Offline delivery is only available for frames that already have been acquired when the RCF-START operation is invoked.

3.4.2.7 `requested-global-VCID`

3.4.2.7.1 The `requested-`**`global-VCID`** parameter shall identify the master channel or virtual channel that is to be provided to the user and shall consist of the TFVN, the SCID, and the VCID.

NOTES

1 The definitions of SCID and VCID depend on the TFVN. If the TFVN indicates that the virtual channel consists of TM Transfer Frames, then the definitions of SCID and VCID are as per reference [3]. If the frame version number indicates that the virtual channel consists of AOS Transfer Frames, then the definitions of SCID and VCID are as per reference [4].

2 The physical channel is not specified directly through the RCF service. Rather, the selection of physical channel is determined through the service package, which specifies the RAF service instance that is consumed by the RFP-FG that is producing the RCF service.

3 Depending on the configuration, for a given service instance, the selection of only one master channel or only one VC from a set of VCs (where the set may have a single member) or a single master channel plus a set of VCs is permitted. In case the permitted GVCID list contains a master channel but no virtual channels from that master channel, the service user is not permitted to request a virtual channel from this master channel.

3.4.2.7.2 The TFVN shall be a valid transfer frame version number defined by CCSDS.

NOTE – At the time of issuance of this Recommended Standard, the only valid TFVN were ‘00’ (version 1) and ‘01’ (version 2) (see references [3] and [4]).

3.4.2.7.3 The SCID shall be a valid spacecraft identifier as defined by CCSDS (see references [3] and [4]).

3.4.2.7.4 The VCID shall be a valid virtual channel identifier as defined by CCSDS (see references [3] and [4]) or it shall be the value ‘any’. The value ‘any’ indicates that a master channel, defined by the TFVN and the SCID, shall be provided by the RCF service. Otherwise, a virtual channel shall be provided by the RCF service.

3.4.2.8 `result`

The **`result`** parameter shall specify the result of the RCF-START operation and shall contain one of the following values:

a) 'positive result'—the RCF-START operation has been performed by the provider, and the provider shall henceforth invoke RCF-TRANSFER-DATA operations as needed to transfer to the user all available frames that meet the specified delivery criteria;

b) 'negative result'—the RCF-START operation has not been performed by the provider, and the provider shall not invoke any RCF-TRANSFER-DATA operations even if frames are available.

3.4.2.9 `diagnostic`

3.4.2.9.1 If `result` is 'negative result', the `diagnostic` parameter shall be present in the return, and its value shall be one of the following:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'out of service'—the provider has been taken out of service for an indefinite period by management action;

c) 'unable to comply'—the provider is unable to transfer data at this time because of a fault affecting the provider;

d) 'invalid start time'—the value of the `start-time` provided in the invocation is not valid;

e) 'invalid stop time'—the value of the `stop-time` provided in the invocation is not valid;

f) 'missing time value'—for the offline delivery mode, the value of `start-time` and/or `stop-time` was 'null';

g) 'invalid global-VCID'—the value specified for the `requested-global-VCID` parameter is not valid;

h) 'other reason'—the reason for the negative result will have to be found by other means.

3.4.2.9.2 If `result` is 'positive result', the `diagnostic` parameter shall not be present in the return.

3.4.3 EFFECTS

3.4.3.1 If `result` is 'positive result', the RCF-START operation shall have the following effects:

a) the provider shall transition to state 3 ('active');

b) in the 'active' state, the provider shall transfer frames to the user whenever they are available and satisfy the delivery criteria.

3.4.3.2 If `result` is 'negative result', the RCF-START operation shall have the following effects:

a) the provider shall remain in state 2 ('ready') and shall not deliver frames even if they are available;

b) if the diagnostic is 'unable to comply':

1) the user may re-invoke the RCF-START operation at a later time within the constraints of the service instance provision period;

2) if the provider's SLE Complex Management determines that being 'unable to comply' is more than a transient problem, the provider may invoke the RCF-PEER-ABORT operation.

3.5 RCF-STOP

3.5.1 PURPOSE

3.5.1.1 The user shall invoke the RCF-STOP operation to request that the provider stop delivering telemetry frames.

NOTE – Within the constraints of the service provision period, the user may re-enable frame delivery by invoking the RCF-START operation.

3.5.1.2 The provider shall provide a report of the outcome of the performance of the RCF-STOP operation to the user.

3.5.1.3 RCF-STOP is valid only in state 3 (‘active’) and shall be invoked only by the user.

3.5.2 INVOCATION, RETURN, AND PARAMETERS

3.5.2.1 General

The parameters of the RCF-STOP operation shall be present in the invocation and return as specified in table 3-5.

Table 3-5: RCF-STOP Parameters

Parameters	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`invoke-ID`	M	M
`result`		M
`diagnostic`		C

3.5.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-STOP invocation (see 3.1.4.1).

3.5.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-STOP (see 3.1.4.1).

3.5.2.4 `invoke-ID`

The RCF service provider shall return unchanged the user-supplied value of the `invoke-ID` parameter (see 3.1.6).

3.5.2.5 `result`

The **`result`** parameter shall specify the result of the RCF-STOP operation and shall contain one of the following values:

a) 'positive result'—the RCF-STOP operation has been performed by the provider, and the delivery of telemetry frames to the user has ceased;

b) 'negative result'—the RCF-STOP operation has not been performed by the provider for the reason specified by the `diagnostic` parameter, and the delivery of telemetry frames to the user continues.

3.5.2.6 `diagnostic`

3.5.2.6.1 If `result` is 'negative result', the `diagnostic` parameter shall be present in the return, and its value shall be one of the following:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'other reason'—the reason for the negative result will have to be found by other means.

3.5.2.6.2 If `result` is 'positive result', the `diagnostic` parameter shall not be present in the return.

3.5.3 EFFECTS

3.5.3.1 If `result` is 'positive result', the RCF-STOP operation shall have the following effects:

a) the provider shall cease invoking RCF-TRANSFER-DATA operations;

b) the provider shall build an RcfTransferBuffer SLE PDU from the transfer buffer contents and pass this SLE PDU to the communication service in accordance with the provision of 3.1.9;

c) the provider shall transition to state 2 ('ready').

3.5.3.2 If `result` is 'negative result', the provider shall remain in state 3 ('active') and shall continue processing unchanged.

3.6 RCF-TRANSFER-DATA

3.6.1 PURPOSE

3.6.1.1 The provider shall invoke the RCF-TRANSFER-DATA operation to deliver a telemetry frame to the user.

3.6.1.2 The RCF-TRANSFER-DATA operation shall be an unconfirmed operation.

NOTE – Although RCF-TRANSFER-DATA is an unconfirmed operation, it is assumed that the communications service provides certain guarantees, as described in 1.3.1.

3.6.1.3 RCF-TRANSFER-DATA is valid only in state 3 (‘active’) and shall be invoked only by the provider.

3.6.2 INVOCATION AND PARAMETERS

3.6.2.1 General

The parameters of the RCF-TRANSFER-DATA operation shall be present in the invocation as specified in table 3-6.

Table 3-6: RCF-TRANSFER-DATA Parameters

Parameters	Invocation
`invoker-credentials`	M
`earth-receive-time`	M
`antenna-ID`	M
`data-link-continuity`	M
`private-annotation`	M
`data`	M

3.6.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the user to authenticate the RCF-TRANSFER-DATA invocation (see 3.1.4.1).

3.6.2.3 `earth-receive-time`

The **`earth-receive-time`** parameter shall contain the UTC time at which the signal event corresponding to the leading edge of the first bit of the attached sync marker that

immediately preceded this telemetry frame was presented at the phase center of the antenna used to acquire the frame.

NOTES

1 The first bit of the frame is the first bit following the attached sync marker.

2 In case of punctured coding, the number of symbols influenced by each information bit is variable, depending on the puncture pattern. Missions applying such coding need to take the resulting jitter of the earth-receive-time annotation with respect to the beginning of the frame into account.

3.6.2.4 `antenna-ID`

3.6.2.4.1 The **`antenna-ID`** parameter shall indicate which antenna of the SLE Complex was used to acquire the frame.

NOTE – `antenna-ID` is provided specifically to identify the physical location used as the reference point for the `earth-receive-time` parameter.

3.6.2.4.2 SLE Complex Management and SLE Utilization Management shall mutually agree upon the allowable values for `antenna-ID` and their interpretation.

NOTE – It is assumed that the value of the `antenna-ID` parameter is a reference to the actual location information, which is provided outside the scope of this service.

3.6.2.5 `data-link-continuity`

3.6.2.5.1 The **`data-link-continuity`** parameter shall indicate whether the frame from which the RCF was extracted was the direct successor of the previous frame on the master or virtual channel selected by means the RCF-START operation.

3.6.2.5.2 The **`data-link-continuity`** parameter shall contain an integer value:

a) a value of '-1' shall indicate that this is the first frame after the start of production or the selected channel is a master channel carrying AOS Transfer Frames and therefore no information regarding a discontinuity on the channel can be provided.

NOTE – AOS Transfer Frames do not contain a master channel frame counter.

b) a value of '([$MCFC_n$ – $MCFC_{n-1}$ – 1] modulo 256)' if the selected channel is a master channel carrying TM Transfer Frames; $MCFC_n$ is the master channel frame count of the frame and $MCFC_{n-1}$ is the master channel frame count of the previous frame delivered by the production process for the given master channel.

c) a value of '([$VCFC_n$ – $VCFC_{n-1}$ – 1] modulo 256)' if the selected channel is a virtual channel carrying TM Transfer Frames; $VCFC_n$ is the virtual channel frame count of the frame and $VCFC_{n-1}$ is the virtual channel frame count of the previous frame delivered by the production process for the given virtual channel.

d) a value of '([$VCFC_n$ – $VCFC_{n-1}$ – 1] modulo 16777216)' if the selected channel is a virtual channel carrying AOS Transfer Frames; $VCFC_n$ is the virtual channel frame count of the frame and $VCFC_{n-1}$ is the virtual channel frame count of the previous frame delivered by the production process for the given virtual channel.

NOTE – The number of missing TM Transfer Frames reported is correct as long as the gap is less than 256 frames. For longer gaps it will normally be possible to resolve the ambiguity resulting from the modulo 256 count based on the ERT of the frames and the nominal frame rate on the given master channel or virtual channel. For AOS Transfer Frames, the likelihood of an incorrectly reported gap size is much lower.

3.6.2.6 `private-annotation`

The **`private-annotation`** parameter shall be used to convey additional information that may be associated with a frame:

a) it may be set to 'null' to indicate that there is no private annotation;

b) if not 'null', there must be a prior arrangement between SLE Complex Management and SLE Utilization Management regarding the contents and interpretation of this parameter.

3.6.2.7 `data`

The value of the **`data`** parameter shall be the telemetry frame acquired by the provider from the RAF channel for delivery to the user. The frame (i.e., a AOS transfer frame) is delivered to the user, without any check symbols.

NOTE – The value of the `data` parameter does not include the attached sync marker.

3.6.3 EFFECTS

The RCF-TRANSFER-DATA operation shall have the following effects:

a) a telemetry frame acquired by the provider from the space link shall be delivered to the user;

b) the provider shall remain in state 3 ('active').

3.7 RCF-SYNC-NOTIFY

3.7.1 PURPOSE

3.7.1.1 The RCF service provider shall invoke the RCF-SYNC-NOTIFY operation to notify the user of the occurrence of an event affecting the production of the RCF service.

NOTE – Notification of events may be of value to the user in understanding specific provider behavior, such as an interruption in frame delivery.

3.7.1.2 The RCF-SYNC-NOTIFY operation shall be an unconfirmed operation.

3.7.1.3 The order in which the RCF-SYNC-NOTIFY and RCF-TRANSFER-DATA operations are invoked shall reflect the actual chronology of events.

