NASA/TM-20240011318

HDTN 7150.2D Class B Software Requirements Specification (SRS)

José Lombay-González
Glenn Research Center, Cleveland, Ohio

Eric A. Brace
HX5, LLC, Brook Park, Ohio

Kevin Carmichael
Kevin Carmichael Consulting LLC, Cleveland, Ohio

Rachel M. Dudukovich, Stephanie Booth, Nadia Kortas, Ethan Schweinsberg,
Prash Choksi, and Shaun M. McKeehan
Glenn Research Center, Cleveland, Ohio

Eugene E. Heard
Bastion Technologies, Inc., Houston, Texas

Amber A. Waid, John J. Nowakowski, Brian J. Tomko, Daniel E. Raible, Joseph G. Ponyik,
Shira Nadile, and Alyssa M. Brewer
Glenn Research Center, Cleveland, Ohio

September 2024

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION.
  Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM.
  Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT.
  Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION.
  Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION.
  Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION.
  English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/TM-20240011318

HDTN 7150.2D Class B Software Requirements Specification (SRS)

José Lombay-González
Glenn Research Center, Cleveland, Ohio

Eric A. Brace
HX5, LLC, Brook Park, Ohio

Kevin Carmichael
Kevin Carmichael Consulting LLC, Cleveland, Ohio

Rachel M. Dudukovich, Stephanie Booth, Nadia Kortas, Ethan Schweinsberg,
Prash Choksi, and Shaun M. McKeehan
Glenn Research Center, Cleveland, Ohio

Eugene E. Heard
Bastion Technologies, Inc., Houston, Texas

Amber A. Waid, John J. Nowakowski, Brian J. Tomko, Daniel E. Raible, Joseph G. Ponyik,
Shira Nadile, and Alyssa M. Brewer
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

September 2024

Level of Review: This material has been technically reviewed by technical management.

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

PREFACE

Space Communications and Navigation (SCaN) is developing new communications technologies to increase the amount of science data returned on future space missions. The High-Rate Delay Tolerant Networking (HDTN) project at NASA Glenn Research Center (GRC) will provide reliable internetworking as a high-speed path for moving data between spacecraft payloads and across communication systems that operate at various rates.

This document describes the requirements that will be implemented in the HDTN software.

DOCUMENT HISTORY LOG

Status (Preliminary/ Baseline/ Revision/ Canceled)	Document Revision	Effective Date	Description
Baseline	N/A	08/21/2024	Initial Release
Revision	A	09/11/2024	Minor Editorial Corrections, HDTN-CR-004

Prepared By:

Jose Lombay-gonzalez
Digitally signed by Jose Lombay-gonzalez
Date: 2024.09.11 14:21:03 -04'00'

José Lombay-González
HDTN Software Lead
NASA John H. Glenn Research Center

Rachel Dudukovich
Digitally signed by Rachel Dudukovich
Date: 2024.09.11 14:49:22 -04'00'

Rachel M. Dudukovich
HDTN Software Engineering Lead
NASA John H. Glenn Research Center

Nadia Kortas
Digitally signed by Nadia Kortas
Date: 2024.09.11 15:06:13 -04'00'

Nadia Kortas
HDTN Software Developer
NASA John H. Glenn Research Center

Stephanie Booth
Digitally signed by Stephanie Booth
Date: 2024.09.11 15:25:44 -04'00'

Stephanie L. Booth
HDTN Software Developer
NASA John H. Glenn Research Center

Concurred By:

Rachel Dudukovich
Digitally signed by Rachel Dudukovich
Date: 2024.09.12 08:43:45 -04'00'

Rachel M. Dudukovich
HDTN Software Engineering Lead
NASA John H. Glenn Research Center

Jose Lombay-gonzalez
Digitally signed by Jose Lombay-gonzalez
Date: 2024.09.12 09:42:06 -04'00'

José Lombay-González
HDTN Software Lead
NASA John H. Glenn Research Center

Eugene Heard (affiliate)
Digitally signed by Eugene Heard (affiliate)
Date: 2024.09.12 11:17:16 -04'00'

Eugene E. Heard, Bastion Technologies
HDTN Software Quality Assurance Lead
NASA John H. Glenn Research Center

Approved By:

John Nowakowski
Digitally signed by John Nowakowski
Date: 2024.09.12 13:03:50 -04'00'

John J. Nowakowski
HDTN Project Manager
NASA John H. Glenn Research Center

Daniel Raible
Digitally signed by Daniel Raible
Date: 2024.09.13 19:00:02 -04'00'

Daniel Raible
HDTN Principal Investigator
NASA John H. Glenn Research Center

TABLE OF CONTENTS

TABLE OF TABLES

1.0 INTRODUCTION

1.1 Identification

Space Communications and Navigation (SCaN) is developing new communications technologies to increase the amount of science data returned on space missions. To expand NASA's exploration and science missions capabilities and satisfy growing requirements on data return, there is a drive to accelerate the infusion of optical communications technology with existing radio frequency (RF) capabilities into one operable network.

Communicating from Earth to any spacecraft is a complex challenge due to the extreme distances involved. When data is transmitted and received across thousands and even millions of miles in space, the delay and potential for disruption or data loss is significant. Delay Tolerant Networking (DTN) is NASA's solution to reliable internetworking for space missions. The High-Rate Data Tolerant Network (HDTN) project at NASA Glenn Research Center (GRC) is developing technology that can act as a high-speed path for moving data between spacecraft payloads and across communication systems that operate at various rates.

1.2 System Overview

The HDTN project aims to develop software to improve space network data throughput to meet future user needs by enhancing communications capability to increase mission science return. The developed software must be robust enough to support robotic and manned missions in NASA's efforts to explore space.

1.3 Document Overview

The Software Requirements Specification (SRS) specifies the requirements for the computer software configuration item (CSCI) being developed by the NASA Glenn Research Center for the HDTN project as defined in the project's Software Development and Management Plan (SDMP, HDTN-PLAN-003) and the methods to ensure each requirement has been met. The intent of the SRS is to document the expected behavior and functionality of the HDTN project needed to fulfill the needs of the stakeholders. This specification will be used to design and verify the HDTN software. In some cases, there are gaps in the project requirements. These gaps are captured within this document as "to be determined" (TBD) or "to be resolved" (TBR). A list of all TBD and TBR references in this document is contained in Appendix D TBD/TBR List. This document follows the guidance provided by the SEPG GRC-SW-7150.5 Requirements Development Process, the HDTN Project document template (HDTN-TPLT-023), and the SEPG's Software Requirements Specification template (GRC-SW-TPLT-SRS).

2.0 APPLICABLE DOCUMENTS

2.1 Reference Documents

This section lists the number and title of all documents referenced in this specification.

Table 2-1 Reference Documents

Document number	Revision	Document title	Release Date
NPR 7150.2	D	NASA Software Engineering Requirements	03/08/2022
GLPR 7150.1	A	Glenn Research Center (GRC) Software Engineering Requirements	07/19/2016
GRC–SW–TPLT–SRS	A	Software Requirements Specification Template	09/01/2011
HDTN-PLAN-003		Software Development and Management Plan	01/18/2024
HDTN-CONOPS-015		HDTN Concept of Operations	***08/08/2024***
HDTN-SWDD-017		HDTN Software Data Dictionary	***TBD-SRS002***
RFC 5050		Bundle Protocol Version 6	11/2007
CCSDS 734.2-B-1		CCSDS Bundle Protocol Specification	9/2015
RFC 9171		Bundle Protocol Version 7	12/14/2022
CCSDS 734.2-P-1.1		CCSDS Bundle Protocol Specification	4/2023
RFC 9172		Bundle Protocol Security (BPSec)	10/11/2023
RFC 9173		Default Security Contexts for Bundle Protocol Security (BPSec)	6/21/2022
CCSDS 734.5-R-2		CCSDS Bundle Protocol Security Specification	9/2023
RFC 5326		Licklider Transmission Protocol	12/8/2022
CCSDS 734.1-B-1		Licklider Transmission Protocol (LTP) for CCSDS	5/2015
RFC 9174		DTN TCP Convergence-Layer Protocol Version 4	1/31/2022
RFC 7122		Datagram Convergence Layers for DTN Bundle Protocol and Licklider Transmission Protocol	3/2014
CCSDS 734.3-B-1		Schedule-Aware Bundle Routing	7/2019
RFC 2119		Keywords for use in RFCs to Indicate Requirement Levels	3/1997
CCSDS 766.3-R-1		Specification for Real-Time Protocol (RTP) as Transport for Audio and Video over DTN	12/2019
CCSDS 766.3-R-2		Specification for RTP as Transport for Audio and Video over DTN	08/2020

3.0 REQUIREMENTS

This section will be composed of sub-sections that capture the requirements generated to satisfy the features allocated to the HDTN project, which are described in the HDTN Concept of Operations (ConOps) (HDTN-CONOPS-015), as well as a description of its modes.

The Delay Tolerant Network environment contains challenges related to intermittent connectivity, long delays, and high error rates. Due to this, using the Bundle Protocol specification is essential to provide robustness and flexibility for communication at an application layer. Bundles contain data from a source to be transmitted to a destination. Convergence Layers facilitate a means to share bundles via the different transport mechanisms that are available in a network.

The HDTN Concept of Operations outlines six key capabilities required for HDTN. Basic Bundle Delivery (Non-Real-time – Store and Forward) is accomplished using Bundle Protocol version 6 or version 7. HDTN continues to support Bundle Protocol version 6 since some existing users, such as the ISS, continue to use version 6. Bundle Protocol version 7 is the latest revision of the specification and is recommended for new users. Bundle Delivery w/ Custody Transfer (Non-Real-time – Store and Forward) is currently only supported in Bundle Protocol version 6 per the CCSDS specification. Bundle Streaming (Real-time – Immediate Forwarding upon receipt) is accomplished using Bundle Protocol with Real-time Transport Protocol. The Concept of Operations also requires quality of service provisions (latency, priority) which are accomplished at the bundle layer and at the convergence layer. LTP and TCPCL are utilized in different latency conditions. LTP is recommended for delays longer than 500 ms. TCPCL is sufficient for lower latency use-cases. Currently, priority is only addressed by Bundle Protocol version 6, although future specifications are being developed for DTN quality of service. Reliability is addressed via store-and-forward in both Bundle Protocol version 6 and version 7, optional custody transfer in Bundle Protocol version 6, and in the LTP convergence layer. Security (encryption and authentication) is addressed in the Bundle Protocol Security (BPSec) requirements. Additional requirements for applications, user interface, and additional convergence layer allow the user to utilize and troubleshoot HDTN in the scenarios outlined in the Concept of Operations.

The document also includes requirements associated with applications developed for users to provide data to the HDTN implementation that will then route the data accordingly throughout a DTN environment.

3.1 States and Modes Descriptions

The HDTN CSCI can be described as having the “Initialization” and “Operational” modes. During the “Initialization” mode, the different CSCs (e.g., BPGen, BPSink, …) will parse the provided configuration data and configure the interfaces necessary to transition into the “Operational” mode. When the CSCs are in the “Operational” mode, they act as a source node, an intermediary node, or a destination node. The CSCs identified as applications below tend to be sources and destination nodes, as they are configured to provide or receive data. Otherwise, they serve as an intermediary node that receives and forwards data towards a destination.

3.2 HDTN Bundle Requirements

3.2.1 Bundle Protocol Version 6 (BPv6) Requirements

The following requirements were decomposed from the CCSDS Bundle Protocol Specification CCSDS 734.2-B-1, which contains the recommended standards definitions for implementing BPv6. BPv6 has been the standard version used on the International Space Station (ISS), the Huntsville Operations Support Center (HOSC), and many other DTN users for years. It encompasses much of DTN's primary functionality, including store-and-forward, custody transfer, and addressing concepts. BPv6 was the first main module implemented in HDTN and is currently maintained to support legacy systems. Bundle Protocol version 7 (BPv7) is expected to eventually supersede version 6; however, there is no set timeline. The BPv6 specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. Several aspects of RFC 5050 were not implemented since they were not being utilized by HDTN's customers. This includes: acting on report request flags, dictionary byte array, cancelling a transmission, and registration polling. BPv6 is maintained only for legacy systems and new users are recommended to use BPv7.

