The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

CCSDS BUNDLE PROTOCOL SPECIFICATION

RECOMMENDED STANDARD

CCSDS 734.2-B-1

BLUE BOOK

September 2015

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

CCSDS BUNDLE PROTOCOL SPECIFICATION

RECOMMENDED STANDARD

CCSDS 734.2-B-1

BLUE BOOK

September 2015

DEDICATION

This book is dedicated to Adrian Hooke, whose end-to-end sensibilities and tireless advocacy for standardization of space data systems directly contributed to the formation of the Consultative Committee for Space Data Systems in 1982. His unique combination of technical skill, management abilities, and vision served CCSDS well for over 30 years. During that time CCSDS solidified the standardization of Physical and Data Link Layer protocols, and developed standards and technologies that had important and wide-ranging impacts in both the space and terrestrial communications industries. In the late 1990s, Adrian envisioned a new era for space communications leveraging a confluence of terrestrial internetworking and space-based data transport technologies. This led to the development of a concept that has come to be known as the Solar System Internetwork (SSI), of which the Bundle Protocol described here is a part.

Adrian will be missed, by CCSDS for the scope of his technical contributions and his leadership, and by his colleagues and friends for the greatness of his spirit and his wit. But his legacy to the space community remains. CCSDS will continue to provide useful and innovative solutions to space communication challenges so that Adrian's vision of an interoperable, standards-based communication system that reduces mission development time, cost, and risk will eventually be realized.

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	September 2015
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
E-mail: 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:

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

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

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

FOREWORD

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

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 *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 e-mail 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.
- Chinese 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.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- 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 734.2-B-1	CCSDS Bundle Protocol Specification, Recommended Standard, Issue 1	September 2015	Original issue

CONTENTS

Section Page

1 INTRODUCTION **1-1**

1.1 PURPOSE 1-1
1.2 SCOPE 1-1
1.3 ORGANIZATION OF THIS RECOMMENDED STANDARD 1-1
1.4 DEFINITIONS 1-2
1.5 REFERENCES 1-5

2 OVERVIEW **2-1**

2.1 GENERAL 2-1
2.2 IMPLEMENTATION ARCHITECTURES 2-3
2.3 SERVICES PROVIDED BY BP 2-3
2.4 QUALITIES OF SERVICE NOT PROVIDED BY BP 2-3

3 CCSDS PROFILE OF RFC 5050 ..... **3-1**

3.1 GENERAL 3-1
3.2 USE OF THE IPN NAMING SCHEME FOR ENDPOINT IDENTIFIERS 3-1
3.3 BUNDLE PROTOCOL EXTENDED CLASS OF SERVICE 3-1
3.4 USE OF TIME IN SECTION 6.1 OF RFC 5050 3-2
3.5 SANA REGISTRY CONSIDERATIONS 3-2

4 SERVICE DESCRIPTION **4-1**

4.1 SERVICES AT THE USER INTERFACE 4-1
4.2 SUMMARY OF PRIMITIVES 4-1
4.3 SUMMARY OF PARAMETERS 4-2
4.4 BP SERVICE PRIMITIVES 4-5

5 SERVICES BP REQUIRES OF THE SYSTEM **5-1**

5.1 RELIABLE STORAGE REQUIREMENTS 5-1
5.2 UNDERLYING COMMUNICATION SERVICE REQUIREMENTS 5-1

6 CONFORMANCE REQUIREMENTS **6-1**

6.1 GENERAL REQUIREMENTS 6-1
6.2 BUNDLE PROTOCOL REQUIREMENTS 6-1

CONTENTS (continued)

Section Page

ANNEX A PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA (NORMATIVE) A-1
ANNEX B CONVERGENCE LAYER ADAPTERS (NORMATIVE) B-1
ANNEX C EXTENDED CLASS OF SERVICE EXTENSION SPECIFICATION (NORMATIVE) C-1
ANNEX D AGGREGATE CUSTODY SIGNAL SPECIFICATION (NORMATIVE) D-1
ANNEX E DELAY-TOLERANT PAYLOAD CONDITIONING SPECIFICATION (NORMATIVE) E-1
ANNEX F BP MANAGED INFORMATION (NORMATIVE) F-1
ANNEX G SECURITY, SANA, AND PATENT CONSIDERATIONS (INFORMATIVE) G-1
ANNEX H INFORMATIVE REFERENCES (INFORMATIVE) H-1
ANNEX I ABBREVIATIONS AND ACRONYMS (INFORMATIVE) I-1

Figure

1-1 Graphical Representation of a Bundle Node 1-3
2-1 The Bundle Protocol Provides an End-to-End Delivery Service 2-2
D-1 ACS Payload Block Definition D-2
D-2 CTEB Block Definition D-3
D-3 ACS Processing Flow D-7

Table

A-1 PICS Notation A-2
A-2 Symbols for PICS ‘Support’ Column A-2
E-1 DPDU Header Fields E-23
E-2 Topic Block Fields E-24
E-3 Payload Record Fields E-24
F-1 Bundle State Information F-2
F-2 Error and Reporting Information F-3
F-3 Registration Information F-4
F-4 Node State Information F-5

1 INTRODUCTION

1.1 PURPOSE

This document defines a Recommended Standard for the CCSDS Bundle Protocol (BP), based on the Bundle Protocol of RFC 5050 (reference [1]), which defines end-to-end protocol, block formats, and abstract service descriptions for the exchange of messages (bundles) that support Delay Tolerant Networking (DTN). BP provides Network Layer service to applications allowing them to utilize BP's capabilities:

- custody-based retransmission;
- ability to cope with intermittent connectivity;
- ability to take advantage of scheduled, predicted, and opportunistic connectivity (in addition to continuous connectivity);
- notional data accountability with built-in status reporting.

1.2 SCOPE

This Recommended Standard is designed to be applicable to any kind of space mission or infrastructure that is communication-resource poor and is subject to long latencies and/or temporary network partitions, regardless of complexity. It is intended that this Recommended Standard become a uniform standard among all CCSDS Agencies. In addition, this specification exists to utilize the underlying service of various internetworking protocols both onboard and in transit between ground and space-based assets.

This Recommended Standard is intended to be applied to all systems that claim conformance to the CCSDS Bundle Protocol. It is agnostic to the choice of underlying transmission protocol in that BP can function over AOS, Space Packet, Proximity-1 Space Link Protocol, and various Internet and ground based protocols.

The CCSDS believes it is important to document the rationale underlying the recommendations chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The concept and rationale for the use of a bundle protocol in space links may be found in reference [H1].

1.3 ORGANIZATION OF THIS RECOMMENDED STANDARD

This Recommended Standard is organized as follows:

- Section 2 contains an overview of the Bundle Protocol and the references from which it is derived.
- Section 3 contains the CCSDS modification to RFC 5050.
- Section 4 contains the service descriptions.
- Section 5 contains services BP requires of the system.

- Section 6 contains conformance requirements.
- Annex A contains the Implementation Conformance Statement for the protocol.
- Annex B contains the Convergence Layer Adapters (CLAs).
- Annex C contains the Extended Class of Service specification.
- Annex D contains the Aggregate Custody Signal specification.
- Annex E contains the Delay Tolerant Payload Conditioning specification.
- Annex F contains BP managed information.
- Annex G contains Security, Space Assigned Numbers Authority (SANA), and Patent Considerations.
- Annex H contains Informative References.
- Annex I contains abbreviations and acronyms used in this document.

1.4 DEFINITIONS

1.4.1 DEFINITIONS FROM OPEN SYSTEMS INTERCONNECTION (OSI) SERVICE DEFINITION CONVENTIONS

This Recommended Standard makes use of a number of terms defined in reference [2]. As used in this Recommended Standard those terms are to be interpreted in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- indication;
- primitive;
- request;
- response.

1.4.2 DEFINITIONS FROM OSI BASIC REFERENCE MODEL

This Recommended Standard makes use of a number of terms defined in reference [3]. As used in this Recommended Standard those terms are to be understood in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- entity;
- Protocol Data Unit (PDU);
- service;
- Service Data Unit (SDU).

1.4.3 DEFINITIONS FROM RFC 5050

1.4.3.1 Overview

This Recommended Standard makes use of a number of terms defined in reference [1]. Some of the definitions needed for section 2 of this document are reproduced here for convenience.

A graphical representation of a bundle node is given in figure 1-1. A bundle node is any entity that can send and/or receive bundles.

Each bundle node has three conceptual components described in more detail below: a 'bundle protocol agent', a set of zero or more 'convergence layer adapters', and an 'application agent'. The major components are illustrated in figure 1-1 and include the addition of storage for enqueued traffic and a Management Information Base (MIB) element.

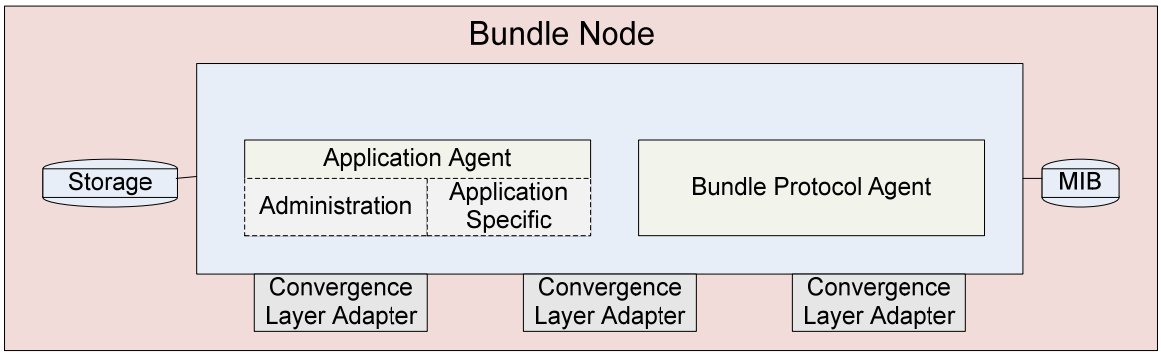

Figure 1-1: Graphical Representation of a Bundle Node

It should be noted that there is *one* application agent per conceptual bundle node. That application may register in multiple endpoints (may provide multiple endpoint identifiers to the bundle protocol agent, requesting delivery of bundles to any of those endpoints).

1.4.3.2 RFC 5050 Terms

bundle: A protocol data unit of the DTN Bundle Protocol.

NOTE – Each bundle comprises a sequence of two or more 'blocks' of protocol data, which serve various purposes. Multiple instances of the same bundle (the same unit of DTN protocol data) might exist concurrently in different parts of a network, possibly in different representations, in the memory local to one or more bundle nodes, and/or in transit between nodes. In the context of the operation of a bundle node, a bundle is an instance of some bundle in the network that is in that node's local memory.

bundle node (also simply 'node' or 'BP node'): Any entity that can send and/or receive bundles.

NOTE – In the most familiar case, a bundle node is instantiated as a single process running on a general-purpose computer, but in general the definition is meant to be broader: a bundle node might alternatively be a thread, an object in an object-oriented operating system, a special-purpose hardware device, etc. Each bundle node has three conceptual components, defined below: a 'bundle protocol agent', a set of zero or more 'convergence layer adapters', and an 'application agent'.

bundle protocol agent, BPA: Node component that offers the BP services and executes the procedures of the Bundle Protocol.

NOTE – The manner in which it does so is an implementation matter. BPA functionality can be coded into individual nodes, as a shared library that is shared by any number of bundle nodes on a single computer, as a daemon whose services are invoked via inter-process or network communication by one or more bundle nodes on one or more computers, or in hardware.

application agent, AA: Node component that utilizes the BP services to effect communication for some purpose.

NOTE – The AA has an application-specific element and administrative element. The application-specific element of an AA constructs, as defined in section 5 of RFC 5050, requests transmission of, accepts delivery of, and processes application-specific application data units; the only interface between the BPA and the application-specific element of the AA is the BP service interface. The administrative element of an AA constructs and requests transmission of administrative records as defined in section 6 of RFC 5050. It accepts delivery of and processes any custody signals that the node receives. In addition to the BP service interface, there is a (conceptual) private control interface between the BPA and the administrative element of the AA that enables each to direct the other to take action under specific circumstances. For a node that serves simply as a 'router' in the overlay network, the AA may have no application-specific element at all. The application-specific elements of other nodes' AAs may perform arbitrarily complex application functions, perhaps even offering multiplexed DTN communication services to a number of other applications. As with the BPA, the way AA performs its functions is wholly an implementation matter; in particular, the administrative element of an AA might be built into the library or daemon or hardware that implements the BPA, and the application-specific element of an AA might be implemented either in software or in hardware.

convergence layer adapter, CLA: Adapter that sends and receives bundles on behalf of the BPA.

NOTE – A CLA enables the BPA to interact with an underlying data transport mechanism such as a link or network to send and receive bundles. The manner in which a CLA sends and receives bundles is an implementation matter and is unique to the underlying transport mechanism. Therefore the BPA may utilize CLAs from a number of different underlying transport mechanisms subject to the routing of traffic.

1.5 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.

[1] K. Scott and S. Burleigh. *Bundle Protocol Specification*. RFC 5050. Reston, Virginia: ISOC, November 2007.

[2] *Information Technology—Open Systems Interconnection—Basic Reference Model—Conventions for the Definition of OSI Services*. International Standard, ISO/IEC 10731:1994. Geneva: ISO, 1994.

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

[4] S. Burleigh. *Compressed Bundle Header Encoding (CBHE)*. RFC 6260. Reston, Virginia: ISOC, May 2011.

[5] Space Assigned Numbers Authority (SANA). http://sanaregistry.org/.

[6] *Licklider Transmission Protocol (LTP) for CCSDS*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 734.1-B-1. Washington, D.C.: CCSDS, May 2015.

[7] L. Eggert and G. Fairhurst. *Unicast UDP Usage Guidelines for Application Designers*. RFC 5405. Reston, Virginia: ISOC, November 2008.

[8] *Encapsulation Service*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.1-B-2. Washington, D.C.: CCSDS, October 2009.

[9] M. Ramadas, S. Burleigh, and S. Farrell. *Licklider Transmission Protocol—Specification*. RFC 5326. Reston, Virginia: ISOC, September 2008.

[10] M. Blanchet. *Delay-Tolerant Networking Bundle Protocol IANA Registries*. RFC 6255. Reston, Virginia: ISOC, May 2011.

2 OVERVIEW

2.1 GENERAL

Delay Tolerant Networking is an end-to-end network service providing communications in and/or through environments characterized by one or more of the following:

- intermittent connectivity;
- variable delays, which may be large and irregular;
- high bit error rates;
- asymmetric and simplex links.

One core element of DTN is the BP. BP provides end-to-end network services, operating above the data transport services provided by links or networks accessed via the CLAs, and forming a store-and-forward network. Key capabilities of the Bundle Protocol include:

- ability to cope with intermittent connectivity;
- ability to take advantage of scheduled and opportunistic connectivity (in addition to 'always up' connectivity);
- custody transfer;
- hop-by-hop security (authentication of transmitting entity);
- end-to-end security (confidentiality, integrity) for data;
- late binding of names to addresses.

Reference [H1] contains descriptions of these capabilities and rationale for the DTN architecture.

The Bundle Protocol uses the 'native' local protocols for communications within a given network. The interface between the Bundle Protocol and a specific lower-layer protocol suite is known as a convergence layer. Figure 2-1 shows an example configuration with the Bundle Protocol and a convergence layer adaptor running above a transport protocol (intended to be interpreted in the context of the Internet stack) on the left, and running directly over a Data Link Layer on the right. The 'CL B' on the right could, for example, be the interface to the Licklider Transmission Protocol with the 'Link B1' representing LTP running over one of the CCSDS Data Link Layer protocols. Alternatively BP could be used to connect together two internets that may exist, such as an on-orbit (or lunar) network and a ground network.