NOTE – For example, if an RCF-SYNC-NOTIFY operation is invoked after one RCF-TRANSFER-DATA operation but before another, then the event indicated by the notification occurred after the ERT of the frame associated with the preceding RCF-TRANSFER-DATA but before the ERT of the frame associated with the following RCF-TRANSFER-DATA.

3.7.1.4 RCF-SYNC-NOTIFY is valid only in state 3 (‘active’) and shall be invoked only by the provider.

3.7.2 INVOCATION AND PARAMETERS

3.7.2.1 General

The parameters of the RCF-SYNC-NOTIFY operation shall be present in the invocation as specified in table 3-7.

Table 3-7: RCF-SYNC-NOTIFY Parameters

Parameter	Invocation
`invoker-credentials`	M
`notification-type`	M
`notification-value`	C

3.7.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the user to authenticate the RCF-SYNC-NOTIFY invocation (see 3.1.4.1).

3.7.2.3 `notification-type`

The **`notification-type`** parameter shall indicate the event that the user is being notified of, and its value shall be one of the following:

a) 'loss of frame synchronization'—the delivery of frames has been interrupted because the frame synchronization process is not able to synchronize to the stream of frames from the space link:

   1) the notification shall be invoked once if the frame synchronizer transitioned from 'in-lock' to 'out-of-lock' at least once during the lock status observation period that the provider applies for lock status monitoring; the length of the lock status observation period applied by the provider shall be documented;

   2) the provider shall minimize the latency from the time the loss of frame synchronization event occurs until the notification is invoked;

   3) there shall be no explicit notification when the frame synchronizer transitions from 'out-of-lock' to 'in-lock'; rather, the next invocation of RCF-TRANSFER-DATA shall implicitly indicate the occurrence of that event;

   4) loss of frame synchronization notifications shall not be invoked in the offline delivery mode;

   NOTE – Because this notification refers to processing of frames from the space link, it may or may not indicate that frames were lost on the master channel or virtual channel being provided by this instance of service.

b) 'production status change'—the status of RCF production has changed:

   1) the notification shall be invoked when the RCF production status changes;

   2) the production status shall be 'running', 'halted', or 'interrupted' (see 3.7.2.4);

   3) production status change notifications shall not be invoked in the offline delivery mode;

c) 'data discarded due to excessive backlog'—some data was discarded by the RCF service provider, either because of timeliness considerations (timely online mode) or because of online frame buffer overflow (complete online mode):

   1) if data are discarded two or more times in a row without a successful intervening delivery of frames to the user, no more than one data discarded notification shall be delivered to the user;

   2) data discarded notifications shall not be invoked in the offline delivery mode;

d) 'end of data'—the provider has no more data to send.

NOTE – The 'end of data' notification is invoked in all delivery modes. For example, for an online service instance, the space link session has ended, and there are no more frames to be delivered; or, regardless of the delivery mode, all available frames between the specified start and stop times (see 3.4) have been delivered.

3.7.2.4 `notification-value`

3.7.2.4.1 The presence of the **`notification-value`** parameter in the return from RCF-SYNC-NOTIFY shall be conditional on the value of `notification-type`.

3.7.2.4.2 If `notification-type` is 'loss of frame synchronization', then `notification-value` shall be present and shall convey the following information:

a) the UTC time when the frame synchronizer transitioned from 'in-lock' to 'out-of-lock';

b) the current status of the carrier demodulation process, which shall be 'in-lock', 'out-of-lock', or 'unknown';

c) the current status of the subcarrier demodulation process, which shall be 'in-lock', 'out-of-lock', 'not in use', or 'unknown';

d) the current status of the symbol synchronization process, which shall be 'in-lock', 'out-of-lock', or 'unknown'.

NOTE – The determinations of the lock statuses of carrier demodulation, subcarrier demodulation, and symbol synchronization typically are based on measurements that are integrated over some time period. To that extent, the values reported here may reflect the statuses of the corresponding processes at a time slightly earlier than the time when the notification is invoked.

3.7.2.4.3 If `notification-type` is 'production status change', then the `notification-value` parameter shall be present, and its value shall indicate the current production status, which shall be one of the following:

a) 'running'—the RCF production process is capable of processing a return space link physical channel, if available;

b) 'halted'—the RCF production process is stopped and production equipment is out of service, due to management action;

c) 'interrupted'—the RCF production process is stopped due to a fault.

3.7.2.4.4 If `notification-type` is 'data discarded due to excessive backlog' or 'end of data', the `notification-value` parameter shall not be present.

3.7.3 EFFECTS

The RCF-SYNC-NOTIFY operation shall have the following effects:

a) information about the occurrence of the specified event shall be delivered to the user;

b) the state of the provider shall not change.

3.8 RCF-SCHEDULE-STATUS-REPORT

3.8.1 PURPOSE

3.8.1.1 The user shall invoke the RCF-SCHEDULE-STATUS-REPORT operation to request that the provider send a status report either immediately or periodically or to stop the sending of such reports.

3.8.1.2 The provider shall return a report of the outcome of the performance of the RCF-SCHEDULE-STATUS-REPORT operation to the user.

3.8.1.3 The provider shall send the requested status report(s) by means of the RCF-STATUS-REPORT operation (see 3.9).

3.8.1.4 Initially (i.e., whenever the RCF-BIND operation is performed and the provider transitions from state 1 to state 2), periodic reporting shall be stopped.

3.8.1.5 When periodic reporting is enabled, the user may change the reporting period by invoking another RCF-SCHEDULE-STATUS-REPORT operation.

3.8.1.6 The RCF-SCHEDULE-STATUS-REPORT operation shall be rejected by the provider if this service instance is configured to the offline delivery mode.

3.8.1.7 The RCF-SCHEDULE-STATUS-REPORT operation is valid only in states 2 (‘ready’) and 3 (‘active’).

3.8.1.8 The RCF-SCHEDULE-STATUS-REPORT operation shall be invoked only by the user.

3.8.2 INVOCATION, RETURN, AND PARAMETERS

3.8.2.1 General

The parameters of the RCF-SCHEDULE-STATUS-REPORT operation shall be present in the invocation and return as specified in table 3-8.

3.8.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-SCHEDULE-STATUS-REPORT invocation (see 3.1.4.1).

3.8.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-SCHEDULE-STATUS-REPORT (see 3.1.5).

Table 3-8: RCF-SCHEDULE-STATUS-REPORT Parameters

Parameters	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invoke-ID	M	M
report-request-type	M	
reporting-cycle	C	
result		M
diagnostic		C

3.8.2.4 invoke-ID

The performer shall return unchanged the invoker-supplied value of the invoke-ID parameter (see 3.1.6).

3.8.2.5 report-request-type

3.8.2.5.1 The **report-request-type** parameter shall specify how reporting shall be done, and its value shall be one of the following:

a) 'immediately'—send a single status report immediately;

b) 'periodically'—send a status report every reporting-cycle seconds;

c) 'stop'—do not send further status reports.

3.8.2.5.2 If report-request-type is 'immediately',

a) the provider shall stop sending status reports after the immediate status report has been sent;

b) periodic reporting may be restarted by means of another RCF-SCHEDULE-STATUS-REPORT operation.

3.8.2.6 reporting-cycle

3.8.2.6.1 If the value of the report-request-type parameter is 'periodically', then the **reporting-cycle** parameter shall be present and shall specify the requested interval between status reports in seconds.

3.8.2.6.2 If the value of the report-request-type parameter is not 'periodically', then the reporting-cycle parameter shall not be present.

3.8.2.7 `result`

The **`result`** parameter shall specify the result of the RCF-SCHEDULE-STATUS-REPORT operation, and its value shall be one of the following:

a) 'positive result'—the RCF-SCHEDULE-STATUS-REPORT operation has been performed, and the provider will send the requested report(s) or stop sending periodic status reports;

b) 'negative result'— the RCF-SCHEDULE-STATUS-REPORT operation has not been performed for the reason specified in the `diagnostic` parameter. The previous setting for status reporting remains in effect.

3.8.2.8 `diagnostic`

3.8.2.8.1 If `result` is 'negative result', the **`diagnostic`** parameter shall be present in the return, and its value shall be one of the following:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'not supported in this delivery mode'—the service instance is configured to the offline delivery mode;

c) 'already stopped'—the provider is not currently providing periodic reports (applicable only when `report-request-type` is 'stop');

d) 'invalid reporting cycle'—the requested `reporting-cycle` value is outside the range mutually agreed upon by SLE Complex Management and SLE Utilization Management;

e) 'other reason'—the reason for rejection of the operation will have to be found by other means.

3.8.2.8.2 If `result` is 'positive result', the `diagnostic` parameter shall not be present in the return.

3.8.3 EFFECTS

3.8.3.1 If `result` is 'positive result', the RCF-SCHEDULE-STATUS-REPORT operation shall have the following effects, depending on the value of the `reporting-request-type` parameter:

a) if the value of `reporting-request-type` is 'immediately':

   1) a status report shall be sent immediately;

   2) the sending of any previously requested periodic status reports shall cease;

b) if the value of `reporting-request-type` is 'periodically':

1) a status report shall sent immediately;

2) subsequent status reports shall be sent at the interval specified in the `reporting-cycle` parameter;

c) if the value of `reporting-request-type` is ‘stop’, periodic status reporting shall cease.

3.8.3.2 If `result` is ‘negative result’, the RCF-SCHEDULE-STATUS-REPORT operation shall have no effect, and the previous setting for status reporting shall not change.

3.8.3.3 The state of the provider shall not change.

3.9 RCF-STATUS-REPORT

3.9.1 PURPOSE

3.9.1.1 The provider shall invoke the RCF-STATUS-REPORT operation to send a status report to the user.

3.9.1.2 RCF-STATUS-REPORT shall be an unconfirmed operation.

3.9.1.3 Status reports shall be sent (or not sent) in accordance with user requests conveyed by means of the RCF-SCHEDULE-STATUS-REPORT operation (see 3.8).

3.9.1.4 The RCF-STATUS-REPORT operation is valid only in states 2 (‘ready’) and 3 (‘active’) and shall be invoked only by the provider.

3.9.2 INVOCATION AND PARAMETERS

3.9.2.1 General

The parameters of the RCF-STATUS-REPORT operation shall be present in the invocation as specified in table 3-9.

Table 3-9: RCF-STATUS-REPORT Parameters

Parameters	Invocation
`invoker-credentials`	M
`number-of-frames-delivered`	M
`frame-sync-lock-status`	M
`symbol-sync-lock-status`	M
`subcarrier-lock-status`	M
`carrier-lock-status`	M
`production-status`	M

3.9.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-STATUS-REPORT invocation (see 3.1.5).

3.9.2.3 number-of-frames-delivered

The **number-of-frames-delivered** parameter shall specify the total number of telemetry frames with the requested-global-VCID value that have been delivered to the user since the start of the service instance provision period.

NOTE – This parameter is equivalent to the number of frames that an RCF service instance with the same requested-global-VCID value would deliver to the user while the service instance is in the active state.

3.9.2.4 frame-sync-lock-status

The **frame-sync-lock-status** parameter shall specify the current lock status of the frame synchronization process, the value of which shall be 'in-lock', 'out-of-lock', or 'unknown'.

3.9.2.5 symbol-sync-lock-status

The **symbol-sync-lock-status** parameter shall specify the current lock status of the symbol (or bit) synchronization process, the value of which shall be 'in-lock', 'out-of-lock', or 'unknown'.

3.9.2.6 subcarrier-lock-status

The **subcarrier-lock-status** parameter shall specify the current lock status of the subcarrier demodulation process, the value of which shall be 'in-lock', 'out-of-lock', or 'unknown'.

3.9.2.7 carrier-lock-status

The **carrier-lock-status** parameter shall specify the current lock status of the carrier demodulation process, the value of which shall be 'in-lock', 'out-of-lock', 'not in use', or 'unknown'.

3.9.2.8 production-status

The **production-status** parameter shall specify the current status of RCF production, the value of which shall be 'running', 'halted', or 'interrupted'.