Table 3-1 BPv6 Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-001	Uniform Resource Identifier (URI) Scheme Specific Part	The Scheme Specific Part of every URI defined within the InterPlaNet ("ipn") scheme shall comprise: 1. The node number of the URI. 2. An ASCII period ('.') character. 3. The service number of the URI.	The scheme-specific part is defined in RFC 6260 section 2.1. Following the same format between bundles will aid in consistency within the network.	Test	This requirement is verified when a test receives a bundle, and the bundle is shown to contain URIs that follow the specified "ipn" scheme.
BPv6-002	Compressed Bundle Header Encoding (CBHE) Unit	A compressed primary block shall contain integers for its fields.	The CBHE-compressed primary block is defined in RFC 6260 section 2.2. If the report-to endpoint value is the null endpoint, then that piece of the primary block is zero.	Inspection	This requirement is verified when a code inspection shows that, when compressing a bundle, the primary block fields are integers.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-003	CBHE Encoding of Endpoint Identifications (EID) by Convergence Layer Adapters (CLAs)	A compressed primary block shall, in order, contain: 1. The node number of the destination endpoint ID. 2. The service number of the destination endpoint ID. 3. The node number of the source endpoint ID. 4. The service number of the source endpoint ID. 5. The node number of the report-to-endpoint ID. 6. The service number of the report-to-endpoint ID. 7. The node number of the current custodian endpoint ID. 8. The service number of the current custodian endpoint ID.	The CBHE-compressed primary block is defined in RFC 6260 section 2.2. If the report-to endpoint value is the null endpoint, then that piece of the primary block is zero.	Test	This requirement is verified when a test receives a bundle and confirms the compressed primary block contains each field in the specified order.
BPv6-004	BP Time	DTN time shall consist of nanosecond precision since the start of the year 2000.	DTN time is defined in CCSDS 734.2-B-1 section 3.4 and RFC 5050 section 6.1. Time must be in the proper format down to the nanoseconds. The onboard system's precision can be used if the node's time system does not provide sufficient accuracy.	Test	This requirement is verified when a test receives a bundle and confirms that the DTN time within the bundle consists of the number of seconds since the start of the year 2000 and the number of nanoseconds since the beginning of the indicated second.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-005	Licklider Transmission Protocol (LTP) Bundle Encapsulation	When utilizing LTP bundles, the bundle shall be entirely contained in a single LTP session of red-data.	LTP transmission is defined in CCSDS 734.2-B-1 section B3.1.2.2. For reliable bundle transmission, bundles are encapsulated in LTP blocks containing only red-part (reliable) data in either of these two different encapsulations. Reliable transmission is a crucial component of a network. LTP blocks are organized according to the Client Operations section 7 of the LTP-for_CCSDS Book.	Test	This requirement is verified when LTP bundles are shown to be encapsulated as an LTP single bundle block without leading and trailing bytes when the bundle is a single LTP bundle.
BPv6-006	Single LTP Bundle Encapsulation	When a single bundle per LTP block is selected, bundles shall be encapsulated as a single bundle.	LTP CL adaptor is detailed in section B3.1.2.2 in CCSDS 734.2-B-1. An LTP bundle block utilizes the Destination LTP Client Service ID for “Bundle Protocol” as specified in the SANA LTP Client Service ID Number Registry.	Test	This requirement is verified when a bundle is shown to be encapsulated as a single bundle with “single bundle per LTP block” selected.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-007	User Datagram Protocol (UDP) Convergence Layer (CL) Adaptor	When using the convergence layer, UDP-encapsulated bundles shall be in UDP datagrams.	UDP convergence layer adapter is defined in CCSDS 734.2-B-1 section B4; ensures network cohesion.	Test	This requirement is verified when a test receives a bundle via the UDP convergence layer and confirms the bundle contains a single UDP datagram with a checksum, and the specified UDP port is utilized.
BPv6-008	BP Bundle Structure	Bundles shall utilize SDNV encoding.	Encoding must be decided upon to understand the header and data within the network. Self-Delimiting Numeric Values (SDNV) are the encoding for bundle fields for BPv6.	Test	This requirement is verified when a test receives a bundle and confirms the bundle is encapsulated with valid SDNV encoding.
BPv6-009	BP Bundle Block Structure	Bundles shall be a concatenated sequence of two or more block structures.	Every bundle needs at least a payload block and a primary block per RFC 5050. A bundle can have additional blocks per request.	Test	This requirement is verified when an initiated bundle contains two or more concatenated block structures.
BPv6-010	Primary Block Location	The first block in the bundle shall be the primary bundle block.	The location of the primary bundle block is defined in RFC 5050 section 4.	Test	This requirement is verified when an initiated bundle contains a primary bundle block in the bundle’s first block.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-011	Number of Primary Blocks	A bundle shall have one primary bundle block.	The number of primary bundle blocks is defined in RFC 5050 section 4. All principal and required header information is placed in the primary bundle block. Therefore, having more would be redundant.	Test	This requirement is verified when a bundle is shown to have only one primary bundle block.
BPv6-012	Last Block Flag	The last block in the sequence shall have the "Last block" Block Processing Control Flag set to TRUE.	The "last block" flag is defined in RFC 5050 section 4. Setting this flag shows that the data stream is ending at the block. Therefore, this flag is set to FALSE for every block in the bundle after the primary block except the final block, which must be set to TRUE.	Test	This requirement is verified when a bundle is analyzed, and the last block in the sequence has the "last block" flags set to TRUE.
BPv6-013	Custody Transfer Request Flag of an Administrative Record	The custody transfer requested flag shall be FALSE when the bundle is an administrative record.	Bundle processing control flags are defined in RFC 5050 section 4.2. Network management is necessary for determining the health and statistics of the network.	Test	This requirement is verified when a test receives an administrative record bundle and confirms the bundle has the custody transfer requested flag set to FALSE.
BPv6-014	Status Report Request Flag of an Administrative Record	Status report request flags shall be FALSE when the bundle is an administrative record.	Bundle processing control flags are defined in RFC 5050 section 4.2. Network management is necessary for determining the health and statistics of the network.	Test	This requirement is verified when a received administrative record bundle has all status report request flags set to FALSE.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-015	Primary Bundle Block	Bundles created by a source bundle protocol agent shall have a unique combination of source endpoint ID and bundle creation timestamp.	The creation timestamp is defined in RFC 5050 section 4.5.1. The combination of source endpoint ID and bundle creation timestamp identifies a single transmission request, enabling it to be acknowledged by the receiving application (provided the source endpoint ID is not “dtn:none”).	Test	This requirement is verified when a test shows multiple bundles created by a bundling agent, each with a unique source endpoint ID and bundle creation timestamp combination.
BPv6-016	Constraining Bundle Fragmentation 1	The concatenation of bundle fragments shall result in a payload identical to the fragmented bundle's payload.	Bundle fragmentation is defined in RFC 5050 section 5.8. The payloads of fragments resulting from different fragmentation episodes in other parts of the network may overlap subsets of the original bundle's payload.	Test	This requirement is verified when a test shows a received reassembled bundle's payload is identical to the original payload before single and nested fragmentation.
BPv6-017	Constraining Bundle Fragmentation 2	The primary block's “Bundle is a fragment” Bundle Processing Control Flag of each fragment shall be set to TRUE.	Bundle fragmentation is defined in RFC 5050 section 5.8. Bundle Processing Control Flags are defined in RFC 5050 section 4.2. Keeping track of fragmented bundles is imperative for network management.	Test	This requirement is verified when a test receives a fragmented bundle and confirms the bundle has the primary block “Bundle is a fragment” Bundle Processing Control Flag set to TRUE.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-018	Constraining Bundle Fragmentation 3	The end of each fragment's primary bundle block shall contain the fragment offset followed by the total application data unit length.	Bundle fragmentation is defined in RFC 5050 section 5.8. Bundle format is defined in RFC 5050 section 4.5. Keeping track of fragmented bundles is imperative for network management. The fragment offset and total application data unit length aid in reconstructing the bundle once it reaches its destination.	Test	This requirement is verified when a test receives a fragmented bundle and confirms that each fragment's end contains the fragment offset followed by the total application data unit length.
BPv6-019	Bundle Fragmentation Replication	When a block's “Block must be replicated in every fragment” Block Processing Control Flag is TRUE, the block shall be replicated in the bundle fragment(s).	Bundle fragmentation is defined in RFC 5050 section 5.8. Block Processing Control Flags are defined in RFC 5050 section 4.3. The "Block must be replicated in every fragment" bit, when TRUE, means that the block shall be replicated in every fragment.	Test	This requirement is verified when a test receives a fragmented bundle and confirms the bundle contains a block with the “Block must be replicated in every fragment” Block Processing Control Flag set to TRUE. Also, the specified block is shown to be replicated in each fragment.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-020	Bundle Fragmentation Order	The relative order of the blocks present in a fragment shall be the same as in the bundle prior to fragmentation.	Bundle fragmentation is defined in RFC 5050 section 5.8. Keeping block order during fragmentation is imperative for network management and aids in reconstructing the bundle once it reaches its destination.	Test	This requirement is verified when fragmenting a bundle shows the blocks are kept in the same order as the bundle before fragmentation.
BPv6-021	SDNV Most Significant Bit (MSB)	An SDNV shall set the MSB of every octet to TRUE, except for the last octet.	SDNVs are defined in RFC 5050 section 4.1. An SDNV is a numeric value. The value encoded in an SDNV is the unsigned binary number obtained by concatenating into a single-bit string the 7 least significant bits of each octet of the SDNV. The last octet of an SDNV has its MSB set to FALSE.	Test	This requirement is verified when a received bundle is encoded in N octets with the MSB of every octet set to TRUE, excluding the last octet.
BPv6-022	Bundle Reception Unintelligible	The bundle protocol agent shall delete the bundle for the reason "Block unintelligible" for any extension block the bundle protocol agent cannot process.	Anything the network doesn't recognize will be deleted to keep the network efficient.	Test	This requirement is verified when a test sends a bundle with a corrupt extension block to the bundle agent, and it is confirmed that the Bundle has been deleted.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-023	Removal of a Block from a Bundle	The bundle protocol agent shall delete a block from a bundle when the block cannot be processed, the "Delete bundle if block can't be processed" Block Processing Control Flag is FALSE, and the "Discard block if it can't be processed" Block Processing Control Flag is TRUE.	Anything the network doesn't recognize will be deleted to keep the network efficient. This indication is shown by the Block Processing Control Flags. The criteria for deleting the block from the bundle is if the flags in that block do NOT indicate that the bundle must be deleted but do indicate that the block must be discarded.	Test	This requirement is verified when a test sends a bundle containing an un-processable block with the "Discard block if it can't be processed" Block Processing Control Flag set to TRUE to the bundle agent, and the block is shown to have been deleted.
BPv6-024	Bundle Reception Block Process Flag	The bundle protocol agent shall set the "Block was forwarded without being processed" Block Processing Control Flag in the block to TRUE when the block cannot be processed.	The "Block was forwarded without being processed" Block Processing Control Flag will not be set to TRUE if the "Delete bundle if block can't be processed" Block Processing Control Flag in that block indicates that the bundle must be deleted or the or "Discard block if it can't be processed" Block Processing Control Flag indicates that the block must be discarded.	Test	This requirement is verified when a bundle with an un-processable block is sent to the bundle agent, and the block is shown to have been updated with the "Block was forwarded without being processed" flag set to TRUE.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-025	Bundle Reception Custody Failure	For a singleton endpoint bundle with custody transfer redundancy, the bundle protocol agent shall set the "Custody transfer succeeded" flag of the Custody Signal Status to FALSE.	Procedures for handling redundancy in custody transfer for a bundle whose destination is not a singleton endpoint are not defined in the RFC 5050 specification. When a redundant single endpoint bundle is received, the “Custody transfer succeeded” flag needs to be set to FALSE. Also, the Custody Signal Reason Code is set to “Redundant reception” in the Custody Signal Status.	Test	This requirement is verified when a test sends a singleton endpoint bundle with custody transfer redundancy to the bundle agent and confirms a custody signal is generated with the “Custody transfer succeeded” flag set to FALSE and the Custody Signal Reason Code set to “Redundant reception” in the Custody Signal Status.
BPv6-026	Reporting of Custodial Delivery	When the bundle's “Custody transfer is requested” Bundle Processing Control Flag is set to TRUE, custodial delivery shall be reported.	Procedures for reporting custodial delivery for a bundle whose destination is not a singleton endpoint are not defined in the RFC 5050 specification. Network management is necessary for determining the health and statistics of the network.	Test	This requirement is verified when a test sends a bundle to the bundle agent with the "Custody transfer is requested" Bundle Processing Control Flag set to TRUE and confirms that custodial delivery is reported.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-027	Local Bundle Delivery Succeeded Response	When a bundle with the “Custody transfer is requested” Bundle Processing Control Flag set to TRUE contains a singleton endpoint destination, the bundle protocol agent shall report custodial delivery by generating a custody signal for the bundle destined for the bundle's current custodian with the "Custody transfer succeeded" flag of the Custody Signal Status set to TRUE.	When custody is enabled, a handshake happens between the two nodes. Details of bundle acceptance are defined in RFC 5050 section 5.7.	Test	This requirement is verified when a test sends a single destination bundle with custody transfer enabled to the bundle agent. Also, it is confirmed a custody signal with the "Custody transfer succeeded" flag of the Custody Signal Status set to TRUE is generated.
BPv6-028	Block Processing Control Flags Zeroing Flags	The “Block must be replicated in every fragment” Block Processing Control Flag shall be set to FALSE on the blocks that follow the payload block.	When a bundle is fragmented, it fragments itself at the payload block. As a result, all blocks after the payload block are fragmented. Bundle Transmission is defined in RFC 5050 section 5.2. Block Processing Control Flags are defined in RFC 5050 section 4.3.	Test	This requirement is verified when a test receives a fragmented bundle and confirms all fragments have the specified bit set to zero for blocks following the payload block.
BPv6-029	Bundle Transmission Commit to Custody	When custody transfer is requested, the bundle protocol agent shall commit to accepting custody of the bundle.	When custody is enabled, custody rules remain through the lifetime of the bundle. Bundle transmission services regarding custody transfer are defined in RFC 5050 section 5.2.	Test	This requirement is verified when a test sends a bundle with custody to the bundle agent and confirms that HDTN commits to accepting custody of the bundle.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-030	Bundle Transmission Source Endpoint	The source endpoint ID of the bundle shall be either the ID of an endpoint of which the node is a member, or the null endpoint ID "dtn:none".	The bundle transmission service details are defined in RFC 5050 section 5.2. The source endpoint ID is the same as the node that created it. This allows network users to pinpoint which node created the bundle.	Test	This requirement is verified when a test receives a bundle with custody and confirms the bundle has a source sending node endpoint ID and a bundle with custody and a null endpoint ID is received.
BPv6-031	Bundle Forwarding Contraindication Lookup	When the bundle's Custody Signal Status Reason Code is set to “No additional information”, the bundle protocol agent shall allow a bundle to be forwarded.	Contraindication reasons are defined in Figure 12 of RFC 5050. Contraindication is placed upon a bundle whenever the bundle protocol agent needs to do something else with the bundle for any reason.	Test	This requirement is verified when a test sends a bundle with its Custody Signal Status Reason code set to “No additional information” to the bundle agent and confirms the bundle is forwarded.
BPv6-032	Bundle Forwarding Choosing Endpoint	The bundle protocol agent shall determine the endpoint(s) to forward the bundle to.	The bundle protocol agent may forward the bundle directly to its destination endpoint (if possible) or some other endpoint(s) for further forwarding. How this decision is made may depend on the scheme name in the destination endpoint ID, but is beyond the scope of this document. If the agent finds it impossible to select any endpoint(s) to forward the bundle to, forwarding is contraindicated.	Test	This requirement is verified when a test sends a bundle to the bundle agent and confirms that the bundle agent sets the bundle’s next hop.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-033	Bundle Forwarding Block Sequencing	The sequencing of the blocks in a forwarded bundle shall remain unchanged as the bundle transits a node.	Bundle forwarding block sequencing is defined in RFC 5050 section 5.4. Keeping the block sequencing between hops will avoid the possibility of invalidating bundle security.	Test	This requirement is verified when a test sends a bundle to the bundle agent and confirms that the bundle agent does not alter the order of the bundle blocks.
BPv6-034	Bundle Forwarding Reason Code	When the bundle requires a reason code without any constraints, the reason code shall be "no additional information".	Bundle forwarding is defined in RFC 5050 section 5.4. Bundles may require reason codes even if it is not contraindicated.	Test	This requirement is verified when a test receives a bundle status report and HDTN confirms the reason code is "no additional information".
BPv6-035	Forwarding Failed Custody Signal Current Custodian	When a bundle with the “Custody transfer is requested” Bundle Processing Control Flag set to TRUE has a singleton endpoint destination, the bundle protocol agent shall handle a custody transfer failure by generating a custody signal for the bundle that is destined for the bundle's current custodian with the "Custody transfer succeeded" flag of the Custody Signal Status set to FALSE.	When custody is enabled, custody rules remain through the lifetime of the bundle. Bundle forwarding failures are defined in RFC 5050 section 5.4.2.	Test	This requirement is verified when a test sends a bundle with custody to the bundle agent that cannot accept custody and confirms that a Custody Signal Status with the "Custody transfer succeeded" flag set to FALSE is generated.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-036	Forwarding Failed Reason Code	When forwarding fails, the Custody Signal Status Reason Code shall indicate the reason for the failure.	Bundle forwarding failures are defined in RFC 5050 section 5.4.2. The custody signal contains, when applicable, the contraindication when a bundle uses reliability.	Test	This requirement is verified when a test sends a bundle with custody to the bundle agent that cannot forward it and confirms a custody signal with the Custody Signal Status “Custody transfer succeeded” flag set to FALSE and Custody Status Reason Code Indicating the reason for the failure is generated.
BPv6-037	Bundle Expiration	When a bundle expires, the bundle protocol agent shall delete the bundle for the reason "lifetime expired".	Bundle forwarding rules for expired bundles are defined in RFC 5050 section 5.5. Network needs not to waste resources moving expired data.	Test	This requirement is verified when a test sends an expired bundle to the bundle agent and confirms the expired bundle is deleted from HDTN storage and a custody signal with the Custody Signal Status “Custody transfer succeeded” flag set to FALSE and Custody Status Reason Code set to “Lifetime expired” is generated.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-038	Bundle Discard Prevented	When deletion of the singleton endpoint bundle is prevented from being discarded, a bundle deletion status report citing the reason for deletion shall be generated.	Bundle deletion is defined in RFC 5050 section 5.13. There are extra procedures for bundles which utilize reliability.	Inspection	This requirement is verified when a code inspection shows a bundle deletion status report is generated when a bundle is prevented from being discarded.
BPv6-039	Intermittent Connectivity Conditions	The DTN implementation shall store data when the link to the next node is unavailable.	DTN is store, carry, and forward. The intermittent nature of space communication is why DTN is used.	Test	This requirement is verified when a test sends a bundle to the bundle agent with no other connectivity and confirms that the bundle is stored.
BPv6-040	Late Binding	The DTN implementation shall provide late Endpoint ID (EID) binding capabilities.	Late binding capabilities are mentioned in RFC 5050 section 1.	Inspection	This requirement is verified when a code inspection shows a late EID binding capability.
BPv6-041	Custody Acceptance Succeeded Signal	The bundle protocol agent shall generate a Custody Signal Status with the "Custody transfer succeeded" flag set to TRUE for the bundle, destined for the bundle's current custodian when accepting custody of a bundle with the “Custody transfer is requested” Bundle Processing Control Flag set to TRUE.	When custody is enabled, a handshake happens between the two nodes. Details of custody signal processing are defined in RFC 5050 section 5.10.1.	Test	This requirement is verified when a test sends a bundle with custody to the bundle agent and confirms a Custody Signal Status with the "Custody transfer succeeded" flag set to TRUE is generated.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv6-042	Retransmit After Custody Transfer Timer Expiration	When custody transfer fails due to the custody transfer timer expiration, the DTN implementation shall retransmit the bundle.	When custody is enabled, a handshake happens between the two nodes.	Test	This requirement is verified when a bundle with custody is retransmitted when the bundle custody transfer timer expires.
BPv6-043	Retransmit After Custody Signal Reception Failure	When a Custody Signal Status with the "Custody transfer succeeded" flag set to FALSE is received for a bundle with the “Custody transfer is requested” Bundle Processing Control Flag set to TRUE has been received, the bundle agent shall retransmit the bundle.	When custody is enabled, a handshake happens between the two nodes.	Test	This requirement is verified when a Custody Signal Status with the "Custody transfer succeeded" flag set to FALSE is received for a bundle with the “Custody transfer is requested” Bundle Processing Control Flag set to TRUE, and the bundle is retransmitted.