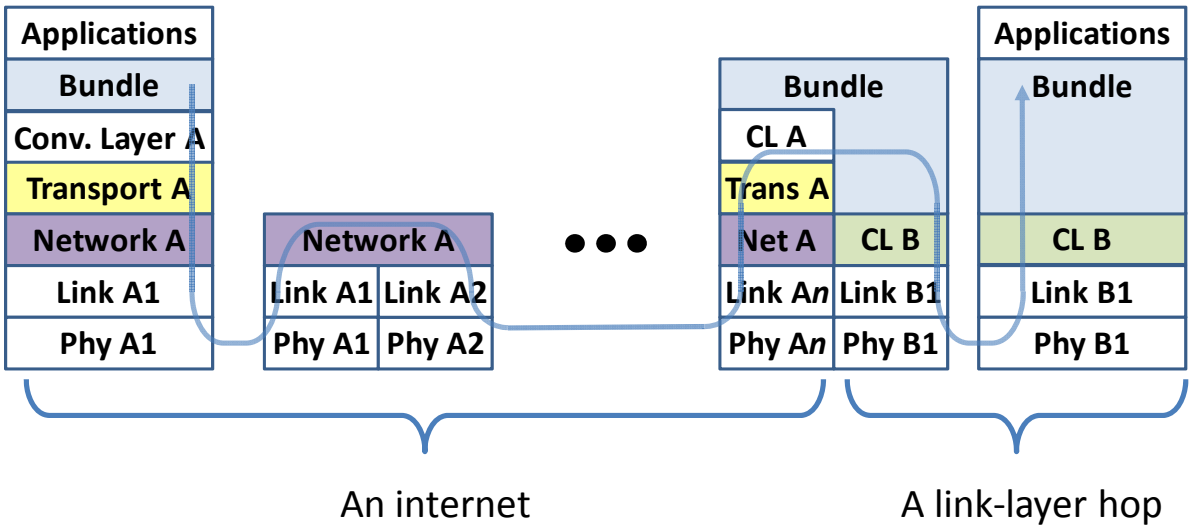

Figure 2-1: The Bundle Protocol Provides an End-to-End Delivery Service

This document describes the format of the messages (called bundles) passed between nodes participating in bundle communications. In addition, this document addresses endpoint naming for the purpose of bundle header compression and describes how the protocol may be extended to support new capabilities while maintaining compatibility with the base protocol. This document does *not* address the bundle routing algorithm or mechanisms for populating the routing or forwarding information bases of bundle nodes.

The IETF's classification of the Bundle Protocol RFC as 'experimental' should be considered in the context of deployment on the global Internet. In fact it is not viewed as an Internet Standard nor is it proposed for any specific application. In the Internet, issues such as scalability to millions of nodes, congestion control, and non-destructive coexistence with other established protocols (in particular the Transmission Control Protocol [TCP]), are of extreme importance. Because the Bundle Protocol has NOT been deployed on a scale of thousands of nodes, and because the specification was the result of an effort by an Internet Research Task Force (IRTF) working group, the protocol's status for global deployment is and should be experimental until it is determined to be suitable for deployment on Internet scales. However, the applicability of the Bundle Protocol in the harsh environment of space makes it an excellent technological innovation allowing multiple internetworking environments to interact.

The SIS-DTN working group has carefully considered the protocol specified in RFC 5050 and has determined that it is suitable for adoption, together with the modifications in section 3 of this document, for use in CCSDS missions. In particular, CCSDS missions do not have the same scalability issues as the Internet, and testing has demonstrated that the profile defined in this document is suitable for CCSDS environments.

2.2 IMPLEMENTATION ARCHITECTURES

There are many ways in which a bundle node can be instantiated. The following are some examples:

- a single process running on a general-purpose computer;
- a thread running as a background process;
- an object in an object-oriented operating system;
- a special-purpose hardware device.

NOTE – No specific instantiation is defined or expected; these decisions are purely an implementation issue.

2.3 SERVICES PROVIDED BY BP

BP provides a data transmission service to move 'bundles' (contiguous groups of octets) of data from one BP node to another:

a) commencing a registration (registering a node in an endpoint);

b) terminating a registration;

c) switching a registration between Active and Passive states;

d) transmitting a bundle to an identified bundle endpoint;

e) canceling a transmission that has been requested;

f) polling a registration that is in the Passive state;

g) delivering a received bundle;

h) reporting bundle status.

2.4 QUALITIES OF SERVICE NOT PROVIDED BY BP

The Bundle Protocol as specified in this document *does not* provide the following services:

a) in-order delivery of bundles;

b) complete delivery of sequences of bundles.

These services may be provided by a layer above BP yet below the end-system applications. These services can exist as shims. Such a shim provides the logic to accomplish the desired functions and is inserted between BP and the Application Layer. This would leave the existing network protocol stack intact. Such a layer is described in annex E of this document.

3 CCSDS PROFILE OF RFC 5050

3.1 GENERAL

This document adopts the Bundle Protocol as specified in Internet RFC 5050 (reference [1]), with the constraints and exceptions specified in section 3 of this document.

3.2 USE OF THE IPN NAMING SCHEME FOR ENDPOINT IDENTIFIERS

3.2.1 Implementations shall support the 'IPN' naming scheme defined in section 2.1 of RFC 6260, *Compressed Bundle Header Encoding (CBHE)* (reference [4]).

NOTE – The scheme-specific part of an IPN name consists of:

1) a sequence of ASCII numeric digits representing an integer in the range 1 to $2^{64}-1$, termed the 'node number' of the URI;

2) an ASCII period ('.') character;

3) a sequence of ASCII numeric digits representing an integer in the range 0 to $2^{64}-1$, termed the 'service number' of the URI.

3.2.2 The IPN node numbers used shall be assigned by SANA from the CCSDS CBHE Node Number Registry.

3.2.3 The Service Numbers used shall be assigned by IANA / SANA from either the IANA CBHE Service Numbers registry or the SANA CBHE Service Numbers Registry.

NOTES

1 CBHE is the compression mechanism enabled by the IPN naming scheme.

2 The SANA CBHE Node Number registry is a portion of the IANA registry that has been delegated to SANA for management by CCSDS.

3.3 BUNDLE PROTOCOL EXTENDED CLASS OF SERVICE

Conformant implementations of the CCSDS Bundle Protocol shall implement the 'Extended Class of Service (ECOS)' block defined in annex C.

NOTE – Spacecraft operations may require additional features beyond those identified in RFC 5050. One such feature is the expansion of the bundle process control flags designated as 'class-of-service'. ECOS provides the capability to prioritize or extend the service classes. Such uses include:

- the creation of emergency or critical traffic;
- expansion of traffic priorities reflective of a diverse user environment;
- special handling of bundles.

3.4 USE OF TIME IN SECTION 6.1 OF RFC 5050

Where the spacecraft time system does not provide sufficient precision to support the requirements of RFC 5050 section 6.1, the precision of the onboard system shall be used.

NOTE – Section 6.1 of RFC 5050 specifies that the time fields in administrative records use seconds and nanoseconds since the start of year 2000. Spacecraft time systems may not be able to provide meaningful values for the nanoseconds fields of these entries. In such a case the administrative time field is required to support the precision of the clock rate to the significant digits of the Command and Data Handling (C&DH) subsystem and will not drive requirements on the precision of the spacecraft clock.

3.5 SANA REGISTRY CONSIDERATIONS

3.5.1 CBHE NODE NUMBERS

3.5.1.1 General

SANA has established the registry

http://sanaregistry.org/r/bp_cbhe_node_numbers/bp_cbhe_node_numbers.html

to manage CBHE Node Number assignments. The registry shall be used to catalog agency-managed BP CBHE Node Numbers and LTP engine IDs that are coincident.

NOTE – The purpose of this registry is to ensure uniqueness of BP CBHE Node Numbers used in space missions.

3.5.1.2 Value Range for SANA BP CBHE Node Numbers

The value range for BP CBHE Node Numbers shall be as assigned by IANA.

3.5.1.3 SANA BP CBHE Node Number Registration Policy

The registration policy for the registry shall be: no engineering review required; request must come from an identified CCSDS representative of a member, observer, or affiliate organization.

NOTE – For missions utilizing LTP and BP protocols, requests to SANA should attempt to utilize identical numbers for LTP Protocol Engine Identifiers and BP CBHE IPN Node Numbers. This allows BP implementations to forego having a CBHE ID-to-LTP Engine ID mapping table for those cases where they know that the two identifiers are the same. Synchronizing the CBHE and LTP Engine identifiers is purely an optimization to aid implementations and is not a requirement.

3.5.2 CBHE SERVICE NUMBERS

3.5.2.1 General

SANA has established the registry

http://sanaregistry.org/r/bp_cbhe_service_numbers/bp_cbhe_service_numbers.html

to manage CBHE Service Number assignments. The registry shall be used by CCSDS to catalog BP CBHE Service Numbers that denote different bundle services.

NOTE – The purpose of this registry is to ensure uniqueness of the CBHE Service Numbers used in space missions.

3.5.2.2 Value Range for BP CBHE Service Numbers assigned via the SANA Registry

The value range for BP CBHE Service Numbers shall be as assigned by IANA.

3.5.2.3 CCSDS BP CBHE Service Numbers Registration Policy

The registration policy for the registry shall be: no engineering review required; request must come from an identified CCSDS representative of a member, observer, or affiliate organization.

4 SERVICE DESCRIPTION

4.1 SERVICES AT THE USER INTERFACE

4.1.1 The services provided by the Bundle Protocol shall be made available to bundle protocol users and include the following:

a) initiate a registration (registering a node in an endpoint);

b) terminate a registration;

c) switch a registration between Active and Passive states;

d) transmit a bundle to an identified bundle endpoint;

e) cancel a transmission;

f) poll a registration that is in the Passive state;

g) deliver a received bundle.

4.1.2 The BP node shall be implemented such that virtually any number of transactions may be conducted concurrently in various stages of transmission or reception at a single BP node.

NOTE – To clarify: the implementation needs to be able to accept a primitive, and thereupon initiate a new transaction prior to the completion of previously initiated transactions. The requirement for concurrent transaction support therefore does not necessarily imply that the implementation needs to be able to begin initial transmission of data for one transaction while initial transmission of file data for one or more other transactions is still in progress. (But neither is support for this functional model precluded.)

4.2 SUMMARY OF PRIMITIVES

4.2.1 The BP service shall consume the following request primitives:

– Register.request;

– Deregister.request;

– ChangeRegistrationState.request;

– Send.request;

– Cancel.request;

– Poll.request.

4.2.2 The BP service shall deliver the following indication primitives:

– LocalBundleID.indication;

– BundleDelivery.indication.

4.3 SUMMARY OF PARAMETERS

4.3.1 DESTINATION COMMUNICATIONS ENDPOINT ID

The *destination communications endpoint ID* parameter shall identify the communications endpoint to which the bundle is to be sent.

NOTE – One can think of a DTN communications endpoint as an application, but in general the definition is meant to be broader. For example, a single BPA (with a single endpoint ID) could service other local nodes such as elements of a sensor network using private protocols.

4.3.2 SOURCE COMMUNICATIONS ENDPOINT ID

The source communications endpoint ID parameter shall uniquely identify the communications endpoint from which the bundle was sent.

4.3.3 REPORT-TO COMMUNICATIONS ENDPOINT ID

The report-to communications endpoint ID parameter shall identify the communications endpoint to which any bundle status reports pertaining to the bundle are sent.

4.3.4 ISSINGLETONEID

The IsSingletonEID parameter shall be ‘True’ if the referenced Endpoint IDentifier (EID) is a singleton, i.e., if there is at most one BP node that is a member of the endpoint identified.

4.3.5 CLASS-OF-SERVICE PARAMETER

4.3.5.1 The class-of-service parameter shall indicate which class of standard procedures is to be followed when transmitting and delivering the bundle.

4.3.5.2 The value of the class-of-service parameter shall be one of the following:

- bulk;
- normal;
- expedited.

4.3.6 DELIVERY OPTIONS PARAMETER

4.3.6.1 The delivery options parameter shall indicate what optional procedures are additionally to be followed when transmitting and delivering the bundle.

4.3.6.2 The value of the delivery options parameter shall be a combination of zero or more of the following:

a) bundle is a fragment;

b) application data unit is an administrative record;

c) bundle must not be fragmented;

d) custody transfer is requested;

e) destination endpoint is a singleton;

f) acknowledgement by application is requested;

g) class of service;

h) request reporting of bundle reception;

i) request reporting of custody acceptance;

j) request reporting of bundle forwarding;

k) request reporting of bundle delivery;

l) request reporting of bundle deletion;

m) extended class of service.

4.3.7 LIFETIME PARAMETER

The lifetime parameter shall indicate the length of time, following initial creation time of a bundle, after which bundle protocol agents may discard the bundle.

4.3.8 APPLICATION DATA UNIT PARAMETER

The application data unit parameter shall indicate the location (in memory or non-volatile storage, a local implementation matter) of the application data conveyed by the bundle.

4.3.9 LOCAL BUNDLE ID

The Local Bundle ID parameter shall identify a particular bundle within the context of a given bundle protocol agent.

NOTE – This identification is provided to the user of the bundle service on submitting a bundle for transmission so that the user may later reference that bundle in other requests, such as cancellation. The form of this identifier is entirely implementation-specific and should not be confused with the Source EID and Creation Timestamp combination (global Bundle ID) used to uniquely identify bundles in the network.

4.3.10 DELIVERY FAILURE ACTION

4.3.10.1 The Delivery Failure Action parameter shall identify the response the node is to take on receipt of a bundle that is deliverable subject to the registration when the registration is in the Passive state (see 4.3.11).

4.3.10.2 The Delivery Failure Action parameter shall signal one of the following possible responses:

- defer delivery of the bundle;
- abandon delivery of the bundle.

NOTE – RFC 5050 section 3.1 contains more on when deferred bundles may be delivered to receiving applications.

4.3.11 REGISTRATION STATE

The Registration State is the state machine characterization of a given node's membership in a given endpoint. A registration state must at any time be in one of two states: Active or Passive.

NOTE – A registration always has an associated 'delivery failure action' which denotes the action to be taken upon receipt of a bundle that is deliverable subject to the registration when the registration is in the Passive state (refer to 4.3.10). Further definition of Registration can be found in section 3.1 of RFC 5050.

4.3.12 HEADER INFORMATION

The Header Information parameter shall uniquely identify the delivered bundle and indicate the delivered bundle's remaining time to live and the time of delivery to the application agent.

4.4 BP SERVICE PRIMITIVES

4.4.1 Register.request

4.4.1.1 Function

The **`Register.request`** primitive shall be used to notify the BP agent of the node's membership in a communications endpoint.

4.4.1.2 Semantics

`Register.request` shall provide parameters as follows:

`Register.request`	(delivery failure action, destination communications endpoint ID)

4.4.1.3 When Generated

`Register.request` may be generated by any BP application at any time.

4.4.1.4 Effect on Receipt

4.4.1.4.1 Receipt of **`Register.request`** shall cause the BP agent to declare the node's registration in the indicated endpoint.

NOTE – The registration is initially in Passive state.

4.4.1.4.2 The indicated failure action shall be taken upon arrival of any bundle destined for this endpoint, as long as the registration remains in Passive state.

