The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE— TRACKING DATA SERVICE

RECOMMENDED STANDARD

CCSDS 922.2-B-1

BLUE BOOK

May 2020

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE— TRACKING DATA SERVICE

RECOMMENDED STANDARD

CCSDS 922.2-B-1

BLUE BOOK

May 2020

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	May 2020
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 *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

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:

- o 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.
- o 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.
- o 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 *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). 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 sent to the CCSDS Secretariat at the email 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.
- 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 (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- 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.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 922.2-B-1	Cross Support Transfer Service—Tracking Data Service, Recommended Standard, Issue 1	May 2020	Original issue

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE OF THIS RECOMMENDED STANDARD

This Recommended Standard defines the Tracking Data Cross Support Transfer Service (TD-CSTS). The TD-CSTS is a transfer service that allows a spaceflight mission to receive periodic measurements of certain tracking data as soon as they are generated by a Cross Support Complex or anytime thereafter. The types of tracking data accessible by means of the TD-CSTS are identified in 2.2.2. Examples of tracking data include, but are not limited to Doppler, range and antenna angles. The TD-CSTS delivers the tracking data as a periodic sequence of data segments formatted in accordance with components of a CCSDS *Tracking Data Message* (TDM), as specified in the TDM Recommended Standard (reference [3]). The TD-CSTS is constructed using procedures and operations defined in the *Cross Support Transfer Service Specification Framework* (reference [1]).

1.2 SCOPE

1.2.1 SCOPE OF THE TD-CSTS

The TD-CSTS transfers tracking data associated with the signal-related and angle-related TDM keyword categories as specified in reference [3].

The TD-CSTS does not deliver data using the time-related keywords, media-related keywords, or meteorological-related keywords of the TDM, which are better suited to a file-based transfer than to a real-time transfer.

The TD-CSTS does not deliver VLBI/Delta-Differential One-way Ranging (DOR)-related data, which typically comprises a single small TDM file (a page or two in length) that is created as a result of substantial post-processing of the raw tracking data.

1.2.2 SCOPE OF THIS RECOMMENDED STANDARD

This Recommended Standard defines the TD-CSTS in terms of:

a) the CSTS Framework procedures (reference [1]) that make up the service;

b) the extensions and refinements of the behavior of those CSTS procedures necessary to provide the transfer service;

c) the extensions and refinements of standard CSTS operations associated with each of the procedures;

d) the relationships among the procedures that make up the service;

e) the requirements on tracking service production to enable the proper operation of the TD-CSTS.

This Recommended Standard 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 measure the values of tracking service parameters;

d) the methods or technologies required for communication;

e) the management activities necessary to schedule, configure, and control the TD-CSTS.

1.3 APPLICABILITY

The applicability and limits of applicability of Cross Support Transfer Services in general, as described in reference [1], pertain to the TD-CSTS.

The TD-CSTS is most applicable in situations where tracking data measurements are needed by the user within some relatively short interval from the time at which they were taken (i.e., in 'near real time'). The applicability of the TD-CSTS is limited to the tracking data provision, i.e., the transfer of tracking data between a TD-CSTS provider, typically a ground station, and a TD-CSTS user, typically an operations control center. The TD-CSTS provider is a *Cross Support Service Element* (CSSE), which is part of a *Cross Support Service System* (CSSS). The TD-CSTS user is located at an *Earth User Node*. These terms are defined in reference [6].

The TD-CSTS can also be used in situations where all tracking data measurements are not required to be delivered in near real time.

NOTE – However, the TDM that results from transfer via the TD-CSTS will likely be larger than a TDM that contains the same information but that is generated at a single time (i.e., after the conclusion of the tracking pass) because the transfer of TDM data via the TD-CSTS involves the repetition of metadata with every tracking data measurement in order to facilitate the incremental transfer of tracking data in near real time.

As specified in reference [3], TDMs can contain raw or processed tracking data, and consequently the TD-CSTS is capable of transferring raw as well as processed data.

1.4 RATIONALE

The goal of this Recommended Standard is to create a standard for interoperability for the exchange of tracking data between the CSSE, as defined in reference [6], of various space Agencies and the users at Earth User Nodes (reference [6]) of the cross support services that the CSSEs provide.

1.5 DOCUMENT STRUCTURE

1.5.1 DOCUMENT ORGANIZATION

Section 2 describes the Tracking Data Cross Support Transfer service in terms of:

- the role of Service Management with respect to the TD-CSTS;
- the allocation of production and provision of the TD-CSTS to Functional Resources;
- the cross support view of the TD-CSTS;
- the functional description of the production and provision of the service; and
- an operational scenario that illustrates some of the more significant aspects of the service.

Section 3 specifies the top-level composition of the TD-CSTS. The service type identifier is declared, the procedures that make up the service are identified, and the CSTS state machine that applies to the TD-CSTS is specified.

Section 4 defines the Buffered Tracking Data Message Delivery procedure as a derivation of the Buffered Data Delivery procedure as specified in 4.5 of reference [1].

Section 5 specifies how the procedure configuration parameters are to be set for the TD-CSTS.

Section 6 specifies the Tracking Data Service-specific versions of the service-generic parameters and events that are defined in reference [1].

Section 7 defines the refinement of parameters and events defined in reference [1] as they apply to the TD-CSTS.

Annex A documents the Implementation Conformance Statement (ICS) Proforma for the TD-CSTS.

Annex B specifies the structure and content of the data components that are used by the TD-CSTS to incrementally transfer TDMs. This annex also includes an example TDM Header and example TDM atomic segments.

Annex C formally specifies the Object Identifiers (OIDs) for the Tracking Data transfer service, the TD-CSTS Provider Functional Resource Type, and the TDM Recording Buffer Functional Resource Type.

Annex D formally specifies the ASN.1 protocol data units (PDUs) for the Buffered Tracking Data Message Delivery procedure.

Annex E formally specifies the ASN.1 parameters, events, and directives for the Buffered Tracking Data Message Delivery procedure.

Annex F defines the tracking data production process. In particular, it specifies how tracking data measurements and associated metadata are to be generated and stored so that, when transferred by TD-CSTS instances, the result will be data structures that conform to the syntactic and semantic requirements for Tracking Data Messages as specified in reference [3].

Annex G addresses the security, Space Assigned Numbers Authority (SANA), and patent considerations associated with the TD-CSTS.

Annex H lists the acronyms used in this Recommended Standard.

Annex I lists the informative references cited in this Recommended Standard.

Annex J lists the specific sections and subsections of the CSTS Specification Framework (reference [1]) and of the Tracking Data Message (reference [3]) that are referenced by this Recommended Standard.

1.5.2 CROSS SUPPORT TRANSFER SERVICES DOCUMENTATION

The basic organization of the CSTS documentation and the relationship to CSTS documentation is shown in figure 1-1.

The Cross Support Services documents that are related to Cross Support Transfer Services are:

a) *Cross Support Concept—Part 1: Space Link Extension Services* (reference [I1]) A report introducing the concepts of cross support and the SLE services. Many of the concepts for the SLE transfer services have been adopted for the CSTSes (refer to c) below);

b) *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [2]): A Recommended Standard that defines the framework and terminology for the specification of SLE services. Much of the framework and terminology of this reference model has been adopted or adapted for CSTSes in reference [1];

c) *Space Communication Cross Support Service Management* suite (references [I6], [I7], and [I8]). Recommended Standards that specify the Service Management Information Entities that are used to configure and schedule CSTSes;

d) *The SLE Transfer Services suite*: The SLE Transfer Services are a suite of cross support transfer services that are used to transfer specific telecommand and telemetry protocol data units. The SLE Transfer Services are closely related to the CSTS suite in that they collectively define the set of operations that are the basis for reference [1]. However, because of history (the SLE Transfer Services were already specified and implemented prior to development of reference [1]) the SLE Transfer Services are separated from CSTSes;

e) *Space Link Extension - Internet Protocol for Transfer Services* (reference [I2]): A Recommended Standard that defines a protocol for transfer of Protocol Data Units (PDU) defined in the Cross Support Transfer Services. This Recommended Standard was originally developed to support SLE transfer services (hence the title), but it is also applicable to (and specified for) use by Cross Support Transfer Services.

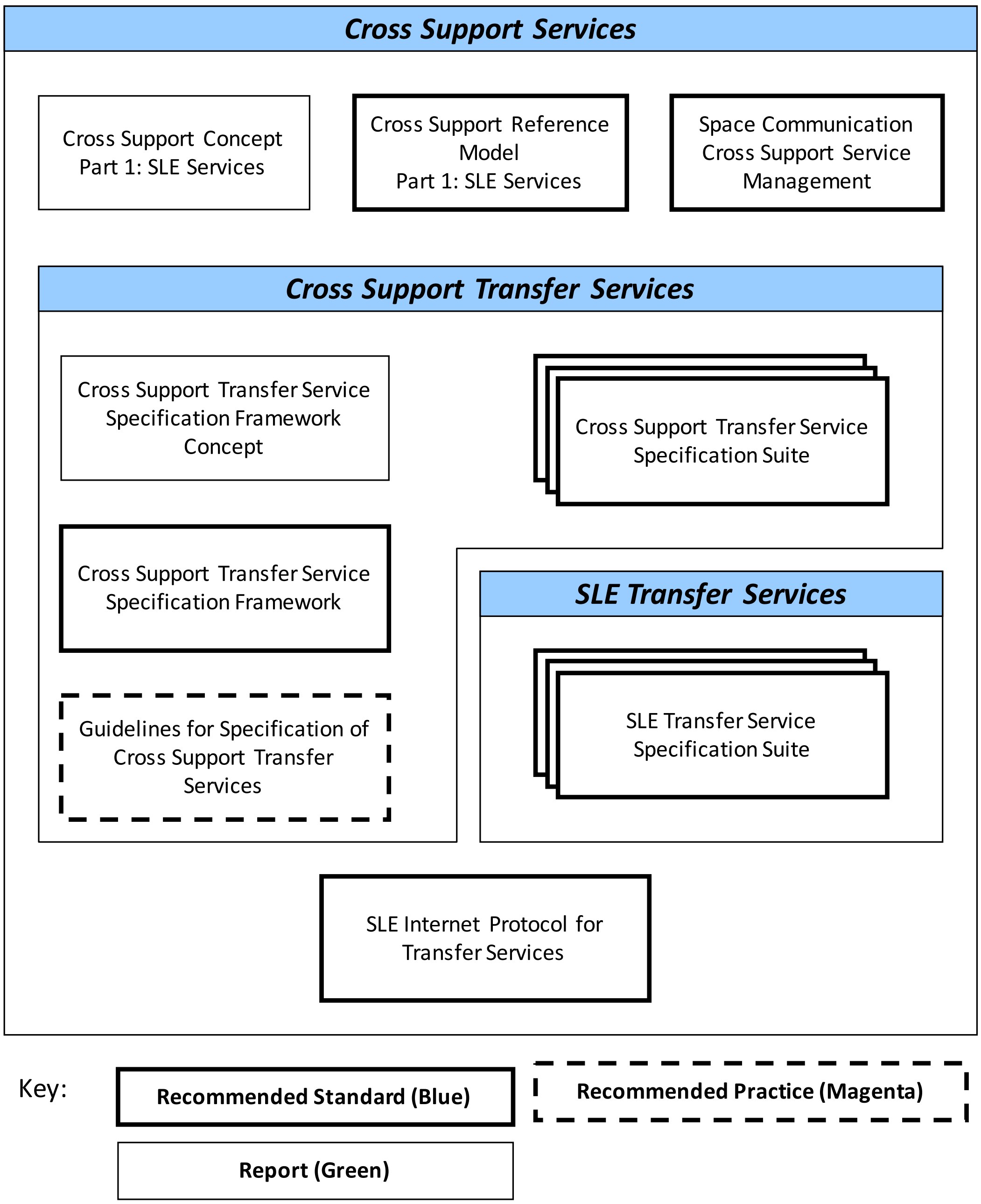

Figure 1-1: Cross Support Services Documentation

The documents specific to Cross Support Transfer Services are:

a) *Cross Support Transfer Services Specification Framework* (reference [1]): A Recommended Standard that specifies the Cross Support Transfer Services procedures and defines their basic building blocks;

b) *Guidelines for the Specification of Cross-Support Transfer Services*: A Recommended Practice that defines the guidelines for construction of a Cross Support Transfer Service based on reference [1];

c) *Cross Support Transfer Services Specification Framework Concepts* (reference [I3]): A report that provides tutorial material on the objectives and concepts of reference [1];

d) Cross Support Transfer Services Suite: The set of specifications for actual CSTSes built from the procedures in reference [1] and in accordance with the CSTS Guidelines. The Cross Support Transfer Service Suite includes this Recommended Standard.

1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS

1.6.1 TERMS

1.6.1.1 Terms Defined in the CSTS Specification Framework (reference [1])

a) Association Control procedure;

b) blocking [operation];

c) Buffered Data Delivery procedure;

d) complete [data delivery mode];

e) Cross Support Complex;

f) Cross Support service production;

g) Cross Support Transfer Service provision;

h) Cross Support Transfer Service;

i) delivery mode;

j) discardable;

k) latency limit;

l) non-blocking [operation];

m) non-discardable;

n) procedure configuration parameter;

o) qualified parameter;

p) real-time [data delivery mode];

q) recording buffer;

r) return-buffer-size;

s) service instance provision period;

t) service management parameter;

u) service production data unit;

v) service-user-responding-timer;

w) start-generation-time;

x) stop-generation-time;

y) TransferDataInvocation.

1.6.1.2 Terms Defined in the Cross Support Reference Model (reference[2])

a) Complex Management (CM);

b) Forward CLTU (F-CLTU);

c) Mission Data Operations System (MDOS);

d) Return All Frames (RAF);

e) Service Package;

f) Space Link Session;

g) Utilization Management (UM).

1.6.1.3 Terms Defined as Keywords in the Tracking Data Message Specification (reference [3])

a) ANGLE_1;

b) ANGLE_2;

c) CARRIER_POWER;

d) DOPPLER_INSTANTANEOUS;

e) DOPPLER_INTEGRATED;

f) ORIGINATOR;

g) PARTICIPANT;

h) PC_N0 (carrier power to noise spectral density);

i) PR_N0 (ranging power to noise spectral density);

j) RANGE;

k) RECEIVE_FREQ, RECEIVE_FREQ_n;

l) TRANSMIT_FREQ_n;

m) TRANSMIT_FREQ_RATE_n.

1.6.1.4 Other Terms Defined in the Tracking Data Message Specification (reference [3])

a) Data Section;

b) differential one-way ranging (DOR);

c) keyword;

d) TDM Header;

e) TDM Metadata Section;

f) TDM Segment;

g) timetag;

h) Tracking Data Message (TDM);

i) Tracking Data Record;

j) very Long Baseline Interferometry (VLBI).

NOTE – Delivery of DOR or VLBI tracking data requires an enormous amount of raw data be sent to and processed by a correlator to generate those tracking data types. Given that the TD-CSTS is primarily intended for delivery of tracking data in (near) real-time, the delivery of DOR or VLBI data is outside the scope of this Recommended Standard (see 1.2.1).

1.6.1.5 Terms Defined in the Extensible Space Communication Cross Support Service Management Concept

This Recommended Standard makes use of the following term defined in reference [I6]:

retrieval service package.

1.6.1.6 Definitions from Abstract Syntax Notation One

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

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

b) Object Identifier, OID.

NOTE – OIDs belong to an ISO/IEC-standardized identifier mechanism for naming any object, concept, or 'thing' with a globally unambiguous persistent name. An OID represents a node in a hierarchically assigned name space defined by reference [4]. OIDs used in the CCSDS context are documented in dedicated registries (reference [5]).

1.6.1.7 Terms Defined in CCSDS Recommended Practice for Space Communication—Cross Support Architecture Requirements Document

This Recommended Standard makes use of the following terms defined in reference [6]:

a) cross support service system, CSSS;

b) cross support service element, CSSE;

c) Earth User Node.

1.6.1.8 Terms Defined in This Specification

TDM atomic segment: a TDM Segment (i.e., Metadata Section and Data Section), constrained to contain a single tracking data measurement and its associated metadata.

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

The following conventions apply for the normative specifications in 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.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.6.2.2 Informative Text

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.6.3 CONVENTIONS

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

1.7 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications. However, this issue of this Recommended Standard is valid only in conjunction with the specifications provided in the issue identified below in reference [1].

[1] *Cross Support Transfer Service—Specification Framework*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 921.1-B-1. Washington, D.C.: CCSDS, April 2017.

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

[3] *Tracking Data Message*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 503.0-B-2. Forthcoming.

[4] *Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. 5th ed. International Standard, ISO/IEC 8825-1:2015. Geneva: ISO, 2015.

[5] "Functional Resources." Space Assigned Numbers Authority. http://sanaregistry.org/r/functional_resources/.

[6] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

2 OVERVIEW OF THE TRACKING DATA CROSS SUPPORT TRANSFER SERVICE

2.1 SERVICE SUMMARY

The Tracking Data CSTS is a CCSDS Cross Support Transfer Service that provides a user with the capability to obtain tracking data in periodic measurements that are taken during the utilization phase of the cross support Service Package.

The tracking data are formatted in accordance with the CCSDS *Tracking Data Message* Recommended Standard (reference [3]). Tracking data measurement types supported by the Tracking Data Message (TDM) format include (among others) antenna angles, range, Doppler, and transmit and receive frequencies. The data is transferred by the TD-CSTS in such a way that reconstitution of a valid TDM is accomplished by direct extraction and concatenation of data fields from TD-CSTS protocol data units, with no format conversions required on the service user side.

Depending on its configuration, one TD-CSTS instance may deliver several tracking data types, e.g. range and Doppler. However, each operation conveying tracking data carries only a single measurement and the associated meta data. This data unit is referred to as *TDM atomic segment* and defined in 2.2.2. An example of a sequence of TDM atomic segments is presented in B3.