NOTE – See 3.7.2.4 for a description of the production-status values.

3.9.3 EFFECTS

The RCF-STATUS-REPORT operation shall have the following effects:

a) status information shall be delivered to the user;

b) the state of the provider shall not change.

3.10 RCF-GET-PARAMETER

3.10.1 PURPOSE

3.10.1.1 The user shall invoke the RCF-GET-PARAMETER operation to ascertain the value of an RCF service parameter.

3.10.1.2 The provider shall return a report of the outcome of the performance of the RCF-GET-PARAMETER operation to the user.

3.10.1.3 If the operation is successful, the current value of the specified RCF service parameter shall be provided to the user in the return from the operation.

3.10.1.4 RCF-GET-PARAMETER is valid in state 2 (‘ready’) and state 3 (‘active’) and shall be invoked only by the user.

3.10.2 INVOCATION, RETURN, AND PARAMETERS

3.10.2.1 General

The parameters of the RCF-GET-PARAMETER operation shall be present in the invocation and return as specified in table 3-10.

Table 3-10: RCF-GET-PARAMETER Parameters

Parameters	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`invoke-ID`	M	M
`rcf-parameter`	M	C
`parameter-value`		C
`result`		M
`diagnostic`		C

3.10.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RCF-GET-PARAMETER invocation (see 3.1.5).

3.10.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RCF-GET-PARAMETER (see 3.1.4.1).

3.10.2.4 `invoke-ID`

The performer shall return unchanged the invoker-supplied value of the **`invoke-ID`** parameter (see 3.1.6).

3.10.2.5 `rcf-parameter`

3.10.2.5.1 The **`rcf-parameter`** parameter shall specify the RCF service parameter whose value is to be returned to the user, and its value shall be one of the values listed in table 3-11.

3.10.2.5.2 The `rcf-parameter` parameter is conditionally present in the return based on the `result` parameter:

a) if the value of `result` is 'positive result', `rcf-parameter` shall be present in the return;

b) if the value of `result` is 'negative result', `rcf-parameter` shall not be present in the return.

3.10.2.6 `parameter-value`

3.10.2.6.1 The **`parameter-value`** parameter shall contain the value for the parameter specified by `rcf-parameter` as described in 3.10.2.5 and table 3-11.

3.10.2.6.2 The `parameter-value` parameter is conditionally present in the return based on the `result` parameter:

a) if the value of `result` is 'positive result', `parameter-value` shall be present;

b) if the value of `result` is 'negative result', `parameter-value` shall not be present.

3.10.2.7 result

The **`result`** parameter shall specify the result of the RCF-GET-PARAMETER operation and shall contain one of the following values:

a) 'positive result'—the RCF-GET-PARAMETER operation has been performed, and the value of the specified RCF service parameter is provided in the return to the user;

b) 'negative result'—the RCF-GET-PARAMETER operation has not been performed for the reason specified in the `diagnostic` parameter.

Table 3-11: RCF Parameters

Parameter	Description
delivery-mode	The delivery mode for this instance of RCF service, which is set by service management (see 3.1.9): its value shall be 'timely online delivery mode', 'complete online delivery mode', or 'offline delivery mode'
latency-limit	The maximum allowable delivery latency time (in seconds) for the online delivery mode, as defined in 3.1.9.1 (i.e., the maximum delay from when the frame is acquired by the provider until the RCF extracted from it is delivered to the user): the value of this parameter shall be 'null' if the delivery mode is offline.
permitted-global-VCID-set	The set of global VCIDs permitted for this RCF service instance (see 3.4.2.7).
reporting-cycle	The current setting of the reporting cycle for status reports (see 3.8 and 3.9): the value is 'null' if cyclic reporting is off, otherwise it is the time (in seconds) between successive RCF-STATUS-REPORT invocations (see 3.8).
requested-global-VCID	If the provider is in state 3 ('active', the GVCID set by the RCF-START operation, used to determine which frames are selected for delivery. If the provider is not in state 3 ('active'), the GVCID value returned shall be 'undefined'.
return-timeout-period	The maximum time period (in seconds) permitted from when a confirmed RCF operation is invoked until the return is received by the invoker (see 4.1.3).
transfer-buffer-size	The size of the transfer buffer: the value of this parameter shall indicate the number of RCF-TRANSFER-DATA and RCF-SYNC-NOTIFY invocations that can be stored in the transfer buffer. The precise specification of the transfer buffer size may be found in 3.1.9.

3.10.2.8 `diagnostic`

3.10.2.8.1 If `result` is 'negative result', the `diagnostic` parameter shall be present in the return, and its value shall be one of the following:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'unknown parameter'—the value of `rcf-parameter` does not identify an RCF parameter that is recognized by the service provider;

c) 'other reason'—the reason for the negative result will have to be found by other means.

3.10.2.8.2 If `result` is 'positive result', the `diagnostic` parameter shall not be present in the return.

3.10.3 EFFECTS

3.10.3.1 If `result` is 'positive result', the value of the RCF parameter specified in the invocation shall be provided to the user in the return.

3.10.3.2 If `result` is 'negative result', no RCF parameter value shall be returned to the user.

3.10.3.3 The state of the provider shall not change.

3.11 RCF-PEER-ABORT

3.11.1 PURPOSE

3.11.1.1 The user or provider shall invoke the RCF-PEER-ABORT operations to notify the peer system that the local application detected an error that requires that the association between them be terminated abnormally.

3.11.1.2 RCF-PEER-REPORT shall be an unconfirmed operation.

3.11.1.3 RCF-PEER-ABORT is valid only in states 2 (‘ready’) and 3 (‘active’) and may be invoked by either the user or the provider.

3.11.2 INVOCATION AND PARAMETERS

3.11.2.1 General

The parameters of the RCF-PEER-ABORT operation shall be present in the invocation as specified in table 3-12.

Table 3-12: RCF-PEER-ABORT Parameters

Parameters	Invocation
`diagnostic`	M

3.11.2.2 `diagnostic`

The **`diagnostic`** parameter shall specify why the RCF-PEER-ABORT is being invoked, and its value shall be one of the following:

a) ‘access denied’—a responder with an identity as presented in the `responder-identifier` parameter of the RCF-BIND return is not known to the initiator (e.g., the value of the `responder-identifier` parameter does not match the authorized responder for any service instance known to the initiator);

b) ‘unexpected responder ID’—the value of the `responder-identifier` parameter in the RCF-BIND return does not match the identity of the authorized responder for this service instance as specified by service management;

c) ‘operational requirement’—the local system had to terminate the association to accommodate some other operational need;

d) ‘protocol error’—the local application detected an error in the sequencing of RCF service operations;

e) ‘communications failure’—the communications service on the other side of a gateway was disrupted;

NOTE – This `diagnostic` value is only applicable when the SLE applications are communicating via a gateway.

f) ‘encoding error’—the local application detected an error in the encoding of one or more operation parameters or did not recognize the operation;

g) ‘return timeout’—the local application detected that the return from a confirmed operation was not received within a specified time limit;

h) ‘end of service instance provision period’—the local application detected that the service instance provision period has ended and the initiator has not invoked the RCF-UNBIND operation;

i) ‘unsolicited invoke-ID’—the local application received a return with an invoke-ID that does not match the invoke-ID of any of the operations for which a return is pending;

j) ‘other reason’—the local application detected an unspecified error during the processing of one or more operations.

NOTE – RCF-PEER-ABORT does not carry an `invoker-credentials` parameter. It is conceivable that an intruder may use the RCF-PER-ABORT operation for a denial-of-service attack. If an intruder has that capability, then a denial-of-service attack can be much more easily accomplished by disrupting communications at a layer lower than the applications layer. Therefore, authentication of RCF-PEER-ABORT would not provide improved protections against such attacks.

3.11.3 EFFECTS

The RCF-PEER-ABORT operation shall have the following effects:

a) the association shall be aborted, and the user and the provider shall cease to communicate with each other;

b) the provider shall transition to state 1 (‘unbound’);

c) the provider shall discard the contents of the transfer buffer;

d) statistical information required for the generation of the status report shall be retained throughout the service instance provision period.

4 RCF PROTOCOL

4.1 GENERIC PROTOCOL CHARACTERISTICS

NOTE – This section specifies the handling of invalid SLE-PDUs and other failures affecting the protocol.

4.1.1 UNEXPECTED PROTOCOL DATA UNIT

If the peer application invokes an operation not allowed in the current state of the performer, the performer shall abort the association by invoking the RCF-PEER-ABORT operation with the `diagnostic` parameter set to 'protocol error'.

4.1.2 INVALID PROTOCOL DATA UNIT

If the application receives an invocation or return that contains an unrecognized operation type, contains a parameter of the wrong type, or is otherwise not decodable, the application shall abort the association by invoking the RCF-PEER-ABORT operation with the `diagnostic` parameter set to 'encoding error'.

4.1.3 MISSING RETURN

For confirmed operations, if the invoker does not receive the return from the performer within a timeout period specified by service management, the invoker shall abort the association by invoking the RCF-PEER-ABORT operation with the `diagnostic` parameter set to 'return timeout'.

NOTES

1 The timeout period shall be chosen taking into account performance of user and provider applications as well as the delays introduced by the underlying communications service.

2 In order to provide responsive service and short timeout periods, the generation of the return from an operation must not depend on any human interaction.

3 After invoking the RCF-UNBIND operation, the initiator must not invoke any further operations with the exception of the case addressed in 3.3.1.4 nor send any returns. The responder is not required to send any pending returns after having received the RCF-UNBIND invocation. Therefore, following an RCF-UNBIND invocation, the 'missing return' event may occur.

4.1.4 UNSOLICITED RETURN

If the application receives a return with an `invoke-ID` parameter value that does not correspond to any invocation for which a return is still pending, the application shall abort the association by invoking the RCF-PEER-ABORT operation with the `diagnostic` parameter set to 'unsolicited Invoke-ID'.

4.1.5 COMMUNICATIONS FAILURE

4.1.5.1 Every SLE entity (i.e., every SLE user or provider) that is in an association (bound) with a peer SLE entity shall maintain knowledge of the health of the communications interface with the peer.

4.1.5.2 Every SLE implementation shall provide that, for every association, the two SLE entities in the association maintain a consistent view of the health of the communications interface between them.

4.1.5.3 If an SLE entity determines that communications with the peer SLE entity have been disrupted (e.g., because of a communications service fault), then the SLE entity shall consider that the association with the peer has been aborted.

NOTE – The exact criteria for determining when communications have been disrupted may depend on the characteristics of the underlying communications service and may be specific to a given implementation. However, every RCF user and provider implementation shall provide for monitoring the health of the communications interface and for ensuring that the user and the provider have a consistent view of the health of the communications interface. If the underlying communications service does not intrinsically provide such a capability, the transmission of a periodic 'heartbeat' indicator or equivalent may need to be implemented.

4.1.5.4 Occurrence of the above-described communications failure event shall be referred to as a 'protocol abort'.

4.1.5.5 Subsequent to a 'protocol abort' event:

a) the RCF provider shall transition to state 1 ('unbound');

b) neither the user nor the provider shall attempt further communications with the peer except that the initiator may attempt to re-establish the association by invoking the RCF-BIND operation;

c) the provider shall discard the contents of the transfer buffer;

d) the values of RCF service parameters shall return to the initial values set by service management for that service instance; and

e) statistical information required for the generation of the status report shall be retained throughout the service instance provision period.

4.1.6 ACCESS CONTROL

4.1.6.1 The initiator of an association shall present its own identity in the `initiator-identifier` parameter of the RCF-BIND invocation.

4.1.6.2 If the RCF-BIND operation is invoked with a value of `initiator-identifier` that is not known to the responder, the responder shall not make any attempt to authenticate that invocation. Instead, the responder shall generate an RCF-BIND return with `result` set to 'negative result', `diagnostic` set to 'access denied', and `performer-credentials` set to 'unused'.