4.4.1.5 Discussion—Additional Comments

None.

4.4.2 Deregister.request

4.4.2.1 Function

The **`Deregister.request`** primitive shall be used to notify the BP agent of the end of the node's membership in the indicated endpoint.

4.4.2.2 Semantics

`Deregister.request` shall provide parameters as follows:

`Deregister.request` (destination communications endpoint ID)

4.4.2.3 When Generated

`Deregister.request` may be generated by any BP application at any time when the node is registered in the indicated endpoint.

4.4.2.4 Effect on Receipt

Receipt of **`Deregister.request`** shall cause the node's registration in the indicated endpoint to be rescinded.

4.4.2.5 Discussion—Additional Comments

Multiple nodes can be members of the same endpoint. One node deregistering from the endpoint does not affect other nodes' delivery or delivery failure behavior.

4.4.3 ChangeRegistrationState.request

4.4.3.1 Function

The **`ChangeRegistrationState.request`** primitive shall be used to notify the BP agent of a desired change in the registration state.

4.4.3.2 Semantics

`ChangeRegistrationState.request` shall provide parameters as follows:

`ChangeRegistrationState.request`	(destination communications endpoint ID, registrationState)

4.4.3.3 When Generated

`ChangeRegistrationState.request` may be generated by any BP application at any time when the node is registered in the indicated endpoint.

4.4.3.4 Effect on Receipt

4.4.3.4.1 Receipt of **`ChangeRegistrationState.request`** shall cause the BP agent to change the state of the registration to the requested state.

4.4.3.4.2 If the new state is Active, receipt of this request shall additionally cause the bundle protocol agent to deliver to the application all bundles, destined for the indicated endpoint, for which delivery was deferred.

4.4.3.5 Discussion—Additional Comments

None.

4.4.4 Send.request

4.4.4.1 Function

The **`Send.request`** primitive shall be used by the application to request transmission of an application data unit from the source communications endpoint to a destination communications endpoint.

4.4.4.2 Semantics

`Send.request` shall provide parameters as follows:

`Send.request`	(source communications endpoint ID, destination communications endpoint ID, report-to communications endpoint ID, class-of-service, IsSingletonEID, delivery options, lifetime, application data unit)

4.4.4.3 When Generated

`Send.request` may be generated by the source BP application at any time.

4.4.4.4 Effect on Receipt

Receipt of **`Send.request`** shall cause the BP agent to initiate bundle transmission procedures.

4.4.4.5 Discussion—Additional Comments

None.

4.4.5 Cancel.request

4.4.5.1 Function

The **`Cancel.request`** primitive shall be used by the application to request termination of transmission of an application data unit for which the application previously requested transmission.

4.4.5.2 Semantics

`Cancel.request` shall provide parameters as follows:

`Cancel.request` (Local Bundle ID)

4.4.5.3 When Generated

`Cancel.request` may be generated by the application at any time after requesting transmission of a bundle.

4.4.5.4 Effect on Receipt

Receipt of **`Cancel.request`** shall cause the BP agent to stop attempting to transmit and to discard the target bundle, if possible.

4.4.5.5 Discussion—Additional Comments

If the bundle has already been transmitted, there is no obligation on the sending BP agent to take any further action. It is an implementation matter whether a bundle that is in the process of being transmitted when a **`Cancel.request`** is received is terminated.

4.4.6 Poll.request

4.4.6.1 Function

The **`Poll.request`** primitive shall be used by the application to request immediate delivery of the least-recently received bundle that is currently deliverable subject to the node's registration in the indicated endpoint.

4.4.6.2 Semantics

`Poll.request` shall provide parameters as follows:

`Poll.request` (destination communications endpoint ID)

4.4.6.3 When Generated

`Poll.request` may be generated by any BP application at any time when the node is registered in the indicated endpoint and that registration is in Passive state.

4.4.6.4 Effect on Receipt

Receipt of **`Poll.request`** shall cause the BP agent to deliver to the BP application the least-recently received bundle, destined for the destination communications endpoint ID, for which delivery was deferred.

NOTE – Prioritization applies only to forwarding of a bundle. Deferred bundles are delivered in the order in which they were received.

4.4.6.5 Discussion—Additional Comments

None.

4.4.7 LocalBundleID.indication

4.4.7.1 Function

The **`LocalBundleID.indication`** primitive shall be used to provide the application a reference to a particular bundle of which the application requested transmission.

4.4.7.2 Semantics

`LocalBundleID.indication` shall provide parameters as follows:

`LocalBundleID.indication` (Local Bundle ID)

4.4.7.3 When Generated

`LocalBundleID.indication` shall be generated by a BP agent once it has consumed a **`Send.request`** from the application.

4.4.7.4 Effect on Receipt

The effect on receipt of **`LocalBundleID.indication`** by a BP application is undefined.

4.4.7.5 Discussion—Additional Comments

On receiving this notice the sending application can, for example, release resources of its own that are allocated to the bundles being transmitted, or remember the Local Bundle ID so that transmission can be canceled in the future if necessary.

4.4.8 BundleDelivery.indication

4.4.8.1 Function

The **`BundleDelivery.indication`** primitive shall be used to indicate to the bundle service user that a bundle has been delivered to the application.

4.4.8.2 Semantics

`BundleDelivery.indication` shall provide parameters as follows:

`BundleDelivery.indication`	(header information, application data unit)

4.4.8.3 When Generated

`BundleDelivery.indication` shall be generated by a BP agent upon delivery of a bundle, either on reception of bundles destined for active registrations or in response to poll requests referencing passive registrations.

4.4.8.4 Effect on Receipt

The effect on receipt is defined by the application.

4.4.8.5 Discussion—Additional Comments

None.

5 SERVICES BP REQUIRES OF THE SYSTEM

5.1 RELIABLE STORAGE REQUIREMENTS

BP nodes shall have access to a reliable storage service.

NOTES

1 This storage mechanism may be in dynamic memory or via a persistent mechanism such as a solid-state recorder and may be organized by various means to include file systems.

2 The implementation of this storage can be shared among multiple elements of the communication stack so that reliability mechanisms at multiple layers do not have to maintain multiple copies of the data being transmitted.

3 Volume of storage required and duration of storage are mission- and implementation-dependent.

5.2 UNDERLYING COMMUNICATION SERVICE REQUIREMENTS

5.2.1 The following information shall be available to BP, either from the local operating environment or from the underlying communication service provider:

- forward advancing time that can be represented as 'DTN time' as defined by RFC 5050 (reference [1]);
- at least one long-lived singleton EID of which the node is a member;
- a unique creation time for BP traffic.

NOTE – The means by which this information is accessed by BP is implementation-dependent.

5.2.2 Each convergence layer adapter is expected to provide the following services to the BP agent:

a) sending a bundle to all bundle nodes in the minimum reception group of the endpoint identified by a specified endpoint ID that are reachable via some convergence layer protocol; and

b) acquiring a bundle that was sent by a remote bundle node via some convergence layer protocol.

NOTE – The convergence layer adaptor service interface specified here is neither exhaustive nor exclusive. That is, supplementary DTN protocol specifications (including, but not restricted to, the Bundle Security Protocol [BSP]) may expect convergence layer adapters that serve BP implementations conforming to those protocols to provide additional services.

5.2.3 The service provided by the protocols beneath BP (not necessarily by the convergence layer protocol itself) shall deliver only complete layer-(N−1) service data units (bundles) to the receiving BP Node.

5.2.4 The service provided by the underlying protocols (not necessarily by the convergence layer protocol itself) shall provide integrity checking of the layer-(N−1) service data units (bundles) and shall discard layer-(N−1) service data units that are determined to be corrupted.

5.2.5 The convergence layer adaptor service may provide a cap on the rate at which a sending BP engine can inject data into the layer-(N−1) service.

NOTE – If such a capability is needed and is not provided by the layer-(N−1) service, it may be possible to provide it as part of the BP interface to the layer-(N−1) service.

5.2.6 Delivery of duplicate BP PDUs to a BPA by the underlying layer shall be acceptable.

6 CONFORMANCE REQUIREMENTS

6.1 GENERAL REQUIREMENTS

6.1.1 PROTOCOL IMPLEMENTATION

A conforming implementation of this protocol shall:

- conform to the BP specification (RFC 5050, reference [1]);
- conform to the ECOS specification of annex C;
- conform to the CBHE specification (RFC 6260, reference [4]);
- implement the modifications in section 3 of this document;
- implement the services described in section 4 of this document.

6.1.2 PICS PROFORMA

An implementer shall prepare a Protocol Implementation Conformance Statement (PICS) based on the defined proforma in annex A of this document.

6.2 BUNDLE PROTOCOL REQUIREMENTS

6.2.1 MAJOR CAPABILITIES

6.2.1.1 All Bundle Protocol Implementations

All BP implementations for CCSDS shall implement and/or conform to the following:

a) bundle structure as described in RFC 5050 sections 3.1, 4.0, 4.2, 4.4, 5.8, and 8;

b) block structure as described in RFC 5050 sections 4.1, 4.5, 4.5.1, 4.5.2, 4.5.3, 4.6, 4.7;

c) administrative record generation and structure as described in RFC 5050 section 5.1, 6.0, 6.1, and 6.2;

d) administrative record processing as described in RFC 5050 sections 6.1.1, 6.1.2, and 6.3;

e) CBHE in accordance with RFC 6260 and section 3 of this document:

   1) BP nodes shall use CBHE endpoint identifiers obtained from SANA;

   2) BP nodes shall use CBHE service numbers obtained from SANA or, in the case of mission-private services, may use service numbers from the range reserved for private/experimental use;

f) ECOS in accordance with annex C and section 3 of this document.

6.2.1.2 Bundle Protocol Senders

6.2.1.2.1 A conforming BP implementation shall support the following in accordance with the base standard:

a) bundle transmission as described in RFC 5050 sections 3.3, 4.3, 5.15, and 5.2;

b) bundle forwarding as defined in RFC 5050 sections 4.2, 5.1, 5.3, 5.4, 5.4.1, 5.4.2, and 5.5).

6.2.1.2.2 In addition, a BP sender shall also support the following capabilities in accordance with the base standard:

a) intermittent connectivity conditions specified in RFC 5050 section 1;

b) late binding as described in RFC 5050 section 1;

c) bundle delivery failure as defined in RFC 5050 section 3.1;

d) bundle priority as defined in RFC 5050 section 4.2 and the ECOS specification of annex C;

e) bundle deletion procedures as defined in RFC 5050 sections 3.1, 4.2, 5.13, and 5.14;

f) dictionary byte array and revision per RFC 5050 sections 4.4 and 4.7.

6.2.1.3 Bundle Protocol Receivers

A conforming BP implementation shall support the following in accordance with the base standard:

a) bundle acceptance in accordance with RFC 5050 sections 4.2, 4.5.1, 4.5.2, 5.6, 5.7, 5.9, 5.10, and 5.13;

b) processing of custody signals as described in RFC 5050 sections 3.1, 4.2, 5.4, 5.4.1, 5.4.2, 5.10.1, 5.10.2, 5.11, 5.12, 6.1, 6.1.2, and 6.3;

c) node registration as defined in RFC 5050 sections 3.3 and 5.16.

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 OVERVIEW

This annex provides the Protocol Implementation Conformance Statement (PICS) Requirements List (RL) for CCSDS-compliant implementations of BP. The PICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements of the base standards referenced in the RL.

An implementation's completed RL is called the PICS. The PICS states which capabilities and options of the protocol have been implemented. The following can use the PICS:

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

b) the supplier and acquirer 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) the 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 protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A2 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the protocol by completing the RL; that is, compliance to all mandatory requirements and the options that are not supported are shown. The resulting completed RL is called a PICS. In the Support column, each response shall be selected either from the indicated set of responses, or it shall comprise one or more parameter values as requested. 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 Xi, where i is a unique identifier, to an accompanying rationale for the noncompliance.

A3 NOTATION

A3.1 The symbols in table A-1 are used in the RL to indicate the status of features.

Table A-1: PICS Notation

Symbol	Meaning
M	Mandatory
O	Optional
O.<n>	Optional, but support of at least one of the group of options labeled by the same numeral <n> is required

A3.2 The symbols in table A-2 shall be used in the 'Support' column of the PICS.

Table A-2: Symbols for PICS 'Support' Column

Symbol	Meaning
Y	Yes, the feature is supported by the implementation.
N	No, the feature is not supported by the implementation.
N/A	The item is not applicable.

A4 REFERENCED BASE STANDARDS

A4.1 The base standards referenced in the RL shall be:

a) CCSDS BP (this document);

b) RFC 5050 (reference [1]);

c) RFC 6260 (reference [4]).

A4.2 In the tables below, the notation in the Reference column combines one of the short-form document identifiers above (e.g., RFC 5050) with applicable subsection numbers in the referenced document. RFC numbers are used to facilitate reference to subsections within the Internet specifications.

A5 GENERAL INFORMATION

A5.1 IDENTIFICATION OF PICS

Ref	Question	Response
1	Date of Statement (DD/MM/YYYY)	
2	PICS serial number	
3	System conformance statement cross-reference	

A5.2 IDENTIFICATION OF IMPLEMENTATION UNDER TEST (IUT)

Ref	Question	Response
1	Implementation name	
2	Implementation version	
3	Name of hardware (machine) used in test	
4	Version of hardware (machine) used in test	
5	Name of operating system used during test	
6	Version of operating system used during test	
7	Additional configuration information pertinent to the test	
8	Other information	

A5.3 IDENTIFICATION

Ref	Question	Response
1	Supplier	
2	Point of contact for queries	
3	Implementation name(s) and version(s)	
4	Other information necessary for full identification (e.g., name(s) and version(s) for machines and/or operating systems	

A5.4 PROTOCOL SUMMARY

Ref	Question	Response
1	Protocol version	
2	Addenda implemented	
3	Amendments implemented	
4	Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the protocol. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming.	a) Yes b) No
5	Date of statement (DD/MM/YYYY)	

A6 BASIC REQUIREMENTS

Item	Protocol Feature	Reference	Status	Support
IPN_naming	Use of 'IPN': EID Naming Scheme	RFC 6260 section 2.1	M	
CBHENodeNo	Use IPN node numbers assigned by SANA	This document: 3.2.2	M	
CBHE Service No	Use CBHE service numbers assigned by IANA / SANA	This document: 3.2.3	M	
CBHE Encoding of EIDs by CLAs	Encoding of BP endpoints via CBHE	RFC 6260 section 2.2; this document section B2.2	O	
bpECOS	Extended Class of Service Block as defined in reference [4]	This document: 3.3, C2, C3	O	
bpTime	Precision of onboard time	RFC 5050 section 6.1; this document: 3.4	M	
LTP 'raw' CL adaptor		This document: B3.1.2.1	O.1	
LTP SDA CL adaptor		This document: B3.1.2.1	O.1	
UDP CL adaptor		This document: B4	O.1	
bpInitRegistration	BP registration initialization service	This document: 4.1	M	