The TD-CSTS supports delivery of periodic tracking data measurements in two data delivery modes: real-time mode and complete mode. The data delivery mode of an instance of the TD-CSTS is established by service management prior to the beginning of the service instance provision period. As for all configuration actions, this mode establishment should preferably be achieved by means of CCSDS-specified Service Management, but it may also be accomplished by means of bilateral or mission-specific arrangements.

In the real-time data delivery mode, the tracking data measurements are guaranteed to be delivered within a worst-case latency that is defined by the service user, respecting the minimum latency supported by the given service provider. In order to stay within this worst-case latency, a real-time TD-CSTS instance can discard some measurements if backpressure in the connection between the service user and service provider makes delivery of all measurements within the specified latency impossible (e.g., if the network that connects the service user and provider becomes congested). The key characteristics of the communications service underlying the TD-CSTS are those specified in 1.3.1 of reference [1].

In the complete data delivery mode, the TD-CSTS is guaranteed to deliver all periodic tracking data measurements, but without a guaranteed limit on the latency of their delivery. The complete delivery mode also allows the user to receive periodic tracking data measurements that were received before the TD-CSTS becomes active, either from an earlier time in the executing Service Package or even from a previously executed Service Package.

NOTE – Even though the complete data delivery mode can be used to deliver the tracking data measurements a significant time after they were observed, the complete data delivery mode still delivers the measurements formatted as though they were to be transferred in real time, i.e., not formatted as a TDM file, which would avoid the overhead caused by the metadata sent for each tracking data measurement (see also the NOTE in 1.3).

2.2 FUNCTIONAL DESCRIPTION

2.2.1 GENERAL

As defined in the Cross Support Reference Model (reference [2]), related cross support services are bundled into Service Packages for the purposes of ensuring that the required relationships among those cross support services are preserved during their production and provision. For example, multiple cross support transfer services might be related to the operation of the same RF link, and the return RF link might be related to the forward RF link. All of those transfer services, as well as the RF links themselves, are treated as a single package for the purposes of scheduling.

Service Packages include functions that produce tracking data measurements and the TD-CSTS instances by which those tracking data measurements are delivered to the user(s) of the tracking data services (e.g., the spaceflight mission).

NOTE – There could be methods other than TD-CSTS by which tracking data are delivered to users, for example, via file transfer. Such other methods, if they exist, are outside the scope of this Recommended Standard.

Typically, for each active pass (i.e., period in which the ground station is in contact with the spacecraft), a Service Package is scheduled. A Service Package can contain multiple instances of the Tracking Data CSTS where different service instances may be accessible by different service users. Each instance of the TD-CSTS in a Service Package in principle is capable of reporting all tracking data of the types supported by TD-CSTS for that Service Package. For example, if a Service Package includes functions that produce two-way Doppler and angle tracking measurements and two instances of TD-CSTS, each of the TD-CSTS instances has access to the two-way Doppler and angle tracking measurements.

The rationale of allowing a TD-CSTS provider to provide tracking data by means of different TD-CSTS service instances is that different types of tracking data may originate from physically different implementations of the functions producing those data. For instance, angular measurements may be produced at a different implementation of such function than Doppler data. To support an efficient TD-CSTS implementation, a TD-CSTS provider may organize the TD-CSTS service instances in a way that fits best the implementation of the functions producing tracking data (see figure 2-1). As a consequence, one TD-CSTS service instance might support only one subset of the tracking data types listed in 2.2.2, while another TD-CSTS service instance could support a different subset of the tracking data types.

In other words, implementations may be based on separate TDM Segment Generation functions supporting disjoint sets of tracking data types.

2.2.2 SERVICE PRODUCTION

The production of the Tracking Data CSTS instances associated with a given Service Package consists of:

- the functions that generate tracking-related data;
- the TDM Segment Generation function; and
- the TDM Recording Buffer function.

Figure 2-1 is a notional representation of the relationships among the tracking-data-generating functions, the TDM Segment Generation function, the TDM Recording Buffer function, and the Tracking Data CSTS instances (real-time and complete delivery mode) in the production and provision of Tracking Data CSTS.

The tracking-data-generating functions depicted in figure 2-1 are Forward Space Link Carrier Transmission, Forward Link Ranging, Return Space Link Carrier Reception, Range and Doppler Extraction, and the Antenna. Operating parameters of and measurements taken by these functions are the basis of the following types of tracking data (refer to reference [3]):

- carrier power;
- carrier power to noise spectral density;
- Doppler (instantaneous);
- Doppler (integrated);
- range;
- range power to noise spectral density;
- receive frequency;
- transmit frequency;
- transmit frequency rate; and
- antenna angles.

Definitions of the above listed tracking data types may be found in reference [3].

NOTE – The functions illustrated in figure 2-1 are notional in the sense that they are abstractions, which in most cases are associated with specific space communication technologies.

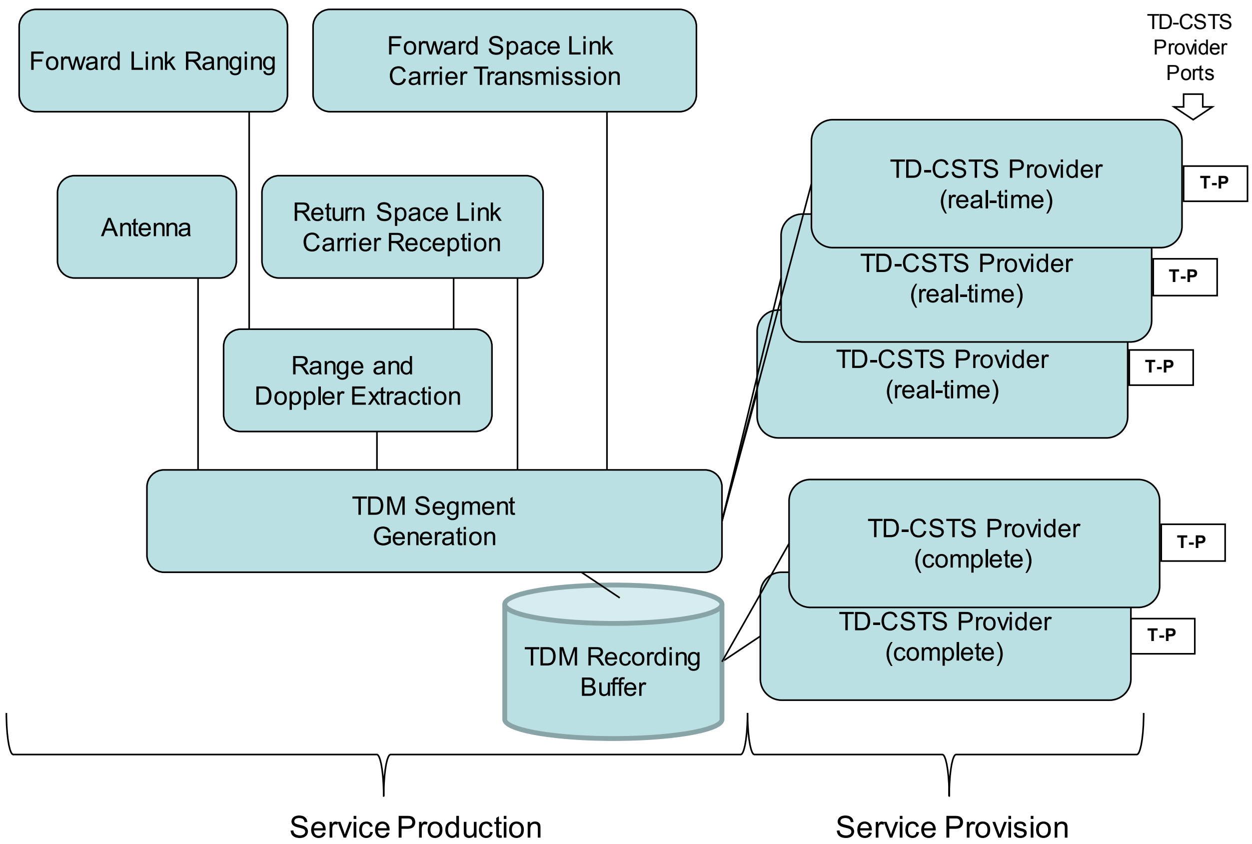

Figure 2-1: Production and Provision of Tracking Data Cross Support Transfer Services

The TDM Segment Generation function combines the tracking measurements from the tracking-data-generating functions to produce a series of *atomic segments* that contain both the metadata that characterizes the tracking data as well as the tracking data measurements themselves, in conformance with the syntactic and semantic rules of the TDM (reference [3]).

Tracking-data-generating functions could occur in different combinations and multiplicities within a Service Package. A Service Package that has both S-band and X-band return links operating simultaneously and both being used for Doppler tracking is an example of a Service Package with multiple concurrent sources of Doppler tracking data. The metadata associated with each tracking data measurement provides the information to discriminate among multiple occurrences of the same tracking data type.

NOTE – In the example above, the lines ‘TRANSMIT_BAND = S’ and ‘RECEIVE_BAND = S’ in the metadata for an atomic segment containing integrated Doppler distinguishes that Doppler measurement from one reported in an atomic segment with the metadata lines ‘TRANSMIT_BAND = S’ and ‘RECEIVE_BAND = X’.

The TDM Recording Buffer stores the tracking data segments for subsequent retrieval by instances of the TD-CSTS operating in complete data delivery mode. The TDM Recording Buffer also stores ‘resource status change’ and ‘resource configuration change’ event notifications regarding the status and configuration of the resources that perform the tracking data-related functions, as defined in 4.5.7 of reference [1]. These notifications are stored in the TDM Recording Buffer synchronously with respect to the time at which the atomic TDM segments are stored. The presence of these notifications may assist the analyst to troubleshoot and/or better understand the environment in which the atomic TDM segments were generated.

NOTE – While the TDM Recording Buffer is capable of receiving, storing, and providing ‘resource status change’ and ‘resource configuration change’ event notifications concerning the resources that perform the tracking data-related functions, the availability of such notifications depends on the instrumentation of the real physical resources that provide those functions.

2.2.3 SERVICE PROVISION

The TD-CSTS operates in both real-time and complete data delivery modes. The data delivery mode of the TD-CSTS is equal to, and configured through, the data delivery mode of the Buffered Tracking Data Message Delivery procedure (refer to section 4), which is the prime procedure of the TD-CSTS. The note in 4.5.2.1 of reference [1] explains why the delivery mode of the prime procedure may be regarded as the delivery mode of the service instance. As stated under 2.1, the data delivery mode of a TD-CSTS instance is established by Service Management prior to the beginning of the service instance provision period. Actually, the prime procedure, i.e., the Buffered Tracking Data Message Delivery procedure, is instantiated as configured by Service Management, and in that way the delivery mode of the TD-CSTS instance is determined.

The Buffered Tracking Data Message Delivery procedure is derived from the CSTS Buffered Data Delivery Procedure defined in 4.5 of reference [1], from which it inherits the two data delivery modes.

In the real-time data delivery mode, the service transfers tracking data measurements as soon as possible after they are taken. In the complete data delivery mode, the service transfers tracking measurements retrieved from the associated TDM Recording Buffer for any specified period before and up to the specified end time. The latest permissible end time is the end of the service instance provision period of the TD-CSTS instance. In case of backpressure in complete delivery mode, delivery of the tracking data covering the time up to the specified end time may not be achieved until the end of the service instance provision period.

Each instance of the TD-CSTS allows the user to select the start and stop times of the tracking data to be delivered.

Each TD-CSTS instance operating in the real-time mode transfers the atomic segments, each of which contains one measurement of one of the selected tracking data types, as they are made available by the TDM Segment Generation function. The user of the service can choose to receive all of the selected atomic segments that are generated between the specified start and stop times for that service instance, or any time-delimited subset thereof.

Each TD-CSTS instance operating in the complete mode retrieves from the TDM Recording Buffer and transfers all atomic segments for the selected tracking data types that have been (or will be) generated by the TDM Segment Generation function between the start and stop generation times specified for that service instance. The start and stop generation times may be any times prior to the end of that service instance provision period for that service instance.

Each TD-CSTS instance allows the user to query the production status of the service and the values of configuration parameters of the procedures that make up the TD-CSTS.

Each TD-CSTS instance notifies the user whenever the production status changes while the service instance is bound during the service instance provision period of the TD-CSTS:

– the production status of a real-time TD-CSTS instance is defined by and equal to the resource status of TDM Segment Generation function that generates the TDM atomic segments;

– the production status of a complete mode TD-CSTS instance is defined by and equal to the resource status of the resource that performs the TDM Recording Buffer function.

Each TD-CSTS instance notifies the user whenever the production configuration changes while the service instance is bound during the service instance provision period of the TD-CSTS:

– the production configuration of a real-time delivery mode TD-CSTS instance encompasses all those resources that perform tracking-data-related functions that provide input to that TD-CSTS instance. This includes the TDM Segment Generation function that generates the TDM atomic segments that are transferred by that TD-CSTS instance. Thus a production configuration change notification is generated and transferred by the real-time TD-CSTS instance whenever a configuration change occurs in any of those resources;

– the production configuration of a complete mode TD-CSTS instance is defined by and equal to the resource configuration of the resource that performs the TDM Recording Buffer function that is associated with that TD-CSTS instance.

NOTE – As noted under 2.2.2, the TDM Recording Buffer records resource status change and resource configuration change notifications that are generated by (or on behalf of) the individual resources that perform tracking-data-related functions. These notifications are also transferred by a complete mode TD-CSTS instance, in generation-time-synchronous order with respect to the recorded atomic TDM segments. However, such resource status change and resource configuration change notifications are independent of the production status change and production configuration change notifications that are related to the complete delivery mode TD-CSTS service instance itself.

2.3 SERVICE MANAGEMENT

Cross support service management both establishes the constraints on the Service Packages to which a given spaceflight mission conforms (e.g., data rate and frequency ranges, types and numbers of cross support transfer service instances) and provides the mechanisms for instantiating conformant Service Packages (e.g., via scheduling).

With regard to the production of tracking data and provision of Tracking Data CSTS instances, cross support service management:

a) schedules the Service Packages that specify the tracking activities that are to be performed and the TD-CSTS instances that transfer the tracking data to the users of those service instances; and

b) establishes the types of tracking data that can be reported by each TD-CSTS instance during the execution of a Service Package.

A Service Package identifies the various space communication and radiometric functions that are to be performed by a Cross Support Complex during a specified period of time. Typically, the Service Package corresponds to the functions performed at a single ground station for one pass/contact/track. More sophisticated cases involving more than one antenna can also be supported. One possibility is that the TDM can contain inputs from different ground stations.

The Service Package also defines – indirectly, through reference to configuration profiles – the configuration parameters that specify the initial configurations of the space communication and radiometric functions, and the interrelationships among them (e.g., the frame length on each return link symbol stream). With respect to the Tracking Data Service, the Service Package identifies which resources perform the functions that produce the radiometric measurements that are reported by the Tracking data service, and pairs those resources to the TDM keyword identifiers (refer to reference [3]) used in the generation of the atomic segments.

The means by which service management schedules the Service Packages is outside the scope of this Recommendation. The CCSDS SCCS-SM suite (references [I7] and [I8]) defines a standard set of service management information entities used in the scheduling of Service Packages.

2.4 CROSS SUPPORT VIEW

Figure 2-2 shows an example configuration of a Cross Support Complex providing instances of Tracking Data CSTS to a Mission Data Operations System (MDOS). Consistent with figure 2-1, the TDM Segment Generation function receives tracking-related measurements from the Return Space Link Carrier Reception function, the Range and Doppler Extraction function, the Forward Space Link Carrier Transmission function, and the Antenna function.

NOTE – The Forward Link Ranging function supplies its ranging timing information to the Range and Doppler Extraction function, which uses that information together with the information obtained from the Return Space Link Carrier Reception function to compute the range and Doppler measurements that are in turn supplied to the TDM Segment Generation function.

The TDM Segment Generation function collects the tracking data from the above-outlined tracking data generating functions and builds for each collected measurement a TDM atomic segment. The built segments are merged to a TDM atomic segments stream and supplied directly to the real-time TD-CSTS instances. For TD-CSTS instances operating in complete delivery mode, the merged segmented tracking data are supplied to the TDM Recording Buffer, which makes them available to the complete mode TD-CSTS instances.

NOTE – For context, figure 2-2 also shows that the Return Space Link Carrier Reception and the Forward Space Link Carrier Transmission functions are also involved (along with the Antenna function) in the production of the Return All Frames (RAF) (reference [I4]) and Forward CLTU (F-CLTU) (reference [I5]) SLE transfer service instances, respectively. However, there are other functions involved in the production of these SLE transfer services that are not illustrated in the figure.

One instance of the TDM Segment Generation function can supply tracking data to multiple instances of real-time TD-CSTSes, and one instance of the TDM Recording Buffer can supply tracking data to multiple instances of complete mode TD-CSTSes. One instance of the TDM Segment Generation function might monitor multiple instances of the same kind of Functional Resource. For example, if a mission spacecraft generates two return space links during a single Space Link Session, (e.g., at S- and X-band) and one-way Doppler tracking is being performed on each of those links, two instances of the Return Space Link Carrier Reception function will be instantiated during the Service Package for that Space Link Session (one for each of S- and X-band). The one-way measurements from both links are provided to the same instance of the TDM Segment Generation function. In such a case, one of the responsibilities of the TDM Segment Generation function is to generate appropriate metadata (refer to reference [3]) to distinguish between the S- and X-band Doppler measurements.