4.1.6.3 If the value of `initiator-identifier` is known to the responder, the responder shall attempt to authenticate the RCF-BIND invocation (see 3.1.5) as required for the given initiator. If authentication succeeds but the initiator is not the authorized initiator for the service instance indicated in the `service-instance-identifier` parameter of the RCF-BIND invocation, the responder shall generate an RCF-BIND return with `result` set to 'negative result' and `diagnostic` set to 'service instance not accessible to this initiator'.

NOTE – If authentication fails, the responder shall behave as specified in 4.1.7. If authentication is not required for the given initiator, it shall be as if authentication was successful.

4.1.6.4 If the initiator receives an RCF-BIND return with a `responder-identifier` value that is not known to the initiator, the initiator shall not make any attempt to authenticate this return but shall abort the association by invoking RCF-PEER-ABORT with `diagnostic` set to 'access denied'.

4.1.6.5 If the initiator receives an RCF-BIND return with a `responder-identifier` value that is known to the initiator, the initiator shall attempt to authenticate the RCF-BIND return (see 3.1.5) as required for the given responder. If authentication succeeds but the `responder-identifier` is not the authorized responder for this service instance as specified by service management, the initiator shall abort the association by means of the RCF-PEER-ABORT operation with `diagnostic` set to 'unexpected responder ID'.

NOTE – If authentication fails, the initiator shall behave as specified in 4.1.7. If authentication is not required for the given responder, it shall be as if authentication was successful.

4.1.7 FAILING AUTHENTICATION

4.1.7.1 An incoming invocation or return shall be ignored if the credentials parameter cannot be authenticated when, by management arrangement, credentials are required.

4.1.7.2 If an invocation is ignored, the operation shall not be performed, and a report of the outcome shall not be returned to the invoker.

4.1.7.3 If a return is ignored, it shall be as if no report of the outcome of the operation has been received.

4.2 RCF SERVICE PROVIDER BEHAVIOR

4.2.1 GENERAL REQUIREMENTS

4.2.1.1 The behavior of the RCF service provider shall conform to the state transition matrix specified in table 4-1.

4.2.1.2 All actions including state transitions specified for given state and a given event shall be performed before a subsequent event is considered.

4.2.1.3 SLE-PDUs shall be sent in the sequence specified in table 4-1.

4.2.1.4 Implementations shall ensure that events are not lost while an earlier event is being processed but are buffered in first-in first-out order for processing as soon as processing of the earlier event has completed.

4.2.1.5 The state transition matrix specified in table 4-1 represents one instance of service and thus one association. Once the association is established, if an RCF-BIND invocation for a different association but for the same service instance is received, it shall be rejected with an RCF-BIND return with the `result` parameter set to 'negative result' and the `diagnostic` parameter set to 'already bound'. This event shall not affect the association already in place.

4.2.2 STATE TRANSITION TABLE

NOTES

1 The state specifies operation interactions and state transitions for the service provider in its role as either initiator or responder.

2 The leftmost column simply numbers the rows of the table.

3 The second column of the state lists all incoming events. Where these events correspond to the arrival of an incoming SLE-PDU, the ASN.1 type defined for this SLE-PDU in annex A is indicated in parentheses ( ).

4 Where an event is internal to the provider, its description is put in single quotation marks ' '. These events are defined in table 4-2.

5 The three columns (one column per state) on the right side of the specify the behavior the provider will exhibit, which depends on the current state and the incoming event. In some cases, the behavior in addition depends on Boolean conditions, also referred to as predicates. Such conditions are put in double quotation marks " ". The

predicates are defined in table 4-3. Predicates that are simple Boolean variables set only by that state machine itself are referred to as Boolean flags and specified in table 4-4. The dependency on a predicate is presented in form of an IF <condition> THEN <action> [ELSEIF <condition> THEN <action>] ELSE <action> clause.

6 If the action given in the table is simply to send a specific SLE-PDU, that is indicated by the appearance of the name of ASN.1 type of the SLE-PDU to be sent in parentheses ( ). If that SLE-PDU is a return, the name may be preceded by the plus symbol ('+') to indicate that `result` is 'positive result' or by the negative symbol ('-') to indicate 'negative result'. Where several actions are to be taken (referred to as a 'compound action'), the name of the compound action is put in curly braces { }. The individual actions making up each compound action are identified in table 4-5.

7 'Not applicable' is stated where the given event can only occur in the given state because of an implementation error on the provider side.

8 Where the consequences of an incoming event are not visible to the user because the provider does not send any SLE-PDU in reaction to the given event, the action is put in square brackets [].

9 State transitions are indicated by an arrow and the number of the state that will be entered; for example, → 1 indicates the transition to state 1.

10 The actions to be taken and the state transition are considered to be one atomic action. The sequence shown in the table is irrelevant except that SLE-PDUs shall be sent in the sequence stated in the table.

11 Whenever the provider invokes a confirmed operation with `invoke-ID` set to <n>, it shall start an associated return <n> timer. Should this timer expire before the return <n> is received, the provider shall invoke RCF-PEER-ABORT.

Table 4-1: Provider Behavior

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
1	'start of service instance provision period'	IF "provider initiated" THEN {invoke bind} → 1 ELSE [ignore] → 1	Not applicable	Not applicable
2	'return <n> timer expired'	IF "bind pending" THEN {return timeout} → 1 IF "provision period" THEN {invoke bind} ELSE [ignore] ELSE Not applicable → 1	{peer abort 'return timeout'} → 1	{peer abort 'return timeout'} → 1
3	(-rcfBindReturn)	IF "bind pending" THEN set "bind pending" FALSE → 1 stop return <n> timer IF "retry permitted" THEN {invoke bind} ELSE release resources ELSE [ignore] → 1	{peer abort 'protocol error'} → 1	{peer abort 'protocol error'} → 1
4	(+rcfBindReturn)	IF "bind pending" THEN set "bind pending" FALSE → 2 stop return <n> timer IF NOT "compatible" THEN {invoke unbind} ELSE [ignore] ELSE [ignore] → 1	{peer abort 'protocol error'} → 1	{peer abort 'protocol error'} → 1
5	(rcfBindInvocation)	IF "provider initiated" THEN [ignore] → 1 ELSE IF "positive result" THEN (+rcfBindReturn) → 2 ELSE (-rcfBindReturn) → 1	{peer abort 'protocol error'} → 1	{peer abort 'protocol error'} → 1

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
6	'end of service instance provision period'	[ignore]	IF "provider initiated" THEN {invoke unbind} → 2 ELSE {peer abort 'end of service instance provision period'} → 1	{peer abort 'end of service instance provision period'} → 1
7	(rcfUnbindReturn)	[ignore]	IF "unbind pending" THEN {provider unbind} → 1 IF "done" THEN release resources ELSE [ignore] ELSE {peer abort 'protocol error'} → 1	{peer abort 'protocol error'} → 1
8	(rcfUnbindInvocation)	[ignore]	IF "provider initiated" THEN {peer abort 'protocol error'} → 1 ELSE {user unbind} → 1 IF "end" THEN release resources ELSE [ignore]	{peer abort 'protocol error'} → 1
9	(rcfStartInvocation)	[ignore]	IF "unbind pending" THEN {peer abort 'protocol error'} → 1 ELSE IF "positive result" THEN (+rcfStartReturn) → 3 initialize transfer buffer ELSE (-rcfStartReturn) → 2	{peer abort 'protocol error'} → 1
10	(rcfStopInvocation) "complete online" or "offline" delivery mode	[ignore]	{peer abort 'protocol error'} → 1	IF "positive result" THEN → 2 IF (NOT "buffer empty") THEN {transmit buffer} (+rcfStopReturn) ELSE (+rcfStopReturn) ELSE (-rcfStopReturn) → 3

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
11	(rcfStopInvocation) “timely online” delivery mode	[ignore]	{peer abort ‘protocol error’} → 1	IF “positive result” THEN → 2 IF (NOT “buffer empty”) THEN {pass buffer contents} (+rcfStopReturn) ELSE (+rcfStopReturn) ELSE (-rcfStopReturn) → 3
12	‘data available’, “offline” delivery mode	Not applicable	Not applicable	IF “buffer full” THEN {transmit buffer} → 3 {insert annotated frame} ELSE {insert annotated frame}→ 3
13	‘data available’, “complete online” delivery mode	Not applicable	Not applicable	IF “buffer full” THEN {transmit buffer} → 3 {insert annotated frame} {start release timer} ELSE IF “buffer empty” THEN {insert annotated frame}→ 3 {start release timer} ELSE {insert annotated frame}→ 3
14	‘data available’, “timely online” delivery mode	Not applicable	Not applicable	IF “buffer full” THEN {pass buffer contents} → 3 IF “congested” THEN {increment buffer size by one} {sync notify ‘data discarded’} {insert annotated frame} {start release timer} ELSE {insert annotated frame} {start release timer} ELSE IF “buffer empty” THEN {insert annotated frame} → 3 {start release timer} ELSE {insert annotated frame}→ 3

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
15	'release timer expired', "timely online" delivery mode	Not applicable	Not applicable	{pass buffer contents} → 3 IF "congested" THEN {increment buffer size by one} {sync notify 'data discarded'} {start release timer} ELSE [ignore]
16	'release timer expired', "complete online" delivery mode	Not applicable	Not applicable	{transmit buffer} → 3
17	'end of data', "timely online" delivery mode	Not applicable	Not applicable	IF "buffer full" THEN {pass buffer contents} → 3 IF "congested" THEN {sync notify 'data discarded'} {sync notify 'end of data'} {transmit buffer} ELSE {sync notify 'end of data'} {transmit buffer} ELSE → 3 {sync notify 'end of data'} {transmit buffer}
18	'end of data', "complete online" delivery mode or "offline" delivery mode	Not applicable	Not applicable	IF "buffer full" THEN {transmit buffer} → 3 {sync notify 'end of data'} {transmit buffer} ELSE {sync notify 'end of data'} → 3 {transmit buffer}

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
19	'loss of frame synchronization', "timely online" delivery mode	Not applicable	[ignore] → 2	IF "buffer full" THEN {pass buffer contents} → 3 IF "congested" THEN {sync notify 'data discarded'} {start release timer} {sync notify 'loss of frame sync'} ELSE {sync notify 'loss of frame sync'} {start release timer} ELSE IF "buffer empty" THEN {sync notify 'loss of frame sync'} → 3 {start release timer} ELSE {sync notify 'loss of frame sync'} → 3
20	'loss of frame synchronization', "complete online" delivery mode	Not applicable	[ignore] → 2	IF "buffer full" THEN {transmit buffer} → 3 {sync notify 'loss of frame sync'} {start release timer} ELSE IF "buffer empty" THEN {sync notify 'loss of frame sync'} → 3 {start release timer} ELSE {sync notify 'loss of frame sync'} → 3
21	'production status change', "timely online" delivery mode or "complete online" delivery mode	Not applicable	IF NOT "unbind pending" THEN {sync notify 'production status change'} → 2 ELSE [ignore] → 2	{sync notify 'production status change'} → 3