Item	Protocol Feature	Reference	Status	Support
bpTerm Registration	BP terminate initialization service	This document: 4.1	M	
bpSwitchRegistration	BP switch registration state service	This document: 4.1	M	
bpTransmitBundle	BP transmit a bundle to an identified bundle endpoint service	This document: 4.1	M	
bpCancelTransmission	BP cancel a transmission service	This document: 4.1	M	
bpPollRegistration	Poll a registration that is in Passive state	This document: 4.1	M	
bpDeliverRecvdBundle	Deliver received bundle service	This document: 4.1	M	
bpBundle	Bundle structure	RFC 5050: sections 3.1, 4.0, 4.2, 4.4, and 5.8.8	M	
bpBlock	Primary Bundle Block structure	RFC 5050: sections 4.1, 4.5, 4.5.1, 4.5.2	M	
bpBlockPayload	Payload Bundle Block Structure	RFC 5050: sections 3.1, 4.5, 4.5.3, and 4.7	M	
SDNV	Self-Delimiting Numeric Values	RFC 5050: section 4.1	M	
SDNVLargeValue	SDNV encoded value larger than $2^{64}-1$	RFC 5050: section 4.1	O	
bpPriority	Bundle Priority	RFC 5050: section 4.2; this document: annex C	M	
bpAccept	Bundle Acceptance	RFC 5050: sections 4.2, 4.5.1, 4.5.2, 5.6, 5.7, 5.9, 5.10, 5.13	M	
bpTransmission	Bundle Transmission Service	RFC 5050: sections 3.3, 4.3, 5.15, 5.2	M	
bpForward	Bundle Forwarding	RFC 5050: sections 4.2, 5.1, 5.3, 5.4, 5.4.1, 5.4.2, 5.5	M	
bpDelete	Bundle deletion procedures	RFC 5050: sections 3.1, 4.2, 5.13, and 5.14	M	
DictionaryArray	Dictionary byte array	RFC 5050: sections 4.4 and 4.7	M	
bpExtnBlk	Implementation supports extension blocks	RFC 5050: section 4.6	M	

Item	Protocol Feature	Reference	Status	Support
Admin Record Structure	Administrative Record Definition	RFC 5050: sections 5.1, 6.0, 6.1, and 6.2	M	
Admin Record Processing	Administrative Record Processing	RFC 5050: sections 6.1.1, 6.1.2, and 6.3	M	
Admin Record Generation	Administrative Record Generation	RFC 5050: sections 5.1 and 6.2	M	
Connectivity	Intermittent connectivity conditions	RFC 5050: section 1	M	
LateBinding	Late Binding	RFC 5050: section 1	M	
bpDeliveryFail	Bundle delivery failure	RFC 5050: section 1	M	
CustodySigProc	Custody Signal Processing	This document: 6.2.1.3b)	M	
	Custody countdown timer		O	
	Custody transfer failure action		O	
EIDRegisterNode	Registration of node in an endpoint	RFC 5050: section 3.3	M	
EIDTerminateNode	Termination of node in an endpoint	RFC 5050: section 3.3.	M	
EIDSwitchNode	Switching node registration between Active & Passive	RFC 5050: section 3.3	M	
EIDPollNode	Polling node registration	RFC 5050: sections 3.3 and 5.16	M	
EIDFormat	EIDs may be expressed in some internationalized manner (IRI)	RFC 5050: section 4.4	O	
BlkFwdError	Blk forward w/o processing flag may be optionally cleared by another node that receives the bundle and can process that block	RFC 5050: section 4.6	O	
Dictionary revision	Modification of the BP dictionary to support changes in custodian	RFC 5050 section 4.7; this document: 6.2.1.2.2	O	

Item	Protocol Feature	Reference	Status	Support
Fragmentation	Splitting a bundle into multiple fragments (each of which is a bundle in its own right)	RFC 5050 section 5.8	O	
Reassembly	Reassembly of bundle fragments into the original bundle (performed at the destination)	RFC 5050 section 5.9	M	
MIB_state	Bundle State Information	This document: table F-1	M	
MIB_errors	Error and Reporting Information	This document: table F-2	M	
MIB_registration	Registration Information	This document: table F-3	M	
MIB_CL_info	Convergence-Layer Information	This document: table F-4	M	
MIB_Config	General Configuration Information	This document: annex F	M	

ANNEX B

CONVERGENCE LAYER ADAPTERS

(NORMATIVE)

B1 OVERVIEW

This annex describes various Convergence Layer Adapters (CLAs) to support mission operations both in space and on the ground. There are many possible convergence layer protocols to support the various communications interfaces with which the Bundle Protocol may interact. This annex is in no manner comprehensive or rigorous but contains CCSDS supported CLAs that have been demonstrated under various environments, have been requested to be included at the time of this writing, and appear applicable to CCSDS users.

When a specific CLA appears to contradict the specification for that specific convergence layer or the Bundle Protocol specification, the CLA will be assumed to be invalid.

B2 CONVERGENCE LAYER ADAPTERS

B2.1 AVAILABLE CL ADAPTERS

Compliant implementations shall implement at least one of the CLAs in this section.

B2.2 COMPRESSED BUNDLE HEADER ENCODING

Convergence Layer Adapters shall support the compressed bundle header encoding mechanisms of RFC 6260.

B3 LTP CONVERGENCE LAYER ADAPTER

B3.1 ENCAPSULATION OF BUNDLES IN LTP BLOCKS

B3.1.1 General

When sending/receiving bundles using LTP (reference [9]) at the convergence layer, bundles shall be encapsulated in LTP blocks as described in the following subsections.

B3.1.2 RELIABLE TRANSMISSION VIA LTP

B3.1.2.1 For reliable bundle transmission, bundles shall be encapsulated in LTP blocks containing only red-part (reliable) data.

B3.1.2.2 Bundles shall be encapsulated either

a) as a single bundle per LTP block with no leading or trailing bytes: in this case the Destination LTP Client Service ID shall be the service ID for 'Bundle Protocol' as specified in the SANA LTP Client Service ID Number Registry (reference [5]);

b) according to the Client Operations section (section 7) of the LTP-for-CCSDS Book (reference [6]): the Destination LTP Client Service ID provided by the LTP CLA to the LTP service shall be the service ID for 'Bundle Protocol' as specified in the SANA LTP Client Service ID Number Registry (reference [5]).

NOTE – In this case the LTP SDA service will use Client Service ID 2 (Service Data Aggregation) as the client service ID for the LTP block; the transmitted LTP block will contain the LTP Client Service ID for 'Bundle Protocol' as the first bytes of the payload.

B3.1.3 UNRELIABLE TRANSMISSION VIA LTP

For unreliable bundle transmission, bundles shall be encapsulated into LTP blocks containing only green-part (unreliable) data. In this case one bundle shall be encapsulated in each LTP block with no leading or trailing bytes. The LTP Client Service ID shall be the service ID for 'Bundle Protocol' as specified in the SANA LTP Client Service ID Number Registry (reference [5]).

B4 UDP CONVERGENCE LAYER ADAPTER—ENCAPSULATION OF BUNDLES IN UDP DATAGRAMS

When sending/receiving bundles using UDP at the convergence layer, bundles shall be encapsulated in UDP datagrams as follows:

a) UDP checksums shall be enabled;

b) each bundle shall be encapsulated into one UDP datagram with no additional bytes;

NOTE – If the bundle to be encapsulated is larger than the maximum UDP MTU size, the bundle needs to be fragmented at the bundle layer before transmission.

c) all implementations should use UDP port 4556/UDP;

d) all implementations should ensure that the traffic sent by the UDP convergence layer adaptor does not adversely affect other traffic on the network;

NOTE – Network characteristics can best be managed on a closed network or a network with reserved bandwidth; or the utilization of congestion control procedures as described in RFC 5405 (reference [7]) can be adopted.

e) bundle protocol agents should endeavor to send bundles of such a size as not to require fragmentation by the IP layer.

NOTE – In practice this generally means keeping the size of the IP datagram (including the IP and UDP headers, plus the bundle) to less than 1500 bytes. When using UDP as the convergence layer protocol, bundles are limited to a maximum size of 65,535 bytes (including all of the bundle blocks, the 8-byte UDP header, and the IP header (20 bytes for IPv4, 40 bytes for IPv6).

B5 CCSDS ENCAPSULATION SERVICE CONVERGENCE LAYER ADAPTER—ENCAPSULATION OF BUNDLES VIA THE CCSDS ENCAPSULATION SERVICE

When sending/receiving bundles using the CCSDS Encapsulation Service (reference [8]) at the convergence layer, bundles shall be encapsulated via the Encapsulation Service specified in the SANA Protocol Identifier for Encapsulation Service Registry (reference [5]) as follows:

a) each bundle shall be presented as the data unit of one invocation of one ENCAPSULATION.request function of the Encapsulation Service with no additional leading or trailing bytes;

b) the Data Unit Loss Flag (if present at the receiver) may be used by the receiving CLA in an implementation-specific manner.

ANNEX C

EXTENDED CLASS OF SERVICE EXTENSION SPECIFICATION

(NORMATIVE)

C1 INTRODUCTION

C1.1 OVERVIEW

This annex describes an extension to the DTN BP (reference [1]) that marks bundles with class-of-service designators beyond those defined for the BP primary block. The extended class-of-service designators consist of an 'ordinal' number that provides fine-grained prioritization of bundles, a 'critical' flag, a 'best-efforts' flag, and an optional flow label.

C1.2 BACKGROUND

Extended Class of Service (ECOS) is an extension to the DTN BP that marks bundles with class-of-service designators beyond those defined for the BP primary block.

The Bundle Protocol specification defines a single designator for a bundle's class of service:

- Priority, a value in the range 0 through 2, with higher values indicating greater urgency: 0 = 'bulk', 1 = 'normal', 2 = 'expedited'. Priority level 3 is reserved for future use.

For some applications, such as space flight operations, additional variation in class of service may be required:

- Many more levels of priority may be needed, enabling more fine-grained control over the precedence of user-selected application data types in the progress of bundles through the network.
- A way of indicating 'emergency' traffic may be needed. Emergency traffic is not merely high-priority: it is so important that the user is willing to incur the network overhead of transmitting the bundle along every potential route to its destination, rather than only on the route that would normally be selected as the 'best' route according to the applicable routing value function. This expedient ensures that the bundle arrives at its destination in the least possible time, regardless of how accurately the routing system reckons end-to-end latency on any given route: the bundle arrives by whatever turns out to be the fastest route, as well as by all others.
- There may be a need to request that all nodes forwarding the bundle use convergence layer protocols that perform retransmission upon detected loss of data.

– There may alternatively be a need to request that all nodes forwarding the bundle use convergence layer protocols that do not perform retransmission upon detected loss of data. This designation may be important for bundles carrying application data for which timeliness of delivery is more important than certainty: retransmitted 'old data' may be a waste of bandwidth that could instead be used to convey new data of greater value, or the out-of-order arrival of retransmitted data may degrade the usefulness of streaming data such as audio or video.

– There may be a need for an opaque 'flow label' that can be used by the application to pass a variety of transmission control parameters to the convergence layer protocol.

The ECOS extension to Bundle Protocol is designed to provide these additional class-of-service designators. This specification defines the BP extension block in which the ECOS service class designators are conveyed, and the procedures to be performed on origination and reception of a bundle containing an ECOS block.

C2 ECOS BLOCK FORMAT

The ECOS extension block shall conform to sections 4.5.2 and 4.6 of reference [1], constrained as follows:

a) Block type code shall be as assigned by IANA.

b) The following block processing control flag shall be set to '1':

Bit 0: block shall be replicated in every fragment.

NOTE – The setting of other block processing control flags, where not mandated by the Bundle Protocol specification, is an implementation matter.

c) The block shall contain no EID references.

d) Block data length shall be 2 + *N*, where *N* is zero if the ECOS block contains no flow label (as described below) and is otherwise the length of the SDNV in which that flow label is represented.

e) The block data of the ECOS block shall comprise at least two and possibly three fields.

f) The first field of the block data shall be an 8-bit 'flags' byte. The bits of the flags byte shall signify the following conditions:

1) The 0x01 bit, if 'True', shall indicate that the bundle is 'critical': the bundle protocol agent is requested to forward one copy of the bundle along every path that might get it to its destination.

2) The 0x02 bit, if 'True', shall indicate that the bundle is 'streaming': the bundle protocol agent is requested to forward the bundle on a 'best-efforts' basis, without retransmission.

3) The 0x04 bit, if 'True', shall indicate that the 'ordinal' byte of this ECOS block (the byte immediately following the flags byte) is followed by a numeric 'flow label' in SDNV representation.

4) The 0x08 bit, if 'True', shall indicate that the bundle requires reliable transmission: the bundle protocol agent is requested to forward the bundle using a convergence layer protocol that automatically detects data loss and retransmits lost data.

5) All other bits of the flags byte are reserved for future use.

g) The flags byte shall be followed by an 8-bit 'ordinal' byte, containing an unsigned 'ordinal' number in the range 0–255. For a bundle whose standard class of service is 2 ('expedited'), the ordinal number shall indicate the relative priority of this bundle among all other expedited bundles: ordinal value 100 indicates greater urgency than ordinal value 99, and so on. Ordinal value 255 is reserved for custody signals.

NOTE – For a bundle whose standard class of service is not 2, the ordinal value has no significance.

h) If the 0x04 bit of the ECOS block's flags byte is 'False' then the ordinal byte shall be the last field of the block data. Otherwise, the third and final field of the block data shall be a numeric 'flow label' value in SDNV representation.

NOTE – The significance of the flow label is an implementation matter. Notionally, the flow label is intended to be used to convey quality-of-service information to the convergence layer adapter.

C3 ECOS BLOCK PROCEDURES

C3.1 STRUCTURAL CONSTRAINTS

C3.1.1 Whenever a bundle contains an ECOS block, the ECOS block shall precede the payload block.

C3.1.2 No bundle shall ever contain more than one ECOS block.

C3.1.3 If the ECOS block contains a flow label, then the 0x04 bit of the block's flags byte shall be set to '1' ('True'), and the flow label shall be a numeric value represented as a valid SDNV. Otherwise the 0x04 bit of the block's flags byte shall be set to '0' ('False').

C3.1.4 The ordinal byte of the ECOS block shall contain an unsigned integer in the range 0–255. If the bundle of which the ECOS block is a part is a custody signal, then the value of the ordinal byte shall be 255; otherwise, the value of the ordinal byte shall be in the range 0–254.

C3.2 BUNDLE ORIGINATION

Inclusion of an ECOS extension block in a newly originated bundle is optional; the decision on whether or not to insert an ECOS block into a new bundle is an implementation matter. In the event that a new ECOS block is inserted into a forwarded bundle, the values of the block data fields of that ECOS block are likewise an implementation matter, provided that they conform to this specification.

NOTE – Some or all of those values might be communicated to the bundle protocol agent by the application on whose behalf the bundle is being originated. In this case, the manner in which the application communicates those values is likewise an implementation matter.

C3.3 BUNDLE FORWARDING

C3.3.1 Inclusion of an ECOS extension block in a received bundle that is to be forwarded but contains no ECOS block is optional; the decision on whether or not to insert a new ECOS block into a forwarded bundle is an implementation matter. In the event that a new ECOS block is inserted into a forwarded bundle, the values of the block data fields of that ECOS block are likewise an implementation matter, provided that they conform to this specification.

C3.3.2 The forwarding of a bundle that contains a valid ECOS block, whether locally originated or locally inserted upon reception from another bundle protocol agent, shall be constrained as follows:

a) If the 0x01 bit of the ECOS block's flags byte is set to '1' ('critical'):

   1) Exactly one copy of the bundle shall be forwarded to every neighboring node that has some plausible prospect of being able to forward the bundle toward its final destination without returning it to the local node, as determined by the bundle protocol agent's route computation mechanism.