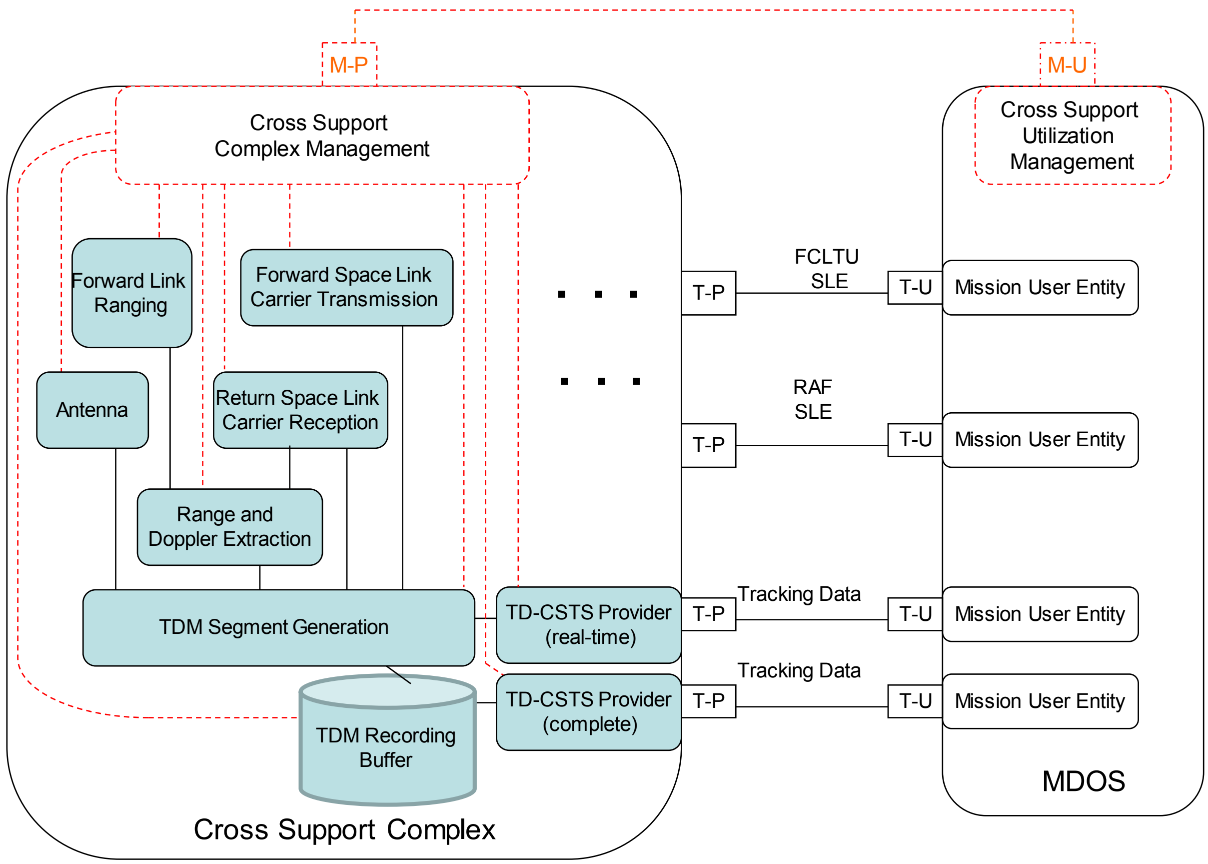

Figure 2-2: Example of the Management and Provision of Tracking Data Service Instances for a Service Package

NOTE – Some real Cross Support Complexes might not be implemented in a way that allows all tracking data to be filtered through a single TDM Segment Generation Functional Resource and/or TDM Recording Buffer. For example, in a particular Complex the antenna angle measurements might not be available for merging with space link frequency data. In such cases the Complex could constrain certain TD-CSTS instances to report only antenna angles and other TD-CSTS instances to report only space link frequency data.

2.5 OPERATIONAL SCENARIO

This section presents an example operational scenario for the TD-CSTS. It does not cover all possibilities, but rather it attempts to provide an example that illustrates the main operational aspects of the TD-CSTS. This scenario is written in the context of service management operating in accordance with the Extensible SCCS-SM Concept (reference [I6]).

2.5.1 SERVICE MANAGEMENT/SERVICE PLANNING ACTIVITIES

As part of Service Management activities that establish the relationship between the spaceflight mission and the Complex, Utilization Management (UM) and Complex Management (CM) negotiate the set of tracking services and associated measurements that will be available to the mission within the context of the Service Agreement. For the purpose of this scenario, the Service Agreement calls for the Complex to provide tracking and communication services on S-Band forward and return links, and tracking measurements consisting of integrated Doppler (range rate), range, and antenna angles. The Service Agreement also specifies how much tracking data can be stored in the TDM Recording Buffer for this mission at each ground station, and the conditions and mechanisms for the purging of that data.

As part of the negotiation process, several Space Link Session (SLS) configuration profiles and several Tracking Data Retrieval configuration profiles are created for use by the mission. Among other things, each SLS configuration profile identifies the resources that are involved in (a) the generation of the Doppler, range, and antenna angle measurements, (b) the generation of atomic TDM segments from those measurements, (c) the recording of those atomic TDM segments for subsequent transfer by complete mode TD-CSTS instances, and (d) the tracking data types to be transferred by each real-time TD-CSTS (if any) that will operate during the course of the SLS (also known as a contact or pass). The SLS configuration profile also specifies the mapping between the identification of those resources and the TDM keywords (refer to reference [3]) that are used to refer to those resources in the atomic TDM segments.

Each Tracking Data Retrieval configuration profile identifies one or more complete mode TD-CSTS instances associated with one TDM Recording Buffer, and for each TD-CSTS instance specifies which tracking data types are to be transferred by that service instance.

At some time after the start time of the Service Agreement, UM causes CM to create two Tracking Data Retrieval Service Packages (refer to reference [I6]), each of which establishes one instance of complete mode TD-CSTS with access to the tracking data collected for the mission for the remaining lifetime of the Service Agreement. Complete mode TD-CSTS instance one (TD-1) is configured to transfer Doppler and range measurements. The user of TD-1 is a Mission flight operations function within the MDOS. Complete mode TD-CSTS instance two (TD-2) is configured to transfer Doppler, range, and antenna angle measurements. The user of TD-2 is the flight dynamics facility that maintains the precision orbit of the spacecraft on behalf of the Mission.

At some (different) time after the start time of the Service Agreement, UM causes CM to create an SLS Service Package (refer to reference [I6]) that specifies an SLS during which (among other things) Doppler and range measurements are to be taken using the S-Band forward and return links. The SLS Service Package includes an instance of real-time TD-CSTS (TD-3) where the user of TD-3 is a Mission flight operations function. The SLS Service Package configures TD-3 to transfer Doppler, range, and antenna angle measurements and specifies the *latency-limit* value for TD-3.

2.5.2 SLS SERVICE PACKAGE EXECUTION

2.5.2.1 Production of TDM Segments During the SLS

At the scheduled start time of the SLS Service Package, the Complex establishes the space links with the spacecraft and begins processing of the signals to and from the spacecraft. When each of the resources that generate or process tracking-related data (including the TDM Segment Generation Function) becomes operational, a time-stamped resource status change event notification, indicating the transition to 'operational', is placed into the TDM Recording Buffer for that resource. In addition, when the TDM Segment Generation Function) becomes operational, the real-time TD-CSTS instance TD-3 generates and transfers a 'production status change' event notification indicating the transition of the production status to 'operational'.

Thereafter, the antenna angle, Doppler, and range measurements are reported to the TDM Segment Generation function. For each single Doppler measurement, single range measurement, and antenna angle pair of measurements, the TDM Segment Generation function uses the mapping information in the applied configuration profile to generate an atomic segment that contains the tracking data measurement and the metadata that characterizes that tracking data in terms of the appropriate TDM keywords. The resultant atomic segments are made available to any and all real-time TD-CSTS instances that are active during the Space Link Session. The atomic segments are also stored by the TDM Recording Buffer for retrieval and transfer by complete mode TD-CSTS instances.

2.5.2.2 Binding of Complete Mode TD-CSTS Instance 1 (TD-1)

At some time before the scheduled start time of the SLS Service Package, the user of the complete mode TD-CSTS instance TD-1 invokes the BIND operation to bind to the service provider and transitions the service instance to the 'bound.ready' state.

2.5.2.3 TD-1 Buffered Tracking Data Message Delivery Procedure

Following the binding of the service instance, the user of TD-1 invokes the START operation of the Buffered Tracking Data Message Delivery (BTDMD) procedure. The BTDMD procedure is the prime procedure of the TD-CSTS. The START invocation has the `start-generation-time` and `stop-generation-time` parameters set equal to the scheduled start and stop times (respectively) of the SLS Service Package, indicating that the service instance is to transfer all Doppler and range tracking data acquired during the execution of the SLS Service Package. TD-1 transitions to the 'bound.active' state and the Buffered Tracking Data Message Delivery instance activates, generates the TDM Header, and transfers the TDM Header to the user in the positive return of the START operation.

TD-1 then retrieves from the TDM Recording Buffer the first 'resource status change' event notification that was put into the TDM Recording Buffer after `start-generation-time`, places it in its Return Buffer, and initiates the release timer. The additional 'resource status change' notifications are subsequently retrieved and put into the Return Buffer. If any new atomic segments containing one of the selected tracking data types (Doppler and range for TD-1) is made available by the TDM Recording Buffer before the expiration of the release timer, TD-1 places them in its Return Buffer. When the Return Buffer fills or the release timer expires, TD-1 transfers the atomic segments and event notifications in the Return Buffer to the service user.

Subsequently, whenever a new atomic segment containing one of the selected tracking data types is made available by the TDM Recording Buffer, TD-1 places it in its Return Buffer and initiates the release timer if the atomic segment is the first one following transmission of the previous Return Buffer.

NOTE – The user of TD-1 is not constrained as to when to start the service instance with respect to the execution of the SLS Service Package: as a complete mode service instance, TD-1 will deliver all designated tracking data for the period between the `start-generation-time` set and `stop-generation-time` even if the service instance is started after the start time of the SLS Service Package. However, starting the service instance at or before the start time of the SLS Service Package ensures that the tracking data measurements are transferred as soon as possible.

2.5.2.4 TD-1 Information Query Procedure

At some time following the activation of the BTDMD procedure, the user invokes the GET operation of the Information Query procedure to query the values of the parameters in the default parameter label list. The `list-of-parameters` parameter of the GET invocation is left empty to indicate the query of the default list. The default parameter list for the TD-CSTS is defined by the service specification to contain all of the configuration parameters of the Association Control and BTDMD procedures. The GET return contains the qualified parameters values for the following set of configuration parameter names:

- service-user-responding-timer (Association Control);
- `initiator-identifier` (Association Control);
- `responder-identifier` (Association Control);
- `service-instance-identifier` (Association Control);
- `return-buffer-size` (Buffered Tracking Data Message Delivery);
- `delivery-mode` (Buffered Tracking Data Message Delivery).

2.5.2.5 Binding of Real-Time TD-CSTS Instance 3 (TD-3)

At the scheduled start of the service instance provision period of the real-time TD-CSTS instance (TD-3) within the SLS Service Package, the service instance exists in the 'unbound' state. Shortly after the scheduled start time of TD-3, the user of that service instance invokes the BIND operation to bind to the service provider and transitions the service instance to the 'bound.ready' state.

2.5.2.6 TD-3 Buffered Tracking Data Message Delivery Procedure Prime Instance

The user of TD-3 then invokes the START operation for the BTDMD procedure. The TD-3 START invocation has 'null' values for the `start-generation-time` and `stop-generation-time` parameters, indicating that the service instance is to begin sending atomic segments and service production change notifications as soon as any are available, and continue sending atomic segments and service production change notifications until the user invokes the STOP operation. TD-3 transitions to the 'bound.active' state and the BTDMD procedure activates, generates the TDM Header, and transfers the TDM Header to the user in the positive return of the START operation.

If the service instance provision period of TD-3 begins with sufficient lead time to allow TD-3 to be bound and started before the TDM Segment Generation function becomes operational, the TD-3 instance generates a 'production status change' event notification when the resource performing the TDM Segment Generation function becomes operational, places it in the Return Buffer, and initiates the release timer.

NOTE – The resource status of the TDM Segment Generation function is determined based on the local status of this function and the status of all other functions being involved in the service production. I.e., the TDM Segment Generation function resource status represents the aggregate status of the service production.

If TD-3 is not started until after the resource performing the TDM Segment Generation function has become operational, whenever the first atomic segment containing one of the subscribed data types is made available by the TDM Segment Generation function, the BTDMD procedure places it in its Return Buffer and initiates the release timer. Subsequently, whenever a new atomic segment containing one of the subscribed data types is made available by the TDM Segment Generation function (or the production status changes), the BTDMD procedure places the atomic segment or event notification in its Return Buffer. When the Return Buffer fills or the release timer expires, TD-3 transfers the atomic segments and event notifications in the Return Buffer to the service user.

NOTE – In order to minimize the latency of transfer of the atomic segments, the Return Buffer latency timer value will normally be configured to be on the order of the shortest sampling period of all of the tracking data sources for the TD-CSTS instance.

2.5.2.7 Backpressure in the Connection between TD Service Provider and User

Partially through the Space Link Session, the communication service underlying TD-1 and TD-3 begins to experience congestion to the point that the resulting backpressure prohibits all of the tracking data from being transferred across the reliable connection in a timely manner. In order to maintain the timeliness of the tracking data measurements, TD-3 discards 'stale' atomic segments and event notifications in order to ensure that the latest measurements are the ones that are delivered. After the backpressure clears, the transfer of timely atomic segments and event notifications resumes, along with a notification that some data has been discarded.

In contrast, TD-1, being a complete mode TD-CSTS instance, does not discard any atomic segments or notifications but rather continues to attempt to send all recorded segments and notifications regardless of the delay. After the backpressure clears, the backlogged atomic segments and notifications are transferred at the maximum rate permitted by the underlying communication service and the performance of the TD-1 user entity.

NOTE – In many cases the nominal data rate of the underlying communication service will be at least several times that needed to transfer the tracking data. In these cases, even if backpressure temporarily slows the transfer of the atomic segments or notifications, a complete mode TD-CSTS instance is likely to quickly 'catch up' once the backpressure has cleared.

2.5.2.8 Stopping and Unbinding of TD-3

Shortly before the scheduled end of the service instance provision period of TD-3, the user invokes the STOP operation of the BTDMD procedure to cease transfer of the tracking data measurements. The BTDMD procedure deactivates and TD-3 transitions to the 'bound.ready' state. The user then invokes the UNBIND operation, which causes TD-3 to transition to the 'unbound' state.

2.5.2.9 Stopping and Unbinding of TD-1

At the specified `stop-generation-time` for complete mode TD-CSTS instance TD-1, the BTDMD procedure of TD-1 generates and sends a notification that the requested data has reached its end. The user invokes the STOP operation of the BTDMD procedure to transition the procedure to the inactive state. The BTDMD procedure deactivates and TD-1 transitions to the 'bound.ready' state. The user then invokes the UNBIND operation, which causes TD-1 to transition to the 'unbound' state.

2.5.3 POST-SPACE LINK SESSION

2.5.3.1 Retrieval of TDM Segments after Completion of the SLS

Following the execution of the SLS Service Package, the antenna angle, Doppler, and range measurements for the SLS have been processed by the TDM Segment Generation function and stored by the TDM Recording Buffer for retrieval and transfer by complete mode TD-CSTS instances.

2.5.3.2 Binding of Complete Mode TD-CSTS Instance 2 (TD-2)

At some time after the stop time of the SLS Service Package, the user of complete mode TD-CSTS instance TD-2 invokes the BIND operation to bind to the service provider and transitions the service instance to the 'bound.ready' state.

2.5.3.3 TD-2 Buffered Tracking Data Message Delivery Procedure Prime Instance

The user of TD-2 then invokes the START operation of the BTDMD procedure. The START invocation has values for the start-generation-time and stop-generation-time parameters that are within the lifetime of the SLS Service Package described above. TD-2 transitions to the 'bound.active' state and the BTDMD procedure activates, generates the TDM Header and transfers the TDM Header to the user in the positive return of the START operation. The BTDMD procedure then retrieves (from the TDM Recording Buffer) and transfers (a) the atomic segments that contain the Doppler, range, and antenna angle measurements and that fall within the start-generation-time/stop-generation-time interval, and (b) recorded event notifications that were generated within the start-generation-time/stop-generation-time interval.

NOTE – The setting of the start-generation-time and stop-generation-time parameters to be within the execution time of the SLS Service Package is arbitrary. A complete mode TD-CSTS instance has access to all tracking data in the TDM Recording Buffer with which it is associated, regardless of when (i.e., during which Space Link Session) the data is captured.

2.5.3.4 Stopping and Unbinding of TD-2

At the specified stop-generation-time of TD-2, the BTDMD procedure of TD-2 generates and sends a notification that the requested data has reached its end. The user subsequently invokes the STOP operation to cease transfer of the tracking data measurements. The BTDMD procedure deactivates and TD-2 transitions to the 'bound.ready' state. The user then invokes the UNBIND operation, which causes TD-2 to transition to the 'unbound' state.

Because TD-2 is a complete delivery mode TD-CSTS instance, TD-2 delivers all recorded tracking data of the selected types and event notifications that were originally generated during the period bounded by the start-generation-time and stop-generation-time parameters, even if the connection between the provider and user of TD-2 experiences backpressure when TD-2 is active.

3 TRACKING DATA CROSS SUPPORT TRANSFER SERVICE COMPOSITION

3.1 DISCUSSION

The Tracking Data CSTS specification is complete and therefore concrete; i.e., it can be implemented as defined herein without need for further extension or refinement.

The service-level OIDs for the Tracking Data CSTS are specified in annex C.

3.2 PROCEDURES OF A TRACKING DATA CROSS SUPPORT TRANSFER SERVICE INSTANCE

3.2.1 The Tracking Data transfer service shall be composed of the Association Control, Buffered Tracking Data Message Delivery, and Information Query procedures.

3.2.2 The Association Control procedure shall conform to the Association Control procedure as defined in 4.3 of reference [1] without derivation.

3.2.3 There shall be one and only one instance of the Association Control procedure.

3.2.4 The Buffered Tracking Data Message Delivery procedure shall be derived from the Buffered Data Delivery procedure as specified in 4.5 of reference [1]. The derivation is specified in section 4.

3.2.5 The Buffered Tracking Data Message Delivery procedure shall be the primary procedure for the Tracking Data service.