3.2.2 Bundle Protocol Version 7 (BPv7) Requirements

The following requirements were decomposed from the Bundle Protocol Version 7 RFC 9171. BPv7 is the latest specification version during requirements development. The CCSDS specification was in draft when BPv7 was implemented in HDTN. It was implemented to meet new customers' requirements who expect to use current standards. There are several changes from BPv6 to BPv7, including the shift from SDNV encoding to CBOR encoding, no specification of custody transfer at the bundle layer, and no definition of priority. These changes were incorporated due to feedback from the DTN community, including CCSDS. Some of the removed functionality is expected to be implemented elsewhere or in other ways. For example, in future standards, priority is expected to be replaced by quality of service. The specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. In particular, version 7 bundles contain the fields for bundle processing control flags, however HDTN does not act on all conditions. Bundle administrative records and status reports are not currently implemented since they are not used by existing HDTN customers. Support for administrative records and status reports may be added in the future.

Table 3-2 BPv7 Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-001	Concise Binary Object Representation (CBOR) Encoding	The data type for bundle block fields shall be CBOR unsigned integers.	CBOR is the encoding for bundle fields. BPv7 only uses unsigned integers, byte strings, and arrays.	Inspection	This requirement is verified when an inspection of the code shows that the data type for bundle block fields is CBOR unsigned integers.
BPv7-002	Bundle Structure	Bundles shall be a CBOR indefinite-length array.	CBOR is the encoding for bundle fields. BPv7 only uses unsigned integers, byte strings, and arrays. The bundle structure is defined in RFC 9171 section 4.1.	Inspection	This requirement is verified when an inspection of the code shows that bundles are a CBOR indefinite-length array.
BPv7-003	Bundle Array Structure	Bundle arrays shall have a length of at least two blocks.	The primary bundle block and payload bundle block are the minimum requirements for a bundle. Bundle structure is defined in RFC 9171 section 4.1.	Test	This requirement is verified when a test receives a bundle and confirms that the bundle array has a length of at least two blocks.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-004	Block Order	Block(s) following the primary block shall be a canonical block.	Every block other than the primary block is called a “canonical” block. Canonical blocks are the extension blocks for adding data to the bundle header total for extra information. Block order is defined in RFC 9171 sections 4.1 and 4.3.2.	Test	This requirement is verified when a test receives a bundle and confirms the block following the primary block conforms to the canonical block format.
BPv7-005	Bundle Structure Stop Code	A CBOR "break" stop code, terminating the array, shall be directly after the payload block.	The stop code tells the node that the bundle is finished. Block order is defined in RFC 9171 section 4.1	Test	This requirement is verified when a test receives a bundle and confirms the payload block is terminated by the CBOR "break" stop code.
BPv7-006	Bundle Encoding	The CBOR encoding of definite-length values of all fields in all blocks shall conform to the core deterministic encoding requirements specified in the data dictionary.	Indefinite-length items are not prohibited within the BPv7 specification. The use of CBOR is discussed in section 4.1 of RFC 9171. CBOR encoding is defined in RFC 8949.	Inspection	This requirement is verified when an inspection of the code shows all definite-length values in block fields are CBOR encodings.
BPv7-007	Primary Bundle Location	The first block in the bundle shall be a primary bundle block.	The location of the primary bundle block is defined in RFC 9171 section 4.1.	Test	This requirement is verified when a test receives a bundle and confirms the first block is a primary bundle block.
BPv7-008	Number of Primary Blocks	A bundle shall have exactly one primary bundle block.	The number of primary bundle blocks is defined in RFC 9171 section 4.1.	Test	This requirement is verified when a test receives a bundle and confirms only one primary bundle block exists.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-009	Payload Block Location	The last bundle block shall be a payload block.	The location of the payload block is defined in RFC 9171 section 4.1.	Test	This requirement is verified when a test receives a bundle and confirms the last block in the bundle is a payload block.
BPv7-010	Number of Payload Blocks	A bundle shall have exactly one payload block.	The payload contains the message, which can be data, information, or communication. The number of payload bundle blocks is defined in RFC 9171 section 4.1.	Test	This requirement is verified when a test receives a bundle and confirms the bundle has exactly one primary bundle block.
BPv7-011	CRC Types	CRC type will specify that no CRC is present, or X-25 CRC-16 or CRC32C CRC-32 are in use.	CRC type is defined in RFC 9171 sections 4.2 and 4.3.1. CRC Type specifies the algorithm used to calculate the CRC, if any.	Inspection	This requirement is verified when an inspection of the code shows the CRC type as omitted, X-25 CRC-16, or CRC32C CRC-32.
BPv7-012	Block Processing Control Flags	The block processing control flags shall be processed as a bit field.	Bit fields are defined in the data dictionary. Bundle processing control flags assert properties of the bundle as a whole. They are conveyed in the primary block of the bundle. Control flag values are defined in RFC 9171 section 4.2.4 with details on page 15 and are required to be at least processed as bit field(s).	Inspection	This requirement is verified when a code inspection shows the block processing control flags are defined as a bit field.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-013	The DTN URI Scheme-SSP	When the string value of a BP endpoint ID's Scheme-Specific Part (SSP) is equal to "none", the SSP shall be set to a value of zero.	The scheme identified by the < scheme name > in an endpoint ID is a set of syntactic and semantic rules that fully explain how to parse and interpret the SSP. URI scheme code definition document requirements are defined in RFC 9171 section 4.2.5.1.1.	Test	This requirement is verified when a test receives a bundle with the endpoint ID of "dtn:none" and confirms the SSP has been set to zero.
BPv7-014	Node ID	The EID of a node's administrative endpoint shall uniquely identify that node.	Network management is necessary for determining the health and statistics of the network. Node ID is defined in RFC 9171 section 4.2.5.2.	Inspection	This requirement is verified when an inspection of the code shows that the EID of a node's administrative endpoint is read from a user-defined configuration file.
BPv7-015	Keep CBOR Encoding Value	The CBOR-encoded values of all fields in the primary block shall remain unchanged from point-to-point.	An immutable primary block provides integrity and accountability of the network. The rule to keep the primary block unchanged is defined in RFC 9171 section 4.3.1.	Test	This requirement is verified when a test confirms the bundle agent does not modify the CBOR-encoded values of any field in the primary block of a bundle from source to destination.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-016	Primary Header Field Order	The fields of the primary bundle block shall be in the following order when present: version, bundle processing control flags, CRC type, destination node EID, source node EID, report-to EID, creation timestamp, lifetime, fragment offset, total Application Data Unit Length, CRC.	Primary bundle block fields explain what information is in what CBOR array. The field order of the primary bundle block is defined in RFC 9171 section 4.3.1.	Test	This requirement is verified when a test receives a bundle and confirms the primary block contents are in the specified order.
BPv7-017	Canonical Block Field Order	The fields of every canonical block shall appear in the following order: Block Type Code, Block Number, Block Processing Control Flags, CRC Type, and Block Type Specific Data.	Canonical bundle block(s) fields explain what information is in what CBOR array. Canonical Bundle Block is defined in RFC 9171 section 4.3.2. Note: When CRC Type indicates “no CRC is present”, the CRC Type field is omitted in the element count.	Test	This requirement is verified when a test receives a bundle with a canonical block and confirms the canonical block fields are in the specified order.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-018	CRC Computation	Computation of a block's CRC shall be performed by temporarily setting the block's CRC field to zero before concatenating all bytes in the block for use as the binary dividend.	CRC calculations aid in verifying the integrity of the bundle. When all bytes in the block are concatenated, the CBOR 'break' stop code and CRC field are included. CRCs are calculated using binary 'long division' with the block data as the dividend, the algorithm-dependent key as the divisor, and the remainder at the end of the calculation as the final CRC value. CRC computation is defined in RFC 9171 section 4.3.1	Test	This requirement is verified when a test shows that the computed CRC value matches the reported CRC value of the block.
BPv7-019	Previous Node Block Format	The Previous Node Block Node ID shall uniquely identify that node.	The Previous Node Block, block type 6, identifies the node that forwarded this bundle to the local node; its block-type-specific data is the node ID of the transmitter node. RFC 9171 defined Previous Node Block Node ID details in Section 4.4.1, explicitly calling out Section 4.2.5.2.	Test	This requirement is verified when a test receives a bundle and confirms the previous node is uniquely identified.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-020	Previous Node Block Limitation	The bundle shall contain a Previous Node Block after its first hop when Previous Node Block is enabled.	A Previous Node Block will not be present when the bundle is first initialized. Previous Node Block details are defined in RFC 9171 section 4.4.1.	Test	This requirement is verified when a bundle with 'Previous Node Block' enabled is shown to a) contain no Previous Node Block at creation and b) contain a Previous Node Block after its first hop.
BPv7-021	Bundle Age Instances	When the bundle's creation time is zero, the bundle shall contain an occurrence of a Bundle Age Block extension block.	Bundle Age Block details are required in CCSDS 734.2-P-1.1; the bundle age units are defined in RFC 9171 section 4.4.2.	Test	This requirement is verified when the bundle's creation time is zero and contains a Bundle Age Block.
BPv7-022	Forwarding Bundle Failure	The Bundle Protocol Agent (BPA) shall declare failure in forwarding the bundle when any reason from the IANA "Bundle Status Report Reason Codes" registry is indicated.	If the bundle was not sent correctly, the system needs to know to correct it; a forwarding bundle failure is defined in RFC 9171 section 5.4.1.	Inspection	This requirement is verified when an inspection of the code shows a check for any indicated IANA 'Bundle Status Report Reason Codes' before forwarding the bundle, resulting in a forwarding failure declaration.
BPv7-023	Bundle Expiration Deletion	When a bundle’s lifetime is exceeded, the BPA shall set the Status Report Reason Code to “Lifetime expired.”	There is no need for the network to hold onto an expired bundle. Bundle expiration is detailed in section 5.5 of RFC 9171.	Test	This requirement is verified when a test sends an expired bundle to the bundle agent and confirms the bundle has been deleted.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPv7-024	Security	The Bundle Protocol Agent shall implement support for BPSec.	The data in the network must keep its integrity, encryption, and authentication when required; security considerations are defined in section 8 of RFC 9171.	Test	This requirement is verified when a test shows that BPSec is used when requested.

3.2.3 Bundle Protocol Security (BPSec) Requirements

The following requirements were decomposed from the Bundle Protocol Security RFC 9172. Bundle Protocol Security (BPSec) provides the delay-tolerant, transport-layer security needed in high latency disrupted space networks where HDTN is meant to operate. BPSec provides both integrity and confidentiality services. HDTN currently implements BPSec as a C++ security library module that requires Open Secure Socket Layer (SSL) support and OpenSSL Federal Information Processing Standards (FIPS) module to comply with NASA cybersecurity requirements.

The BPsec library can encrypt, decrypt, and modify the bundle memory in place, avoiding the need for copying, allocating, and deallocating memory, thus increasing efficiency.

The BPsec specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. In particular, bundle fragmentation is not supported in HDTN's BPSec.