   NOTE – A conformant bundle protocol agent can at any time arbitrarily decide that there is only one neighboring node that has some plausible prospect of being able to forward a given bundle toward its final destination without returning it to the local node, in effect disabling this feature of ECOS. Such an implementation decision should not be taken lightly, however, as it might put network assets at risk; this expedient should be avoided unless necessary to preserve the continued operation of the bundle protocol agent.

   2) The bundle shall be queued for transmission as if its class of service were 2 ('expedited') and its ordinal value were 254, regardless of the actual values of these fields.

3) The bundle shall not be reforwarded in response to custody refusal, the expiration of a custody transfer timer, the presence of a routing loop in the network, or any other condition. The manner in which this constraint is enforced is an implementation matter.

NOTES

1 Such reforwarding could result in unbounded bundle transmission explosions.

2 One possible implementation approach is to manage a list of the IDs and expiration times of all critical bundles received, removing bundles from the list only as the associated expiration times are reached; since 'critical' bundles should be issued rarely; managing such a list should not be a severe processing burden.

b) If the 0x02 bit of the ECOS block's flags byte is set to '1' ('streaming'), then the bundle protocol agent shall forward the bundle by invoking an adapter for a convergence layer protocol that does *not* perform retransmission of data lost in transit, provided the bundle protocol agent has access to such a convergence layer adapter; otherwise, this flag shall be ignored.

NOTE – Inability to accommodate this request for streaming transmission may cause application data units to arrive out of transmission order at the destination (possibly degrading application performance) and/or cause transmission bandwidth to be wasted on unnecessary retransmission, reducing the effective throughput of the network.

c) If the 0x08 bit of the ECOS block's flags byte is set to '1' ('reliable'), then the bundle protocol agent shall forward the bundle by invoking an adapter for a convergence layer protocol that does perform retransmission of data lost in transit, provided the bundle protocol agent has access to such a convergence layer adapter; otherwise, this flag shall be ignored.

d) If the 0x02 bit and 0x08 bit of the ECOS block's flags byte are both set to '1', then the bundle protocol agent shall forward the bundle by invoking an adapter for a convergence layer protocol that functions as a 'bundle streaming service', provided the bundle protocol agent has access to such a convergence layer adapter: whenever loss is detected in 'best-efforts' transmission, the lost data are retransmitted for eventual out-of-order delivery in background. If the bundle protocol agent has no access to such a convergence layer adapter then the bundle protocol adapter shall forward the bundle using a 'best-efforts' convergence layer protocol if one is available, otherwise using a 'reliable' convergence layer protocol.

e) If the bundle's class of service is 2 (expedited), then:

1) The *effective ordinal byte value* of a given bundle is zero if the bundle has no ECOS block; otherwise it is the ordinal byte value in the bundle's ECOS block.

2) The bundle protocol agent shall forward this bundle only after forwarding all other bundles that are to be forwarded to the same node and have class of service 2 and have effective ordinal byte value that is higher than or equal to the ECOS block's ordinal byte value.

3) The bundle protocol agent shall forward this bundle before forwarding any other bundle that is to be forwarded to the same node and either (a) has class of service 2 and effective ordinal byte value lower than the ECOS block's ordinal byte value or (b) has class of service less than 2.

f) The ECOS block of a received bundle that is to be forwarded to another node shall not be deleted from the bundle.

C4 BUNDLE DELIVERY

When a bundle that contains an ECOS block is delivered to its final destination, the values of ECOS block fields shall have no impact on bundle delivery procedures.

C5 DISCUSSION—SECURITY CONSIDERATIONS

The origination of bundles whose ECOS blocks have the 'critical' flag set to 'True' could increase the impact of a denial of service attack. As with all such attacks, probably the best available defense is to require valid Bundle Authentication Blocks on all received bundles.

C6 IANA CONSIDERATIONS

This specification requests that IANA allocate a codepoint from the Bundle Block Types registry defined in RFC 6255 (reference [10]) for the 'Extended Class of Service Block' defined in this annex.

ANNEX D

AGGREGATE CUSTODY SIGNAL SPECIFICATION

(NORMATIVE)

D1 OVERVIEW

D1.1 GENERAL

This annex defines a new administrative bundle type and a complementary bundle block that carries extra information and enhances custody transfer of BP as defined in RFC 5050. Enhanced custody transfer is particularly important with asymmetric routing and data rates between forward (uplink) and return (downlink) paths in many environments including human and robotic spaceflight.

D1.2 INTRODUCTION

In order to guarantee delivery of data, BP provides for the capability to positively identify a bundle and acknowledge the receipt of that bundle. This capability is provided for each bundle discretely. For links with asymmetric data rates, the acknowledgement on a bundle-by-bundle basis with link asymmetries of two or three orders of magnitude may be onerous. By the aggregation of custody signals, link efficiencies of more than one order of magnitude can be realized.

An Aggregate Custody Signal (ACS) is similar to a normal custody signal in that it signals acceptance or rejection of custody and a reason for this acceptance or rejection. An ACS extends the BP custody mechanism by identifying one or more bundles in a compressed format.

The identification is in the form of blocks or fills, like TCP selective acknowledgments. The ACS block is a contiguous sequence of custody IDs that identify specific bundles. These custody IDs are provided in each bundle's Custody Transfer Enhancement Block (CTEB). An aggregate custody signal is the payload of a bundle with the 'Administrative Record' flag set. As a payload, it is contained inside a payload block. The aggregate custody signal is a new Administrative Record, number 4.

D2 DEFINITION OF TERMS

A pending ACS is a logical entity that shall contain the following:

a) custodian EID;

b) reason code;

c) ‘successful’ flag;

d) ACS generation countdown timer;

e) bundle IDs.

An Aggregate Custody Signal is an administrative record that shall have:

a) an administrative record type 4 for ‘Aggregate Custody Signal’;

b) Administrative Record Flag ‘record is for a fragment’ cleared.

D3 ACS COMPONENT FORMATS

D3.1 OVERVIEW

An ACS function is similar to a normal custody signal in signaling acceptance or rejection of custody and a reason for this acceptance or rejection. An ACS provides additional benefits by providing custody signals for one or more bundles in a compressed format.

The identification is in the form of blocks or fills, like TCP selective acknowledgments. A block is a contiguous sequence of custody IDs that identify bundles. The custody IDs come from Custody Transfer Enhancement Blocks.

All block formats and behavior are consistent with RFC 5050.

D3.2 AGGREGATE CUSTODY SIGNAL

D3.2.1 An aggregate custody signal shall be the payload of a bundle with the ‘Administrative Record’ flag set.

NOTE – As a payload, it is contained inside a payload block.

D3.2.2 The aggregate custody signal shall be the Administrative Record, number 4.

0x04	Status
Left edge of first fill*	Length of first fill*
Difference between right edge of first fill and left edge of second fill*	Length of second fill*
• • •	
Difference between right edge first N−1 and left edge of fill N*	Length of fill N*

* Field is an SDNV

Figure D-1: ACS Payload Block Definition

D3.2.3 The first field shall identify administrative record type 4.

D3.2.4 The second field shall be a 'Status' byte encoded in the same way as the status byte for administrative records in RFC 5050, using the same reason codes.

D3.2.5 The third field shall be the custody ID of the left edge of the first fill, encoded as an SDNV.

D3.2.6 The fourth field shall be the length of the first fill, encoded as an SDNV, indicating the number of contiguous custody IDs described by this fill.

NOTE – All fields beyond this are optional: if there is only one fill in this ACS, then the payload block ends. After this, fills continue with a difference field and a length field.

D3.2.7 The next field shall be the difference between the right edge of the first block and the left edge of the second block. It is encoded as an SDNV.

D3.2.8 The next field shall be the length of the second block, encoded as an SDNV.

D3.3 CUSTODY TRANSFER ENHANCEMENT BLOCK

D3.3.1 A CTEB is required for each bundle that is to be supported by ACS and is the responsibility of the accepting bundle protocol agent that supports ACS processing.

D3.3.2 For bundle protocol agents in the network which are not ACS aware, the block flags must be set to ensure that the block is passed through the network unimpeded.

0x0a	Block Flags*	Block Length*
Custody ID*		CTEB Creator Custodian EID†

* Field is an SDNV † Field is variable length

Figure D-2: CTEB Block Definition

D3.3.3 There shall be only one CTEB per bundle.

D3.3.4 The first field shall identify block type 0x0a.

D3.3.5 The second field shall be block flags encoded as an SDNV.

D3.3.6 The third field shall be the block length encoded as an SDNV.

D3.3.7 The fourth field shall be a non-negative integer, an identifier encoded as an SDNV, which uniquely identifies a bundle for this custodian at this instant in time.

D3.3.8 The fifth field shall be the custodian ID of the creator of the CTEB.

D4 ACS BEHAVIOR

D4.1 DISCUSSION

Custody signals are a verbose form of bundle acknowledgement. Whereas a ***minimum*** custody signal is composed of 43 bytes, of which 20 bytes are the primary bundle block, 22 bytes are custody signal for a single bundle, and a single byte can represent a scheme specific node, as defined by RFC 5050.

The use of ACS allows for a high degree of compression because of two factors: Custody ID from a CTEB, and the exclusion of the overhead of a primary bundle block for each custody signal. Time is omitted since it is not used for processing but is logged and can be approximated by the ACS bundle creation time. Therefore an ACS bundle single acknowledgement is at most 13 bytes:

- 1 byte for the Administrative Record Type and Flags per RFC 5050;
- 1 byte for status per RFC 5050;
- 10 bytes for the left edge of the first fill assuming that $2^{62}-1$ is the largest Custody ID;
- 1 byte for the length of the first fill of a single aggregated bundle.

Additional bundles aggregated will increase the compression by only adding the aggregated bundle custody signal without the overhead of the primary bundle block.

Fragmentation of a bundle creates additional bundles of a smaller size with proportionate fragment offsets and fragment lengths as identified in RFC 5050. Therefore a fragmented ACS is consistent with a fragmented bundle where the payload is the ACS 'Failed' or 'Succeeded' list fragment.

D4.2 ACS IMPLEMENTATION

D4.2.1 Discussion

There are four possibilities when a bundle protocol agent accepts a bundle with a CTEB.

a) For an intermediate node which is not ACS capable and accepts custody, the bundle protocol agent ignores the CTEB and updates the custodian field in the primary bundle block. Since the CTEB custodian is not updated, the CTEB is invalid, and the next ACS-capable bundle protocol agent will delete the CTEB.

b) For an intermediate node which is not ACS capable and does not accept custody, the bundle protocol agent forwards the bundle without change. The CTEB is not recognized.

c) For an intermediate node which is ACS capable and does not accept custody, possibility b) is the mode of operation. A node may not accept a bundle for any of a number of reasons as defined in RFC 5050.

d) For an intermediate node which is ACS capable and accepts custody, the bundle protocol agent compares the CTEB custodian with the primary bundle block custodian. If they are different, the CTEB is invalid and deleted. For identical custodians, the primary bundle block and CTEB are updated with the new custodian by the bundle protocol agent, and custody aggregation is utilized to improve link efficiency.

Item d) bounds the defining set of capabilities unique to ACS. By accepting a bundle for custody transfer, an ACS-capable bundle protocol agent will process the bundle per RFC 5050, section 5.10.1. However, instead of the normal custody signaling, the CTEB identifies in shortened form the specific bundle by custody ID. Figure D-3 features a detailed ACS processing flow of the following requirements.

D4.2.2 Requirements

D4.2.2.1 Non-ACS-aware bundle protocol agents shall process ACS-supporting bundles per RFC 5050 section 5.10.

D4.2.2.2 For ACS-aware bundle protocol agents which do not accept custody of ACS:

a) for bundles without a valid CTEB block as identified in RFC 5050 section 5.10, the bundle protocol agent shall generate a ‘Failed’ status;

b) for bundles with a valid CTEB:

1) the bundle protocol agent shall aggregate ‘Failed’ status into a single bundle as identified in D3.2:

i) the aggregation of ‘Failed’ status shall not exceed the maximum allowed bundle size;

ii) the time period for aggregation of bundle status shall not exceed the maximum allowed;

2) the bundle protocol agent shall transmit an ACS as identified in RFC 5050 section 5.10;

3) the bundle protocol agent shall delete, upon successful transmission of an ACS signal, the associated timer and pending ACS ‘Failed’.

D4.2.2.3 For ACS-aware bundle protocol agents which do accept custody of ACS:

a) the bundle protocol agent shall generate a ‘Succeeded’ status for bundles without a valid CTEB block as identified in RFC 5050 section 5.10;

b) the bundle protocol agent shall update the custodian of the Primary Bundle Block and the CTEB as identified in D3.3;

c) for bundles with a valid CTEB:

1) the bundle protocol agent shall aggregate ‘Succeeded’ status into a single bundle as identified in D3.2:

   i) the aggregation of ‘Succeeded’ status shall not exceed maximum allowed bundle size;

   ii) the time period for aggregation of bundle status shall not exceed the maximum allowed;

2) the bundle protocol agent shall delete, upon successful transmission of an ACS signal, the associated timer and pending ACS ‘Succeeded’.

D4.2.2.4 A non-ACS-aware bundle protocol agent shall forward unchanged an ACS to the originating Custody EID.

D4.2.2.5 An ACS-aware bundle protocol agent that receives an ACS shall retrieve each bundle ID associated with each Custody ID.

D4.2.2.6 An ACS-aware bundle protocol agent that receives an ACS shall execute RFC 5050 section 6.3 for each bundle ID of the ACS signal.

D4.2.2.7 An ACS-aware bundle protocol agent shall utilize the ACS bundle timestamp time as the ‘Time of Signal’ when executing RFC 5050 section 6.3.