3.2.6 There shall be one and only one instance of the Buffered Tracking Data Message Delivery procedure.

3.2.7 The version number of the Buffered Tracking Data Message Delivery procedure is 1.

3.2.8 The Information Query procedure shall be adopted directly from the Information Query procedure defined in 4.9 of reference [1].

3.2.9 There shall be one secondary procedure instance of the Information Query procedure.

NOTE – Table 3-1 summarizes the procedures that make up the Tracking Data transfer service, where (a) the '[P]' designates Buffered Tracking Data Message Delivery as the primary procedure; (b) *Version* = '-' indicates that the version of the service procedure is the same as that of the procedure specified in reference [1] for the procedures that are directly adopted (Association Control and Information Query), and *Version* = '1' indicates the version of the refined and/or extended service procedures (Buffered Tracking Data Message Delivery); (c) *No. of Instances* indicates the minimum and maximum number of allowed instances of each procedure type; (d) *Specification Approach* indicates which procedures are directly adopted or refined and extended; and *Source* indicates the procedure specified in reference [1] from which the service procedure is adopted, refined and/or extended.

Table 3-1: Procedures of the Tracking Data CSTS

Procedure	Association Control	Buffered Tracking Data Message Delivery [P]	Information Query
Version	-	1	-
No. of Instances	1..1	1..1	1..1
Specification Approach	adopted	refined-and-extended	adopted
Source	Subsection 4.3 of reference [1]: Association Control	Subsection 4.5 of reference [1]: Buffered Data Delivery	Subsection 4.9 of reference [1]: Information Query

3.3 TRACKING DATA CROSS SUPPORT TRANSFER SERVICE STATE MACHINE

The Tracking Data Cross Support Transfer Service state machine conforms to the state machine for a CSTS with a stateful prime procedure instance, as defined in G3 of reference [1].

4 BUFFERED TRACKING DATA MESSAGE DELIVERY PROCEDURE

4.1 DISCUSSION

4.1.1 PURPOSE

The Buffered Tracking Data Message Delivery procedure supports transfer from the Provider to the User of CCSDS Tracking Data Messages as specified in reference [3], structured into data units using one of the real-time or complete delivery modes, as defined in 4.5 of reference [1].

4.1.2 CONCEPT

The Buffered Tracking Data Message Delivery (BTDMD) procedure is derived from the Buffered Data Delivery procedure specified in reference [1] so that it conveys the tracking data in a way that satisfies the syntactic and semantic requirements for valid CCSDS Tracking Data Messages.

a) The contents of the `data` parameter of the TransferDataInvocations are refined to be TDM atomic segments, as defined in annex B of this specification;

b) The START positive return is extended to include a TDM Header, as defined in annex B of this specification.

A BTDMD procedure instance can deliver one or more tracking data types, as configured by Service Management. For each tracking data type that is specified for a BTDMD instance, the BTDMD delivers the TDM atomic segments for all instances of that tracking data type that are generated by the tracking-related resources that are associated with the TD-CSTS instance that executes the BTDMD procedure. The association of TD-CSTS instances with specific tracking-related resources is a function of Service Management.

NOTE – The parent Buffered Data Delivery procedure in reference [1] specifies that the START operation includes selection criteria for the selection of the TransferDataInvocations to be delivered. For the case of the BTDMD procedure, the selection criteria applied in the START operation are limited to the `start-generation-time` and `stop-generation-time` parameters specified in the parent Buffered Data Delivery procedure of reference [1]. Selection of the types of tracking data to be delivered is configured in the Service Package prior to the start of the TD-CSTS service instance provision period and applies for the duration of the service instance provision period.

4.2 PROCEDURE TYPE IDENTIFIER

The procedure identifier `buffTrkDataMsgDel`, as specified in annex C, shall be used for this procedure.

4.3 EXTENSION

The Buffered Tracking Data Message Delivery shall extend the Buffered Data Delivery procedure by modification of the behavior of the procedure, addition of a parameter to the START operation, and refinement of a parameter of the TRANSFER-DATA operation.

4.4 BEHAVIOR

4.4.1 OVERVIEW

The overall activities of the Buffered Tracking Data Message Delivery procedure are the same as those of the standard Buffered Data Delivery procedure as defined in 4.5 of reference [1]. The detailed behavior of the Buffered Tracking Data Message Delivery procedure with respect to starting, transferring data and notifications, stopping, and aborting is the same as that of the standard Buffered Data Delivery procedure as defined in 4.5 of reference [1], with the exceptions to the Starting behavior specified in 4.4.2 and the Transferring Data and Notifications behavior defined in 4.4.3.

4.4.2 DERIVED BEHAVIOR—STARTING

In successfully performing the START operation, the service provider shall return a result that contains a TDM Header (refer to annex B) in addition to the data contained in the Buffered Data Delivery START positive return.

4.4.3 DERIVED BEHAVIOR—TRANSFERRING DATA AND NOTIFICATIONS

4.4.3.1 In addition to the `start-generation-time` and `end-generation-time` selection criteria specified in the START invocation, the selection of data to be delivered by the TRANSFER-DATA invocations shall be further limited to the atomic segments (refer to 4.5.3) that contain tracking data that conform to the `tracking-data-types` configuration parameter.

4.4.3.2 The `tracking-data-types` configuration parameter shall specify one or more of the following types:

– a pair of antenna angle Tracking Data Records;

– carrier power;

– carrier power to noise spectral density;

– Doppler (instantaneous);

– Doppler (integrated);

– range;

– ranging power to noise spectral density;

– receive frequency;

– transmit frequency;

– transmit frequency rate.

NOTE – The capabilities of individual tracking service providers may restrict the types of tracking data that are available from those providers.

4.5 REQUIRED OPERATIONS

4.5.1 GENERAL

4.5.1.1 The Buffered Tracking Data Message Delivery procedure shall use the STOP and NOTIFY operations of the Buffered Data Delivery procedure of reference [1] without extension or refinement.

4.5.1.2 The Buffered Tracking Data Message Delivery procedure shall extend the START operation of the Buffered Data Delivery procedure of reference [1] as specified in 4.5.2.

4.5.1.3 The Buffered Tracking Data Message Delivery procedure shall refine the TRANSFER-DATA operation of the Buffered Data Delivery procedure of reference [1] as specified in 4.5.3.

4.5.1.4 The START and STOP operations of the Buffered Tracking Data Message Delivery procedure shall be Blocking, as defined for the parent Buffered Data Delivery procedure in reference [1].

4.5.1.5 The TRANSFER-DATA and NOTIFY operations of the Buffered Tracking Data Message Delivery procedure shall be Non-Blocking, as defined for the parent Buffered Data Delivery procedure in reference [1].

NOTE – Table 4-1 summarizes the operations of the Buffered Tracking Data Message Delivery procedure.

Table 4-1: Buffered Tracking Data Message Delivery Required Operations

Operations	Extended	Refined	Procedure Blocking/Non-Blocking
START	Y	N	Blocking
STOP	N	N	Blocking
TRANSFER-DATA	N	Y	Non Blocking
NOTIFY	N	N	Non Blocking

4.5.2 START (CONFIRMED)

4.5.2.1 Invocation, Return, and Parameters

4.5.2.1.1 In addition to the parameters of the START invocation and return for the Buffered Data Delivery procedure as defined in 4.5 of reference [1], the extension parameter specified in table 4-2 shall be present in the START return of the Buffered Tracking Data Message Delivery procedure.

Table 4-2: START Extension Parameter

Extension Parameters	Invocation	Return
`tdm-header`		C

4.5.2.1.2 Extension Parameter Syntax

The type `BuffTrkDataDelStartPosReturnExt`, as defined in annex D, shall define the syntax of the extension parameter of the START positive return.

4.5.2.1.3 `tdm-header`

4.5.2.1.3.1 The `tdm-header` parameter shall contain a TDM Header, as specified in annex B.

4.5.2.1.3.2 The `tdm-header` parameter shall be present in the return if and only if the value of the `result` parameter is 'positive result' (refer to the `result` parameter of the Standard Operation Header as defined in 3.3 of reference [1]).

4.5.3 TRANSFER-DATA (UNCONFIRMED)

The **`data`** parameter shall be of type octet string, formatted as an atomic segment as defined in annex B.

4.6 CONFIGURATION PARAMETERS

4.6.1 Table 4-3 defines the configuration parameters of the Buffered Tracking Data Message Delivery procedure that need to be configured in the context of this procedure. For each configuration parameter, the table provides the engineering unit (if applicable), a cross reference to the use of the parameter in the specification of the procedure, identifies whether the parameter may be read and/or dynamically modified, and also identifies the Parameter Identifier to be used in reporting the value of the parameter.

4.6.2 The setting of service management and configuration parameters inherited from operations and procedures defined in reference [1] is specified in chapter 5.

4.6.3 The `tracking-data-types` (refer to table 4-3) shall be configured by a service management parameter with the classifier `tdTrackingDataTypes`.

Table 4-3: Buffered Tracking Data Message Delivery Configuration Parameters

Parameters	Cross-Reference	Readable	Dynamically modifiable	Configuration Parameter Identifier and Type
`return-buffer-size` (in number of TRANSFER-DATA and/or NOTIFY invocations the buffer will accommodate)	Subsection 4.5.3.2.7.3 of reference [1]	Yes	Yes	`pBDDreturnBufferSize` `PBDDreturnBufferSizeType` (subsection F4.16 of reference [1])
`delivery-latency-limit` (in seconds)	Subsection 4.5.3.2.7.2 of reference [1]	Yes	Yes	`pBDDdeliveryLatencyLimit` `PBDDdeliveryLatencyLimitType` (subsection F4.16 of reference [1])
`delivery-mode`	Subsections 4.5.2.2.2.1 and 4.5.3.2.6 of reference [1]	Yes	No	`pBDDdeliveryMode` `PBDDdeliveryModeType` (subsection F4.16 of reference [1])
`tracking-data-types`	4.4.3.2	Yes	No	`pBTDMDtrackingDataTypes` `PBTDMDtrackingDataTypesType` (annex E)

4.7 PROCEDURE STATE TABLE (PROVIDER SIDE)

The state table for the Buffered Tracking Data Message Delivery procedure shall be the same as that for the Buffered Data Delivery procedure as specified in 4.5.6 of reference [1].

5 SETTING OF SERVICE MANAGEMENT AND CONFIGURATION PARAMETERS INHERITED FROM FRAMEWORK OPERATIONS AND PROCEDURES

5.1 OVERVIEW

The BIND operation defines the `responder-port-identifier` parameter (refer to 3.4.2.2.4.3 of reference [1]) to be a service management parameter of each CSTS. Subsection 5.2, below, specifies the classifier to be used for the `responder-port-identifier` parameter for the TD-CSTS. The `parameterId` corresponding to this classifier is defined in the SANA Functional Resource Registry (reference [5]) subtree for the Tracking Data CSTS Provider Functional Resource.

NOTE – As described in the specification of the `responder-port-identifier` parameter in reference [1], the contents of the parameter are not used by the procedures of the CSTS provider itself, but rather by the underlying communication service that delivers the incoming PDUs to a CSTS provider. The purpose of assigning a classifier and `parameterId` to this parameter is to allow its value to be reported or queried.

The procedures of reference [1] define configuration parameters for those procedures, but defer to the derived procedures the specification of the method by which each of those configuration parameters is to be set. This section specifies the method by which each of the procedure configuration parameters defined in reference [1] is to be set for the TD-CSTS.

For each of the procedure configuration parameters that are specified to be a service management parameter, the classifier for each parameter is also specified. The `parameterId` corresponding to each such classifier is defined in the SANA Functional Resource Registry (reference [5]) subtree for the Tracking Data CSTS Provider Functional Resource.

5.2 `responder-port-identifier` SERVICE MANAGEMENT PARAMETER

The `responder-port-identifier` service management parameter (subsection 3.4.2.2.4.3 of reference [1]) shall have the classifier `tdResponderPortId`.

5.3 ASSOCIATION CONTROL PROCEDURE CONFIGURATION PARAMETERS

5.3.1 The service-user-responding-timer (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdServiceUserRespondingTimer`.

5.3.2 The `initiator-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdInitiatorId`.

5.3.3 The `responder-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdResponderId`.

5.3.4 The `service-instance-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdServiceInstanceId`.

5.4 BUFFERED TRACKING DATA MESSAGE DELIVERY PROCEDURE CONFIGURATION PARAMETERS

NOTE – Reference [1] defers the setting of the `delivery-mode`, `delivery-latency-limit`, and `return-buffer-size` configuration parameters to the service that uses the Buffered Data Delivery procedure or a procedure derived from it. For the TD-CSTS this is the Buffered Tracking Data Message Delivery procedure.

5.4.1 The `delivery-mode` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdDeliveryMode`.

5.4.2 The `return-buffer-size` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdReturnBufferSize`.

5.4.3 The `delivery-latency-limit` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdDeliveryLatencyLimit`.

5.5 INFORMATION QUERY PROCEDURE CONFIGURATION PARAMETERS

NOTE – Reference [1] defers the setting of the `default list of parameters` and `named label lists` configuration parameters to the service that uses the Information Query procedure. The Tracking Data CSTS defines the contents and name of the default list of parameter labels.

5.5.1 The `default list of parameters` (refer to table 4-52 in 4.9.5 of reference [1]) shall be named 'defaultParameterList' and contain the following Parameter Labels:

– `[associationControl: pACserviceUserRespTimer]`, where `associationControl` is the Procedure Identifier of the Association Control procedure as defined in the CCSDS-CSTS-OBJECT-IDENTIFIERS module in F4.1 of reference [1], and `pACserviceUserRespTimer` is the Parameter Identifier of the service-user-responding-timer configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1];

- [associationControl: pACinitiatorId], where associationControl is the Procedure Identifier of the Association Control procedure as defined in the CCSDS-CSTS-OBJECT-IDENTIFIERS module in F4.1 of reference [1], and pACinitiatorId is the Parameter Identifier of the initiator-identifier configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1];

- [associationControl: pACresponderId], where associationControl is the Procedure Identifier of the Association Control procedure as defined in the CCSDS-CSTS-OBJECT-IDENTIFIERS module in F4.1 of reference [1], and pACresponderId is the Parameter Identifier of the responder-identifier configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1];

- [associationControl: pACserviceInstanceId], where associationControl is the Procedure Identifier of the Association Control procedure as defined in the CCSDS-CSTS-OBJECT-IDENTIFIERS module in F4.1 of reference [1], and pACserviceInstanceId is the Parameter Identifier of the service-instance-identifier configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1];

- [buffTrkDataMsgDel: pBDDreturnBufferSize], where buffTrkDataMsgDel is the Procedure Identifier of the TD-CSTS Buffered Tracking Data Message Delivery procedure as defined in annex C, and pBDDreturnBufferSize is the Parameter Identifier of the return-buffer-size configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1];

- [buffTrkDataMsgDel: pBDDrecordingBufferSize], where buffTrkDataMsgDel is the Procedure Identifier of the TD-CSTS Buffered Tracking Data Message Delivery procedure as defined in annex C, and pBDDrecordingBufferSize is the Parameter Identifier of the recording-buffer-size configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1]; and

- [buffTrkDataMsgDel: pBDDdeliveryMode], where buffTrkDataMsgDel is the Procedure Identifier of the TD-CSTS Buffered Tracking Data Message Delivery procedure as defined in annex C, and pBDDdeliveryMode is the Parameter Identifier of the delivery-mode configuration parameter as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES module in F4.16 of reference [1].

5.5.2 The named-label-lists (refer to table 5-52 in 4.9.5 of reference [1]) shall be configured by a service management parameter with the classifier tdNamedLabelLists.

6 TRACKING DATA SERVICE-SPECIFIC VERSIONS OF SERVICE-GENERIC PARAMETERS AND EVENTS

6.1 OVERVIEW

Annex B of reference [1] specifies the following service-generic parameters and events for use by any CSTS:

a) A production status that can be monitored. The OID to be used for the parameter that contains the production status for every CSTS is specified in F4.17 of reference [1] with the classifier `svcProductionStatusVersion1`.

b) A production status change event that is to be emitted when the production status changes, as specified in 3.11.2.2.3.2 a) of reference [1]. The OID to be used for the production status change event for every CSTS is specified in F4.17 of reference [1] with the classifier `svcProductionStatusChangeVersion1`.

c) A production configuration change event that is to be emitted when any Functional Resource in the production experiences a configuration change, as specified in 3.11.2.2.3.2 b) of reference [1]. The OID to be used for the production configuration change event for every CSTS is specified in F4.17 of reference [1] with the classifier `svcProductionConfigurationChangeVersion1`.

Each CSTS is to provide its own label for the production status parameter, production status change event, and production configuration change event.

The Tracking Data service supports the production status parameter as well as the production status change and production configuration change events.

6.2 `TdSvcProductionStatus` PARAMETER

The `tdSvcProductionStatus` parameter shall contain the production status, with the Published Identifier `svcProductionStatusVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F4.17 of reference [1].

6.3 `TdSvcProductionStatusChange` EVENT

The `tdSvcProductionStatusChange` event shall report production status changes, with the Published Identifier `svcProductionStatusChangeVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F4.17 of reference [1].

6.4 `tdSvcProductionConfigurationChange` EVENT

The `tdSvcProductionConfigurationEvent` event shall report production configuration changes, with the Published Identifier `svcProductionConfigurationChangeVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F4.17 of reference [1].

7 REFINEMENT OF DEFINITIONS OF CSTS SPECIFICATION FRAMEWORK PARAMETERS, EVENTS, DIRECTIVES, AND DIAGNOSTIC VALUES USED BY THE TRACKING DATA SERVICE

7.1 OVERVIEW

Except where explicitly refined in this section, the definitions of the parameters, events, directives, and diagnostic values of the operations of the procedures defined in reference [1] that are used by the Tracking Data service are the same as their definitions in reference [1].

7.2 `tdSvcProductionStatus` PARAMETER DEFINITION REFINEMENT

NOTE – This refined definition applies to the `tdSvcProductionStatus` parameter, which has the Published Identifier `svcProductionStatusVersion1` in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F4.17 of reference [1].