No.	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
22	(rcfScheduleStatusReportInvocation)	[ignore]	IF “positive result” THEN (+rcfScheduleStatusReport) → 2 IF “immediately” THEN {immediate report} ELSE IF “periodically” THEN {periodic report} ELSE stop reporting-cycle timer ELSE → 2 (-rcfScheduleStatusReportReturn)	IF “positive result” THEN (+rcfScheduleStatusReport) → 3 IF “immediately” THEN {immediate report} ELSE IF “periodically” THEN {periodic report} ELSE stop reporting-cycle timer ELSE → 3 (-rcfScheduleStatusReportReturn)
23	‘reporting-cycle timer expired’	Not applicable	{periodic report} → 2	{periodic report} → 3
24	(rcfGetParameterInvocation)	[ignore]	IF “positive result” THEN (+rcfGetParameterReturn) → 2 ELSE (-rcfGetParameterReturn) → 2	IF “positive result” THEN (+rcfGetParameterReturn) → 3 ELSE (-rcfGetParameterReturn) → 3
25	(rcfPeerAbortInvocation)	[ignore]	{clean up} → 1	{clean up} → 1
26	‘invalid SLE-PDU’	[ignore]	{peer abort (‘encoding error’)} → 1	{peer abort (‘encoding error’)} → 1
27	‘return SLE-PDU with unsolicited Invoke-ID’	[ignore]	{peer abort (‘unsolicited Invoke-ID’)} → 1	{peer abort (‘unsolicited Invoke-ID’)} → 1
28	‘protocol abort’	[ignore]	{clean up} → 1	{clean up} → 1
29	‘not authenticated SLE-PDU’	[ignore] → 1	[ignore] → 2	[ignore] → 3

Table 4-2: Event Description References

Event	Reference
'data available'	3.1.9.1.2, 3.1.9.2.2, 3.1.9.3.2
'end of data'	3.7.2.3
'end of service instance provision period'	3.11.2.2
'invalid protocol data unit'	4.1.2
'loss of frame synchronization'	3.7.2.3
'not authenticated SLE-PDU'	4.1.7
'production status change'	3.7.2.3
'release timer expired'	3.1.9.1.4, 3.1.9.2.6
'reporting-cycle timer expired'	3.8.2.6
'return SLE-PDU with unsolicited Invoke-ID'	4.1.4
'return <n> timer expired'	4.1.3
'start of service instance provision period'	1.6.1.7.15

Table 4-3: Predicate Descriptions

Predicate	Evaluates to TRUE if
"buffer empty"	There are no RCF SLE-PDUs in the transfer buffer
"buffer full"	The transfer buffer cannot accommodate the currently available annotated frame or synchronous notification
"compatible"	The version number contained in (+rcfBindReturn) is supported by the provider
"complete online"	Delivery mode is complete online
"done"	The unbind-reason parameter value in the provider-initiated BIND invocation was 'end'
"end"	All checks on the UNBIND invocation are passed and the unbind-reason parameter value is 'end'
"immediately"	All parameter checks on the RCF-SCHEDULE-STATUS-REPORT are passed and the `report-request-type` value is 'immediately'
"offline"	Delivery mode is offline
"online"	Delivery mode is timely online or complete online
"periodically"	All parameter checks on the RCF-SCHEDULE-STATUS-REPORT are passed and the `report-request-type` value is 'periodically'
"positive result"	All checks on the invocation are passed
"provider initiated"	The RCF-BIND operation is specified to be initiated by the provider for this service instance
"provision period"	Current time is inside the service instance provision period

Predicate	Evaluates to TRUE if
“retry permitted”	The diagnostic value contained in the (-rcfBindReturn) is ‘unable to comply’ or ‘other’, and the service instance provision period is still active
“timely online”	Delivery mode is timely online

Table 4-4: Boolean Flags

Flag Name	Initial Value
“bind pending”	FALSE
“congested”	FALSE
“unbind pending”	FALSE

Table 4-5: Compound Action Definitions

Name	Actions Performed
{clean up}	stop release timer stop all return timers stop reporting-cycle timer reinitialize transfer buffer reset parameter values to those specified in service package
{immediate report}	(rcfStatusReportInvocation) stop reporting-cycle timer
{insert annotated frame}	annotate the available frame with the parameters of the RCF-TRANSFER-DATA operation insert the annotated frame into the transfer buffer
{invoke bind}	(rcfBindInvocation) set “bind pending” to TRUE start return <n> timer
{invoke unbind}	(rcfUnbindInvocation) stop reporting-cycle timer set “unbind pending” to TRUE start return <n> timer
{pass buffer contents}	stop release timer submit contents of transfer buffer to underlying communications service IF successful THEN set “congested” to FALSE ELSE set “congested” to TRUE reinitialize transfer buffer using the nominal size

Name	Actions Performed
{peer abort 'xxxx'}	stop release timer stop all return timers stop reporting-cycle timer reinitialize transfer buffer (rcfPeerAbortInvocation) with diagnostic set to 'xxxx'
{periodic report}	(rcfStatusReportInvocation) set reporting-cycle timer to the `reporting-cycle` value in the most recent SCHEDULE-STATUS-REPORT invocation start reporting-cycle timer
{provider unbind}	set "unbind pending" to FALSE stop all return timers
{return timeout}	(rcfPeerAbortInvocation) with diagnostic 'return timeout' set "bind pending" to FALSE set "unbind pending" to FALSE
{start release timer}	set release timer to latency limit start release timer
{sync notify 'xxxx'}	create an RCF synchronous notification with `notification-type` set to 'xxxx' insert the notification into the transfer buffer
{transmit buffer}	stop release timer submit the contents of transfer buffer to underlying communications service until accepted by that service reinitialize transfer buffer using nominal size
{user unbind}	stop reporting-cycle timer stop all return timers (rcfUnbindReturn)

ANNEX A

DATA TYPE DEFINITIONS

(NORMATIVE)

A1 INTRODUCTION

A1.1 This annex defines the data types that are used by the RCF service. It is intended to provide a clear specification of these data types and to avoid ambiguity. It is not intended to constrain how these data types are implemented or encoded. These definitions are suitable for inclusion in any type of ASN.1-based protocol that implements the RCF service.

A1.2 The data type definitions are presented in seven ASN.1 modules.

A1.3 Subsection A2.1 contains basic types that are common with other SLE Transfer Services. As more services become specified by CCSDS, further types may be added to this module or existing types may be extended. However, that eventuality is not expected to invalidate the module in its present form because it is expected that an implementation compliant with a future extended version of this module will be interoperable with an implementation based on its present version.

A1.4 Subsection A2.2 specifies the SLE-PDUs exchanged between an SLE user and an SLE provider application in order to establish, release or abort an association. They are common among SLE transfer service types.

A1.5 Subsection A2.3 specifies SLE-PDUs related to invocations and returns that are common to SLE transfer service types.

A1.6 Subsection A2.5 specifies data types specific to the RCF service. In part, these specific types are derived from types specified in A2.1 by means of subtyping.

A1.7 Subsection A2.4 specifies the format of the Service Instance Identifiers.

A1.8 Subsection A2.6 specifies all incoming (from a provider point of view) SLE-PDUs. Where applicable, these SLE-PDUs are mapped to the generic SLE-PDUs defined in A2.2 and A2.3.

A1.9 Subsection A2.7 specifies in the same way the outgoing SLE-PDUs.

A1.10 Although subsections A2.2, A2.3, A2.6 and A2.7 define the SLE-PDUs that will be exchanged between the SLE provider and user applications, they shall not be interpreted as requiring that these SLE-PDUs shall be completely mapped to the user data field of the underlying communications protocol. For example, depending on the communications protocol(s) used, part of the SLE-PDUs may be used to determine the appropriate setting of protocol control information.

A2 RCF DATA TYPE SPECIFICATION

A2.1 SLE TRANSFER SERVICE—COMMON TYPES

```
CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

{  iso identified-organization(3)
   standards-producing-organization(112) ccsds(4)
   space-link-extension(3) sle-transfer-services(1)
   modules(1) common-modules(99) version-four(4) asn1-common-types(1)
}

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS  ConditionalTime
,        Credentials
,        DeliveryMode
,        Diagnostics
,        Duration
,        ForwardDuStatus
,        IntPosLong
,        IntPosShort
,        IntUnsignedLong
,        IntUnsignedShort
,        InvokeId
,        ParameterName
,        SlduStatusNotification
,        SpaceLinkDataUnit
,        Time
;

ConditionalTime         ::=   CHOICE
{  undefined         [0]   NULL
,  known             [1]   Time
}

-- If credentials are used, it will be necessary that
-- the internal structure of the octet string is known
-- to both parties. Since the structure will depend on
-- algorithm used, it is not specified here. However,
-- the peer entities may use ASN.1 encoding to make the
-- internal structure visible.
Credentials             ::=   CHOICE
{  unused            [0]   NULL
,  used              [1]   OCTET STRING (SIZE (8 .. 256))
}

DeliveryMode            ::=   INTEGER
{  rtnTimelyOnline               (0)
,  rtnCompleteOnline             (1)
,  rtnOffline                    (2)
,  fwdOnline                     (3)
,  fwdOffline                    (4)
}
```

```
Diagnostics             ::=   INTEGER
{  duplicateInvokeId             (100)
,  otherReason                   (127)
}

-- The Duration is expressed in microseconds
Duration                ::=   IntUnsignedLong

ForwardDuStatus         ::=   INTEGER
{  radiated                      (0)
,  expired                       (1)
,  interrupted                   (2)
,  acknowledged                  (3)   -- FSP
,  productionStarted             (4)   -- CLTU: 'radiation started'
,  productionNotStarted          (5)   -- CLTU: 'radiation not started'
,  unsupportedTransmissionMode   (6)   -- FSP
}

-- 1 to (2^32)-1
IntPosLong              ::=   INTEGER (1 .. 4294967295)

-- 1 to (2^16)-1
IntPosShort             ::=   INTEGER (1 .. 65535)

-- 0 to (2^32)-1
IntUnsignedLong         ::=   INTEGER (0 .. 4294967295)

-- 0 to (2^16)-1
IntUnsignedShort        ::=   INTEGER (0 .. 65535)

InvokeId                ::=   IntUnsignedShort
```

```
ParameterName                          ::=   INTEGER
{  acquisitionSequenceLength                 (201)
,  apidList                                  (2)
,  bitLockRequired                           (3)
,  blockingTimeoutPeriod                     (0)
,  blockingUsage                             (1)
,  bufferSize                                (4)
,  clcwGlobalVcid                            (202)
,  clcwPhysicalChannel                       (203)
,  deliveryMode                              (6)
,  directiveInvocation                       (7)
,  directiveInvocationOnline                 (108)
,  expectedDirectiveIdentification           (8)
,  expectedEventInvocationIdentification     (9)
,  expectedSlduIdentification                (10)
,  fopSlidingWindow                          (11)
,  fopState                                  (12)
,  latencyLimit                              (15)
,  mapList                                   (16)
,  mapMuxControl                             (17)
,  mapMuxScheme                              (18)
,  maximumFrameLength                        (19)
,  maximumPacketLength                       (20)
,  maximumSlduLength                         (21)
,  minimumDelayTime                          (204)
,  modulationFrequency                       (22)
,  modulationIndex                           (23)
,  notificationMode                          (205)
,  permittedControlWordTypeSet               (101)
,  permittedGvcidSet                         (24)
,  permittedTcVcidSet                        (102)
,  permittedTransmissionMode                 (107)
,  permittedUpdateModeSet                    (103)
,  plop1IdleSequenceLength                   (206)
,  plopInEffect                              (25)
,  protocolAbortMode                         (207)
,  reportingCycle                            (26)
,  requestedControlWordType                  (104)
,  requestedFrameQuality                     (27)
,  requestedGvcid                            (28)
,  requestedTcVcid                           (105)
,  requestedUpdateMode                       (106)
,  returnTimeoutPeriod                       (29)
,  rfAvailable                               (30)
,  rfAvailableRequired                       (31)
,  segmentHeader                             (32)
,  subcarrierToBitRateRatio                  (34)
,  timeoutType                               (35)
,  timerInitial                              (36)
,  transmissionLimit                         (37)
,  transmitterFrameSequenceNumber            (38)
,  vcMuxControl                              (39)
,  vcMuxScheme                               (40)
,  virtualChannel                            (41)
}
```

```
SlduStatusNotification  ::=   INTEGER
{  produceNotification           (0)
,  doNotProduceNotification      (1)
}

SpaceLinkDataUnit       ::=   OCTET STRING (SIZE (1 .. 65536))

Time                    ::=   CHOICE
{  ccsdsFormat       [0]   TimeCCSDS
,  ccsdsPicoFormat   [1]   TimeCCSDSpico
}

TimeCCSDS               ::=   OCTET STRING (SIZE(8))
-- P-field is implicit (not present, defaulted to 41 hex
-- T-field:
-- 2 octets: number of days since 1958/01/01 00:00:00
-- 4 octets: number of milliseconds of the day
-- 2 octets: number of microseconds of the millisecond
--        (set to 0 if not used)
-- This definition reflects exactly the format of the CCSDS defined
-- time tag as used in spacelink data units (see Time Code Formats.
-- Recommendation for Space Data System Standards, CCSDS 301.0-B-3.
-- Blue Book.  Issue 3.  Washington, D.C.: CCSDS, January 2002).

TimeCCSDSpico           ::=   OCTET STRING (SIZE(10))
-- P-field is implicit (not present, defaulted to 42 hex
-- T-field:
-- 2 octets: number of days since 1958/01/01 00:00:00
-- 4 octets: number of milliseconds of the day
-- 4 octets: number of picoseconds of the millisecond
--        (set to 0 if not used)
-- This definition reflects exactly the format of the CCSDS defined
-- time tag as used in spacelink data units (see Time Code Formats.
-- Recommendation for Space Data System Standards, CCSDS 301.0-B-3.
-- Blue Book.  Issue 3.  Washington, D.C.: CCSDS, January 2002).

END
```