Table 3-3 BPSec Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-001	Security Operation Uniqueness	BPSec shall limit the application of any given security service to a security target to once per bundle.	This uniqueness requirement ensures no ambiguity related to the order in which security blocks are processed or how security policy can be specified to require certain security services in a bundle.	Test	This requirement is verified when the user configuration for one target block indicates that the same security service is applied twice, generating an "duplicate security service for target block" error message in the log file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-002	Security Context 1	BPSec shall support the BCB-AES-GCM security context.	HDTN currently implements all default security contexts as per RFC 9173. These are the minimum security contexts that need to be supported for interoperability.	Test	This requirement is verified when the default confidentiality security context BCB-AES-GCM is part of the options security context parameters in the BPSec configuration file, and the security context can be used to encrypt and decrypt a bundle.
BPSec-003	Operation Multiplicity	BPSec shall utilize a single security block to represent multiple security operations.	A single security block MAY represent multiple security operations to reduce the security block number in a bundle. Reducing the security block number in a bundle reduces the redundant bundle information.	Inspection	This requirement is verified when code inspection shows that BPSec security blocks (Bpv7AbstractSecurityBlock) support multiple results per target.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-004	Target Identification	BPSec shall set security target values in a security block to the block numbers of the target blocks.	A security target must be uniquely and unambiguously identifiable when processing a security block. The extension block number field is used for this purpose. Placing the set of target blocks covered by operations in an easily accessible field allows Bundle Protocol Agents (BPAs) to scan this field and assess whether new Security Operations (SOps) can be added to a bundle and if the block has operations that must be processed.	Test	This requirement is verified when displaying the block value to which the security operation is applied matches the security target field in the corresponding security extension block.
BPSec-005	BIB Block Type	The block-type-specific data field of a Block Integrity Block (BIB) shall follow the Abstract Security Block (ASB) structure.	All security blocks share the same block-type-specific data structure, as these blocks have common aspects. The ASB data structure is defined in section 3.6 of RFC 9172.	Test	This requirement is verified when the inspection of a bundle with BIB block shows the contents of the BIB block-type-specific data field conforms to the ASB data structure.
BPSec-006	BIB Security Targets	The Block Integrity Blocks (BIB) and Block Confidentiality Blocks (BCB) shall be considered invalid security targets for BIB operations.	An appropriate target block that a BIB should be able to reference is any block that may have its block-type-specific data signed.	Test	This requirement is verified when a configuration file has a BIB with the security target as BIB or BCB and generates an “invalid security target block” error message in the log file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-007	BIB Integrity Mechanism	The security context shall use a signed integrity mechanism for authentication or an unsigned mechanism for error detection.	The integrity mechanism used by the BIB is given by the security context associated with the BIB. It may represent either signed or unsigned integrity. This way, BIB can represent authentication (with a signed integrity mechanism) or error detection (with an unsigned integrity mechanism).	Test	This requirement is verified when a test shows that the software successfully authenticates a block using a signed integrity mechanism, and a test demonstrates that the software successfully detects an error in a block using an unsigned mechanism.
BPSec-008	BCB Processing	BPSec shall set the flag “Block must be removed from bundle if it cannot be processed” to false for a BCB.	Removing a BCB from a bundle has significant consequences since the BCB is the sole indication that the BCB target block(s) have had their block-type-specific data field encrypted. Removing a BCB would make it impossible for future BPAs to decrypt the block.	Test	This requirement is verified when a test shows that BCB has the "Block must be removed from bundle if it cannot be processed” flag set to FALSE.
BPSec-009	BCB Block Type	The block-type-specific data fields of a BCB shall follow the structure of the ASB.	All security blocks share the same block-type-specific data structure, as these blocks have common aspects. The ASB data structure is defined in section 3.6 of RFC 9172.	Test	This requirement is verified when the inspection of a bundle with BCB block shows that the BCB block-type-specific data field conforms to the ASB data structure.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-010	BCB Security Targets	BCBs shall be considered invalid security targets for other BCB operations.	An appropriate target block that a BCB should be able to reference is any block that may have its block-type-specific data encrypted. Only payload, non-security extension blocks, and BIB blocks can be encrypted to properly process bundles. A BCB MUST NOT include another BCB as a security target, as other BCBs in a bundle cannot be encrypted. Doing so would hide other blocks in an encrypted bundle and remove the ability to decrypt them.	Test	This requirement is verified when each of the following conditions has been satisfied: 1) A test shows that a BCB operation is performed on a payload block without generating an error. 2) A test shows that a BCB operation is performed on a non-security extension block without generating an error. 3) A test shows a BCB operation is performed on a BIB without generating an error. 4) A test shows that specifying a BCB as the target of a BCB operation generates an error.
BPSec-011	BCB Disallowed Security Targets 1	A BCB shall target blocks excluding the primary block.	Encrypting the primary block hides bundle identity.	Test	This requirement is verified when the user cannot add the payload block as a target for a BCB.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-012	BCB Disallowed Security Targets 2	A BCB shall target a BIB only when sharing a security target with that BIB.	Encrypting a BIB and not the target block of the BIB removes the ability to check the integrity of that target block.	Test	This requirement is verified when a test shows that a BIB with shared security targets can be specified as the security target of a BCB operation without generating an error, and specifying a BIB without shared security targets as the target of a BCB operation generates an error.
BPSec-013	Authenticated Encryption with Associated Data	A BCB shall utilize a confidentiality cipher that provides Authenticated Encryption with Associated Data (AEAD).	Using a confidentiality cipher confirms that ciphertext, block processing flags, and other blocks in the bundle have not been modified.	Test	This requirement is verified when a test shows that performing an AEAD operation on unaltered ciphertext, block processing flags, and other blocks in a bundle does not result in an error, and performing an AEAD operation on altered ciphertext, block processing flags, and other blocks in a bundle results in an error.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-014	Authentication Tag Placement	Additional information created by the cipher suite shall be placed in one of the following locations, as specified by the BCB: 1. The security result field. 2. The generated ciphertext.	Security contexts used by the BCB must specify whether the authentication tag is included in the BCB as a security result or is represented with the ciphertext that replaces the security target block's plaintext.	Test	This requirement is verified when a test shows that the user can configure an authentication tag to be included either in the BCB as a security result or in the ciphertext that replaces the security target block's plaintext.
BPSec-015	Encryption in Place	When applying a BCB, BPSec shall overwrite unencrypted security target body data with the encrypted security target body data.	This eliminates the need to move the data, making the encryption faster and more reliable and provides better performance.	Inspection	This requirement is verified when an inspection of the encryption function code shows that encryption is performed in place.
BPSec-016	BCB and BIB Blocks Interactions 1	When adding a BCB to a bundle matching all the existing BIB target(s)security targets, BPSec shall encrypt the existing BIB.	This is needed to handle the special case of protecting the plaintext integrity of the target block when that plaintext has been replaced by ciphertext.	Test	This requirement is verified when a test shows that adding a BCB with some (or all) of its security targets matching all of the security targets of an existing unencrypted BIB results in the encryption of the BIB.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-017	BCB and BIB Blocks Interactions 2	When adding a BCB to a bundle with security target(s) matching some of the security target(s) of an existing BIB, BPSec shall remove the security results associated with the BCB security target(s) from that BIB and place them in a new encrypted BIB.	This processing rule prevents security information about the unencrypted target block from persisting after the target block has been encrypted.	Test	This requirement is verified when adding a BCB with some (or all) of its security targets matching some (but not all) of the security targets of an existing unencrypted BIB causes the removal of the BIB operation results on security targets shard with the BCB and placement of those results in a newly encrypted BIB.
BPSec-018	BCB and BIB Blocks Interactions 3	BPSec shall prevent the addition of a BIB with a security target already identified as the security target(s) of a BCB.	This is needed to prevent ambiguity in block processing order.	Test	This requirement is verified when the user is prevented from adding a BIB for a security target that is already the security target of a BCB.
BPSec-019	BCB and BIB Blocks Interactions 4	BPSec shall prevent checking the BIB integrity value of a BIB that is the security target of an existing BCB.	When a BIB is the security target of a BCB, the BIB data is encrypted. BIB integrity checks MUST NOT be performed on encrypted BIB data.	Test	This requirement is verified when a test shows that BPSec performs a BIB integrity check on unencrypted BIB data and BPSec does not perform a BIB integrity check on encrypted BIB data.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-020	BCB and BIB Blocks Interactions 5	BPSec shall prevent checking the BIB integrity value of a security target that is also the security target of a BCB.	When a BIB security target is also the security target of a BCB, the security target data is encrypted. BIB integrity checks MUST NOT be performed on encrypted security target data.	Test	This requirement is verified when a test shows that BPSec performs a BIB integrity check on unencrypted security target data, and BPSec does not perform a BIB integrity check on encrypted security target data.
BPSec-021	BCB and BIB Blocks Interactions 6	A BIB shall not be added as a BCB security target when that BIB is a BCB security target.	An appropriate target block that a BIB should be able to reference is any block that may have its block-type-specific data signed.	Test	This requirement is verified when the user is prevented from adding a BIB as a security target if it already is a security target for a BCB.
BPSec-022	Canonical Form of the Primary Block	Concise Binary Object Representation (CBOR) values from the primary block shall be canonicalized using the rules for Deterministically Encoded CBOR.	Canonicalization algorithms are discussed in section 3.7 of RFC 9173. Canonicalization algorithms transcode the contents of a security target into a canonical form. Security services require consistency and determinism in how information is presented to cipher suites at security sources, verifiers, and acceptors.	Inspection, Test	This requirement is verified when an inspection of the code and bundle with BPSec enabled shows that the primary block is canonicalized using the rules for Deterministically Encoded CBOR.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-023	Canonical Block Structure	Canonical blocks shall share the same block structure as specified in RFC 9171 section 4.3.2.	Security services require consistency and determine how information is presented to cipher suites at security sources, verifiers, and acceptors. RFC 9172 does not define a standard canonical block structure, leaving the structure to be defined by each implementation.	Inspection	This requirement is verified when code inspection shows that all canonical blocks share the same structure defined in HDTN-SWDD-017 HDTN Data Dictionary.
BPSec-024	Canonical Form of the Canonical Block	Canonical blocks shall be canonicalized using the canonicalization algorithms specified in section 3.7 of RFC 9173	Canonicalization algorithms are discussed in section 3.7 of RFC 9173. To ensure that if the security target values are unchanged, the canonical form of that target will be the same even if the encoding of those values for wire transmission is different.	Test	This requirement is verified when an inspection of the bundle with BPSec enabled shows that all its canonical blocks are canonicalized.
BPSec-025	Canonical Form Canonical Block CBOR Values	CBOR values from the canonical block shall be canonicalized using the rules for Deterministically Encoded CBOR.	Canonicalization algorithms are discussed in section 3.7 of RFC 9173. Security services require consistency and determinism in how information is presented to cipher suites.	Test and Inspection	This requirement is verified when an inspection of the code shows that CBOR values from the canonical block shall be canonicalized using the rules for Deterministically Encoded CBOR, and tests indicate that CBOR values in a generated bundle have been canonicalized.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-026	Blocks Encryption	Only the block-type-specific data field of a block shall be provided to a cipher suite for encryption of the block.	BPSec operates on data fields within bundle blocks (e.g., the block-type-specific data field). In their canonical form, these fields include their own CBOR encoding and no other encapsulating CBOR encoding.	Inspection	This requirement is verified when an encryption crypto function code inspection shows that only the block-type-specific data field is provided for block encryption.
BPSec-027	Canonical Form of Canonical Blocks Decryption	Only the block-type-specific data within a canonical block shall be decrypted and included in the canonical form used by the cipher suite for decryption.	BPSec operates on data fields within bundle blocks (e.g., the block-type-specific data field). In their canonical form, these fields include their own CBOR encoding and no other encapsulating CBOR encoding.	Test	This requirement is verified when a bundle inspection with BCB only has the block-type-specific data within a canonical block decrypted and included in the canonical form used by the cipher suite for decryption.
BPSec-028	Associated Authenticated Data	When a non-block-type-specific data field within a canonical block is tagged for authentication by user configuration, BPSec shall apply integrity-protection to the block.	An integrity protection mechanism allows confirmation that blocks in the bundle have not been modified.	Test	This requirement is verified when a test shows that a non-block-type specific data field within a canonical block is tagged for authentication in the configuration file, and integrity-protection is applied to the block.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-029	Receiving BCBs 1	When a received bundle contains a BCB, the receiving node shall determine whether it is the security acceptor for any security operations in the BCB.	The security acceptor identified in a BCB performs the final processing of the security operation and is the only node that decrypts and processes the security operations in the BCB.	Inspection	This requirement is verified when a code inspection shows that a check of the BCB security acceptor node ID is performed, the BCB is processed when the BCB security acceptor node ID and processing node ID match, and the BCB is not processed when the BCB security acceptor node ID and processing node ID do not match.