7.2.1 PRODUCTION CONFIGURED

7.2.1.1 For a TD-CSTS operating in real-time mode, the definition of the 'production configured' value of the `tdSvcProductionStatus` parameter shall be refined to mean that configuration of the resource performing the TDM Segment Generation function has been completed.

7.2.1.2 For a TD-CSTS operating in complete mode, the definition of the 'production configured' value of the `tdSvcProductionStatus` parameter shall be refined to mean that configuration of the resource performing the TDM Recording Buffer function has been completed.

7.2.2 PRODUCTION INTERRUPTED

7.2.2.1 For a TD-CSTS operating in real-time mode, the definition of the 'production interrupted' value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Segment Generation function has been stopped because of a condition that may be temporary.

7.2.2.2 For a TD-CSTS operating in complete mode, the definition of the 'production interrupted' value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Recording Buffer function has been stopped because of a condition that may be temporary.

7.2.3 PRODUCTION HALTED

7.2.3.1 For a TD-CSTS operating in real-time mode, the definition of the ‘production halted’ value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Segment Generation function has been stopped by management action.

7.2.3.2 For a TD-CSTS operating in complete mode, the definition of the ‘production halted’ value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Recording Buffer function has been stopped by management action.

7.2.4 PRODUCTION OPERATIONAL

7.2.4.1 For a TD-CSTS operating in real-time mode, the definition of the ‘production operational’ value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Segment Generation function has changed to ‘operational’.

7.2.4.2 For a TD-CSTS operating in complete mode, the definition of the ‘production operational’ value of the `tdSvcProductionStatus` parameter shall be refined to mean that the resource performing the TDM Recording Buffer function has changed to ‘operational’.

7.3 `tdSvcProductionStatusChange` EVENT DEFINITION REFINEMENT

NOTE – This refined definition applies to the `tdSvcProductionStatusChange` event, which has the Published Identifier `svcProductionStatusChangeVersion1` defined in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F4.17 of reference [1].

7.3.1 For a TD-CSTS operating in real-time mode, the definition of the `event-value` of the `tdSvcProductionStatusChange` event shall be refined to mean that the change refers to the resource performing the TDM Segment Generation function.

7.3.2 For a TD-CSTS operating in complete mode, the definition of the `event-value` of the `tdSvcProductionStatusChange` event shall be refined to mean that the change refers to the resource performing the TDM Recording Buffer function.

ANNEX A

IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of the *Cross Support Transfer Services – Tracking Data Service*, CCSDS 922.2-B-1. CCSDS 922.2-B-1 specifies the requirements on the provider of the Tracking Data Cross Support Transfer Service.

The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements reference in the RL.

The RL support column in this annex is blank. An implementation's completed RL is called the PICS. The PICS states which capabilities and options have been implemented. The following can use the PICS:

a) the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);

d) a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

<u>Item Column</u>

The item column contains a prefix identifying the element the given table is referring to and sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means 'Is this feature supported by the implementation?'

Status Column

The status column uses the following notations:

a) M mandatory;

b) O optional;

c) O.<n> optional, but support of at least one of the group of options labeled by the same numeral <n> is required;

d) C<n> conditional as defined in corresponding expression below the table;

e) X prohibited;

f) N/A not applicable.

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

a) Y Yes, supported by the implementation;

b) N No, not supported by the implementation;

c) N/A Not applicable.

The support column should also be used, when appropriate, to enter values supported for a given capability.

Allowed Values Column

All PDU parameter types are specified in annex F of reference [1] or in this Recommended Standard using ASN.1. The ASN.1 data type specifications constrain among others the permissible value range and therefore such constraints are not repeated in the Allowed Values column in the tables contained in this ICS annex. However, if a parameter is constrained for all instances of the given PDU to a subset of the range or set specified for that parameter type, then the subset is identified in the tables that contain PDU parameters.

Allowed Values Column Symbols

If the allowed values are too large to fit in the Allowed Values cell, the Allowed Values column uses the notation 'AV<n>' as an indication that the allowed values are specified below the table.

Supported Values Column

The Supported Values column is to be used by the implementer to state whether the specified range or set of values for the parameter is supported by entering Y or SV<n>, indicating:

a) Y — Yes, the range/set defined in the Recommended Standard is fully supported by the implementation;
b) SV<n> — The range/set defined in the Recommended Standard is not fully supported by the implementation. The supported subset is documented below the table.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called a PICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 PICS PROFORMA FOR THE TRACKING DATA CSTS PROTOCOL (CCSDS 922.2-B-1)

A2.1 GENERAL INFORMATION

The PICS for a TD-CSTS implementation shall encompass the filled in tables A-1 to A-4.

Table A-1: Identification of PICS

Date of Statement (DD/MM/YYYY)	
PICS serial number	
System Conformance statement cross-reference	

Table A-2: Identification of Implementation Under Test

Implementation name	
Implementation version	
Special Configuration	
Other Information	

Table A-3: Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	

Table A-4: Identification of Specification

CCSDS 922.2-B-1		
Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is nonconforming.	Yes [ ]	No [ ]

A2.2 REQUIREMENTS LIST

This subsection provides the Requirement Lists for the elements specified in this Recommended Standard.

Table A-5: Required Procedures

Procedures				
Item	**Description**	**Reference**	**Status**	**Support**
proc-1	Association Control	Subsection 4.3 of reference [1]	M	
proc-2	Buffered Data Delivery	Subsection 4.5 of reference [1]	M	
proc-3	Buffered Tracking Data Message Delivery	Subsection 4 of this Recommended Standard	M	
proc-4	Information Query	Subsection 4.9 of reference [1]	M	

The Buffered Data Delivery procedure is mandatory in the sense that the Buffered Tracking Data Message Delivery procedure (which is mandatory) is derived from the Buffered Data Delivery procedure. In this TD-CSTS ICS, all requirements for the Buffered Data Delivery procedure are covered by the requirements for the Buffered Tracking Data Message Delivery procedure.

Table A-6: Required PDUs

Item	PDU	Reference	Service-Provider-System		Service-User-System	
			Status	Support	Status	Support
pdu-1	BindInvocation	Subsection F4.5 of reference [1]	M		M	
pdu-2	BindReturn	Subsection F4.5 of reference [1]	M		M	
pdu-3	PeerAbortInvocation	Subsection F4.5 of reference [1]	M		M	
pdu-4	UnbindInvocation	Subsection F4.5 of reference [1]	M		M	
pdu-5	UnbindReturn	Subsection F4.5 of reference [1]	M		M	
pdu-6	GetInvocation	Subsection F4.4 of reference [1]	M		M	
pdu-7	GetReturn	Subsection F4.4 of reference [1]	M		M	
pdu-8	NotifyInvocation	Subsection F4.4 of reference [1]	M		M	
pdu-9	StartInvocation	Subsection F4.4 of reference [1]	M		M	
pdu-10	StartReturn	Subsection F4.4 of reference [1]	M		M	
pdu-11	StopInvocation	Subsection F4.4 of reference [1]	M		M	
pdu-12	StopReturn	Subsection F4.4 of reference [1]	M		M	
pdu-13	TransferDataInvocation	Subsection F4.4 of reference [1]	M		M	
pdu-14	ReturnBuffer	Subsection F4.7 of reference [1]	M		M	

Table A-7: BIND Invocation Parameters

Parameters of the BindInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
bindInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
bindInv-2	invokeId	Subsection F4.3 of reference [1]	M			
bindInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV1	
bindInv-4	initiatorIdentifier	Subsection F4.5 of reference [1]	M			
bindInv-5	responderPortIdentifier	Subsection F4.5 of reference [1]	M			
bindInv-6	serviceType	Subsection F4.5 of reference [1]	M			
bindInv-7	versionNumber	Subsection F4.5 of reference [1]	M			
bindInv-8	serviceInstanceIdentifier	Subsection F4.5 of reference [1]	M			
bindInv-9	bindInvocationExtension	Subsection F4.5 of reference [1]	M		‘notUsed’	

AV1 For the BIND invocation the procedureRole element of the parameter bindInv-3 must be set to ‘associationControl’.

The parameters bindInv-1, bindInv-2, and bindInv-3 are contained in the complex parameter standardInvocationHeader shown in F4.5 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of reference [1].

Table A-8: BIND Return Parameters

Parameters of the BindReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
bindRet-1	performerCredentials	Subsection F4.3 of reference [1]	M			
bindRet-2	invokeId	Subsection F4.3 of reference [1]	M			
bindRet-3	result	Subsection F4.3 of reference [1]	M			
bindRet-4	positive	Subsection F4.3 of reference [1]	C1		'notUsed'	
bindRet-5	diagnostics	Subsection F4.3 of reference [1]	C2		AV2	
bindRet-6	negExtension	Subsection F4.3 of reference [1]	C2		'notUsed'	
bindRet-7	responderIdentifier	Subsection F4.5 of reference [1]	M			

C1 IF bindRet-3 = 'positive' THEN M ELSE X

C2 IF bindRet-3 = 'negative' THEN M ELSE X

AV2 For the negative BIND return the parameter bindRet-5 is extended by the type AssocBindDiagnosticExt defined in F4.5 of reference [1]. Therefore the parameter bindRet-5 may have (a) any value defined for the `Diagnostic` type in F4.3 of reference [1] except 'diagnosticExtension'; or (b) any value defined by 'diagnosticExtension': 'acBindDiagExt': 'AssocBindDiagnosticExt' defined in F4.5 of reference [1] except 'assocBindDiagnosticExtExtension'.

All parameters of the BIND return PDU except bindRet-7 are contained the complex parameter of the type StandardReturnHeader that is specified in F4.3 of reference [1]. Specific extensions are, however, specified in F4.5 of that document.

Table A-9: PEER-ABORT Invocation Parameters

Parameters of the PeerAbortInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
peerAbortInv-1	diagnostic	Subsection F4.5 of reference [1]	M		40..126	

Table A-10: UNBIND Invocation Parameters

Parameters of the UnbindInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
unbindInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
unbindInv-2	invokeId	Subsection F4.3 of reference [1]	M			
unbindInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV3	
unbindInv-4	unbindInvocationExtension	Subsection F4.5 of reference [1]	M		'notUsed'	

AV3 For the UNBIND invocation the procedureRole element of the parameter unbindInv-3 must be set to 'associationControl'.

The parameters unbindInv-1, unbindInv-2 and unbindInv-3 are contained in the complex parameter standardInvocationHeader shown in F4.5 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

Table A-11: UNBIND Return Parameters

Parameters of the UnbindReturn PDU						
					Values	
Item	Parameter	Reference	Status	Support	Allowed	Supported
unbindRet-1	performerCredentials	Subsection F4.3 of reference [1]	M			
unbindRet-2	invokeId	Subsection F4.3 of reference [1]	M			
unbindRet-3	result	Subsection F4.3 of reference [1]	M		AV4	

AV4 The value of the parameter unbindRet-3 shall always be set to ‘positive’: ‘notUsed’; i.e., the result is always positive and not extended.

All parameters of the UNBIND return PDU are contained the complex parameter of the type StandardReturnHeader that is specified in F4.3 of reference [1].

Table A-12: GET Invocation Parameters

Parameters of the GetInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
getInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
getInv-2	invokeId	Subsection F4.3 of reference [1]	M			
getInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV5	
getInv-4	listOfParameters	Subsection F4.4 of reference [1]	M			
getInv-5	getInvocationExtension	Subsection F4.4 of reference [1]	M		‘notUsed’	

AV5 The value of the procedureRole element of the parameter getInv-3 must be set to ‘secondary procedure’.

The parameters getInv-1, getInv-2 and getInv-3 contained in the complex parameter standardInvocationHeader shown in F4.4 reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

Table A-13: GET Return Parameters

Parameters of the GetReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
getRet-1	performerCredentials	Subsection F4.3 of reference [1]	M			
getRet-2	invokeId	Subsection F4.3 of reference [1]	M			
getRet-3	result	Subsection F4.3 of reference [1]	M			
getRet-4	positive	Subsection F4.3 of reference [1]	C3		AV6	
getRet-5	qualifiedParameters	Subsection F4.4 of reference [1]	C3		AV7	
getRet-6	getPosReturnExtExtension	Subsection F4.4 of reference [1]	C3		'notUsed'	
getRet-7	diagnostics	Subsection F4.3 of reference [1]	C4		AV8	
getRet-8	negExtension	Subsection F4.3 of reference [1]	C4		'notUsed'	

C3 IF getRet-3 = 'positive' THEN M ELSE X

C4 IF getRet-3 = 'negative' THEN M ELSE X

AV6 For the positive GET return the parameter getRet-4 is set to 'getPosReturnExt': 'GetPosReturnExt' defined in F4.4 of reference [1].

AV7 For the positive GET return the parameter getRet-5 is specified by 'qualifiedParameters': 'QualifiedParametersSequence'. The type QualifiedParametersSequence is defined in F4.4 of reference [1].

AV8 For the negative GET return the parameter getRet-7 is extended by the type GetDiagnosticExt defined in F4.4 of reference [1]. Therefore the parameter getRet-7 may have (a) any standard value defined for the Diagnostic type in F4.3 of that document except 'diagnosticExtension'; or (b) any value defined by the extension 'diagnosticExtension': 'getDiagnosticExt': 'GetDiagnosticExt' defined in F4.4 of that document except 'getDiagnosticExtExtension'.

All parameters of the GET return PDU are contained the complex parameter of the type StandardReturnHeader that is specified in F4.3 of reference [1]. Specific extensions are, however, specified in F4.4 of that document.

Table A-14: START Invocation Parameters

Parameters of the StartInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
startInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
startInv-2	invokeId	Subsection F4.3 of reference [1]	M			
startInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV9	
startInv-4	startInvocationExtension	Subsection F4.4 of reference [1]	M		AV10	
startInv-5	startGenerationTime	Subsection F4.7 of reference [1]	M			
startInv-6	stopGenerationTime	Subsection F4.7 of reference [1]	M			
startInv-7	buffDataDelStartInvocExt Extension	Subsection F4.7 of reference [1]	M		‘notUsed’	

AV9 The value of the procedureRole element of the parameter startInv-3 must be set to ‘prime procedure’.

AV10 The parameter startInv-4 shall be set to the value ‘bddStartInvocExt’: ‘BuffDataDelStartInvocExt’.

The parameters startInv-1, startInv-2 and startInv-3 are contained in the complex parameter standardInvocationHeader shown in F4.4 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

Table A-15: START Return Parameters

Parameters of the StartReturn PDU						
					Values	
Item	Parameter	Reference	Status	Support	Allowed	Supported
startRet-1	performerCredentials	Subsection F4.3 of reference [1]	M			
startRet-2	invokeId	Subsection F4.3 of reference [1]	M			
startRet-3	result	Subsection F4.3 of reference [1]	M			
startRet-4	positive	Subsection F4.3 of reference [1]	C5		AV11	
startRet-5	diagnostics	Subsection F4.3 of reference [1]	C6		AV12	
startRet-6	negExtension	Subsection F4.3 of reference [1]	C6		'notUsed'	
startRet-7	buffTrkDataDelStartPos ReturnExtExtension	Annex D	C5		'notUsed'	

C5 IF startRet-3 = 'positive' THEN M ELSE X

C6 IF startRet-3 = 'negative' THEN M ELSE X

AV11 The parameter startRet-4 has the value 'buffTrkDataDelStartPosReturnExt': 'BuffTrkDataDelStartPosReturnExt', as defined in annex D.

AV12 For the START return PDU the parameter startRet-5 is extended by the type StartDiagnosticExt defined in F4.4 and BuffDataDelStartDiagnosticExt defined in F4.7 of reference [1]. Therefore the parameter startRet-5 may have (a) any standard value defined for the `Diagnostic` type in F4.3 of that document except 'diagnosticExtension'; (b) any value defined by the extension 'diagnosticExtension': 'startDiagnosticExt': 'StartDiagnosticExt' in F4.4 of that document except 'startDiagnosticExtExtension'; or (c) any value defined by the extension 'diagnosticExtension': 'startDiagnosticExt': 'StartDiagnosticExt': 'startDiagnosticExtExtension': bddStartDiagExt': 'BuffDataDelStartDiagnosticExt' in F4.7 of that document except 'buffDataDelStartDiagnosticExtExtension'.

Table A-16: STOP Invocation Parameters

Parameters of the StopInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
stopInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
stopInv-2	invokeId	Subsection F4.3 of reference [1]	M			
stopInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV13	
stopInv-4	stopInvocationExtension	Subsection F4.4 of reference [1]	M		'notUsed'	

AV13 The value of the procedureRole element of the parameter stopInv-3 must be set to 'prime procedure'.

The parameters stopInv-1, stopInv-2 and stopInv-3 are contained in the complex parameter standardInvocationHeader shown in F4.4 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

Table A-17: STOP Return Parameters

Parameters of the StopReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
stopRet-1	performerCredentials	Subsection F4.3 of reference [1]	M			
stopRet-2	invokeId	Subsection F4.3 of reference [1]	M			
stopRet-3	result	Subsection F4.3 of reference [1]	M			
stopRet-4	positive	Subsection F4.3 of reference [1]	C7		'notUsed'	
stopRet-5	diagnostics	Subsection F4.3 of reference [1]	C8		AV14	
stopRet-6	negExtension	Subsection F4.3 of reference [1]	C8		'notUsed'	

C7 IF stopRet-3 = 'positive' THEN M ELSE X

C8 IF stopRet-3 = 'negative' THEN M ELSE X

AV14 The parameter stopRet-5 may have any standard value defined for the `Diagnostic` type in F4.3 of reference [1] except 'diagnosticExtension'.

All parameters of the STOP return PDU are contained in the complex parameter of the type StandardReturnHeader that is specified in F4.3 of reference [1].