A2.2 SLE TRANSFER SERVICE—BIND TYPES

```
CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES

{  iso identified-organization(3)
   standards-producing-organization(112) ccsds(4)
   space-link-extension(3) sle-transfer-services(1)
   modules(1) common-modules(99) version-four(4) asn1-bind-types(2)
}

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS  SleBindInvocation
,        SleBindReturn
,        SlePeerAbort
,        SleUnbindInvocation
,        SleUnbindReturn
;

IMPORTS  Credentials
,        IntPosShort
   FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

         ServiceInstanceIdentifier
   FROM CCSDS-SLE-TRANSFER-SERVICE-SERVICE-INSTANCE-ID
;

-- ============================================================
-- The first part of the module definition contains the SLE-PDUs
-- ============================================================

SleBindInvocation          ::=   SEQUENCE
{  invokerCredentials         Credentials
,  initiatorIdentifier        AuthorityIdentifier
,  responderPortIdentifier    PortId
,  serviceType                ApplicationIdentifier
,  versionNumber              VersionNumber
,  serviceInstanceIdentifier  ServiceInstanceIdentifier
}

SleBindReturn           ::=   SEQUENCE
{  performerCredentials    Credentials
,  responderIdentifier     AuthorityIdentifier
,  result                  CHOICE
   {  positive          [0]   VersionNumber
   ,  negative          [1]   BindDiagnostic
   }
}

SlePeerAbort            ::=   PeerAbortDiagnostic

SleUnbindInvocation     ::=   SEQUENCE
{  invokerCredentials      Credentials
,  unbindReason            UnbindReason
}
```

```
SleUnbindReturn         ::=   SEQUENCE
{  responderCredentials    Credentials
,  result                  CHOICE
   {  positive          [0]   NULL
   }
}

-- ==============================================================
-- The second part of the module definition contains the types
-- used by the SLE-PDUs declared in the first part.
-- ==============================================================

ApplicationIdentifier      ::=   INTEGER
{  rtnAllFrames               (0)
,  rtnInsert                  (1)
,  rtnChFrames                (2)
-- rtnChFrames includes rtnMcFrames and rtnVcFrames
,  rtnChFsh                   (3)
-- rtnChFsh includes rtnMcFsh and rtnVcFsh
,  rtnChOcf                   (4)
-- rtnChOcf includes rtnMcOcf and rtnVcOcf
,  rtnBitstr                  (5)   -- AOS
,  rtnSpacePkt                (6)
,  fwdAosSpacePkt             (7)
,  fwdAosVca                  (8)
,  fwdBitstr                  (9)
,  fwdProtoVcdu               (10)
,  fwdInsert                  (11)
,  fwdCVcdu                   (12)
,  fwdTcSpacePkt              (13)  -- conventional telecommand
,  fwdTcVca                   (14)  -- conventional telecommand
,  fwdTcFrame                 (15)
,  fwdCltu                    (16)
}

AuthorityIdentifier     ::=   IdentifierString (SIZE (3 .. 16))

BindDiagnostic                 ::=   INTEGER
{  accessDenied                   (0)
,  serviceTypeNotSupported        (1)
,  versionNotSupported            (2)
,  noSuchServiceInstance          (3)
,  alreadyBound                   (4)
,  siNotAccessibleToThisInitiator (5)
,  inconsistentServiceType        (6)
,  invalidTime                    (7)
,  outOfService                   (8)
,  otherReason                    (127)
}

IdentifierString        ::=   VisibleString (FROM (ALL EXCEPT " "))

LogicalPortName         ::=   IdentifierString (SIZE (1 .. 128))
```

```
PeerAbortDiagnostic           ::=   INTEGER
{  accessDenied                  (0)
,  unexpectedResponderId         (1)
,  operationalRequirement        (2)
,  protocolError                 (3)
,  communicationsFailure         (4)
,  encodingError                 (5)
,  returnTimeout                 (6)
,  endOfServiceProvisionPeriod   (7)
,  unsolicitedInvokeId           (8)
,  otherReason                   (127)
-- The range of this INTEGER shall be (0 .. 255).
-- (128 .. 255) is reserved for diagnostic codes
-- that are specific to the communications technology used.
}

PortId               ::=   LogicalPortName

UnbindReason         ::=   INTEGER
{  end                   (0)
,  suspend               (1)
,  versionNotSupported   (2)
,  other                 (127)
}

VersionNumber        ::=   IntPosShort

END
```

A2.3 SLE TRANSFER SERVICE—COMMON PDUS

```
CCSDS-SLE-TRANSFER-SERVICE-COMMON-PDUS

{  iso identified-organization(3)
   standards-producing-organization(112) ccsds(4)
   space-link-extension(3) sle-transfer-services(1)
   modules(1) common-modules(99) version-four(4) asn1-common-pdu(3)
}

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS  ReportingCycle
,        SleAcknowledgement
,        SleScheduleStatusReportInvocation
,        SleScheduleStatusReportReturn
,        SleStopInvocation
;

IMPORTS  Credentials
,        Diagnostics
,        InvokeId
   FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
;

-- ============================================================
-- The first part of the module definition contains the SLE-PDUs
-- ============================================================

SleAcknowledgement      ::=   SEQUENCE
{  credentials             Credentials
,  invokeId                InvokeId
,  result                  CHOICE
   {  positiveResult    [0]   NULL
   ,  negativeResult    [1]   Diagnostics
   }
}

SleScheduleStatusReportInvocation   ::=   SEQUENCE
{  invokerCredentials                  Credentials
,  invokeId                            InvokeId
,  reportRequestType                   ReportRequestType
}

SleScheduleStatusReportReturn ::=   SEQUENCE
{  performerCredentials          Credentials
,  invokeId                      InvokeId
,  result                        CHOICE
   {  positiveResult          [0]   NULL
   ,  negativeResult          [1]   DiagnosticScheduleStatusReport
   }
}
```

```
SleStopInvocation       ::=   SEQUENCE
{  invokerCredentials      Credentials
,  invokeId                InvokeId
}

-- ============================================================
-- The second part of the module definition contains the types
-- used by the SLE-PDUs declared in the first part.
-- ============================================================

DiagnosticScheduleStatusReport   ::=   CHOICE
{  common                     [0]   Diagnostics
,  specific                   [1]   INTEGER
   {  notSupportedInThisDeliveryMode   (0)   -- not used for
                                             -- version 1 forward transfer
                                             -- services
   ,     alreadyStopped                (1)
   ,     invalidReportingCycle         (2)
   }
}

-- The cycle duration is expressed in seconds
ReportingCycle          ::=   INTEGER (2 .. 600)

ReportRequestType       ::=   CHOICE
{  immediately       [0]   NULL
,  periodically      [1]   ReportingCycle
,  stop              [2]   NULL
}

END
```

A2.4 SLE TRANSFER SERVICE— SERVICE-INSTANCE-ID

```
CCSDS-SLE-TRANSFER-SERVICE-SERVICE-INSTANCE-ID

{  iso identified-organization (3) standards-producing-organization(112)
   ccsds(4) space-link-extension(3) sle-transfer-services(1)
   modules(1) common-modules(99) version-four(4)
   asn1-service-id-type(4)
}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

EXPORTS ServiceInstanceIdentifier
;

-- Object Identifiers definition
sagr     OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 52}
spack    OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 53}
fsl-fg   OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 14}
rsl-fg   OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 38}
cltu     OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 7}
fsp      OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 10}
raf      OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 22}
rcf      OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 46}
rcfsh    OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 44}
rocf     OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 49}
rsp      OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 40}
tcf      OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 12}
tcva     OBJECT IDENTIFIER    ::=   {iso 3 112 4 3 1 2 16}
```

```
-- ++++++++++++++++++++++++++++++++++
-- ++++++++++++++++++++++++++++++++++
-- The Service Instance Identifier is a concatenation of Attributes.
-- Each Attribute is composed of a name and a value.
-- Note that the Attribute names are not specified by the ASN.1 proper,
-- but by means of comments in this module.
-- The following Attributes identify the Service Instance:
-- Service agreement
--    the name of which is:   "sagr",
--    the value of which is to be agreed between the user and the provider;
-- Service package
--    the name of which is:   "spack",
--    the value of which is to be agreed between the user and the provider;
-- Forward/Return Service Functional Group
--    the name of which is:
--       "fsl-fg" for the forward services,
--    or:
--       "rsl-fg" for the return services,
--    the value of which is to be agreed between the user and the provider;
-- Service Name identifier,
--       the name of which is:   "cltu"
--       the value of which is:  "cltu" plus an instance number;
--    or:
--       the name of which is:   "fsp",
--       the value of which is:  "fsp" plus an instance number;
--    or:
--       the name of which is:   "tcva",
--       the value of which is:  "tcva" plus an instance number;
--    or:
--       the name of which is:   "tcf",
--       the value of which is:  "tcf" plus an instance number;
--    or:
--       the name of which is:   "raf",
--       the value ofwhich is:   "onlc" or "onlt" or "offl" plus an
--       instance number;
--    or:
--       the name of which is:   "rcf",
--       the value of which is:  "onlc" or "onlt" or "offl" plus an
--       instance number;
--    or:
--       the name of which is:   "rocf",
--       the value of which is:  "onlc" or "onlt" or "offl" plus an
--       instance number;
--    or:
--       the name of which is:   "rcfsh",
--       the value of which is:  "onlc" or "onlt" or "offl" plus an
--       instance number;
--    or:
--       the name of which is:   "rsp",
--       the value of which is:  "onlc" or "onlt" or "offl" plus an
--       instance number.
--
-- To obtain the string correponding to the instance identifier, one must
-- concatenate the Attributes values and Names as follows:
-- ‘Atribute Name"="Attrbute value’.
-- Two Attributes must be separated by a ".".
--
-- Example: ‘sagr=xyz.spack=abcdef.rsl-fg=gfjdy.raf=onlc2’
-- Note that the quotes are not part of the string.
```

```
ATTRIBUTE            ::=   CLASS
{  &id                   OBJECT IDENTIFIER UNIQUE
}
WITH SYNTAX { ID  &id }

GeneralAttributes ATTRIBUTE   ::=
{  serviceAgreement
|  servicePackage
|  forwardService
|  returnService
}

ServiceInstanceAttribute   ::=      SET SIZE(1) OF SEQUENCE
{  identifier                       ATTRIBUTE.&id
                                    ({ServiceInstanceAttributes})
,  siAttributeValue                 VisibleString (SIZE (1 .. 256))
}

ServiceInstanceAttributes ATTRIBUTE ::=
{  GeneralAttributes
|  ServiceNames
}

ServiceInstanceIdentifier  ::=   SEQUENCE OF ServiceInstanceAttribute

ServiceNames ATTRIBUTE  ::=
{  rafService
|  rcfService
|  rcfshService
|  rocfService
|  rspService
|  cltuService
|  fspService
|  tcfService
|  tcvaService
}

-- ++++++++++++++++++++++++++++++++++++
-- CLTU Service Definition
-- The cltu Service Id starts with "cltu" and is
-- followed by the instance number.
cltuService ATTRIBUTE      ::=   {  ID cltu  }

-- ++++++++++++++++++++++++++++++++++++
-- Forward Service Definition
forwardService ATTRIBUTE   ::=   {  ID fsl-fg   }

-- ++++++++++++++++++++++++++++++++++++
-- FSP Service Definition
-- The fsp Service Id starts with "fsp" and is
-- followed by the instance number.
fspService ATTRIBUTE       ::=   {  ID fsp   }

-- ++++++++++++++++++++++++++++++++++++
-- RAF Service Definition
-- The raf Service Id starts with "onlc" or "onlt" or "offl" and is
-- followed by the instance number.
rafService ATTRIBUTE       ::=   {  ID raf   }
```

```
-- ++++++++++++++++++++++++++++++++++++
-- RCF Service Definition
-- The rcf Service Id starts with "onlc" or "onlt" or "offl" and is
-- followed by the instance number.
rcfService ATTRIBUTE       ::=   {  ID rcf   }

-- ++++++++++++++++++++++++++++++++++++
-- RCFSH Service Definition
-- The rcfsh Service Id starts with "onlc" or "onlt" or "offl" and is
-- followed by the instance number.
rcfshService ATTRIBUTE     ::=   {  ID rcfsh }

-- ++++++++++++++++++++++++++++++++++++
-- Return Service Definition
returnService ATTRIBUTE    ::=   {  ID rsl-fg   }

-- ++++++++++++++++++++++++++++++++++++
-- ROCF Service Definition
-- The rocf Service Id starts with "onlc" or "onlt" or "offl" and is
-- followed by the instance number.
rocfService ATTRIBUTE      ::=   {  ID rocf  }

-- ++++++++++++++++++++++++++++++++++++
-- RSP Service Definition
-- The rocf Service Id starts with "onlc" or "onlt" or "offl" and is
-- followed by the instance number.
rspServiceId ATTRIBUTE     ::=   {  ID rsp   }

-- ++++++++++++++++++++++++++++++++++++
-- Service Agreement Definition
serviceAgreement ATTRIBUTE ::=   {  ID sagr  }

-- ++++++++++++++++++++++++++++++++++++
-- Service Package Definition
servicePackage ATTRIBUTE   ::=   {  ID spack }

-- ++++++++++++++++++++++++++++++++++++
-- TCF Service Definition
-- The tcf Service Id starts with "tcf" and is
-- followed by the instance number.
tcfServiceId ATTRIBUTE     ::=   {  ID tcf   }

-- ++++++++++++++++++++++++++++++++++++
-- TCVA Service Definition
-- The tcva Service Id starts with "tcva" and is
-- followed by the instance number.
tcvaServiceId ATTRIBUTE    ::=   {  ID tcva  }

END
```