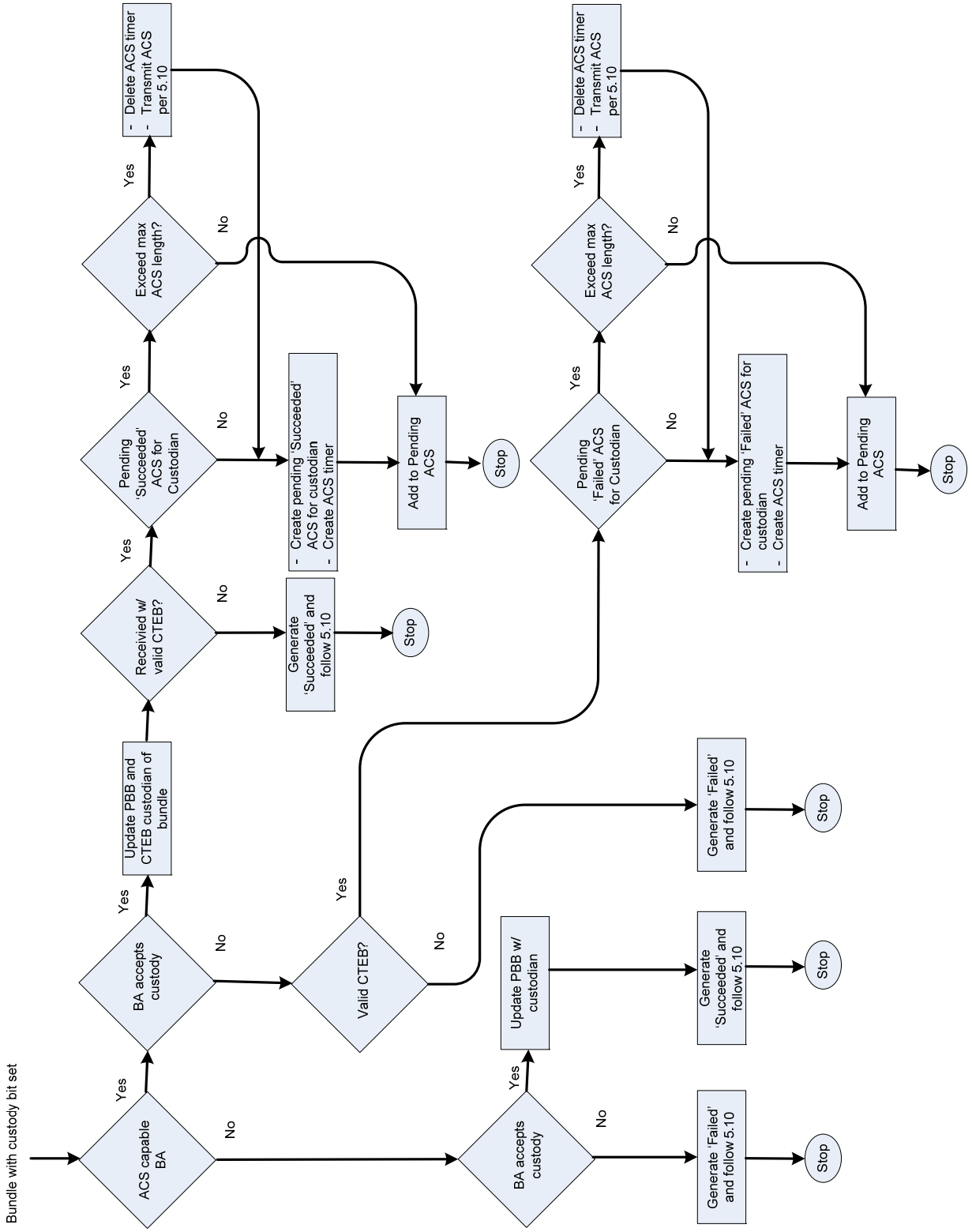

Figure D-3: ACS Processing Flow

ANNEX E

DELAY-TOLERANT PAYLOAD CONDITIONING SPECIFICATION

(NORMATIVE)

E1 OVERVIEW

E1.1 GENERAL

This annex describes an application service that utilizes the Delay-Tolerant Networking (DTN) Bundle Protocol in order to provide DTN end-to-end services that are similar to those provided by 'transport' protocols such as the Transmission Control Protocol (TCP) of the Internet.

BP accomplishes source-to-destination bundle delivery, possibly over heterogeneous networks, in a manner that is functionally analogous to the Internet Protocol (IP); as such it supports neither end-to-end acknowledgment and retransmission nor delivery of data in transmission order without omission or duplication. The Delay Tolerant Payload Conditioning (DTPC) protocol offers delay-tolerant support for these services.

E1.2 INTRODUCTION

A core principle of the design of the Internet architecture is the 'End to End Argument' first articulated by Salzer, Reed, and Clark in 1981 (reference [H2]). The Argument, in essence, is that functionality required by the applications at the endpoints of a data exchange should normally be provided by mechanisms implemented at those endpoints rather than at intermediate points in the end-to-end path. This is not only because it is inefficient to impose the costs of those mechanisms on all applications (by requiring that shared intermediate nodes support them) when only a subset benefit from them, but also because no standard infrastructural mechanisms can be guaranteed to offer all required levels of performance to all imaginable applications, so the mechanisms will in some cases need to be implemented at the endpoints anyway.

The Argument has been validated countless times throughout the history of the Internet's development and deployment. Nonetheless it is qualified:

> Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.

In a network that includes links that are subject to extremely long signal propagation delays and/or transient but frequent and lengthy lapses in connectivity, the end-to-end data interchange required to implement such functions may be so time-consuming as to impose

unacceptable performance degradation. In that case—that is, in a delay-tolerant network—it may in practice be mandatory to implement functions such as acknowledgment and retransmission at layers of the stack below the Network Layer. The resulting performance enhancement is not merely advantageous; it is a precondition to effective operation of the network.

However, even in a delay-tolerant network some end-to-end functionality may be possible, and where possible it may be highly desirable. This annex describes Delay Tolerant Payload Conditioning (DTPC), an application service protocol that offers delay-tolerant support for several end-to-end services to applications that may require them:

- delivery of application data items in transmission (rather than reception) order;
- detection of reception gaps in the sequence of transmitted application data items;
- end-to-end positive acknowledgment of successfully received data;
- end-to-end retransmission of missing data, driven by timer expiration;
- suppression of duplicate application data items;
- aggregation of small application data items into large bundle payloads, to reduce bundle protocol overhead;
- application-controlled elision of redundant data items in aggregated payloads, to improve link utilization.

The high round-trip data exchange latencies that characterize delay-tolerant networks may result in DTPC data delivery that is neither as efficient nor as timely as the delivery of data by 'raw' BP. The purpose of the present specification is to provide standardized support for those DTN applications that require Transport Layer services and can tolerate those performance limits.

E1.3 DEFINITIONS OF TERMS

E1.3.1 DTPC Endpoint

DTPC is a BP application; more precisely, each DTPC protocol entity is one component of the application-specific element of some BP node's application agent. As such:

- A DTPC protocol entity sends data by requesting that its node's BP agent send a bundle whose payload is a DTPC protocol data unit.
- When requesting this transmission, the DTPC protocol entity must identify the BP endpoint that is the destination of the bundle. That destination BP endpoint should be a *DTPC endpoint*, i.e., a BP endpoint that is configured such that delivery of a bundle subject to the delivering node's membership in this endpoint will result in reception of the bundle's payload by a DTPC protocol entity.

– A DTPC protocol entity receives data when a bundle whose payload is a DTPC protocol data unit is delivered at its BP node, subject to the node's membership in the BP endpoint that was identified as the bundle's destination.

At most one (1) DTPC protocol entity may reside in the application-specific element of any single BP node's application agent.

E1.3.2 Topic

A *topic* is a semantic association between peer instances of DTPC user applications, identified by a single number called a *topic ID*. Topics characterize types of data exchanged during the operation of DTPC user applications. They are globally defined; all topic definitions must be provided to all DTPC protocol entities.

E1.3.3 Application Data Item

An *application data item* is a bounded array of octets, characterized by a topic ID, that encodes some data item of significance to user applications that have an interest in the indicated topic. DTPC communication is initiated when an application data item is presented to DTPC by some DTPC user application instance for transmission to a specified peer DTPC user application instance (implicitly identified by a destination DTPC endpoint ID, as discussed below), subject to a specified transmission profile (identified by a profile number, as discussed below).

E1.3.4 DTPC Payload

A *DTPC payload* (or, for this specification, simply 'payload') is a collection of application data items, possibly on a variety of topics, that were all presented for transmission at the same BP node and are all destined for the same DTPC endpoint and subject to the same transmission profile.

E1.3.5 DTPC PDU

A *DTPC Protocol Data Unit* (DPDU) is a unit of DTPC protocol activity, the exchange of DTPC data between two DTPC protocol entities.

DPDUs are of two types: data and acknowledgment. A *data PDU* contains a single DTPC payload. An *acknowledgement PDU* serves to announce reception of a single data PDU.

Each data PDU is tagged with a DTPC payload sequence number, assigned by the 'payload aggregator' (discussed below) that created the DPDU.

Every DPDU is encapsulated in a single BP bundle.

E1.3.6 Supported DTPC Services

The conduct of each DTPC transmission is constrained by the services requested for that transmission. The two *supported DTPC services* are:

- Transport service: data are delivered in transmission order without duplications or omissions, except that under extreme conditions some data omissions may be unavoidable.
- Optimization service: application data items are aggregated into larger payloads to reduce bundle protocol overhead, and redundant application data items in an aggregated DTPC payload are deleted under application control to reduce waste of transmission bandwidth.

Either one, or both, of the supported DTPC services may be requested whenever an application presents an application data item to DTPC for transmission in the context of a given *transmission profile*, as discussed below.

Whenever transmission is requested in the context of a transmission profile that requests transport service, as discussed below, the destination endpoint must be a 'singleton' endpoint.

Acknowledgment PDUs are transmitted only upon reception of data PDUs for which transport service was requested, which for brevity are here termed *transport PDUs*.

The operation of DTPC's optimization service may result in the aggregation of multiple application data items in the payload of a single data PDU.

E1.3.7 Transmission Profile

A *transmission profile* (or, for this specification, simply 'profile') is a set of BP transmission request parameter values and DTPC control parameter values. Included in the definition of each profile is a number, termed a 'Profile ID', which uniquely identifies this profile.

Values for the following BP transmission request parameters are included in each transmission profile:

- Custody Transfer Requested (a Boolean value);
- Lifetime (in seconds);
- Report-to Endpoint ID;
- BP Class-of-service;
- BP Extended Class-of-Service;
- Types of Status Reports that are requested.

These profile values will constrain bundle issuance whenever DTPC asks BP to transmit a data PDU in the context of this profile.

Values for the following DTPC control parameters are also included in each transmission profile:

- Retransmission Limit. The maximum number of times any single DTPC data PDU may be retransmitted by the DTPC entity. A value of zero indicates that DTPC transport service is not requested.
- Aggregation Size Limit. The size threshold for concluding aggregation of a DTPC payload and requesting transmission of that payload: when the size of an aggregated DTPC payload equals or exceeds this limit, aggregation of that payload must cease and the aggregated PDU must be handed down to BP for transmission. A value of zero indicates that DTPC optimization service is not requested.
- Aggregation Time Limit. The time threshold for concluding aggregation of a DTPC payload and requesting transmission of that payload: when the number of seconds that have elapsed since initialization of an aggregated DTPC payload equals or exceeds this limit, aggregation of that payload must cease (regardless of the size of the aggregated payload) and the aggregated PDU must be handed down to BP for transmission. A value of zero indicates that DTPC optimization service is not requested.

E1.3.8 Topic Registration

A *topic registration* is a state machine that characterizes a given DTPC user application instance's claim in a given topic. In each bundle node, any number of registrations may be concurrently associated with a given application, but no more than one registration may be associated with a given topic at any time.

E1.3.9 Topic Aggregator

A *topic aggregator* is a notional element of a payload aggregator (explained below) that manages outbound application data items. A topic aggregator is characterized by a topic ID; no two topic aggregators in any single payload aggregator may share the same topic.

A topic aggregator manages a conceptual list, for its associated topic, of all application data items that have been presented to the payload aggregator for transmission to a specified destination endpoint, subject to a specified transmission profile, and have not yet been transmitted.

E1.3.10 Payload Aggregator

A *payload aggregator* is a notional DTPC protocol entity element that controls the transmission of application data items. A payload aggregator is characterized by a destination DTPC endpoint ID and a transmission profile ID; no two payload aggregators in any single DTPC protocol entity may share the same destination endpoint and profile.

Each payload aggregator constructs a series of DTPC payloads, one at a time. When an application data item is presented to DTPC for transmission, it is passed to the payload aggregator for the destination endpoint and profile that are specified for that application data item. The aggregator in turn passes the item to the topic aggregator for the topic that is specified for that application data item, effecting insertion of the application data item into the payload that is currently under construction at this payload aggregator.

If the size of the payload (the sum of the lengths of all application data items currently managed by all topic aggregators for this payload aggregator) exceeds the aggregation size limit for the payload aggregator's transmission profile, the payload aggregator encapsulates the payload in a DTPC data PDU, asks BP to transmit a bundle whose payload is that PDU, and begins constructing its next payload.

E1.3.11 Data PDU Collector

A data PDU collector (or simply 'collector') is a notional DTPC protocol entity element that controls the delivery of application data items. A collector is characterized by a source endpoint ID and a profile ID; no two collectors in any single DTPC protocol entity may share the same source endpoint and profile.

Each collector manages a list of received transport PDUs and has a payload sequence counter that indicates the payload sequence number of the next transport PDU that is to be delivered to user applications from this collector.

When a received data PDU is delivered to DTPC by the BP agent, it is passed to the collector for the source endpoint ID and profile ID that characterize the PDU. If transport service was not requested for this PDU, then the collector immediately delivers to user applications all application data items in the PDU. Otherwise, the collector inserts the data PDU into its transport PDU list, in payload sequence number order (unless the PDU for that payload sequence number is already in the list, in which case the data PDU is simply discarded as a duplicate). If the payload sequence number of the oldest transport PDU in the list matches the current value of the collector's payload sequence counter, then the collector delivers to user applications all application data items in the oldest transport PDU in the list and all subsequent transport PDUs that are in numerically uninterrupted sequence.

E1.3.12 Elision Function

An *elision function* is an application callback function which is provided by an application in the course of a topic registration. When an application data item is presented to a payload aggregator whose associated profile ID indicates that optimization service is requested, the elision function for the data item's topic is invoked after the data item has been inserted into the payload, prior to the check for payload size in excess of the profile's aggregation size limit, to give the application an opportunity to manage the application data items in the payload. It should be noted that only the data items for the applicable topic are exposed to the elision function; the intent of the elision callback capability is to enable the application to, for example, delete or replace time-sensitive payload records which are no longer valid and meaningful.

E2 SERVICE DESCRIPTION

E2.1 SERVICES PROVIDED TO THE APPLICATION

E2.1.1 SUMMARY OF PRIMITIVES

E2.1.1.1 DTPC shall consume the following request primitives:

a) Register.request;

b) Unregister.request;

c) Send.request.

E2.1.1.2 DTPC shall deliver the following indication primitives:

a) Item.indication;

b) Fault.indication.

E2.1.2 SUMMARY OF PARAMETERS

E2.1.2.1 Overview

The availability and use of parameters for each primitive are enumerated in the definitions of primitives below, where optional parameters are identified with square brackets [thus]. The following parameter definitions apply.

E2.1.2.2 User Application

The *user application* parameter shall be a reference to an instance of a user application, such as a task, thread, or process. The syntax of such references is an implementation matter.

E2.1.2.3 Topic ID

The *topic ID* parameter shall be a number that uniquely identifies a defined DTPC topic, as noted in E1.3.2 above.

E2.1.2.4 Elision function

The *elision function* parameter shall be a reference to an elision function, as defined in E1.3.12 above.

E2.1.2.5 Application data item

The *application data item* parameter shall be an application data item as defined in E1.3.3 above.

E2.1.2.6 Application data item length

The *application data item length* parameter shall be the length, in octets, of the application data item.

E2.1.2.7 Destination endpoint ID

The *destination endpoint ID* parameter shall be the ID of the BP endpoint containing the BP node(s) hosting the user application instance(s) to which the application data item is to be delivered, as described in E1.3.1 above.

E2.1.3 Profile ID

The *profile ID* parameter shall be a number that uniquely identifies a defined transmission profile, as noted in E1.3.7 above.

E2.1.4 Source endpoint ID

E2.1.4.1 General

The *source endpoint ID* parameter shall be the ID of the BP endpoint containing the BP node(s) hosting the user application instance(s) from which the application data item was sent.

E2.1.4.2 Fault expression

The *fault expression* parameter shall indicate the nature of an operational fault encountered by DTPC. The syntax of fault expressions is an implementation matter.

E2.1.5 SERVICE PRIMITIVES

E2.1.5.1 Register.request

E2.1.5.1.1 Function

The Register.request primitive shall be used to establish the user application as the sole authorized client for application data items on a specified topic.

E2.1.5.1.2 Semantics

Register.request shall provide parameters as follows:

Register.request	(user application, topic ID, [elision function])

E2.1.5.1.3 When Generated

Register.request may be generated at any time.

E2.1.5.1.4 Effect on Receipt

Receipt of Register.request shall, if approved, cause the DTPC entity to register the indicated user application as the sole associated client for the indicated topic and link the indicated elision function (if any) to that topic.