Table A-18: TRANSFER-DATA Invocation Parameters

Parameters of the TransferDataInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
transferDataInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
transferDataInv-2	invokeId	Subsection F4.3 of reference [1]	M			
transferDataInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV15	
transferDataInv-4	generationTime	Subsection F4.4 of reference [1]	M			
transferDataInv-5	sequenceCounter	Subsection F4.4 of reference [1]	M			
transferDataInv-6	data	Subsection F4.4 of reference [1]	M		AV16	
transferDataInv-7	transferDataInvocation Extension	Subsection F4.4 of reference [1]	M		‘notUsed’	

AV15 The value of the procedureRole element of the parameter transferDataInv-3 must be set to ‘prime procedure’.

AV16 The value of transferDataInv-6 is refined to be an octet string formatted as an atomic segment as defined in B3.

The parameters transferDataInv-1, transferDataInv-2 and transferDataInv-3 are contained in the complex parameter standardInvocationHeader shown in F4.4 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

Table A-19: NOTIFY Invocation Parameters

Parameters of the NotifyInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
notifyInv-1	invokerCredentials	Subsection F4.3 of reference [1]	M			
notifyInv-2	invokeId	Subsection F4.3 of reference [1]	M			
notifyInv-3	procedureInstanceId	Subsection F4.3 of reference [1]	M		AV17	
notifyInv-4	eventTime	Subsection F4.4 of reference [1]	M			
notifyInv-5	eventName	Subsection F4.4 of reference [1]	M			
notifyInv-6	eventValue	Subsection F4.4 of reference [1]	M		AV18	
notifyInv-7	notifyInvocationExtension	Subsection F4.4 of reference [1]	M		'notUsed'	

AV17 The value of the procedureRole element of the parameter notifyInv-3 must be set to 'prime procedure'.

AV18 The value of the notifyInv-6 parameter can be any value that can be expressed using the type SequenceOfQualifiedValues defined in F4.3 of reference [1] or 'empty'. The value of 'eventValue' must not be set to 'eventValueExtension'.

The parameters notifyInv-1, notifyInv-2 and notifyInv-3 are contained in the complex parameter standardInvocationHeader in the NotifyInvocation type shown in F4.4 of reference [1]. This parameter is of the type StandardInvocationHeader that is specified in F4.3 of that document.

ANNEX B

SPECIFICATION OF TD-CSTS TDM COMPONENTS

(NORMATIVE)

B1 GENERAL

This section defines the contents of the TDM Header and TDM atomic segment.

B2 TDM HEADER

The format and semantic content of the TDM Header shall conform to the TDM Header as specified in 3.2 of reference [3]. Figure B-1 is an example of the TDM Header. (Refer to 3.2.3 of reference [3] for the definitions of the TDM Header keywords.)

```
CCSDS_TDM_VERS = 1.0

CREATION_DATE = 2007-075T11:42:28.000
ORIGINATOR = JPL
```

Figure B-1: Example TDM Header

B3 TDM ATOMIC SEGMENT

B3.1 The TDM atomic segment shall comprise a TDM Metadata Section, as specified in 3.3 of reference [3], followed by a TDM Data Section, as defined in 3.4 and 3.5 of reference [3], with the constraint that a TDM Data Section shall contain either:

a) a single Tracking Data Record for one of the following tracking data types:

   1) carrier power;

   2) carrier power to noise spectral density;

   3) Doppler (instantaneous);

   4) Doppler (integrated);

   5) range;

   6) ranging power to noise spectral density;

   7) receive frequency;

8) transmit frequency;

9) transmit frequency rate; or

b) a pair of antenna angle Tracking Data Records.

B3.2 The format and semantic content of the TDM Metadata Section of the TDM atomic segment shall conform to those specified in 3.3 of reference [3].

B3.3 The format and semantic content of the Tracking Data Records of the TDM atomic segment shall conform to those specified in 3.4 and 3.5 of reference [3].

NOTE – Figure B-2 contains TDM atomic segments for the TDM with the TDM Header presented in figure B-1. In this example, the transmit frequency and the transmit frequency rate are measured every 5 seconds, while the range is measured every 10 seconds. The lines separating the segments are not part of either segment. (Refer to 3.3 and 3.5 of reference [3] for the definitions of the TDM Metadata and TDM Data keywords.)

```
META_START
TIME_SYSTEM          = UTC
START_TIME           = 2007-075T11:50:43.000
STOP_TIME            = 2007-075T11:50:43.000
PARTICIPANT_1        = DSS-26
MODE                 = SEQUENTIAL
TRANSMIT_BAND        = X
META_STOP

DATA_START
TRANSMIT_FREQ_1        = 2007-075T11:50:43.000   7175510611.700343
DATA_STOP
```

```
META_START
TIME_SYSTEM          = UTC
START_TIME           = 2007-075T11:50:43.000
STOP_TIME            = 2007-075T11:50:43.000
PARTICIPANT_1        = DSS-26
MODE                 = SEQUENTIAL
TRANSMIT_BAND        = X
META_STOP

DATA_START
TRANSMIT_FREQ_RATE_1   = 2007-075T11:50:43.000   0.0
DATA_STOP
```

Figure B-2: Example Sequence of TDM Atomic Segments

```
META_START
TIME_SYSTEM           = UTC
START_TIME            = 2007-075T11:50:48.000
STOP_TIME             = 2007-075T11:50:48.000
PARTICIPANT_1         = DSS-26
MODE                  = SEQUENTIAL
TRANSMIT_BAND         = X
META_STOP

DATA_START
TRANSMIT_FREQ_1         = 2007-075T11:50:48.000   7175510611.700343
DATA_STOP
```

```
META_START
TIME_SYSTEM           = UTC
START_TIME            = 2007-075T11:50:48.000
STOP_TIME             = 2007-075T11:50:48.000
PARTICIPANT_1         = DSS-26
MODE                  = SEQUENTIAL
TRANSMIT_BAND         = X
META_STOP

DATA_START
TRANSMIT_FREQ_RATE_1    = 2007-075T11:50:48.000   0.0
DATA_STOP
```

```
META_START
TIME_SYSTEM = UTC
START_TIME            = 2007-075T11:50:43.000
STOP_TIME             = 2007-075T11:50:53.000
PARTICIPANT_1 = DSS-26
PARTICIPANT_2 = XENOSAT
MODE = SEQUENTIAL
PATH = 1,2,1
INTEGRATION_REF = START
RANGE_MODE = COHERENT
RANGE_MODULUS = 2.0e+26
RANGE_UNITS = km
META_STOP

DATA_START
RANGE = 2007-075T11:50:43.000   80452.7025
DATA_STOP
```

Figure B-2: Example Sequence of TDM Atomic Segments (continued)

```
META_START
TIME_SYSTEM          = UTC
START_TIME           = 2007-075T11:50:53.000
STOP_TIME            = 2007-075T11:50:53.000
PARTICIPANT_1        = DSS-26
MODE                 = SEQUENTIAL
TRANSMIT_BAND        = X
META_STOP

DATA_START
TRANSMIT_FREQ_1        = 2007-075T11:50:53.000   7175510611.700343
DATA_STOP
```

```
META_START
TIME_SYSTEM          = UTC
START_TIME           = 2007-075T11:50:53.000
STOP_TIME            = 2007-075T11:50:53.000
PARTICIPANT_1        = DSS-26
MODE                 = SEQUENTIAL
TRANSMIT_BAND        = X
META_STOP

DATA_START
TRANSMIT_FREQ_RATE_1   = 2007-075T11:50:53.000   0.0
DATA_STOP
```

Figure B-2: Example Sequence of TDM Atomic Segments (continued)

B4 DISCUSSION: ILLUSTRATIONS OF NOMINAL TD-CSTS DATA TRANSFORMATIONS

B4.1 OVERVIEW

This subsection provides illustrated examples of TD-CSTS data transformations in three cases: (1) a TD-CSTS instance operating in real-time delivery mode, with user-specified `start-generation-time` and `stop-generation-time` parameters, (2) a TD-CSTS instance operating in real-time delivery mode, with no specified `start-generation-time` and `stop-generation-time` parameters, and (3) a TD-CSTS instance operating in complete data delivery mode.

B4.2 REAL-TIME DELIVERY MODE TD-CSTS WITH USER-SPECIFIED `start-generation-time` AND `stop-Generation-Time` PARAMETERS

Figure B-3 illustrates some of the data transformations from the TDM Segment Generation Function, through the TD-CSTS operations to the resulting Tracking Data Message document for this case.

The production of atomic segments by the TDM Segment Generation function starts at the beginning of tracking services, which, as illustrated in the figure, may precede the START of a TD-CSTS instance. In this example, the START invocation specifies a `start-`

generation-time that occurs sometime after the START operation is invoked. Therefore the TD-CSTS instance sends the START return and then waits until the specified start-generation-time before beginning to transfer atomic segments produced as of start-generation-time specified in the START operation and ignores any that occurred before.

NOTE – In this example case, explicit values are provided in the START invocation for both the start-generation-time and the stop-generation-time, but in general whether or not a parameter contains a specified time value is independent of the setting of the other.

The TD-CSTS instance transfers only those atomic segments that each contain one of the tracking data types selected as part of the managed configuration of the TD-CSTS instance. As shown in the figure, atomic segments that do not meet the selection criteria are ignored by the TD-CSTS instance and are not included in the TDM that is ultimately created from the atomic segments that are transferred by the TD-CSTS instance.

The TD-CSTS instance may also generate tdSvcProductionStatusChange event notifications if the production status – which for real-time TD service instances is defined as the status of the TDM Segment Generation function – changes. Figure B-3 illustrates the generation of two such notifications, a tdSvcProductionStatusChange event with value ‘interrupted’ followed (after some time) by a tdSvcProductionStatusChange notification with event-value ‘operational’. As defined in reference [1], these notifications are *discardable*, meaning that a TD-CSTS in real-time data delivery mode may not deliver these notifications if the connection experiences backpressure (a TD-CSTS in complete delivery mode delivers all notifications, whether they are discardable or non-discardable). However, even when these notifications are transferred, they are not included in the resulting TDM – they are provided to the TD-CSTS User entity for purposes of monitoring the execution of the TD-CSTS. In real-time mode, TDM Segment Generation generates atomic segments only when it is operational; the gap in between segments M+7 and N represents lost tracking data.

A TD-CSTS provider instance may also inject notifications regarding the execution of the transfer service itself (e.g., ‘data discarded due to excessive backlog’) into the transfer. The example shows the discarding of TRANSFER-DATA invocations due to backpressure. Once the backpressure clears tracking data begins to flow again, and the notification ‘data discarded due to excessive backlog’ is also sent. As with the production-generated notifications, these provider-generated notifications do not affect the content of the resultant TDM.

Finally, when the user-specified stop-generation-time is reached, the TD service provider injects an ‘end of data’ notification. The ‘end of data’ notification (which is non-discardable) is used by the TD-CSTS user to identify the end of the TDM.

The net result is a TDM with atomic segments for the time from the start-generation-time through the stop-generation-time, minus segments corresponding to the period of time where the production of tracking data was interrupted and the period of time during which network backpressure caused the TD service to discard segments.

B4.3 REAL-TIME DELIVERY MODE TD-CSTS WITH UNSPECIFIED STOP-GENERATION-TIME

Figure B-4 illustrates some of the data transformations from Tracking Data Production, through the TD-CSTS operations to the resulting Tracking Data Message document for the case where both the `start-generation-time` and the `stop-generation-time` are left unspecified in the START invocation. Figure B-4 is identical to figure B-3 except for two differences. The first difference is that at the beginning of the TD service, the transfer of atomic segments begins as soon as possible after the START operation is performed. The second difference is that the user signals the end of the TDM by sending the STOP invocation.

The net result is a TDM with atomic segments for the time from the performance of the START operation through the performance of the STOP operation, minus the segments corresponding to the period of time where the production of tracking data was interrupted and the period of time during which network backpressure caused the TD service to discard segments.

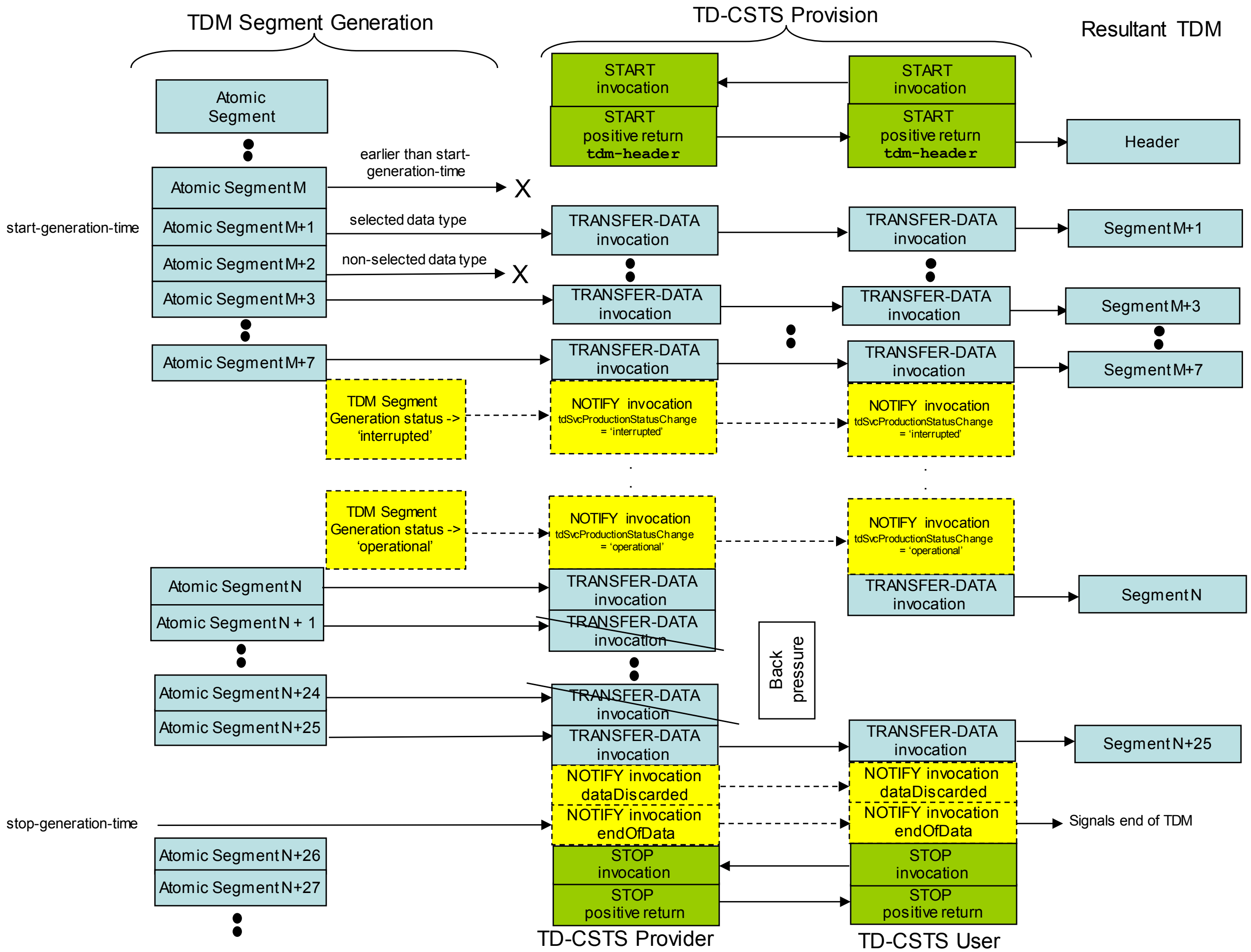

Figure B-3: TDM Segment Generation, TD-CSTS Operations, and Tracking Data Message Document for the Real-Time Delivery Mode with User-Specified Start and Stop Generation Times

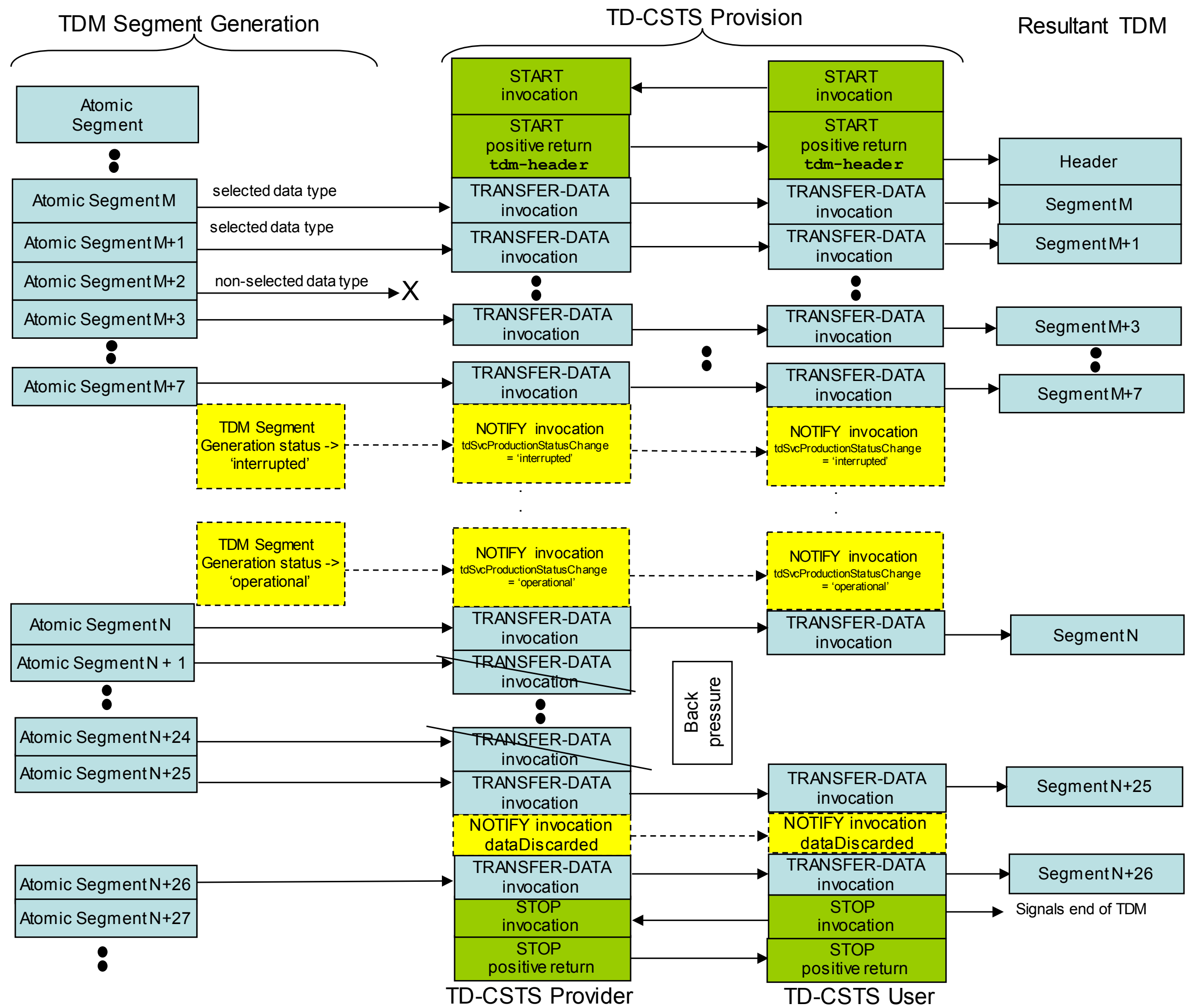

Figure B-4: TDM Segment Generation, TD-CSTS Operations, and Tracking Data Message Document for the Real-Time Delivery Mode with Unspecified Start and Stop Generation Times

B4.4 COMPLETE DELIVERY MODE TD-CSTS

Figure B-5 illustrates some of the data transformations from Tracking Data Production, through the TD-CSTS operations to the resulting Tracking Data Message document for the complete delivery mode case. In the complete delivery mode, both the `start-generation-time` and the `stop-generation-time` must be specified by the user in the START invocation.