BPSec-030	Receiving BCBs 2	BPSec shall process BCB security targets according to a user-configured security policy after confidentiality security operation failures.	In case of security operation failure, the action for target processing is based on the security policy configured by the user.	Test	This requirement is verified when a test shows that a node identified as a BCB security acceptor processes and fails a security target and processes the failed security target as specified by the security policy defined in the BPSec config file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-031	Receiving BCBs 3	BPSec shall report any confidentiality security operation failures by generating a bundle status report indicating the failure.	In case of security operation failure, a bundle status report generation will help debug and track these failures.	Test	This requirement is verified when a test shows that a node identified as a BCB security acceptor performs and fails a confidentiality security operation on a security target and reports the failed confidentiality security operation by generating a bundle status report indicating the failed operation.
BPSec-032	Receiving BCBs 4	When the receiving node is the bundle's destination node, the node shall process all BCBs remaining in the bundle.	The bundle destination is, by necessity, the acceptor of any block remaining in the bundle. Since a bundle will no longer exist after processing at this BPA, all blocks must be accepted before passing the bundle payload to applications resident on the destination BPA.	Test	This requirement is verified when a test shows that if a node that is the bundle's final destination receives a bundle with multiple BCBs, it successfully processes all BCBs in the bundle.
BPSec-033	Receiving BCBs 5	When the receiving node is identified as a verifier for a BCB, the node shall process the BCB per the node’s user-configured security policy.	If the receiving node is a verifier, the BCB shall be processed based on the configured security policy.	Test	This requirement is verified when a test shows that a BCB received by a verifier node is processed according to the security policy defined in the BPSec config file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-034	Receiving BCBs 6	When the security policy of a node requires a node to apply confidentiality to a specific security target and no such BCB is present in the bundle, the node shall remove this security target.	HDTN removes the security target from the bundle because the confidentiality (and possibly the integrity) of the security target cannot be guaranteed.	Test	This requirement is verified when a test shows the security policy of a node specifies that the node should apply confidentiality to a specific security target, no such BCB is present in the bundle, and the security target is removed from the bundle.
BPSec-035	Receiving BCBs 7	When the security processing results in removing the payload block, BPSec shall discard the bundle.	The payload is required in a bundle.	Test	This requirement is verified when a test shows that a bundle is discarded after security processing, resulting in the bundle's payload block removal.
BPSec-036	Receiving BCBs 8	When BPSec fails to decrypt the payload block of a bundle, the payload block shall be removed.	Failure to decrypt a payload block indicates the payload could be compromised and should be discarded.	Test	This requirement is verified when a test shows that a payload block that cannot be decrypted is removed from a bundle.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-037	Receiving BCBs 9	When an encrypted security target other than a payload block cannot be decrypted, BPSec shall discard the target and the corresponding security blocks, identifying the failed block as a target.	Failure to decrypt a block indicates the block could be compromised and should be discarded along with corresponding security blocks.	Test	This requirement is verified when a test shows that a BCB decryption operation on a non-payload block security target fails, the security target block is removed from the bundle, and all BCBs identifying the removed block as a security target are also removed from the bundle.
BPSec-038	Receiving BCBs 10	When the security block is deleted from a bundle, BPSec shall generate a status report with the "Reporting node deleted the bundle" Bundle Status Report Flag set to TRUE.	Status reports help with debugging and reflect the exact root cause of a failure.	Test	This requirement is verified when a test shows that BPSec deletes a security block from a bundle and reports the block deletion with the "Reporting node deleted the bundle" Bundle Status Report Flag set to TRUE.
BPSec-039	Receiving BCBs 11	When a failed BCB operation results in the deletion of a bundle, BPSec shall generate a bundle status report with the "Reporting node deleted the bundle" Bundle Status Report Flag set to TRUE.	Status reports help with debugging and reflect the exact root cause of a failure.	Test	This requirement is verified when a test shows that a BCB operation fails, the failed BCB operation results in the deletion of a bundle, and BPSec generates a bundle status report with the "Reporting node deleted the bundle" Bundle Status Report Flag set to TRUE.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-040	Receiving BCBs 12	When BPSec decrypts a BCB, the recovered plaintext for the security targets shall replace the ciphertext in the corresponding security targets' block-type specific data fields.	The final bundle received by the application should only have decrypted data.	Test	This requirement is verified when a bundle inspection at the acceptor node for a confidentiality test shows that the ciphertext in the bundle was replaced with the correct plain text for the security target block-type specific data field.
BPSec-041	Receiving BCBs 14	When a BCB contains multiple security operations, each operation processed by the node shall be treated as if the security operation has been represented by a single BCB with a single security operation.	This helps with report generation and policy processing.	Inspection	This requirement is verified when a code inspection shows that if the BCB data structure contains multiple security operations, each operation processed by the node shall be treated as if the security operation has been represented by a single BCB with a single security operation.
BPSec-042	Receiving BIBs 1	When a received bundle contains a BIB, the receiving node shall determine whether it is the security acceptor for any of the security operations in the BIB.	The security acceptor identified in a BIB is responsible for processing the BIB.	Inspection	This requirement is verified when a code inspection shows that a bundle with a BIB is received by the HDTN ingress module and a security acceptor node ID check is performed against the processing node ID.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-043	Receiving BIBs 2	When the receiving node is the security acceptor for any of the security operations in a BIB: 1. The node shall process those operations. 2. Remove any operation-specific information from the BIB before forwarding the bundle.	The security acceptor identified in a BIB performs the final processing of the security operation.	Test	This requirement is verified when a test shows that a check of the BIB security acceptor node ID against the processing node ID is performed, the BIB is processed, and operation-specific information is removed from the processed BIB when the BIB security acceptor node ID and processing node ID match, and the BIB is not processed when the BIB security acceptor node ID and processing node ID do not match.
BPSec-044	Receiving BIBs 3	BPSec shall process BIB security targets according to a user-configured security policy after integrity security operation failures.	In case of security operation failure, the action for target processing is based on the security policy configured by the user.	Test	This requirement is verified when a test shows that a node identified as a BIB security acceptor processes and fails a security target and processes the failed security target specified by the security policy defined in the BPSec config file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-045	Receiving BIBs 4	When a node performs and fails an integrity operation on a security target, BPSec shall generate a bundle status report to indicate an integrity security operation failure.	In case of security operation failure, a bundle status report generation will help debug and track these failures.	Test	This requirement is verified when a test shows that a node performs and fails an integrity operation on a security target and reports the failed integrity security operation by generating a bundle status report indicating the failed operation.
BPSec-046	Receiving BIBs 5	When BPSec removes all the security operations for a given BIB, that BIB shall be removed from the bundle.	When all security operations in a BIB have been removed, the BIB is no longer needed and removed.	Test	This requirement is verified when a test shows that when all security operations for a BIB have been removed from the BIB, the BIB is removed from the bundle.
BPSec-047	Receiving BIBs 6	BPSec shall process a BIB only if the security target of the BIB is not the security target of a BCB in the bundle.	When a BIB and BCB share a security target, the target is encrypted after it is integrity signed. As a result, the BIB cannot be verified until the security target is decrypted by processing the BCB.	Test	This requirement is verified when a test shows when a BIB and BCB share a security target; the BIB is not processed until the security target is decrypted by BCB.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-048	Receiving BIBs 7	When the security policy of a node requires a node to apply integrity to a specific security target and no such BIB is present in the bundle, the node shall process this security target per the security policy.	The BIB is processed per the configured security policy.	Test	This requirement is verified when a test shows that when a node's user-specified security policy for an integrity operation is performed on a BIB security target that does not exist, the security target is processed per the security policy.
BPSec-049	Receiving BIBs 8	When the security policy of a node specifies the application of a BIB not present in the bundle to a security target that is not the payload or primary block, the node shall remove the security target from the bundle.	Application of an integrity operation to a non-BIB is considered an integrity security failure. This indicates that the security target may be compromised. Potentially compromised blocks are to be removed from the bundle; however, payload and primary blocks cannot be removed.	Test	This requirement is verified when a test shows that if a node's security policy specifies a node should have applied integrity to a specific security target, and no such BIB exists in the bundle, the node removes the security target from the bundle if the security target is not the payload or primary block.
BPSec-050	Receiving BIBs 9	When the target block of the failed integrity security operation is the primary block, BPSec shall discard the bundle.	The primary block is required in a bundle and cannot be discarded.	Test	This requirement is verified when a test shows that the bundle is discarded when an integrity security operation targeting the primary block fails.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-051	Receiving BIBs 10	When an integrity security operation performed on the payload block fails, BPSec shall discard the bundle.	The payload block is required in a bundle. Removing the payload block invalidates the entire bundle. As a result, the bundle is discarded.	Test	This requirement is verified when a test shows that the bundle is discarded when an integrity security operation targeting the payload block fails.
BPSec-052	Receiving BIBs 11	When a receiving node is designated as a security verifier of a security operation in a BIB, the node shall verify the security operation.	Verification of BIB security operations prevents nodes from forwarding corrupt data.	Test	This requirement is verified when a test shows that a node is the security verifier of a security operation in a BIB and the node attempts to verify the security operation.
BPSec-053	Receiving BIBs 12	BPSec shall process BIB security targets per a user-configured security policy after verification of security operation failures.	In case of security operation failure, the action for target processing is based on the security policy configured by the user.	Test	This requirement is verified when a test shows that when there's a BIB security operation failure, the node processes the security target per the security policy configured in the BPSec config file.
BPSec-054	Receiving BIBs 13	When a payload integrity check fails at a waypoint, BPSec shall process the payload according to user-defined preferences in the security configuration file.	Waypoints will prevent forwarding corrupt payloads.	Test	This requirement is verified when a test shows that 1) a BIB security operation processed on a payload security target at a waypoint node fails, and 2) the payload is processed according to user-defined preferences in the security configuration file.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-055	Receiving BIBs 14	When a BIB integrity check passes at a waypoint, the node shall retain the security operation in the BIB before forwarding.	If the integrity check passes at the waypoint, the bundle is verified and should be forwarded to the next hop. Only the acceptor removes the BIB extension block.	Test	This requirement is verified when a test shows that a) A BIB integrity security operation is performed by a node that is not the security acceptor, b) The BIB integrity security operation passes, and c) The bundle is forwarded to the next hop with the BIB integrity security operation intact.
BPSec-056	Receiving BIBs 15	When a BIB contains multiple security operations, each operation processed by the node shall be treated as if the security operation is represented by a single BIB with a single security operation.	Processing each security operation as a single operation with a single security target simplifies reporting and policy compliance.	Inspection	This requirement is verified when code inspection shows that if the BIB data structure contains multiple security operations, each operation processed by the node shall be treated as if the security operation has been represented by a single BIB with a single security operation.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-057	BCB Blocks Addition to a Bundle	BPSec shall prevent the addition of BCB to a bundle if the “Bundle is a fragment” flag is set to true in the bundle processing control flags field.	Specific security processes cause fragmentation delays until the bundle is reassembled. This results in some extension blocks being duplicated and security failures being ambiguous in the presence of fragmentation.	Test	This requirement is verified when a BCB is allowed to be added when then the "Bundle is a fragment" flag is set to FALSE and not allowed to be added to a bundle when the "Bundle is a fragment" flag is set to TRUE (Bundle fragmentation for BPv7 is not implemented at this time, but will be added soon.)
BPSec-058	BIB Blocks Addition to a Bundle	BPSec shall prevent the addition of BIB to a bundle if the “Bundle is a fragment” flag is set to true in the bundle processing control flags field.	Specific security processes cause fragmentation delays until the bundle is reassembled. This results in some extension blocks being duplicated and security failures being ambiguous in the presence of fragmentation.	Test	This requirement is verified when a BIB is allowed to be added when then the "Bundle is a fragment" flag is set to FALSE and not allowed to be added to a bundle when the "Bundle is a fragment" flag is set to TRUE (Bundle fragmentation for BPv7 is not implemented at this time, but will be added soon.)