A2.5 SLE TRANSFER SERVICE—RCF STRUCTURES

```
CCSDS-SLE-TRANSFER-SERVICE-RCF-STRUCTURES
{iso identified-organization(3) standards-producing-organization(112)
    ccsds(4) space-link-extension(3) sle-transfer-services(1)
    modules(1) return-channel-frames-service(13) version-four(4)
    asn1-rcf-structures(1)}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

EXPORTS     AntennaId
,           CarrierLockStatus
,           DiagnosticRcfGet
,           DiagnosticRcfStart
,           FrameSyncLockStatus
,           GvcId
,           LockStatus
,           Notification
,           RcfGetParameter
,           RcfParameterName
,           RcfProductionStatus
,           SymbolLockStatus
;

IMPORTS     DeliveryMode
,           Diagnostics
,           IntPosShort
,           ParameterName
,           Time
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
            ReportingCycle
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-PDUS
;

AntennaId                   ::= CHOICE
{   globalForm                      [0] OBJECT IDENTIFIER
,   localForm                       [1] OCTET STRING (SIZE (1 .. 16))
}

CarrierLockStatus    ::=   LockStatus
(  inLock
|  outOfLock
|  unknown
)


CurrentReportingCycle       ::= CHOICE
{   periodicReportingOff            [0] NULL
,   periodicReportingOn             [1] ReportingCycle
}

DiagnosticRcfGet            ::= CHOICE
{   common                          [0] Diagnostics
,   specific                        [1] INTEGER
    {   unknownParameter                    (0)
```

```
    }
}

DiagnosticRcfStart          ::= CHOICE
{   common                          [0] Diagnostics
,   specific                        [1] INTEGER
    {   outOfService                        (0)
    ,   unableToComply                      (1)
    ,   invalidStartTime                    (2)
    ,   invalidStopTime                     (3)
    ,   missingTimeValue                    (4)
    ,   invalidGvcId                        (5)
    }
}

FrameSyncLockStatus     ::=   LockStatus
(  inLock
|  outOfLock
|  unknown
)

GvcId                       ::= SEQUENCE
{   spacecraftId                   INTEGER (0 .. 1023)
,   versionNumber                  INTEGER (0 .. 3)
,   vcId                           CHOICE
    {  masterChannel                [0]   NULL
 , virtualChannel                   [1]   VcId
    }
}
-- Notes:
-- The permissible range of the spacecraftId depends on the versionNumber.
-- PTM and AOS frames are supported.
-- In case of AOS the permissible range is 0 to 255 (8 bits).
-- In case of PTM the permissible range is 0 to 1023 (10 bits).


GvcIdSet                    ::= SEQUENCE OF MasterChannelComposition

LockStatus                  ::= INTEGER
{   inLock                          (0)
,   outOfLock                       (1)
,   notInUse                        (2)
,   unknown                         (3)
}

LockStatusReport            ::= SEQUENCE
{   time                            Time
,   carrierLockStatus               CarrierLockStatus
,   subcarrierLockStatus            LockStatus
,   symbolSyncLockStatus            SymbolLockStatus
}

MasterChannelComposition    ::= SEQUENCE
{   spacecraftId                   INTEGER (0 .. 1023)
,   versionNumber                  INTEGER (0 .. 3)
,   mcOrVcList                     CHOICE
 { masterChannel                  [0] NULL
 , vcList                         [1] SEQUENCE OF VcId
 }
}
```

```
RequestedGvcId             ::=   CHOICE
{  gvcid                [0]   GvcId
,  undefined            [1]   NULL
}


Notification                ::= CHOICE
{   lossFrameSync                   [0] LockStatusReport
,   productionStatusChange          [1] RcfProductionStatus
,   excessiveDataBacklog            [2] NULL
,   endOfData                       [3] NULL
}

RcfDeliveryMode             ::= DeliveryMode
(   rtnTimelyOnline
|   rtnCompleteOnline
|   rtnOffline
)

RcfGetParameter     ::= CHOICE
{   parBufferSize           [0] SEQUENCE
    {   parameterName               ParameterName (bufferSize)
    ,   parameterValue              IntPosShort
                                    -- bufferSize in number of invocations
                                    -- that can be held in buffer
    }
,   parDeliveryMode         [1] SEQUENCE
    {   parameterName               ParameterName (deliveryMode)
    ,   parameterValue              RcfDeliveryMode
    }
,   parLatencyLimit         [2] SEQUENCE
    {   parameterName               ParameterName (latencyLimit)
    ,   parameterValue              CHOICE
        {   online                [0] IntPosShort
                                          -- latencyLimit in seconds
        ,   offline               [1] NULL
        }
    }
,   parPermittedGvcidSet    [3] SEQUENCE
    { parameterName                  ParameterName (permittedGvcidSet)
    , parameterValue                 GvcIdSet
    }
,   parReportingCycle       [4] SEQUENCE
  { parameterName                  ParameterName (reportingCycle)
  , parameterValue                 CurrentReportingCycle
  }
,   parReqGvcId             [5] SEQUENCE
    { parameterName                  ParameterName (requestedGvcid)
    , parameterValue                 RequestedGvcId
    }
,   parReturnTimeout        [6] SEQUENCE
    {   parameterName               ParameterName (returnTimeoutPeriod)
    ,   parameterValue              TimeoutPeriod
    }
}

RcfParameterName            ::= ParameterName
(   bufferSize
|   deliveryMode
```

```
|   latencyLimit
|   permittedGvcidSet
|   reportingCycle
|   requestedGvcid
|   returnTimeoutPeriod
)

RcfProductionStatus         ::= INTEGER
{   running                         (0)
,   interrupted                     (1)
,   halted                          (2)
}

SymbolLockStatus        ::=   LockStatus
(  inLock
|  outOfLock
|  unknown
)

TimeoutPeriod               ::= INTEGER (1 .. 600)
                                -- measured in seconds

VcId                        ::= INTEGER (0 .. 63)
-- The actual permissible range depends on version number of the associated
-- GVCID
--
END
```

A2.6 SLE TRANSFER SERVICE—RCF INCOMING PDUS

```
CCSDS-SLE-TRANSFER-SERVICE-RCF-INCOMING-PDUS
{iso identified-organization(3) standards-producing-organization(112)
    ccsds(4) space-link-extension(3) sle-transfer-services(1)
    modules(1) return-channel-frames-service(13) version-four(4)
    asn1-rcf-incoming-pdu(2)}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

IMPORTS     ConditionalTime
,           Credentials
,           InvokeId
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
            SleScheduleStatusReportInvocation
,           SleStopInvocation
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-PDUS
            GvcId
,           RcfParameterName
    FROM CCSDS-SLE-TRANSFER-SERVICE-RCF-STRUCTURES
            SleBindInvocation
,           SleBindReturn
,           SlePeerAbort
,           SleUnbindInvocation
,           SleUnbindReturn
    FROM CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES
;

-- ============================================================
-- The first part of the module definition contains the RCF type
-- that contains all the possible PDUs the provider may receive.
-- ============================================================

RcfUserToProviderPdu            ::= CHOICE
{ rcfBindInvocation                 [100] SleBindInvocation
, rcfBindReturn                     [101] SleBindReturn
, rcfUnbindInvocation               [102] SleUnbindInvocation
, rcfUnbindReturn                   [103] SleUnbindReturn
, rcfStartInvocation                [0]   RcfStartInvocation
, rcfStopInvocation                 [2]   SleStopInvocation
, rcfScheduleStatusReportInvocation [4]   SleScheduleStatusReportInvocation
, rcfGetParameterInvocation         [6]   RcfGetParameterInvocation
, rcfPeerAbortInvocation            [104] SlePeerAbort
}

-- ============================================================
-- The second part of the module definition contains the types
-- used by the RCF-PDUs declared in the first part.
-- ============================================================

RcfGetParameterInvocation       ::= SEQUENCE
{   invokerCredentials                  Credentials
,   invokeId                            InvokeId
,   rcfParameter                        RcfParameterName
}

RcfStartInvocation              ::= SEQUENCE
```

```
{   invokerCredentials              Credentials
,   invokeId                        InvokeId
,   startTime                       ConditionalTime
,   stopTime                        ConditionalTime
,   requestedGvcId                          GvcId
}

END
```