E2.1.5.1.5 Additional Comments

Only one user application shall be the registered client for a given topic at any time.

E2.1.5.2 Unregister.request

E2.1.5.2.1 Function

The Unregister.request primitive shall be used to terminate the user application's association with a specified topic.

E2.1.5.2.2 Semantics

Unregister.request shall provide parameters as follows:

Unregister.request (topic ID)

E2.1.5.2.3 When Generated

Unregister.request may be generated by any DTPC application at any time.

E2.1.5.2.4 Effect on Receipt

Receipt of Unregister.request shall, if approved, cause the DTPC entity to disassociate the indicated topic from its current registered client, if any, and detach the current elision function (if any) from that topic.

E2.1.5.2.5 Discussion—Additional Comments

Unregistering from a topic enables another user application to register as the client for that topic.

E2.1.5.3 Send.request

E2.1.5.3.1 Function

The Send.request primitive shall be used to present an application data item to DTPC for transmission.

E2.1.5.3.2 Semantics

Send.request shall provide parameters as follows:

Send.request (application data item,
topic ID,
destination endpoint ID,
profile ID)

E2.1.5.3.3 When Generated

Send.request may be generated by any DTPC application at any time.

E2.1.5.3.4 Effect on Receipt

Receipt of Send.request shall, if approved, cause the DTPC entity to insert the indicated application data item into the current payload of the payload aggregator identified by the indicated destination endpoint ID and profile ID.

E2.1.5.3.5 Discussion—Additional Comments

None.

E2.1.5.4 Item.indication

E2.1.5.4.1 Function

The Item.indication primitive shall be used to deliver an application data item that has been received by DTPC.

E2.1.5.4.2 Semantics

Item.indication shall provide parameters as follows:

Item.indication	(application data item, topic ID, source endpoint ID)

E2.1.5.4.3 When Generated

Item.indication shall be generated by a payload collector upon extraction of an application data item from a DTPC payload whose payload sequence number exceeds by one the payload sequence number of the previously received payload (from the indicated source endpoint ID) that was most recently delivered by this payload collector.

E2.1.5.4.4 Effect on Receipt

The effect on reception of Item.indication by a user application is undefined.

E2.1.5.4.5 Discussion—Additional Comments

None.

E2.1.5.5 Fault.indication

E2.1.5.5.1 Function

The Fault.indication primitive shall be used to indicate a DTPC fault condition to the user application.

E2.1.5.5.2 Semantics

Fault.indication shall provide parameters as follows:

Fault.indication (fault expression)

E2.1.5.5.3 When Generated

Fault.indication shall be generated when DTPC encounters a fault condition.

E2.1.5.5.4 Effect on Receipt

The effect on reception of Fault.indication by a user application is undefined.

E2.1.5.5.5 Discussion—Additional Comments

None.

E2.2 SERVICES REQUIRED OF BP

The service primitives and parameters required to access the services of the Bundle Protocol are:

BP.request (Source EID,
Destination EID,
Report-to Endpoint ID,
application data unit,
'custody transfer requested' flag,
lifetime,
class of service,
extended class of service,
types of Status Reports that are requested)

BP.indication (Source EID,
creation time timestamp,
remaining time to live,
application data unit)

E3 PROTOCOL SPECIFICATION

E3.1 GENERAL

E3.1.1 All DTPC Protocol Data Units (DPDUs) shall be transmitted using the Bundle Protocol.

E3.1.2 Reception of a DPDU which is determined to be ill-formed or inappropriate shall cause that DPDU to be discarded immediately and processed no further.

E3.1.3 An 'ill-formed' DPDU shall be one whose structure and/or content do not conform to the DTPC specification. An 'inappropriate' DPDU shall be an acknowledgment PDU whose profile ID and sequence number do not identify a payload that (a) was transmitted by the receiving DTPC entity and (b) has not yet been acknowledged by its destination DTPC entity.

E3.2 REGISTRATION PROCEDURES

E3.2.1 Register

Upon reception of a Register.request primitive:

a) If some user application other than the indicated user application is currently associated with the indicated topic, then a Fault.indication primitive shall be issued.

b) Otherwise, the DTPC entity shall associate the indicated user application with the indicated topic and shall link the indicated elision function to that topic.

E3.2.2 Unregister

Upon reception of an Unregister.request primitive, the DTPC entity shall

a) detach the indicated topic from the user application with which it is currently associated, if any; and

b) detach the indicated topic from the elision function to which it is currently linked, if any.

E3.3 TRANSMISSION PROCEDURES

E3.3.1 General

Upon reception of a Send.request primitive, the DTPC entity shall handle the outbound application data item as follows.

E3.3.2 Payload Aggregator Instantiation

E3.3.2.1 If no payload aggregator identified by the indicated destination endpoint ID and profile ID exists within the DTPC entity, then such a payload aggregator shall be instantiated.

E3.3.2.2 In the course of instantiating the payload aggregator,

a) the payload counter for this payload aggregator shall be initialized to ‘1’;

b) an initial current payload of length zero shall be instantiated for this payload aggregator; and

c) the aggregation deadline for this payload shall be computed and noted.

E3.3.2.3 The aggregation deadline for the payload shall be the sum of the time at which the payload was instantiated and the aggregation time limit for the payload aggregator’s transmission profile.

NOTE – The payload aggregator is an architectural concept, described here as an aid in expressing DTPC operational requirements; the instantiation and operation of payload aggregation are implementation matters.

E3.3.3 Topic Aggregator Instantiation

E3.3.3.1 If no topic aggregator identified by the indicated topic ID exists within the current payload of the payload aggregator for the indicated destination endpoint ID and profile ID, then such a topic aggregator shall be instantiated.

E3.3.3.2 In the course of instantiating the topic aggregator, an empty (conceptual) list of application data items shall be instantiated for this topic aggregator.

NOTE – The topic aggregator is an architectural concept, described here as an aid in expressing DTPC operational requirements; the instantiation and operation of topic aggregation are implementation matters.

E3.3.4 Payload Aggregation

The indicated application data item shall be appended to the application data item list managed by the topic aggregator for the indicated topic, within the payload aggregator for the indicated destination endpoint and transmission profile.

E3.3.5 Elision

If an elision function is currently linked to the indicated topic, then the list of application data items for that topic shall be passed to that function and processing shall be suspended until the conclusion of the elision function.

NOTE – The intent of elision function support is to enable the user application currently registered for the indicated topic to examine the currently aggregated application data items and delete those that are now redundant or invalid. However, the operation of elision functions is opaque to DTPC and the processing performed by an elision function need not be limited to (or even include) the deletion of currently aggregated application data items.

E3.3.6 Payload Size Limit Check

E3.3.6.1 Following conclusion of the elision function (as applicable), the revised size of the current payload, for the payload aggregator for the indicated destination endpoint ID and profile ID, shall be computed. This size shall be the sum of the lengths of all application data items in lists managed by all topic aggregators for this payload aggregator.

E3.3.6.2 If the computed revised size of the current payload exceeds the aggregation size limit for the payload aggregator's transmission profile, then the payload shall be deemed 'complete' and the payload completion procedure described in E3.4 below shall be performed.

E3.4 PAYLOAD COMPLETION

E3.4.1 When the current payload of a payload aggregator is deemed 'complete', the DTPC entity shall conclude the aggregation of that payload as follows.

E3.4.2 A data PDU shall be issued by the payload aggregator:

a) The profile ID in the data PDU shall be the ID of the payload aggregator's transmission profile.

b) The payload sequence number in the data PDU shall be zero if the retransmission limit for the transmission profile of the payload aggregator is zero (indicating that no transport service is requested). Otherwise it shall be the current value of the payload aggregator's payload counter.

c) One topic block shall be appended to the data PDU's content for each of the payload aggregator's topic aggregators that has a non-empty list of application data items.

d) The order of appearance of topic blocks in the content of the data PDU is undefined.

E3.4.3 If the resulting data PDU has zero octets of content, then the data PDU shall simply be discarded and no further data PDU construction procedures shall be performed.

E3.4.4 Otherwise:

a) The data PDU shall be transmitted as described in E3.5 below.

b) If the retransmission limit for the transmission profile of the payload aggregator is zero, then the data PDU shall be discarded; otherwise, the payload aggregator's payload counter shall be increased by 1 and the acknowledgment deadline for this data PDU shall be computed and noted.

E3.4.5 The acknowledgment deadline for the data PDU shall be the sum of the time at which the data PDU was transmitted and the *nominal round-trip time* for the payload aggregator. The nominal round-trip time for the payload aggregator shall be the lifetime noted in the payload aggregator's transmission profile, divided by the retransmission limit plus one.

E3.5 DATA PDU TRANSMISSION

E3.5.1 The DTPC entity shall effect data PDU transmission by delivering a BP.request primitive to BP.

E3.5.2 The source endpoint ID of that request shall be the DTPC endpoint ID asserted for the BP node, an implementation matter.

E3.5.3 The destination endpoint ID of the request shall be the destination endpoint ID of the issuing payload aggregator.

E3.5.4 The application data unit of the request shall be the data PDU.

E3.5.5 The lifetime of the BP.request shall be whichever is greater: either 1 second or else the lifetime noted in the transmission profile of the issuing payload aggregator, less the product of (a) the nominal round-trip time for the payload aggregator and (b) the number of times transmission of this data PDU has previously been requested.

E3.5.6 The other parameter values for the request shall be as indicated in the transmission profile of the issuing payload aggregator.

E3.6 PAYLOAD TIME LIMIT CHECK

When the current time equals or exceeds the aggregation deadline for the current payload of some payload aggregator, that payload shall be deemed 'complete' and the payload completion procedure described in E3.4 above shall be performed.

E3.7 RETRANSMISSION CHECK

E3.7.1 When the current time equals or exceeds the acknowledgment deadline for some outbound data PDU that has not yet been discarded, that data PDU shall be deemed eligible for retransmission.

E3.7.2 If the number of times the data PDU has previously been retransmitted is equal to the retransmission limit, then the data PDU shall be discarded.

E3.7.3 Otherwise:

a) The data PDU shall be retransmitted as described in E3.5 above. Mechanisms for recalling the issuing payload aggregator for a data PDU that is to be retransmitted are an implementation matter.

b) The acknowledgment deadline for this data PDU shall be computed and noted.

E3.7.4 The acknowledgment deadline for the data PDU shall be the sum of the time at which the data PDU was retransmitted and the nominal round-trip time for the payload aggregator that issued the data PDU. The nominal round-trip time for the payload aggregator shall be the lifetime noted in the payload aggregator's transmission profile, divided by the retransmission limit plus one.

E3.8 RECEPTION PROCEDURES

E3.8.1 Upon reception of a BP.indication primitive from BP, the DTPC entity shall interpret the indication's application data unit as a DPDU and shall handle that DPDU as follows.

E3.8.2 If the DPDU's PDU type is 1,

a) the acknowledgment handling procedure described in E3.9 below shall be performed;

b) otherwise, the data PDU handling procedure described in E3.10 below shall be performed.

E3.9 ACKNOWLEDGMENT HANDLING

The outbound data PDU identified by the acknowledgment PDU's payload sequence number, produced by the payload aggregator identified by the acknowledgement PDU's source endpoint ID (as provided in the BP.indication primitive) and the acknowledgment PDU's profile ID, shall be discarded.

E3.10 DATA PDU HANDLING

E3.10.1 Data PDU Collector Instantiation

E3.10.1.1 If no data PDU collector identified by the indicated source endpoint ID (provided in the BP.indication primitive) and the received data PDU's profile ID exists within the DTPC entity, then such a collector shall be instantiated.

E3.10.1.2 In the course of instantiating the collector, the payload counter for this collector shall be initialized to '1' and an empty (conceptual) list of data PDUs shall be instantiated for this payload collector.

NOTE – The payload collector is an architectural concept, described here as an aid in expressing DTPC operational requirements; the instantiation and operation of payload collection are implementation matters.

E3.10.2 Acknowledgment

E3.10.2.1 If the received data PDU's payload sequence number is greater than zero (indicating that transport service is requested), then an acknowledgment PDU shall be transmitted as follows.

E3.10.2.2 An acknowledgment PDU shall be constructed. The profile ID in the acknowledgment PDU shall be the received data PDU's profile ID, and the payload sequence number in the data PDU shall be the payload sequence number in the received data PDU.

E3.10.2.3 The DTPC entity shall effect acknowledgment PDU transmission by delivering a BP.request primitive to BP.

E3.10.2.4 The source endpoint ID of that request shall be the DTPC endpoint ID asserted for the BP node, an implementation matter.

E3.10.2.5 The destination endpoint ID of the request shall be the source endpoint ID as provided in the BP.indication primitive.

E3.10.2.6 The application data unit of the request shall be the acknowledgment PDU.

E3.10.2.7 Determination of the values of all other parameters for the request shall be an implementation matter.

E3.10.3 Unacknowledged Payload Delivery

If the received data PDU's payload sequence number is equal to zero (indicating that transport service is not requested), the data PDU's payload shall be delivered as follows:

a) For each topic block in the content of the data PDU, for which a user application is currently associated with the indicated topic, one Item.indication primitive shall be delivered to that user application for each data item block in the topic block.

b) The source endpoint ID provided in each indication shall be the collector's source endpoint ID.

c) The data PDU shall be deleted.

E3.10.4 Payload Collection

E3.10.4.1 General

If the received data PDU's payload sequence number is greater than zero (indicating that transport service is requested), payload collection procedures shall be performed as specified in E3.10.4.2 through E3.10.4.5.

E3.10.4.2 Duplicate Data Suppression

If the (conceptual) list of data PDUs associated with this collector already contains a data PDU whose payload sequence number is the same as that of the received data PDU, then the received data PDU shall be discarded and no further payload collection procedures shall be performed with regard to this PDU.

E3.10.4.3 Data PDU Insertion

The received data PDU shall be inserted into the list of data PDUs associated with this collector, maintaining ascending payload sequence number order (i.e., oldest first) among the data PDUs in the list.

E3.10.4.4 Placeholder Management

NOTE – The collector's list of data PDUs may conceptually contain not only received data PDUs but also 'placeholder' PDUs representing PDUs that are believed to have been transmitted but have not yet been received. (A placeholder PDU consists of a payload sequence number and no other information. Placeholder PDUs are a conceptual mechanism for dealing with gaps in data reception that may be repaired by data PDU retransmission.)

E3.10.4.4.1 If some placeholder PDU in the list has the same payload sequence number as the received data PDU, then that placeholder PDU shall be deleted.

E3.10.4.4.2 The *PDU collection gap* for the received data PDU shall be the range of all payload sequence numbers that are greater than the payload sequence number of the youngest PDU in the collector's list of data PDUs that is older than this newly received data PDU—or, if there is no such data PDU, that are greater than the current value of the collector's payload counter minus 1—and that are less than the payload sequence number of the received data PDU.

E3.10.4.4.3 All placeholder PDUs with payload sequence number within the PDU collection gap for the received data PDU shall be retained but with revised expiration times. The expiration time of each such placeholder PDU shall be changed to the sum of the creation time timestamp of the bundle in which the received data PDU was delivered (provided in the BP.indication primitive) and the original lifetime of that bundle, minus one second, or the current time, whichever is later. The original lifetime of the bundle in which the received data PDU was delivered shall be computed by adding that bundle's remaining time to live (as provided in the BP.indication primitive) to the time at which the PDU was delivered and then subtracting from this sum the creation time timestamp of the bundle.