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
BPSec-059	Security Context 2	BPSec shall support the BIB-HMAC-SHA2 integrity security context.	HDTN currently implements all default security contexts as per RFC 9173. These are the minimum security contexts that need to be supported for interoperability.	Test	This requirement is verified when the default integrity and confidentiality security contexts BIB-HMAC-SHA2 are part of the options security context parameters in the BPSec config file and are working as expected.
BPSec-060	Security Acceptor	BPSec shall be the security acceptor for bundles originating from node(s) identified as the security source(s).	HDTN currently defines the BPSec policy rules, which include the node's role (source, acceptor, or verifier) in JSON config files.	Test	This requirement is verified when a node identified as a security acceptor for an originating node in the security configuration file processes a bundle sent from its specified originating node.

3.2.4 Real-Time Protocol (RTP) Bundle Requirements

The following requirements were decomposed from the CCSDS Draft Recommended Standard 766.3-R-1 and 766.3-R-2, which specify audio and video transmission methods over DTN using RTP. The specification may contain additional definitions that were not incorporated into the requirements for the HDTN project.

Table 3-4 RTP Bundle Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
RTPBP-001	RTP Packets Concatenation	Bundles containing concatenations of RTP packets must be transmitted as soon as concatenation is completed.	When the number of RTP packets in the outgoing queue reaches numRtpPacketsPerBundle, a bundle containing concatenations of RTP packets is immediately transmitted. There is no attempt to smooth the bitrate.	Test	This requirement is verified when a test shows when the number of RTP packets in the outgoing queue reaches the maximum number of RTP packets per bundle and the bundle is successfully sent.
RTPBP-002	RTP Bundle Size Limit	When a bundle size limit is required, it shall be specified by the user.	BPSendStream implements a bundle size limit as an application parameter.	Test	This requirement is verified when a test shows that BPSendStream, configured with a maximum bundle size, sends bundles that do not exceed the specified size.
RTPBP-003	RTP Bundle Endpoint ID	The connection URI shall be the destination node endpoint ID for sources transmitted via unicast.	BPSendStream implements the destination node endpoint ID using the des-uri-eid parameter. BPReceiveStream implements the corresponding destination node endpoint ID using the my-uri-eid parameter.	Demonstration	This requirement is verified when a demonstration shows BPSendStream's des-uri-eid matches BPReceiveStream my-uri-eid and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
RTPBP-004	RTP Bundle EID Delimiter	Unless otherwise specified, the delimiting character shall be ".".	Wildcard characters are permitted when setting source and destination URI EIDs in HDTN.	Demonstration	This requirement is verified when a demonstration shows ipn:2.* is used for the des-uri-eid parameter on BPSendStream, and BPReceiveStream has a my-uri-eid of ipn:2.1 and bundles are received successfully.

3.3 Convergence Layer Requirements

3.3.1 Transmission Control Protocol (TCP) Convergence Layer (TCPCL) Requirements

The following requirements were decomposed from the Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4 RFC 9174, describing the specification definitions for using Bundle Protocol version 7 over a TCP convergence-layer adapter. The TCP convergence layer is implemented in HDTN to support local area networks and ground networks that are part of a larger DTN. The TCP convergence layer is not recommended for networks with longer than 0.5-second delays. Still, it connects non-disrupted networks to long-delay networks using Bundle Protocol as an overlay. TCP convergence layer is also often used in preliminary laboratory testing, prototype development, and related non-flight applications. The RFC 9174 specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. In particular, HDTN does not generate transfer refuse messages since this feature was not requested by HDTN customers and does not impact interoperability. Certificates for TCPCL are implemented by a third-party library rather than in HDTN TCPCL codebase.