The first difference between figure B-5 and the previous two figures B-3 and B-4 is that the Tracking Data Production functionality that is ‘seen’ by the TD service instance is the Tracking Data Recording Buffer (instead of TDM Segment Generation), which may hold up to several weeks or more worth of tracking data. The `start-generation-time` parameter of the TD-CSTS START invocation can contain any time before the end of the service instance provision period of the TD-CSTS instance. If the `start-generation-time` is before the time that the START operation is performed, then once the START return is sent the selected tracking data types and any production-related notifications that are stored in the Recording Buffer for the requested period are transferred by the service.

The second difference is that in complete mode, the service production status is determined by the status of the TDM Recording Buffer. In this example, when the status of the TDM Recording Buffer is temporarily interrupted, transfer of atomic segments is suspended. However, as soon as the TDM Recording Buffer becomes operational again, the transfer of atomic segments resumes with the segment immediately following the one that was previously sent.

The third, and most important, difference is that unlike the real-time delivery mode, if network backpressure is encountered the service continues to attempt to send all requested tracking data. When a tracking data segment or notification with a timestamp later than the specified `stop-generation-time` is encountered in the Recording Buffer, the TD service provider transfers an ‘end of data’ notification, which signals the TD-CSTS user that there are no more data in the requested timespan, and therefore the TDM can be closed.

The net result is a TDM with atomic segments for the time from the `start-generation-time` through the `stop-generation-time`, minus segments corresponding to any period of time where the production of tracking data was interrupted, i.e. the resource status of the TDM Segment Generation function was ‘interrupted’ and therefore those segments did not get forwarded to the TDM Recording Buffer function. However, no TDM segments will be lost if the resource status of the TDM Recording Buffer temporarily changes to ‘interrupted’ while TDM segments are being retrieved from it. Rather, the transfer of TDM segments to the TD-CSTS user is resumed with the TDM segment following the segment that had been transferred before the TDM Recording Buffer resource status changed to ‘interrupted’.

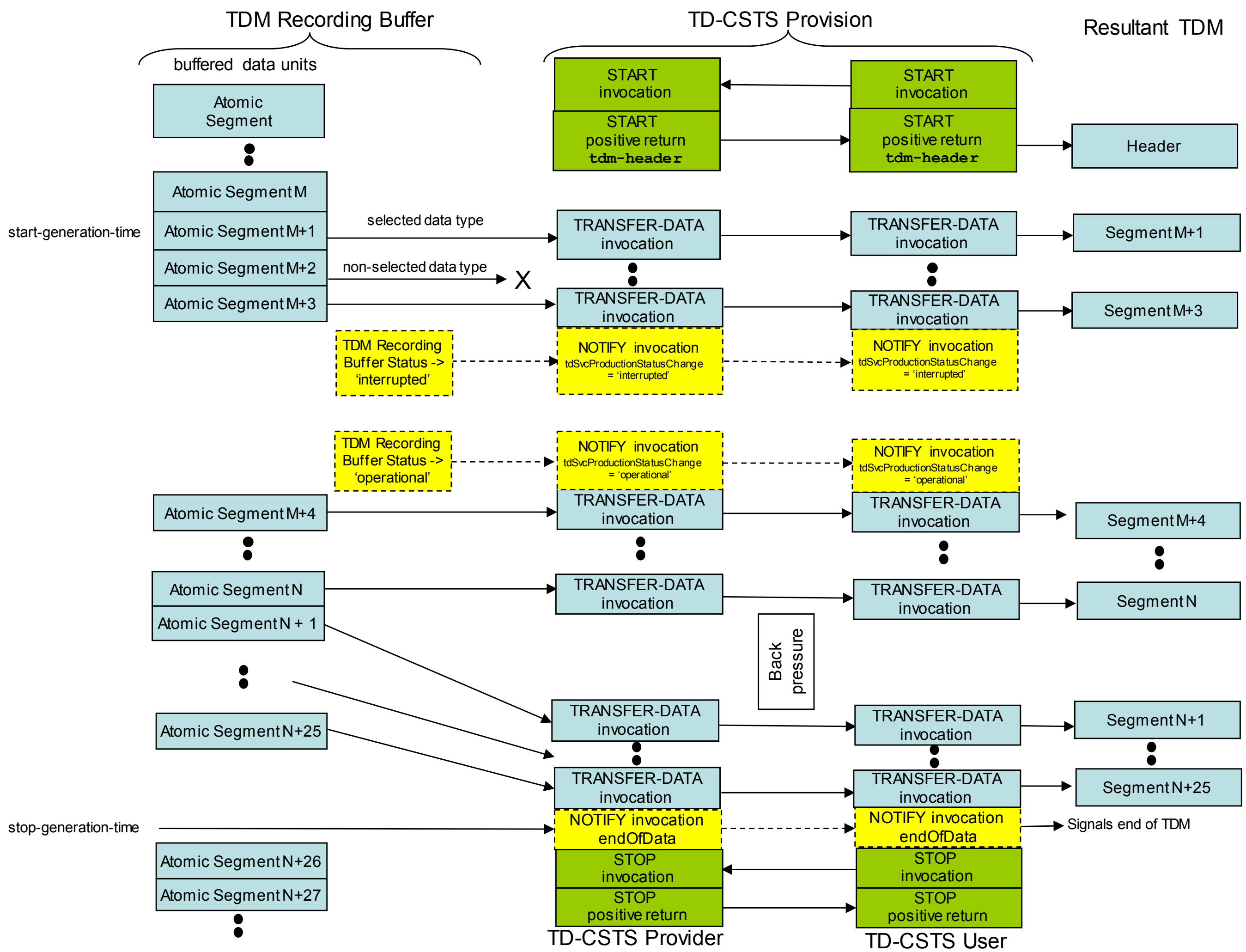

Figure B-5: TDM Recording Buffer, TD-CSTS Operations, and Tracking Data Message Document for the Complete Delivery Mode

ANNEX C

SERVICE OBJECT IDENTIFIERS MODULE

(NORMATIVE)

```
CCSDS-TRACKING-DATA-OBJECT-IDENTIFIERS
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) trackingDataService(2)
   trackingDataServiceModules(4) object-identifiers(1) version(1)
}

DEFINITIONS

IMPLICIT TAGS
::= BEGIN

EXPORTS  buffTrkDataMsgDelExtendedOpsParam
,        tdCstsProvider
,        tdCstsProviderDirectivesId
,        tdCstsProviderEventsId
,        tdCstsProviderParametersId
,        trackingDataDerivedServices
,        trackingDataExtendedServiceParameters
,        trackingDataServiceProcedures
;

IMPORTS  services
,        crossSupportFunctionalities
   FROM CCSDS-CSTS-OBJECT-IDENTIFIERS

         BufferSize
   FROM CCSDS-CSTS-COMMON-TYPES
;

-- The CCSDS-CSTS-OBJECT-IDENTIFIERS and CCSDS-CSTS-COMMON-TYPES modules
-- are defined in F4.1 and F4.3 of reference [1].

-- **********************************************************
-- Root Object Identifiers of the Service

trackingDataService  OBJECT IDENTIFIER ::= {services 2}
trackingDataDerivedServices   OBJECT IDENTIFIER ::=
                                             {trackingDataService 1}
trackingDataExtendedServiceParameters OBJECT IDENTIFIER ::=
{trackingDataService 2}
trackingDataServiceProcedures OBJECT IDENTIFIER ::=
                                                {trackingDataService 3}
trackingDataServiceModules OBJECT IDENTIFIER ::= {trackingDataService 4}
```

```
-- *******************************************************
-- Procedure Type Identifier:
buffTrkDataMsgDel    OBJECT IDENTIFIER ::=
                                    {trackingDataServiceProcedures 1}

buffTrkDataMsgDelExtendedOpsParam   OBJECT IDENTIFIER ::=
                                     {buffTrkDataMsgDel     1}
buffTrkDataMsgDelExtendedProcParam  OBJECT IDENTIFIER ::=
                                     {buffTrkDataMsgDel     2}
buffTrkDataMsgDelEventsId           OBJECT IDENTIFIER ::=
                                     {buffTrkDataMsgDel     3}
buffTrkDataMsgDelDirectivesId       OBJECT IDENTIFIER ::=
                                     {buffTrkDataMsgDel     4}

-- *******************************************************
-- Root Object Identifiers of the TD-CSTS Provider Functional
-- Resource Type

tdCstsProvider    OBJECT IDENTIFIER ::=
                                 {crossSupportFunctionalities 19}
tdCstsProviderParametersId OBJECT IDENTIFIER ::=      {tdCstsProvider 1}
tdCstsProviderEventsId     OBJECT IDENTIFIER ::=      {tdCstsProvider 2}
tdCstsProviderDirectivesId OBJECT IDENTIFIER ::=      {tdCstsProvider 3}

-- *******************************************************
-- Object Identifiers of procedures specified in reference [1] used by the
-- Tracking Data service
-- The Information Query procedures may be used to request the current
-- values of procedure configuration parameters with the OIDs
--    pACserviceUserRespTimer
--    pACinitiatorId
--    pACresponderId
--    pACresponderPortId
--    pACserviceInstanceId
--    pBDDreturnBufferSize
--    pBDDdeliveryMode
--    pIQlistNames
-- as defined in the CCSDS-CSTS-FW-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES
-- module in F4.16 of reference [1].

END
```

ANNEX D

PROCEDURE – BUFFERED TRACKING DATA MESSAGE DATA DELIVERY PDUS

(NORMATIVE)

D1 TRACKING DATA BUFFERED-TRACKING-DATA-MESSAGE-DELIVERY-PDUS MODULE

```
CCSDS-TRACKING-DATA-BUFFERED-TRACKING-DATA-MESSAGE-DELIVERY-PDUS
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) trackingDataServiceModules(4)
   extensions(2) buffTrkDataMsgDelPdus(1)
   version(1)
}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

EXPORTS  BufferedTrackingDataMessageDeliveryPdu

IMPORTS  BufferedDataDeliveryPdu
   FROM CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS
-- CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS is defined in F4.7 of reference
-- [1]).
         Extended
   FROM CCSDS-CSTS-COMMON-TYPES
         buffTrkDataMsgDelExtendedOpsParam
   FROM CCSDS-TRACKING-DATA-OBJECT-IDENTIFIERS
;

-- ===============================================================
-- The Buffered Tracking Data Message Delivery procedure is derived from
-- the Buffered Data Delivery procedure. It reuses the PDU defined in the
-- Buffered Data Delivery procedure: BufferedDataDeliveryPdu type defined
-- in the CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS module in F4.7 of
-- reference [1].

BufferedTrackingDataMessageDeliveryPdu  ::=  BufferedDataDeliveryPdu

-- ===============================================================
-- *****
-- START Invocation extension parameters
-- No extension parameters are added to the START Invocation of the
-- Buffered Data Delivery procedure. Therefore 'StartInvocation':
-- 'startInvocationExtension': 'bddStartInvocExt':
-- 'BuffDataDelStartInvocExt': 'buffDataDelStartInvocExtExtension' (refer
-- to the CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS module in F4.7 of
-- reference [1]) shall be set to 'notUsed'.
```

```
-- START positive return parameters
-- START positive return is extended with the tdm-header parameter. This
-- extension is defined by ‘StartReturn’: ‘StandardReturnHeader’: ‘result’:
-- ‘positive’: ‘BuffTrkDataDelStartPosReturnExt’. No further parameters are
-- added, i.e., ‘StartReturn’: ‘StandardReturnHeader’: ‘result’:
-- ‘positive’: ‘BuffTrkDataDelStartPosReturnExt’:
-- ‘buffTrkDataDelStartPosReturnExtExtension’ shall be set to ‘notUsed’.

BuffTrkDataDelStartPosReturnExt  ::=   SEQUENCE
{  tdmHeader                                 VisibleString
,  buffTrkDataDelStartPosReturnExtExtension  Extended
}

buffTrkDataDelStartPosReturnExt   OBJECT IDENTIFIER  ::=
                              {buffTrkDataMsgDelExtenendedOpsParam      1}

-- START negative return extension parameters
-- No extension parameters are added to the START negative return of the
-- BufferedDataDelivery procedure. Therefore ‘StartReturn’:
-- ‘StandardReturnHeader’: ‘result’: ‘negative’: ‘negExtension’ (refer to
-- the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F4.4 of reference [1])
-- shall be set to ‘notUsed’.

-- START negative return extension diagnostics
-- No extension diagnostics are added to the START negative return of the
-- BufferedDataDelivery procedure. Therefore the negative StartReturn is
-- returned using one of the common diagnostics of ‘StandardReturnHeader’:
-- ‘result’: ‘negative’: ‘diagnostic’: ‘Diagnostic’ (refer to 3.3.2.7 of
-- reference [1]) or one of the additional diagnostics defined by
-- ‘StartReturn’: ‘StandardReturnHeader’: ‘result’: ‘negative’:
-- ‘diagnostic’: ‘Diagnostic’: ‘diagnosticExtension’:
-- ‘startDiagnosticExt’: ‘StartDiagnosticExt’ or any of the additional
-- values defined by ‘StartReturn’: ‘StandardReturnHeader’: ‘result’:
-- ‘negative’: ‘diagnostic’: ‘Diagnostic’: ‘diagnosticExtension’:
-- ‘startDiagnosticExt’: ‘StartDiagnosticExt’:
-- ‘startDiagnosticExtExtension’: ‘bddStartDiagExt’:
-- ‘BuffDataDelStartDiagnosticExt’ except
-- buffDataDelStartDiagnosticExtExtension.

-- *****
-- STOP Invocation extension parameters
-- No extension parameters are added to the STOP Invocation of the
-- BufferedDataDelivery procedure. Therefore ‘StopInvocation’:
-- ‘stopInvocationExtension’(refer to the CCSDS-CSTS-COMMON-OPERATIONS-PDUS
-- module in F4.4 of reference [1]) shall be set to ‘notUsed’.

-- STOP positive return extension parameters
-- No extension parameters are added to the STOP positive return of the
-- BufferedDataDelivery procedure. Therefore ‘StopReturn’:
-- ‘StandardReturnHeader’: ‘result’: ‘positive’ (refer to the
-- CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F4.4 of reference [1]) shall
-- be set to ‘notUsed’.

-- STOP negative return extension parameters
-- No extension parameters are added to the STOP negative return of the
-- BufferedDataDelivery procedure. Therefore ‘StopReturn’:
-- ‘StandardReturnHeader’: ‘result’: ‘negative’: ‘negExtension’ (refer to
-- the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F4.4 of reference [1])
-- shall be set to ‘notUsed’.
```

```
-- STOP negative return extension diagnostics
-- The negative StopReturn is returned using one of the common diagnostics
-- of 'StandardReturnHeader': 'result': 'negative': 'diagnostic':
-- 'Diagnostic' (refer to 3.3.2.7 of reference [1]). No additional
-- diagnostics are specified, i.e., 'StopReturn': 'StandardReturnHeader':
-- 'result': 'negative': 'diagnostic': 'Diagnostic' must not be set to
-- 'diagnosticExtension'.

-- *****
-- TRANSFER-DATA Invocation extension parameters
-- No extension parameters are added to the TRANSFER-DATA Invocation of the
-- BufferedDataDelivery procedure. Therefore 'TransferDataInvocation':
-- transferDataInvocationExtension (refer to the
-- CCSDS-CSTS-COMMON- OPERATIONS-PDUS module in F4.4 of reference [1])
-- shall be set to 'notUsed'.

-- TRANSFER-DATA Invocation data parameter resolution
-- The data parameter of the TRANSFER-DATA Invocation is resolved as an
-- octet string. Therefore 'TransferDataInvocation': data':
-- 'AbstractChoice' (refer to the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module
-- in F4.4 of reference [1]) shall be set to 'opaqueString'.

-- *****
-- NOTIFY Invocation extension
-- No extension parameters are added to the NOTIFY Invocation of the
-- BufferedDataDelivery. Therefore 'NotifyInvocation':
-- 'notifyInvocationExtension' (refer to the
-- CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F4.4 of reference [1]) shall
-- be set to 'notUsed'.

END
```

D2 TRANSFER SYNTAX

The `BuffTrkDataDelStartPosReturnExt` type specified in this module shall be encoded for transfer using the Basic Encoding Rules specified in reference [4].