E3.10.4.4.4 One placeholder PDU shall be (in concept) inserted into the list for every payload sequence number value that is within the PDU collection gap for the received data PDU.

E3.10.4.4.5 The expiration time of each inserted placeholder PDU shall be noted. The expiration time of each inserted placeholder PDU shall be the sum of the creation time timestamp of the bundle in which the received data PDU was delivered (provided in the BP.indication primitive) and the original lifetime of that bundle (computed as noted above), minus one second, or the current time, whichever is later.

E3.10.4.5 Sequence Check

If the payload sequence number of the first (oldest) data PDU in the collector's list of data PDUs is equal to the current value of the collector's payload counter, then the payload delivery procedure described in E3.11 shall be performed.

E3.11 PAYLOAD DELIVERY

E3.11.1 General

When payload delivery is initiated for a data PDU collector, the two procedures described below shall be performed repeatedly until either the collector's list of data PDUs is empty or the current value of the collector's payload counter is not equal to the payload sequence number of the first (oldest) data PDU in the collector's list of data PDUs.

E3.11.2 Application Data Item Delivery

E3.11.2.1 For each topic block in the content of the first (oldest) data PDU in the collector's list of data PDUs, for which a user application is currently associated with the indicated topic, one Item.indication primitive shall be delivered to that user application for each data item block in the topic block.

E3.11.2.2 The source endpoint ID provided in each indication shall be the collector's source endpoint ID.

E3.11.3 Data PDU Deletion

E3.11.3.1 The first (oldest) data PDU in the collector's list of data PDUs shall be deleted.

E3.11.3.2 The value of the collector's payload counter shall be increased by one.

E3.12 PLACEHOLDER EXPIRATION

E3.12.1 When the current time equals or exceeds the expiration time for a placeholder PDU, then that placeholder PDU shall be deleted from the list of data PDUs in which it resides.

E3.12.2 If, as a result of this deletion, the first item in that list is now a data PDU rather than a placeholder PDU, then the value of the payload counter for the data PDU collector to which that list belongs shall be set to the payload sequence number of the first data PDU in the list, and the payload delivery procedure described in E3.11 shall be performed.

E4 PROTOCOL DATA UNITS

E4.1 DTPC PROTOCOL DATA UNIT FORMAT

E4.1.1 Each DTPC PDU (DPDU) shall consist of a header in fixed format followed by zero or more octets of content.

E4.1.2 DPDUs are of two types:

- data PDUs flow from the sender to the receiver and carry one or more payload records subject to one or more topics;
- acknowledgment PDUs flow from the receiver to the sender and carry zero octets of content. An acknowledgement block acknowledges the successful reception of one DTPC data PDU.

E4.2 DTPC PDU HEADER

E4.2.1 The DPDU header shall consist of the fields shown in table E-1.

E4.2.2 The DPDU header fields shall be transmitted in the order of presentation in table E-1.

Table E-1: DPDU Header Fields

Field	Length (bits)	Values	Comment
Version number	2	'00' for this version of DTPC.	
Reserved	5	Always '00000'	May be redefined in future versions.
PDU type	1	'0' = data '1' = acknowledgment	
Profile ID	variable		Profile ID is an unsigned integer in SDNV representation as defined earlier in this document.
Payload sequence number	variable		Sequence number is an unsigned integer in SDNV representation.

E4.3 DTPC PDU CONTENT

E4.3.1 The content of an acknowledgement PDU shall be an array of zero octets.

E4.3.2 The content of a data PDU shall be a sequence of one or more concatenated *topic blocks* as defined below.

E4.4 TOPIC BLOCK

E4.4.1 A topic block shall consist of the fields shown in table E-2.

E4.4.2 The fields of each topic block shall be transmitted in the order of presentation in table E-2.

Table E-2: Topic Block Fields

Field	Length (bits)	Values	Comment
Topic ID	variable		Topic ID is an unsigned integer in SDNV representation as defined earlier in this document.
Payload record count	variable		Payload record count is an unsigned integer in SDNV representation.
Topic block body	variable		Topic block body is a sequence of *data item blocks* as defined below.

E4.5 DATA ITEM BLOCK

E4.5.1 A data item block shall consist of the fields shown in table E-3.

E4.5.2 The fields of each data item block shall be transmitted in the order of presentation in table E-3.

Table E-3: Payload Record Fields

Field	Length (bits)	Values	Comment
Application data item length	variable		Application data item length is an unsigned integer in SDNV representation as defined earlier in this document.
Application data item	variable		Application data item is an array of octets, of the length noted in the application data item length field.

ANNEX F

BP MANAGED INFORMATION

(NORMATIVE)

F1 BASIC REQUIREMENTS

F1.1 Each BP node shall support a set of managed information that represents the state of the node at a particular time. The minimal set of such information contains those data items identified by RFC 5050 and collected in this annex. This collection of managed information is shown as the MIB on the right of figure 1-1.

F1.2 BP nodes shall support five types of managed information:

a) bundle state information;

b) error and reporting information;

c) registration information;

d) convergence layer information;

e) node state information.

F1.3 In addition to required information, each BP node may choose to provide supplementary information. Each identified managed information item shall identify whether its collection and accurate reporting is required or recommended.

NOTES

1 Representation of, and mechanisms for access to, managed information items will be implementation matters.

2 Individual pieces of managed information may describe related events. Care must be taken when modifying these data to ensure that related data sets remain coherent. For example, when a cumulative counter 'rolls over' or is otherwise reset, related counters should also be reset.

F2 BUNDLE STATE INFORMATION

F2.1 OVERVIEW

Bundles do not have a natural end state within a node: they are forwarded, delivered, or deleted. As such, bundles at rest within a node exist pending a particular action. This set of managed information describes these bundle states and the transitions between them.

F2.2 SUPPORTED TYPES OF BUNDLE STATE INFORMATION

BP nodes shall support the bundle state information itemized in table F-1.

Table F-1: Bundle State Information

Managed Information Item	Description		Req?
Retention Constraints			
Bundles Retained for Forwarding	The number of bundles/bytes associated with the retention constraint ***forward pending*** at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Bundles Retained for Transmission	The number of bundles/bytes associated with the retention constraint ***dispatch pending*** at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Bundles Retained for Custody Acceptance	The number of bundles/bytes associated with the retention constraint ***custody accepted*** at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Bundles Retained for Reassembly	The number of bundles/bytes associated with the retention constraint ***reassembly pending*** at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Priority Counters			
Bulk Bundles Sourced	The number of bundles/bytes generated by this node with the ***bulk*** priority.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Normal Bundles Sourced	The number of bundles/bytes generated by this node with the ***normal*** priority.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Expedited Bundles Sourced	The number of bundles/bytes generated by this node with the ***expedited*** priority.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Bulk Bundles Queued	The number of bundles/bytes with the ***bulk*** priority currently resident on this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Normal Bundles Queued	The number of bundles/bytes with the ***normal*** priority currently resident on this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Expedited Bundles Queued	The number of bundles/bytes with the ***expedited*** priority currently resident on this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Fragmentation			
Fragmentation	The number of bundles that have been fragmented by this node.	Cumulative Bundles	Yes
Number of Fragments	The number of fragments created by this bundle.	Cumulative Bundles	Yes

F3 NODE ERROR AND REPORTING INFORMATION

F3.1 OVERVIEW

Nodes generate reports in response to both anomalous and special events. This set of managed information reports on the number of errors and reports constructed at the node.

F3.2 SUPPORTED TYPES OF ERROR AND REPORTING INFORMATION

BP nodes shall support the error and reporting information itemized in table F-2.

Table F-2: Error and Reporting Information

Managed Information Item	Description		Req?
Bundle Deletions			
No Info Deletions	The number of bundles deleted with the ***No additional information*** reason code.	Cumulative Bundles	No
Expired Deletions	The number of bundles deleted with the ***Lifetime expired*** reason code.	Cumulative Bundles	No
Uni-forwarded Deletions	The number of bundles deleted with the ***Forwarded over unidirectional link*** reason code.	Cumulative Bundles	No
Cancellation Deletions	The number of bundles deleted with the ***Transmission canceled*** reason code.	Cumulative Bundles	No
No Storage Deletions	The number of bundles deleted with the ***Depleted Storage*** reason code.	Cumulative Bundles	No
Bad EID Deletions	The number of bundles deleted with the ***Destination endpoint ID unintelligible*** reason code.	Cumulative Bundles	No
No Route Deletions	The number of bundles deleted with the ***No known route to destination from here*** reason code.	Cumulative Bundles	No
No Timely Contact Deletions	The number of bundles deleted with the ***No timely contact with next node on route*** reason code.	Cumulative Bundles	No
Bad Block Deletions	The number of bundles deleted with the ***Block unintelligible*** reason code.	Cumulative Bundles	No
Bytes deleted	The total number of bytes in all bundles deleted at this node.	Cumulative Bytes	No

Managed Information Item	Description		Req?
Bundle Processing Errors			
Failed Custody Transfers	The number of incoming bundles/bytes whose request for custody was not successful at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Failed Forwards	The number of bundles/bytes that have experienced a forwarding failure at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Abandoned Delivery	The number of bundles/bytes whose delivery has been abandoned at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes
Discarded Bundles	The number of bundles/bytes discarded at this node.	Cumulative Bytes	No
		Cumulative Bundles	Yes

F4 REGISTRATION INFORMATION

F4.1 OVERVIEW

Each node registers in one or more endpoints. These registrations allow for the reception and processing of bundles in the context of the endpoints to which they are addressed.

F4.2 SUPPORTED TYPES OF REGISTRATION INFORMATION

BP nodes shall support the registration information itemized in table F-3.

Table F-3: Registration Information

Managed Information Item	Description	Req?
Identity Information		
Endpoint Identifier	The Endpoint ID of this registered endpoint.	Yes
Activity State	The current state of the EID, at the time the managed information was queried. One of: ACTIVE or PASSIVE.	Yes
Singleton State	Whether this EID is a singleton EID. One of: YES or NO.	Yes
Default Failure Action	The default action to be taken when delivery is not possible. One of: ABANDON or DEFER.	Yes

F5 NODE STATE INFORMATION

F5.1 OVERVIEW

Global node state information provides the context for using other managed information items.

F5.2 SUPPORTED TYPES OF NODE STATE INFORMATION

BP nodes shall support the node state information itemized in table F-4.

Table F-4: Node State Information

Managed Information Item	Description	Req?
Identity Information		
Node Identifier	The Endpoint ID that uniquely and permanently identifies this node.	Yes
Bundle Protocol version number	The number of the version of the Bundle Protocol that is supported at this node.	Yes
Available storage	The number of kilobytes of storage allocated to bundle retention at this node and not currently occupied by bundles.	Yes
Last up time	The most recent time at which the operation of this node was started or restarted.	Yes
Registration count	The number of different endpoints in which this node has been registered since it was last started or restarted.	No
Extension Information (one occurrence per extension)		
Extension name	The name identifying one of the BP extensions supported at this node.	Yes

ANNEX G

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

G1 SECURITY

G1.1 OVERVIEW

The Bundle Protocol as defined by RFC 5050 has factored in security from the outset of its design. The necessary security architecture and services have been developed in an accompanying RFC, the Bundle Security Protocol specification. Because BP was designed for a resource-constrained environment, it is essential to ensure that only those entities authorized to utilize those resources be allowed to do so.

Also, because of the long latencies and delays in the constrained environments which utilize BP, integrity and confidentiality are essential. Without adequate protections in place to ensure that data integrity and confidentiality are maintained, the difficulty in identifying compromised data will be compounded as a result of the unique environment of CCSDS missions.

G1.2 SECURITY CONCERNS WITH RESPECT TO THE CCSDS DOCUMENT

The BP specification (reference [1]) contains a security section (8), which addresses necessary measures to protect bundle protocol data and recommends the use of the Bundle Security Protocol (BSP) of RFC 6257. Four types of security blocks are defined in RFC 6257:

a) Bundle Authentication Blocks (BAB) ensure the authenticity and integrity of bundles on a hop-by-hop basis between bundle nodes.

b) Payload Integrity Blocks (PIBs) ensure the authenticity and integrity of the payload from the PIB security-source to the PIB security-destination.

c) Payload Confidentiality Blocks (PCBs) indicate that the associated payload has been encrypted, and provide for confidential communications between the PCB security-source and the PCB security-destination.

d) Extension Security Blocks (ESBs) provide security for non-payload blocks in a bundle.

G1.3 AUDITING OF RESOURCE USAGE

No mechanisms are defined in this specification to audit or assist with the auditing of resource usage by the protocol.

G1.4 POTENTIAL THREATS AND ATTACK SCENARIOS

No potential threat or attacks scenarios are discussed.

G1.5 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

By not applying the native security of the BP protocol and the extended security of BSP allowed by BP, the system must rely on security measures provided at the CLA interfaces. For space applications these may be non-existent or merely physical because of the lack of integration between payload and ground systems interfaces. If no security is applied at the BP or lower layers, then applications may be open to man-in-the middle attacks, replay attacks, or a general loss of integrity of transported bundles.

G2 SANA CONSIDERATIONS

The recommendations of this document request SANA to create the following registries based on the recommendation of 3.5:

a) the registry named Bundle Protocol Compressed Bundle Header Encoding Node Numbers which consists of a table of parameters:

   1) the initial registry values are not defined;

   2) the registration rule for new values of this registry requires no engineering review, but the request must come from the official representative of a space agency, member of the CCSDS;

b) the registry named Bundle Protocol Compressed Bundle Header Encoding Service Numbers which consists of a table of parameters:

   1) limited initial registry values are defined;

   2) the registration rule for new values of this registry requires an official representative of a space agency member of the CCSDS request followed by expert CESG review.

G3 PATENT CONSIDERATIONS

There are no known patents covering the Bundle Protocol as described in this document and its normative references.

ANNEX H

INFORMATIVE REFERENCES

(INFORMATIVE)

[H1] *Rationale, Scenarios, and Requirements for DTN in Space*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 734.0-G-1. Washington, D.C.: CCSDS, August 2010.

[H2] J.H. Saltzer, D.P. Reed, and D.D. Clark. "End-to-End Arguments in System Design." In *Proceedings of the 2nd International Conference on Distributed Computing Systems (April 8-10, 1981, Paris, France)*, 509-512. Los Alamitos, CA, USA: IEEE Computer Society, 1981.

[H3] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 4. CCSDS Record (Yellow Book), CCSDS A02.1-Y-4. Washington, D.C.: CCSDS, April 2014.

ANNEX I

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

Term	Meaning
AA	application agent
ACS	aggregate custody signal
API	application programming interface
BAB	bundle authentication block
BP	Bundle Protocol
BPA	bundle protocol agent
BSP	Bundle Security Protocol
CBHE	compressed bundle header encoding
CTEB	custody transfer enhancement block
DPDU	DTPC protocol data unit
DTPC	delay tolerant payload conditioning
CLA	convergence layer adapter
DTN	delay tolerant network
ECOS	extended class of service
EID	endpoint identifier
ESB	extension security block
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
IPN	InterPlanetary Network
IRTF	Internet Task Research Task Force
LTP	Licklider Transmission Protocol
MIB	management information base
OSI	Open Systems Interconnection
PICS	protocol implementation conformance statement
PCB	payload confidentiality block
PDU	payload data unit

PIB	payload integrity block
RL	requirement list
RFC	Request For Comment
SANA	Space Assigned Numbers Authority
SDA	service data aggregation
SDNV	self-delimiting numeric values
SDU	service data unit
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	uniform resource identifier