Table 3-5 TCPCL Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-001	Network Byte Order	The data encodings shall transmit in big-endian byte order.	This is the byte order for data per RFC 9174. This means the bits are transmitted in this order: bits 0-7 first, then bits 8-15, then 16-23, and bits 24-31 last.	Test	This requirement is verified when a test shows that data encodings are received in big-endian byte order.
TCPCL-002	Session Establishment	When using TCPCL for bundle transmissions, communicating entities shall establish a TCPCL session.	TCPCL requires a handshake with the other node to ensure both sides are compatible for transmission and reception.	Test	This requirement is verified when a test shows two TCPCL nodes establishing a session and successfully exchanging data.
TCPCL-003	Contact Header	When a TCP connection is established, the active entity shall transmit its Contact Header to the passive entity.	The TCPCL contact header requires the TCPCL protocol version and a Transport Layer Security presence indication for the two nodes to establish a TCPCL connection.	Test	This requirement is verified after a test shows the active entity transmitting its Contact Header.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-004	Contact Header Reception	When the passive entity receives the Contact Header of an active entity, the passive entity shall transmit its Contact Header to the active entity.	To complete a handshake, the passive/receiver node must acknowledge receipt of the active/transmitter node's contact header by sending out the passive node's contact header. Exchanging contact headers allows each node to ensure that it uses the correct protocol version and negotiates Transport Layer Security use.	Test	This requirement is verified when a test shows that the passive entity transmits its Contact Header after receiving the active entity's Contact Header.
TCPCL-005	TCP Idle Timeout	The TCP connection shall close when the entity timer matches or exceeds the specified timeout.	Open network connections consume memory and processing resources. Timeouts and keep-alive messages allow nodes to detect defunct or idle connections and close them to free up resources.	Test	This requirement is verified when a test shows that the TCP connection closes when the timeout timer is reached or exceeded.
TCPCL-006	TCP FIN	The entity shall use the TCP FIN mechanism when closing a TCP connection.	TCP has two mechanisms for opening and closing connections: RST and FIN. FIN is the standard clean close for a TCP connection.	Test	This requirement is verified when a test shows an entity initiating TCP connection closure by transmitting a TCP FIN packet to the node on the other end of the connection.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-007	Enable Transport Layer Security (TLS) by CAN_TLS Flag	TLS shall set the CAN_TLS flag within its Contact Header to TRUE when TLS is enabled.	TLS is designed to facilitate privacy and data security for communications over the network. CAN_TLS indicates that the entity that generated this contact header has enabled TLS security. Enabling TLS for all sessions is recommended in RFC 9174.	Test	This requirement is verified when a test shows that TLS is enabled by setting the following fields in the HDTN outduct config: tryUseTls, tlsIsRequired, useTlsVersion1_3 or useTlsVersion1_4, doX509CertificateVerification, verifySubjectAltNameInX509Certificate, certificationAuthorityPemFileForVerification.
TCPCL-008	Magic String Validation	The connection shall terminate when the "dtn!" string is not in the contact header.	Magic is a four-octet field that always contains the octet sequence 0x64 0x74 0x6E 0x21, i.e., the text string "dtn!" in US-ASCII (and UTF-8). Magic is one of the 3 fields within a TCP contact header.	Test	This requirement is verified when a test shows that the receiving entity terminates the connection upon receiving a message without the "dtn!" string in its contact header.
TCPCL-009	Version Number Negotiation	The active entity shall terminate the TCP connection when the passive entity's TCPCL protocol version is lower than the active entity's version provided in the Contact Header.	Both Contact Headers of a successful contact negotiation must have identical TCPCL version numbers. TCPCL versions are not backward and forward compatible.	Test	This requirement is verified when a test shows that the active entity terminates the TCP connection when the active entity receives a TCPCL protocol version lower than the version provided in the Contact Header.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-010	Version Mismatch Reason Code	When the active entity terminates the TCP connection from TCPCL protocol version mismatch, the passive entity shall terminate the TCP connection with a reason code of "Version Mismatch".	Both Contact Headers of a successful contact negotiation must have identical TCPCL version numbers. TCPCL versions are not backward and forward compatible.	Test	This requirement is verified when a test shows that the active entity terminated the TCP connection from version mismatch with a reason code of "Version Mismatch" when the TCPCL protocol version in the contact header is not supported.
TCPCL-011	TLS Lifetime	The underlying TCP connection's lifetime shall match the TLS connection's lifetime.	The TLS connection will also be terminated when the TCP connection is terminated. This is because TLS resides on the TCP connection when it is enabled.	Test	This requirement is verified when a test with TLS enabled shows that the underlying TCP connection’s lifetime matches the TLS connection’s lifetime during the handshake.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-012	Node ID Matching	The implementation shall terminate the session when a contact header contains a NODE-ID that does not match the Uniform Resource Identifier (URI) "ipn:nextHopNodeId.0".	This is for compatibility between legacy network implementations. If the validation result fails or is absent and the security policy requires an authenticated node ID the session will terminate (reason code: "Contact Failure").	Test	This requirement is considered verified when a test proves that the entity will terminate the session when a contact header is received without the URI "ipn:nextHopNosdeId.0".
TCPCL-013	Version 3 Certificates	TCPCL shall require TCPCL version 3 certificates.	The TCPCL requires version 3 certificates due to the extensions used by the TCPCL certificate profile. It will reject version 1 and version 2 end-entity certificates.	Test	This requirement is verified when a test shows that TCPCL version 3 certificates have been used.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-014	Node ID Certificate	The TCPCL end-entity certificate shall contain a NODE-ID when applicable by Certificate Authority (CA) policy.	When assigned one or more stable DNS names, a TCPCL end-entity certificate contains a DNS-ID that authenticates those (fully qualified) names. When assigned one or more stable network addresses, a TCPCL end-entity certificate contains an IPADDR-ID that authenticates those addresses.	Test	This requirement is verified when the CA policy is enabled, and a test shows a TCPCL end-entity certificate containing a NODE-ID.
TCPCL-015	Type-ID	The active entity shall terminate the session if the type-id id-on-bundleEID does not match the passive entity's next hop endpoint ID.	Network integrity is lost when messages are delivered to the wrong node.	Test	This requirement is verified when a bundle with a bad certificate is sent and rejected at the receiving node.
TCPCL-016	TLS Authentication	The requested Transport Layer Security (TLS) handshake shall authenticate the TLS.	Enabling TLS for all sessions is recommended. TLS is designed to facilitate privacy and data security for communications over the network.	Inspection	This requirement is verified when a code inspection shows that the TLS authentication happens during the TLS handshake.
TCPCL-017	Node ID Mismatch	When an active entity receives a SESS_INIT that differs from the intended node ID, the TCPCL session shall reject the SESS_INIT.	Network integrity is lost when messages are delivered to the wrong node.	Test	This requirement is verified when a test shows that an active entity rejects a SESS_INIT message when the received node ID does not match the intended one.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-018	Transfer MRU	When the Transfer MRU provided during negotiation is unacceptable, the entity shall terminate the session with a reason code of "Contact Failure”.	Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Test	This requirement is verified when a test shows a termination and reason code of "Contact Failure" happens when the Transfer MRU is unacceptable.
TCPCL-019	Segment MRU Contact Failure	When the Segment Maximum Receive Unit (MRU) provided during negotiation is unacceptable, the entity shall terminate the session with a reason code of "Contact Failure".	A receiving entity can set the Segment Maximum Receive Unit (MRU) in its SESS_INIT message to determine the largest acceptable segment size. A transmitting entity can segment a transfer into sizes smaller than the receiver's Segment MRU. Determining an appropriate segmentation policy for entities using the TCPCL protocol is a Network Administration matter.	Test	This requirement is verified when a test shows a termination and reason code of "Contact Failure" happens when the Segment MRU is unacceptable.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-020	Session Extension Items Contact Failure	When a TCPCL entity receives a Session Extension Item with an unknown Item Type, and the CRITICAL flag is 1, the entity shall refuse the TCPCL session with a SESS_TERM reason code of "Contact Failure".	Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Test	This requirement is verified when a test shows a TCPCL entity receiving a Session Extension Item with an unknown Item Type and the CRITICAL flag = 1 and the entity refusing the TCPCL session with a SESS_TERM reason code of "Contact Failure".
TCPCL-021	Session Extension Items Encoding	Session Extension Items shall use Type-Length-Value (TLV) container encoding.	This is a requirement from RFC 9174 section 4.8. The fields of Session Extension Items are: 1. item flags, 2. item type, 3. item length, and 4. item value.	Inspection	This requirement is verified when a code inspection shows Session Extension Items encoded as TLV containers.
TCPCL-022	Keep-alive	Nodes shall send a keep-alive message when no message transmission reception happens during the negotiated interval.	Timeouts avoid an idle network. Timeouts ensure the network will not be 'hung up' in any particular node. Keep-alive messages are the network's last chance before a timeout occurs.	Test	This requirement is verified when a test shows an entity sending a keep-alive message when no message transmission reception happens during the negotiated interval between entities.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-023	SESS_TERM Idle Timeout	The entity shall terminate the session by transmitting a SESS_TERM message with a reason code of "Idle timeout" when no message transmission reception happens during the negotiated interval.	Messages can be keep-alive or other. Reason codes enable network statistics. These statistics can be used to understand how the network is operating. The timeout interval is 2.5 times the negotiated keep-alive interval.	Test	This requirement is verified when a test shows an entity terminating the TCP connection from no message transmission reception during the negotiated interval with a reason code of "Idle timeout".
TCPCL-024	Message Unsupported	When a TCPCL entity receives an inappropriate message type for the negotiated session parameters, the entity shall send an MSG_REJECT message with a reason code of "Message Unsupported".	This requirement is a catch statement. An inappropriate message can be due to an incorrectly negotiated session extension. Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Test	This requirement is verified when a test shows an entity sending an MSG_REJECT message from a wrong message type for the negotiated session parameters with a reason code of "Message Unsupported".
TCPCL-025	TCPCL Segment to Segment MRU	The size of a TCPCL segment shall be less than or equal to the receiving entity's Segment MRU.	A receiving entity can set the Segment Maximum Receive Unit (MRU) in its SESS_INIT message to determine the largest acceptable segment size. A transmitting entity can segment a transfer into sizes smaller than the receiver's Segment MRU. Determining an appropriate segmentation policy for entities using the TCPCL protocol is a Network Administration matter.	Inspection	This requirement is verified when a code inspection shows the comparison of a TCPCL segment is less than or equal to the receiving entity's Segment MRU.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-026	Single Transfer	A single transfer shall contain a single bundle.	This requirement is imposed on the agent using the TCPCL rather than the TCPCL itself. Since TCPCL cannot handle fragmentation like the bundle protocol layer, data must be as one unit.	Test	This requirement is verified when a test shows that a single bundle is always captured in a single TCP transfer.
TCPCL-027	Multiple Bundles	Multiple bundles on a single TCPCL connection shall transmit contiguously.	This is a requirement from RFC 9174 section 5.2. Bundle loss is reduced by transmitting multiple bundles contiguously.	Test	This requirement is verified when a test shows that a single TCPCL connection transmits contiguously when multiple bundles are sent.
TCPCL-028	Unique Transfer ID	Transfer IDs shall differ between endpoint entities within a single TCPCL session and direction.	When transfer IDs are not unique within a TCPCL session, communications can be accepted at another node. When this happens, the network will lose its validity and integrity.	Test	This requirement is verified when a test of multiple bundles within a single TCPCL session contains different transfer IDs between endpoint entities.
TCPCL-029	Reserved Message Header Flag Set	The sender shall set the reserved message header flag bits to 0.	Reserved message header flag bits are found within the Contact Header.	Test	This requirement is verified when a test shows that the sender's reserved message header flag bits are zero.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-030	Transfer Extension Items Length	The Transfer Extension Items Length and Transfer Extension Items list shall only be present when the START flag is 1 on the message.	Transfer Extension Items Length and Transfer Extension Items list: These fields represent protocol extension data for this specification.	Test	This requirement is verified when a test shows the Transfer Extension Items Length and Transfer Extension Items are present when the START = 1 on the message and a test showing that the Transfer Extension Items Length and Transfer Extension Items list are NOT present when the START = 0.
TCPCL-031	START Flag	The first segment of a transfer shall set the START flag = 1.	The flags portion of the message contains two flag values in the two low-order bits, denoted START and END in XFER_SEGMENT flags. These flags are to start and stop a handshake.	Test	This requirement is verified when a test shows that the first transfer segment contains a START = 1 and END = 0.
TCPCL-032	END Flag	The last segment of a transfer shall set the END flag = 1.	The flags portion of the message contains two flag values in the two low-order bits, denoted START and END in XFER_SEGMENT flags. These flags are to start and stop a handshake.	Test	This requirement is verified when a test shows that the last transfer segment contains a START = 0 and END = 1.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-033	Bundle Transfer	When a bundle transfer has commenced, the entity shall only send segments containing sequential portions of that bundle to the END flag = 1 segment.	The flags portion of the message contains two flag values in the two low-order bits, denoted START and END in XFER_SEGMENT flags. These flags are to start and stop a handshake.	Test	This requirement shall be considered verified when a test shows that, once a transfer of a bundle has commenced, the entity only sends segments containing sequential portions of that bundle until the segment with END flag = 1
TCPCL-034	Transfer ACK	A receiving TCPCL entity shall send XFER_ACK message(s) in response to receiving processed XFER_SEGMENT message segment(s).	These acknowledgments enable the transmitting entity to determine how much of the bundle has been received, so if the session is interrupted, a reactive fragmentation can be performed to avoid resending the already transmitted part of the bundle. In addition, there is no explicit flow control on the TCPCL.	Test	This requirement is verified when a test shows a receiving TCPCL entity sending XFER_ACK messages in response to the same amount of receiving processed XFER_SEGMENT message segments.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-035	XFER_ACK Flags	The flags portion of the XFER_ACK header shall match the acknowledged corresponding XFER_SEGMENT message.	The flags portion includes flags that are not decodable to the entity. The rationale behind these acknowledgments is to enable the transmitting entity to determine how much of the bundle has been received so that if the session is interrupted, it can perform reactive fragmentation to avoid resending the already transmitted part of the bundle. In addition, there is no explicit flow control on the TCPCL.	Test	This requirement is verified when a test shows that the flags portion of the XFER_ACK header matches the same number of receiving processed XFER_SEGMENT message segments.
TCPCL-036	Transfer Refuse ID	The transfer sender shall indicate the transfer ID of a refused transfer.	By indicating the transfer ID of a refused transfer, the network statistics can aid the overall network health, including the health of each individual node.	Test	This requirement is verified when a test shows that the transfer sender indicates the transfer ID of a refused transfer that was sent.
TCPCL-037	Complete Transmission	When a sender receives an XFER_REFUSE message, the sender shall complete the transmission of partially sent XFER_SEGMENT message(s).	This handshake lets the sender know what was received.	Test	This requirement is verified when a test shows a sender completing its transmission of partially sent XFER_SEGMENT messages after receiving an XFER_REFUSE message.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-038	Transfer Extension Items	Transfer Extension Items shall use Type-Length-Value (TLV) container encoding.	Transfer Extension Items are defined in RFC 9174 section 5.2.5. Transfer Extension item fields are: 1. item flags, 2. item type, 3. item length, and 4. item value.	Inspection	This requirement is verified when a code inspection shows Transfer Extension Items encoded as TLV containers.
TCPCL-039	CRITICAL Flag 1	When a TCPCL entity receives a Transfer Extension Item with an unknown Item Type, and the CRITICAL flag is 1, the entity shall refuse the transfer with an XFER_REFUSE reason code of "Extension Failure".	Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Test	This requirement is verified when a test shows a TCPCL entity receiving a Transfer Extension Item with an unknown Item Type and the CRITICAL flag = 1 and the entity refusing the transfer with an XFER_REFUSE reason code of "Extension Failure".
TCPCL-040	Single Transfer Length Extension Items	A transfer shall contain a single Transfer Length Extension Item at the beginning of the transfer when it contains more than one segment.	This requirement is imposed on the agent using the TCPCL rather than the TCPCL itself.	Test	This requirement is verified when a test showing a bundle transfer with multiple segments contains a single Transfer Length Extension Item.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-041	IANA Code Point	The Transfer Length Extension shall use the IANA-assigned code point.	IANA has created the "Bundle Protocol TCP Convergence-Layer Version 4 Transfer Extension Types" registry and populated it with the contents of the Transfer Length Extension Codes. Values in the range 0x8000-0xFFFF are reserved for Private or Experimental Use and not recorded by IANA.	Inspection	This requirement is verified when a code inspection shows the Transfer Length Extension value is assigned by the IANA-assigned code point options.
TCPCL-042	Total Length Field	The receiver shall accept a received bundle when the Total Length value matches the length of the bundle data received.	The total length mandates what the length is going to be. The authority that the receiver has is to accept it or not.	Test	This requirement is verified when a test shows a received bundle being accepted based on the Total Length value matching the length of the bundle data received.
TCPCL-043	Actual Total Length	When the actual total length of bundle data received is different from the value indicated by the Total Length value, the receiver shall: 1. Invalidate the transmitted data. 2. Send an XFER_REFUSE with a reason code of "Not Acceptable".	Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Test	This requirement is verified when a test shows a receiver invalidating transmitted data and sending an XFER_REFUSE message with a reason code of "Not Acceptable" when the total length of bundle data received differs from the value indicated by the Total Length value.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-044	Session Termination	To terminate a session, an entity shall 1. Complete transmission of other message(s). 2. Transmit a SESS_TERM message.	The handshake must be finished for both parties to terminate the session.	Test	This requirement is verified when a test shows one entity terminating a session (by SESS_TERM message) and the other entity completing transmission of its message and transmitting its SESS_TERM message.
TCPCL-045	SESS_TERM Message	When initiating a termination, the REPLY flag of a SESS_TERM message shall be set to 0.	Setting the REPLY flag of a SESS_TERM message lets the other node know that its connection wants a clean closeout.	Test	This requirement is verified when a test shows an entity terminating a session by sending the REPLY = 0 in a SESS_TERM message and a test showing that an entity with REPLY = 1 in a SESS_TERM message does not terminate the session.
TCPCL-046	Acknowledging SESS_TERM	Upon receiving an initial SESS_TERM message in the current session, an entity shall send an acknowledging SESS_TERM message.	The receiver node must acknowledge the sender to complete a handshake.	Test	This requirement is verified when a test shows that a receiving SESS_TERM message entity sends its acknowledging SESS_TERM message.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-047	XFER_REFUSE	While the Ending state attempts a new incoming transfer, the receiving entity shall send an XFER_REFUSE with a "Session Terminating" reason code.	Reason codes enable network statistics. These statistics can be used to understand how the network is operating.	Inspection	This requirement is verified when a code inspection shows an entity sending an XFER_REFUSE message from being in an Ending state with a reason code of "Session Terminating".
TCPCL-048	TCP Closed Failed Transfer	When the underlying TCP connection closes during a transmission, the BPA shall indicate the failed transfer.	TCP connections can be closed in either transfer stream.	Test	This requirement is verified when HDTN receives an error message that the TCP socket is closed when the underlying TCP connection is closed.
TCPCL-049	Contact Header Fails TCP Closes	When reception of the Contact Header fails, an entity shall close the TCP connection without sending a SESS_TERM message.	The Contact Header is sent first to ensure the two nodes are compatible. The SESS_TERM message sets the final settings of the communication. A contact header reception can fail by receiving an invalid magic string.	Test	This requirement is verified when a test shows an entity closing the TCP connection without sending a SESS_TERM message after receiving a failed Contact Header.
TCPCL-050	Ending State	While the session is in the Ending state, an entity shall complete the termination transfer procedure for the remainder of the session.	When the handshake/session is in the termination process, the two nodes are focused on a clean close-out. Therefore, no new transmission and receiver messages can be handled.	Test	This requirement is verified when a test shows that when a session is in the ending state, no new incoming and outgoing transfers are ignored.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
TCPCL-051	TCP Timeout	The timeout value for a TCP connection shall be configurable by the user.	Timeouts and keep-alive messages allow nodes to detect defunct or idle connections and close them to free up resources.	Test	This requirement is verified when a test shows that the TCP connection closes when the timeout timer is reached or exceeded.