A2.7 SLE TRANSFER SERVICE—RCF OUTGOING PDUS

```
CCSDS-SLE-TRANSFER-SERVICE-RCF-OUTGOING-PDUS
{iso identified-organization(3) standards-producing-organization(112)
    ccsds(4) space-link-extension(3) sle-transfer-services(1)
    modules(1) return-channel-frames-service(13) version-four(4)
    asn1-outgoing-pdu(3)}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

IMPORTS     Credentials
,           IntUnsignedLong
,           InvokeId
,           SpaceLinkDataUnit
,           Time
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
            SleAcknowledgement
,           SleScheduleStatusReportReturn
    FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-PDUS
            AntennaId
,           CarrierLockStatus
,           DiagnosticRcfGet
,           DiagnosticRcfStart
,           FrameSyncLockStatus
,           LockStatus
,           Notification
,           RcfGetParameter
,           RcfProductionStatus
,           SymbolLockStatus
    FROM CCSDS-SLE-TRANSFER-SERVICE-RCF-STRUCTURES
            SleBindInvocation
,           SleBindReturn
,           SlePeerAbort
,           SleUnbindInvocation
,           SleUnbindReturn
    FROM CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES
;

-- ============================================================
-- The first part of the module definition contains the RCF type
-- that contains all the possible PDUs the provider may send.
-- ============================================================

RcfProviderToUserPdu        ::= CHOICE
{ rcfBindInvocation                 [100]   SleBindInvocation
, rcfBindReturn                     [101]   SleBindReturn
, rcfUnbindInvocation               [102]   SleUnbindInvocation
, rcfUnbindReturn                   [103]   SleUnbindReturn
, rcfStartReturn                    [1]     RcfStartReturn
, rcfStopReturn                     [3]     SleAcknowledgement
, rcfTransferBuffer                 [8]     RcfTransferBuffer
, rcfScheduleStatusReportReturn     [5]     SleScheduleStatusReportReturn
, rcfStatusReportInvocation         [9]     RcfStatusReportInvocation
, rcfGetParameterReturn             [7]     RcfGetParameterReturn
, rcfPeerAbortInvocation            [104]   SlePeerAbort
```

```
}

-- ============================================================
-- The second part of the module definition contains the types
-- used by the RCF-PDUs declared in the first part.
-- ============================================================

FrameOrNotification             ::= CHOICE
{   annotatedFrame                      [0] RcfTransferDataInvocation
,   syncNotification                    [1] RcfSyncNotifyInvocation
}

RcfGetParameterReturn           ::= SEQUENCE
{   performerCredentials                Credentials
,   invokeId                            InvokeId
,   result                              CHOICE
    {   positiveResult                      [0] RcfGetParameter
    ,   negativeResult                      [1] DiagnosticRcfGet
    }
}

RcfStartReturn                  ::= SEQUENCE
{   performerCredentials                Credentials
,   invokeId                            InvokeId
,   result                              CHOICE
    {   positiveResult                      [0] NULL
    ,   negativeResult                      [1] DiagnosticRcfStart
    }
}

RcfStatusReportInvocation       ::= SEQUENCE
{   invokerCredentials                  Credentials
,   deliveredFrameNumber                IntUnsignedLong
,   frameSyncLockStatus                 FrameSyncLockStatus
,   symbolSyncLockStatus                SymbolLockStatus
,   subcarrierLockStatus                LockStatus
,   carrierLockStatus                   CarrierLockStatus
,   productionStatus                    RcfProductionStatus
}

RcfSyncNotifyInvocation         ::= SEQUENCE
{   invokerCredentials                  Credentials
,   notification                        Notification
}

RcfTransferBuffer               ::= SEQUENCE OF FrameOrNotification

RcfTransferDataInvocation   ::= SEQUENCE
{   invokerCredentials              Credentials
,   earthReceiveTime                Time
,   antennaId                       AntennaId
,   dataLinkContinuity              INTEGER (-1 .. 16777215)
,   privateAnnotation               CHOICE
    {   null                            [0] NULL
    ,   notNull                         [1] OCTET STRING (SIZE (1 .. 128))
    }
,   data                            SpaceLinkDataUnit
}

END
```

ANNEX B

CONFORMANCE MATRIX

(NORMATIVE)

This annex provides the Conformance Matrix for implementations of Space Link Extension (SLE) Return Channel Frames (RCF) Service systems. An SLE RCF Service system will be considered to conform to this Recommended Standard if the mandatory elements identified in tables B-1 and B-2 are implemented as described in this Recommended Standard.

Table B-1: Conformance Matrix for RCF Service (Operations)

RCF Service Operation	Optional/Mandatory

RCF-BIND
RCF-UNBIND
	These operations are mandatory. All parameters are also mandatory with the exception of:
	`invoker-credentials`
	`performer-credentials`

RCF-START
RCF-STOP
RCF-TRANSFER-DATA
RCF-SYNC-NOTIFY
RCF-SCHEDULE-STATUS-REPORT
RCF-STATUS-REPORT
RCF-GET-PARAMETER
	These operations are mandatory. All parameters are also mandatory with the exception of:
	`invoker-credentials`
	`performer-credentials`

RCF-PEER-ABORT	This operation is mandatory. All parameters are also mandatory with the exception of: `invoker-credentials`

Table B-2: Conformance Matrix for RCF Service (Other Requirements)

Requirement	Optional / Mandatory
State transition (see 4.2)	Mandatory.
ASN.1 protocol specification (see annex A)	All data types are mandatory. Implementations may vary as described in the annex.
Transfer buffer size (see 3.1.9.1)	Mandatory minimum size shall be sufficient to contain 200 RCF-TRANSFER-DATA invocations (or the equivalent thereof).
Online buffer size (see 3.1.9.2)	Mandatory minimum size shall be sufficient to contain 100,000 RCF-TRANSFER-DATA invocations (or the equivalent thereof).
Offline buffer size (see 3.1.9.3)	Mandatory minimum size shall be sufficient to contain 1,000,000 RCF-TRANSFER-DATA invocations (or the equivalent thereof).

ANNEX C

INDEX TO DEFINITIONS

(INFORMATIVE)

This annex lists terms used in this Recommended Standard and, for each term, provides a reference to the definition of that term.

Term	Reference
(data) type	reference [7]
(data) value	reference [7]
abstract binding	reference [1]
abstract object	reference [1]
abstract port	reference [1]
abstract service	reference [1]
abstract syntax	reference [6]
Abstract Syntax Notation One (ASN.1)	reference [7]
active (state)	subsection 2.6.4.2
AOS Transfer Frame	reference [4]
application entity	reference [6]
application layer	reference [6]
association	subsection 1.6.1.7.1
attached sync marker	reference [2]
communications service	subsection 1.6.1.7.2
complete (online delivery mode)	subsections 2.3, 3.1.9.2
confirmed operation	subsection 1.6.1.7.3
delivery criteria	subsection 1.6.1.7.4
error control field	subsection 1.6.1.7.5
flow control	reference [6]
Frame Error Control Field (FECF)	references [3] and [4]
frame version number	subsection 1.6.1.7.6
initiator	subsection 1.6.1.7.7
invocation	subsection 1.6.1.7.8
invoker	reference [1]
latency limit	subsections 2.6.4.6.2, 3.1.9.1
master channel	subsection 1.6.1.7.9

Term	Reference
Mission Data Operation System (MDOS)	reference [1]
Mission User Entity (MUE)	reference [1]
object identifier	reference [7]
offline delivery mode	reference [1]
offline frame buffer	subsections 2.6.4.6.3, 3.1.9.3
online delivery mode	reference [1]
online frame buffer	subsections 2.6.4.6.3, 3.1.9.2
Open Systems Interconnection (OSI)	reference [6]
operation	reference [1]
parameter	subsection 1.6.1.7.9
performance	subsection 1.6.1.7.11
performer	reference [1]
physical channel	reference [1]
port identifier	subsection 1.6.1.7.12
provider-initiated	subsections 2.3, 3.2.1
ready (state)	subsection 2.6.4.2
real system	reference [6]
Reed-Solomon check symbols	reference [2]
Reed-Solomon code	reference [2]
release timer	subsections 3.1.9.1.4, 3.1.9.2.6
responder	subsection 1.6.1.7.13
return (of an operation)	subsection 1.6.1.7.14
Return All Frames channel	reference [1]
Return All Frames service	reference [1]
Return Channel Frames (RCF) service	subsection 1.1
Return Master Channel Frames (RMCF) service	reference [1]
Return Virtual Channel Frames (RVCF) service	reference [1]
return data	reference [1]
Service Access Point (SAP)	reference [6]
service agreement	reference [1]
service instance provision period	subsection 1.6.1.7.15
service provider (provider)	reference [1]
service user (user)	reference [1]
SLE Complex	reference [1]
SLE Complex Management	reference [1]

Term	Reference
SLE data channel	reference [1]
SLE Functional Group (SLE-FG)	reference [1]
SLE Protocol Data Unit (SLE-PDU)	reference [1]
SLE Service Data Unit (SLE-SDU)	reference [1]
SLE service package	reference [1]
SLE transfer service instance	reference [1]
SLE transfer service production	reference [1]
SLE transfer service provision	reference [1]
SLE Utilization Management	reference [1]
spacecraft identifier (SCID)	subsection 1.6.1.7.16
space link	reference [1]
space link data channel	reference [1]
Space Link Data Unit (SL-DU)	reference [1]
space link session	reference [1]
telemetry frame	subsection 1.6.1.7.16
timely (online delivery mode)	subsections 2.3, 3.1.9.1
TM Transfer Frame	reference [3]
transfer buffer	subsections 2.6.4.6.2, 3.1.9
Transfer Frame Version Number	subsection 1.6.1.7.18
unbound (state)	subsection 2.6.4.2
unconfirmed operation	subsection 1.6.1.7.19
user-initiated	subsections 2.3, 3.2.1
virtual channel	subsection 1.6.1.7.20
virtual channel identifier (VCID)	subsection 1.6.1.7.21

ANNEX D

ACRONYMS

(INFORMATIVE)

This annex lists the acronyms used in this Recommended Standard.

ASN.1	Abstract Syntax Notation One
C	Conditional
CCSDS	Consultative Committee for Space Data Systems
CDS	CCSDS Day Segmented (time code)
CM-P	Complex Management Port
CRC	Cyclic Redundant Code
ERT	Earth Receive Time
FECF	Frame Error Control Field
FG	Functional Group
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
M	Mandatory
MDOS	Mission Data Operation System
MUE	Mission User Entity
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RAF	Return All Frames

RCF	Return Channel Frames
RFP-FG	Return Frame Processing Functional Group
SAP	Service Access Point
SCID	Spacecraft Identifier
SDU	Service Data Unit
SL-DU	Space Link Data Unit
SLE	Space Link Extension
SLE-FG	SLE Functional Group
SLE-PDU	SLE Protocol Data Unit
SLE-SDU	SLE Service Data Unit
SL-DU	Space Link Data Unit
TCP	Transmission Control Protocol
TFVN	Transfer Frame Version Number
TM	Telemetry
T-P	Transfer Provider Port
T-U	Transfer User Port
UTC	Coordinated Universal Time
VC	Virtual Channel
VCID	Virtual Channel Identifier
WAN	Wide Area Network

ANNEX E

INFORMATIVE REFERENCES

(INFORMATIVE)

[E1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-9. Yellow Book. Issue 9. Washington, D.C.: CCSDS, November 2003.

[E2] *Cross Support Concept — Part 1: Space Link Extension Services*. Report Concerning Space Data System Standards, CCSDS 910.3-G-3. Green Book. Issue 3. Washington, D.C.: CCSDS, March 2006.

[E3] *Telemetry Channel Coding*. Recommendation for Space Data System Standards, CCSDS 101.0-B-6-S. Historical Recommendation. Issue 6-S. Washington, D.C.: CCSDS, (October 2002) August 2005.

[E4] *Packet Telemetry*. Recommendation for Space Data System Standards, CCSDS 102.0-B-5-S. Historical Recommendation. Issue 5-S. Washington, D.C.: CCSDS, (November 2000) August 2005.

[E5] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data System Standards, CCSDS 701.0-B-3-S. Historical Recommendation. Issue 3-S. Washington, D.C.: CCSDS, (June 2001) August 2005.

[E6] *Space Link Extension—Internet Protocol for Transfer Services*. Recommendation for Space Data System Standards, CCSDS 913.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2008.