ANNEX E

TRACKING DATA SERVICE PROCEDURE PARAMETERS, EVENTS, AND DIRECTIVES

(NORMATIVE)

```
CCSDS-TRACKING-DATA-SERVICE-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) serviceIdentifiers(2)
   trackingDataService(2) trackingDataServiceModules(4)
   procedureParamEventDirective(3) version(1)
}

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

IMPORTS  buffTrkDataMsgDelExtendedOpsParam
FROM CCSDS-TRACKING-DATA-OBJECT-IDENTIFIERS
;

-- ============================================================
-- BUFFERED TRACKING DATA MESSAGE DELIVERY

pBTDMDtrackingDataTypes OBJECT IDENTIFIER ::-
                           {buffTrkDataMsgDelExtendedOpsParam     2}

PBTDMDtrackingDataTypesType ::= SET OF TrackingDataType

TrackingDataType  ::=   INTEGER
{  dopplerInstantaneous                (0)
,  dopplerIntegrated                   (1)
,  range                               (2)
,  carrierPower                        (3)
,  carrierPowerToNoiseSpectralDensity  (4)
,  rangingPowerToNoiseSpectralDensity  (5)
,  receiveFrequency                    (6)
,  transmitFrequency                   (7)
,  transmitFrequencyRate               (8)
,  antennaAngles                       (9)
}

END
```

ANNEX F

TRACKING DATA PRODUCTION

(NORMATIVE)

F1 GENERAL

F1.1 The production of tracking data for transfer via the TD-CSTS is performed by two functions: the TDM Segment Generation function and the TDM Recording Buffer function.

F1.2 The TDM Segment Generation function generates a time-ordered sequence of atomic segments as defined in B3, which correspond to Segments of the CCSDS Tracking Data Message specified in reference [3] with the additional constraint that each segment may contain only one tracking data measurement (except in the case of antenna angles, in which case the pair of angles is contained in the atomic segment).

F1.3 The TDM Recording Buffer function stores the atomic segments generated by the TDM Segment Generation function, for subsequent retrieval by TD-CSTS instances operating in the complete delivery mode.

F2 SPECIFICATION OF THE TDM SEGMENT GENERATION FUNCTION

F2.1 TDM SEGMENT GENERATION FUNCTION BEHAVIOR

F2.1.1 The TDM Segment Generation function shall collect from the radiometric data-generating Functional Resources of the Service Package the following tracking data measurements (as appropriate to the tracking activities being performed in the context of the Service Package):

– Angles;
– carrier power;
– carrier power to noise spectral density;
– Doppler (instantaneous);
– Doppler (integrated);
– range;
– ranging power to noise spectral density;
– receive frequency;
– transmit frequency; and
– transmit frequency rate.

(Refer to reference [3] for the definitions of these terms.)

F2.1.2 Each tracking data measurement shall be accompanied by the time at which it was measured at its respective Functional Resource.

NOTE – The periodicity at which the tracking data measurements are generated, and the degree to which Service Management may control that periodicity, may vary by Functional Resource instance and individual implementation and is outside the scope of this Recommended Standard.

F2.1.3 The TDM Segment Generation function shall format the tracking measurements into TDM Tracking Data Records.

F2.1.4 The TDM Segment Generation function shall combine the Tracking Data Records and their associated metadata into atomic segments, as defined in annex B.

F2.1.5 Each atomic segment shall constitute a Service Production Data Unit (refer to the Buffered Data Delivery procedure as specified in 4.5 of reference [1]).

F2.1.6 The TDM Segment Generation function shall make its current status available through the `tdmSegmentGenStatus` parameter, with values 'configured', 'operational', 'interrupted', and 'halted'.

F2.1.7 The TDM Segment Generation function shall emit a `tdmSegmentGenStatusChange` event for each change of status, with the event-value equal to the value of the `tdmSegmentGenStatus` parameter after the status change.

F2.2 TDM SEGMENT GENERATION FUNCTION SERVICE MANAGEMENT INFORMATION

For each TDM Segment Generation function instance, Service Management shall establish the relationships among each tracking signal path (including the names of the PARTICIPANTs (refer to 3.3.1 of reference [3]) involved with that signal path), the radiometric data-generating Functional Resources of the Service Package that constitute each signal path, and the TDM Metadata Keywords specified in 3.3 of reference [3], as necessary to form the TDM Metadata Sections of the atomic segments.

F2.3 TDM SEGMENT GENERATION FUNCTIONAL RESOURCE TYPE REGISTRATION DESCRIPTION

The TD-CSTS relies on the SANA Functional Resource Registry (reference [5]) to provide the identification and definition of Functional Resources and their parameters, events and directives.

Functional Resource Types are registered under the

```
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) crossSupportResources(2)
}
```

node of the OID registration tree.

Under the crossSupportResources node, there is the crossSupportFunctionalities subnode used to register CCSDS-standard Functional Resource Types. Under each Functional Resource Type OID, the parameters, events, and directives are registered under dedicated subnodes.

Maintenance of the SANA registry of the Functional Resource Types, parameters, events, and directives under the crossSupportFunctionalities subnode is under the purview of the CCSDS Cross Support Services Area in accordance with the process and procedures identified in the CSTS Specification Framework (reference [1]).

F2.3.1 The OID for the Functional Resource type that represents the TDM Segment Generation function is specified in the SANA Functional Resource Registry (reference [5]), using the FR classifier `tdmSegmentGen`.

F2.3.2 The `tdmSegmentGenStatus` parameter is registered under the `tdmSegmentGenParametersId ({tdmSegmentGen 1}` node of the SANA FR registry.

F2.3.3 The `tdmSegmentGenStatusChange` event is registered under the `tdmSegmentGenEventsId ({tdmSegmentGen 2}` node of the SANA FR registry.

F2.3.4 Any other monitored parameters, notifiable events, or directives that may be defined for the TDM Segment Generation FR type will be registered with SANA under the `tdmSegmentGenParametersId`, `tdmSegGenEventsId`, or `tdmSegGenDirectivesId ({tdmSegmentGen 3})` nodes, respectively, of the SANA FR registry.

F3 SPECIFICATION OF THE TDM RECORDING BUFFER FUNCTION

F3.1 TDM RECORDING BUFFER FUNCTION BEHAVIOR

F3.1.1 The functionality of the TDM Recording Buffer shall be as that specified for Recording Buffer in 4.5.7 of reference [1].

F3.1.2 The Service Production Data Units recorded by the TDM Recording Buffer function shall be the atomic segments generated by the TDM Segment Generation function (F2).

F3.1.3 The TDM Recording Buffer function shall make its current status available through the `tdmRecordingBufferStatus` parameter, with values ‘configured’, ‘operational’, interrupted’, and ‘halted’.

F3.1.4 The TDM Recording Buffer function shall generate a `tdmRecordingBufferStatusChange` event for each change of status, with the `event-value` equal to the value of the `tdmRecordingBufferStatus` parameter after the status change.

F3.1.5 The TDM Recording Buffer shall make the size of the recording buffer available through the `tdmRecordingBufferSize` parameter, which conforms to the specification of the queriable `recording-buffer-size` parameter defined in 4.5.7.10 of reference [1].

F3.1.6 In accordance with 4.5.7.5 b) of reference [1], if the recording buffer overflows, the TDM Recording Buffer FR type shall emit a `tdmFrRecordingBufferOverflow` event which conforms to the specification of the 'fr recording buffer overflow' event. The `event-value` for the `tdmFrRecordingBufferOverflow` event shall be empty.

F3.2 TDM RECORDING BUFFER FUNCTION SERVICE MANAGEMENT INFORMATION

NOTE – Reference [1] states in 4.5.7.5, "time span over which data is retained in the recording buffer, the policy for deleting data from the recording buffer, and the conditions under which the recording buffer begins to accept data following an overflow condition are outside the scope of" reference [1]. They are also outside the scope of this Recommended Standard.

F3.3 TDM RECORDING BUFFER FUNCTIONAL RESOURCE TYPE REGISTRATION DESCRIPTION

F3.3.1 The OID for the Functional Resource type that represents the TDM Recording Buffer function is specified in the SANA Functional Resource Registry (reference [5]), using the FR classifier `tdmRecordingBuffer.`

F3.3.2 The tdmRecordingBufferStatus and tdmRecordingBufferSize parameters are registered under the tdmRecordingBufferParametersId (`{tdmRecordingBuffer 1}`) node of the SANA FR registry.

F3.3.3 The tdmRecordingBufferStatusChange and tdmFrRecordingBufferOverflow events are registered under the tdmSegmentGeneventsId (`{tdmSegmentGen 2}`) node of the SANA FR registry.

F3.3.4 Any other monitored parameters, notifiable events or directives that may be defined for the TDM Recording Buffer FR type will be registered under the tdmRecordingBufferParametersId, tdmRecordingBufferEventsId, or tdmRecordingBufferDirectivesId (`{tdmRecordingBuffer 3}`) nodes, respectively, of the SANA FR registry.

ANNEX G

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

G1 SECURITY CONSIDERATIONS

G1.1 INTRODUCTION

This subsection describes security aspects of the Tracking Data service.

Reference [1] explicitly provides authentication and access control for CSTSes. As one of the suite of CSTSes, the Tracking Data service inherits the authentication and access control capabilities defined in reference [1]. The Tracking Data service provides no service-specific security capabilities. As specified in reference [1], additional security capabilities, if required, are levied on the underlying communications services that support the TD-CSTS. Specification of the various underlying communications technologies, and in particular their associated security provisions, are outside the scope of this Recommended Standard.

G1.2 SECURITY CONCERNS WITH RESPECT TO THE TRACKING DATA SERVICE

The Statements of Security Concerns subsection (refer to H1 of reference [1]) identifies the support for capabilities that respond to security concerns in the areas of data privacy (also known as confidentiality), data integrity, authentication, access control, availability of resources, and auditing.

G1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

As a member of the suite of CSTSes, the Tracking Data service depends on unspecified mechanisms operating in the underlying communications service, or on privacy-ensuring capabilities in the service-specific application processes that interoperate through the procedures defined in reference [1], 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 data contained in the TD-CSTS protocol data units as they traverse the WAN between service user and service provider.

Reference [1] constrains the ability of a third party to seize control of an active CSTS instance, but it does not specify mechanisms that would prevent an attacker from intercepting the protocol data units. Interception of tracking data could assist an attacker in establishing the orbit/trajectory of the Mission spacecraft, which could assist the attacker in subsequent attempts to acquire return link data or jam the forward link. The prevention of such interception attacks depends on unspecified mechanisms in the underlying communications service. 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 intercept data transferred between the service user and the service provider without detection.

If the CSTS authentication capability is not used and if authentication is not ensured by the underlying communications service, attackers could somehow obtain valid `initiator-identifier` values and use them to initiate TD-CSTS instances by which they could gain access to the tracking data transferred via the service.

The TD-CSTS depends on unspecified mechanisms operating in the underlying communications service to ensure that the supporting network has sufficient resources to provide sufficient support to legitimate service 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 service users from using the TD-CSTS.

If the service provider of the TD-CSTS provides no security auditing capabilities, or if a service user chooses not to employ auditing capabilities that do exist, then attackers may delay or escape detection while stealing data exchanged via the service.

G1.4 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the TD-CSTS are possible degradation and loss of ability to use the service, or the interception of tracking data that could aid in the determination of the orbit/trajectory of the spacecraft, acquisition of the space link, and/or jamming of the space link. The ability to actually acquire the space link (that is, either transmit to the spacecraft or receive and interpret data from the spacecraft) depends on the security supplied by the protocols used on that space link.

G2 SANA CONSIDERATIONS

The TD-CSTS relies on the SANA Functional Resource Registry (reference [5]) to provide the identification and definition of Functional Resource parameters and events.

As described in this Recommended Standard, the TD-CSTS reports parameters and events that are named in the context of Functional Resources. Functional Resource Types are registered under the

```
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) crossSupportResources(2)
}
```

node of the OID registration tree.

There are two subnodes under the `crossSupportResources` node: `crossSupportFunctionalities` and `agencyFunctionalities`, used to register CCSDS-standard Functional Resource Types and agency-unique Functional Resource Types, respectively. Agency-unique Functional Resource types are not relevant in the TD-CSTS context. Under each Functional Resource Type OID, the parameters, events, and directives are registered under dedicated subnodes.

Maintenance of the SANA registry of the Functional Resource Types, parameters, events, and directives under the `crossSupportFunctionalities` subnode is under the purview of the CCSDS Cross Support Services Area in accordance with the process and procedures identified in the CSTS Specification Framework (reference [1]).

The positions in the OID tree where the OIDs of the Functional Resources associated with the TD-CSTS are registered are specified in F2.3 and F3.3. The OID values are defined in annex C.

G3 PATENT CONSIDERATIONS

There are no patents that are known to apply to the technology used in the Tracking Data service.

ANNEX H

ACRONYMS

(INFORMATIVE)

CCSDS	Consultative Committee for Space Data Systems
CLTU	Communication Link Transmission Unit
CM	Complex Management
CSSE	cross support service element
CSSS	cross support service system
CSTS	Cross Support Transfer Service
DOR	Differential One-way Ranging
F-CLTU	Forward CLTU
ISO	International Organization for Standardization
MDOS	Mission Data Operation System
OID	Object Identifier
PDU	Protocol Data Unit
RAF	Return All Frames
RF	Radio Frequency
SANA	Space Assigned Numbers Authority
SCCS-SM	Space Communication Cross Support Service Management
SM	Service Management
SLE	Space Link Extension
TD-CSTS	Tracking Data Cross Support Transfer Service
TDM	Tracking Data Message
UM	Utilization Management
VLBI	Very Long Baseline Interferometry

ANNEX I

INFORMATIVE REFERENCES

(INFORMATIVE)

[I1] *Cross Support Concept—Part 1: Space Link Extension Services*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 910.3-G-3. Washington, D.C.: CCSDS, March 2006.

[I2] *Space Link Extension—Internet Protocol for Transfer Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 913.1-B-2. Washington, D.C.: CCSDS, September 2015.

[I3] *Cross Support Transfer Service Specification Framework Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 920.0-G-1. Forthcoming.

[I4] *Space Link Extension—Return All Frames Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.1-B-4. Washington, D.C.: CCSDS, August 2016.

[I5] *Space Link Extension—Forward CLTU Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-4. Washington, D.C.: CCSDS, August 2016.

[I6] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.

[I7] *Cross Support Service Management—Service Management Utilization Request Formats* CCSDS 902.9, proposed.

[I8] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.1-B-1. Washington, D.C.: CCSDS, May 2018.

[I9] *Navigation Data—Definitions and Conventions*. Issue 4. Report Concerning Space Data System Standards (Green Book), CCSDS 500.0-G-4. Washington, D.C.: CCSDS, November 2019.

ANNEX J

CROSS REFERENCES TO THE CSTS SPECIFICATION FRAMEWORK AND TO TRACKING DATA MESSAGE

(INFORMATIVE)

Table J-1 lists the specific sections and subsections of the CSTS Specification Framework (reference [1]) that are referenced by this Recommended Standard, and identifies the sections and subsections of this Recommended Standard that make specific reference to each of those sections/subsections of reference [1].

Table J-2 lists the specific sections and subsections of the Tracking Data Message Recommended Standard (reference [3]) that are referenced by this Recommended Standard, and identifies the sections and subsections of this Recommended Standard that make specific reference to each of those sections/subsections of reference [3].

Table J-1: Cross Reference to Reference [1] Sections and Subsections

Reference [1] Section/Subsection	Referencing Sections/Subsections of TD-CSTS
1.3.1	2.1
3.3	4.5.2.1.3.2
3.3.2.7	D1
3.4.2.2.4.3	5.1, 5.2
3.11.2.2.3.2 a)	6.1 b)
3.11.2.2.3.2 b)	6.1 c)
4.3	3.2.2, 3.2.9, 3.2.9 (table 3-1), A2.2 (table A-5)
4.3.5 (table 4-2)	5.3.1, 5.3.2, 5.3.3, 5.3.4
4.5	1.5.1, 2.2.3, 3.2.4, 3.2.9, 3.2.9 (table 3-1), 4.1.1, 4.4, 4.5.2.1.1, F2.1.5, A2.2 (table A-5)
4.5.2.1 NOTE	2.2.3
4.5.2.2.2.1	4.6 (table 4-3)
4.5.3.2.6	4.6 (table 4-3)
4.5.3.2.7.2	4.6 (table 4-3)
4.5.3.2.7.3	4.6 (table 4-3)
4.5.5 (table 4-16)	5.4.1, 5.4.2, 5.4.3
4.5.6	4.7
4.5.7	2.2.2, F3.1.1

Reference [1] Section/Subsection	Referencing Sections/Subsections of TD-CSTS
4.5.7.5	F3.2
4.5.7.5 b)	F3.1.6
4.5.7.10	F3.1.5
4.9	3.2.8, 3.2.9, 3.2.9 (table 3-1), A2.2 (table A-5)
4.9.5 (table 4-52)	5.5.1, 5.5.2
A	6.1
E	A1.2
F4.1	5.5.1, C
F4.3	C, E, A2.2, A2.2 (tables A-7, A-8, A-10, A-11, A-12, A-13, A-14, A-15, A-16, A-17, A-18, A-19)
F4.4	D1, A2.2, A2.2 (tables A-6, A-12, A-13, A-14, A-16, A-18, A-19)
F4.5	A2.2, A2.2 (tables A-5, A-6, A-7, A-8, A-9, A-10)
F4.7	D1, A2.2, A2.2 (table A-6, table A-14)
F4.16	4.6 (table 4-3), 5.5.1, C
F4.17	6.1, 6.2, 6.3, 6.4, 7.2, 7.3, C
F3	3.3
H1	G1.2

Table J-2: Cross Reference to Reference [3] Sections and Subsections

Reference [3] Section/Subsection	Referencing Sections/Subsections of TD-CSTS
3.2	B2
3.2.3	B2
3.3	B3.1, B3.2
3.3.1	F2.2
3.4	B3.1, B3.3
3.5	B3.1, B3.3