3.3.2 Simple Transmission Control Protocol (STCP) Requirements

Simple Transmission Control Protocol (STCP) is a non-standard DTN protocol. It was implemented in HDTN to support the International Space Station Joint Station LAN, which uses STCP for the onboard DTN network during requirements development. STCP is primarily defined in legacy implementation as simplifying the TCP convergence layer. HDTN has implemented STCP to maintain legacy compatibility.

Table 3-6 STCP Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
STCP-001	Bundle Reception	Bundles shall be received over the connected TCP socket.	STCPCL was implemented to ensure compatibility with legacy implementations.	Test	This requirement is verified when a test initializes an STCPCL outduct and confirms a TCP connection is used as the protocol for bundle transfer.
STCP-002	Bundle Reception Length	STCP shall interpret a 32-bit unsigned integer preceding a bundle in network byte order as the length of the bundle.	The 32-bit length frames the bundle within a TCP stream.	Test	This requirement is verified when a test sends a bundle to an STCPCL induct and confirms the initial 32-bit integer of the bundle is read as the bundle length and is used to read the correct number of bytes from the TCP stream.
STCP-003	Unidirectional Link	Each STCPCL link shall be unidirectional.	STCPCL is a simplified version of TCPCL. It is only implemented to support legacy systems.	Test	This requirement is verified when an STCP link has been shown as unidirectional.
STCP-004	Bundle Transmission	Bundles shall be transmitted over the connected TCP socket.	STCP was implemented to ensure compatibility with legacy implementation.	Test	This requirement is verified when bundles are transmitted over the TCP connection.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
STCP-005	Bundle Transmission Length	Bundles transmitted on the connection shall be preceded by a 32−bit unsigned integer in network byte order indicating the length of the bundle.	The 32-bit length frames the bundle within a TCP stream.	Test	This requirement is verified when a test receives a bundle from an STCPCL outduct and confirms the value of the 32-bit integer preceding the bundle matches the length of the bundle.
STCP-006	STCPCL Keep-alive	Keep-alive packets shall be indicated by a 32-bit unsigned integer with all bits set to zero.	Keep-alive packets have been implemented to ensure compatibility with legacy implementation STCPCL.	Test	This requirement is verified when a test monitors an STCPCL outduct and confirms the TCP session is maintained by periodic reception of data or a 32-bit unsigned integer with all bits set to zero.
STCP-007	STCPCL Keep-alive Interval	Keep-alive packets shall be transmitted at a user-specified interval when no data is transmitted.	Keep-alive packets have been implemented to ensure compatibility with legacy implementation STCPCL.	Test	This requirement is verified when a test monitors an STCPCL outduct and confirms that keep-alives are transmitted at the user-specified interval.
STCP-008	Maximum Bundle Length	Accepted bundles shall be less than or equal to a user-specified maximum length.	Very large bundles could consume a large amount of memory.	Test	This requirement is verified when a test sends an oversized bundle to an STCPCL induct and confirms the connection is terminated.

3.3.3 User Datagram Protocol (UDP) Requirements

The following requirements were decomposed from the Datagram Convergence Layers for the Delay- and Disruption-Tolerant Networking (DTN) Bundle Protocol and Licklider Transmission Protocol (LTP) RFC 7122. The UDP convergence layer is among the most

uncomplicated DTN convergence layers. It can be used for unidirectional links, such as some radios, as an underlying transport for LTP, and as a simple prototype development and testing implementation. The specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. In particular, HDTN's UDPCL does not implement keep-alives or fragmentation. The UDPCL is intended to be a simplified implementation used for testing purposes.

Table 3-7 UDP Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
UDPCL-001	BP Over a Datagram CL	A UDP datagram shall contain one bundle.	To utilize DTN protocols across the Internet, encapsulating them into standard protocols is necessary. UDP is a unidirectional protocol with no congestion control. One bundle per datagram simplifies the convergence layer.	Test	This requirement is verified when a test receives a datagram from a UDPCL outduct and confirms the datagram contains exactly one bundle.
UDPCL-002	Rate Limit	UDP outduct transmission rates shall be limited by a user-specific rate limit.	UDP packets will be dropped if they exceed the expected link rate.	Test	This requirement is verified when a test receives data from a UDPCL outduct and confirms the received data rate does not exceed the configured rate.
UDPCL-003	Port Configuration Listening	UDP induct shall be bound to a user-provided port.	UDP connections are to be established via a known port.	Test	This requirement is verified when a test successfully connects to a UDPCL induct utilizing the expected port.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
UDPCL-004	Port Configuration Transmission	UDP outduct shall connect to a user-provided port.	UDP connections are to be established via a known port.	Test	This requirement is verified when a test receives a connection from a UDPCL outduct utilizing the expected port.

3.3.4 Licklider Transmission Protocol (LTP) Requirements

The following requirements were decomposed from the Licklider Transmission Protocol - Specification RFC 5326, which describes the Licklider Transmission Protocol (LTP), designed to provide retransmission-based reliability over links characterized by extremely long message round-trip times (RTTs) and/or frequent interruptions in connectivity. LTP is the most recommended convergence layer for long-delay links with 0.5-second and longer delays. LTP provides reliability using a checkpoint and acknowledgment-based system. It is also designed to split data and acknowledgments between channels common in space communication systems. The specification may contain additional definitions that were not incorporated into the requirements for the HDTN project. In particular, green part data is not fully supported by HDTN. This is due to the fact that data not requiring reliable transport can be sent using the UDP convergence layer. HDTN does not utilize the suspend timers discussed in RFC 5326 section 6.5. This functionality is not required by HDTN customers and does not impact interoperability with other LTP implementations.

Table 3-8 LTP Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-001	Session ID	A session ID shall consist of the sender's engine ID and a session number randomly generated by the sender.	This is the definition of the session ID.	Test	This requirement is verified when a test shows that a series of session IDs consists of the sender's engine ID and a session number randomly generated by the sender.
LTP-002	Unique Session ID	A session ID shall uniquely identify every session initiated by an LTP engine.	Each session needs a unique ID to distinguish it from other sessions.	Test	This requirement is verified when a test shows that a series of bundles transmitted from an LTP outduct contains unique session IDs.
LTP-003	Incrementing Checkpoint Serial Numbers	Any subsequent checkpoints issued by the sender shall have the serial number value found by incrementing the prior checkpoint serial number by 1.	The checkpoint serial number uniquely identifies the checkpoint among all checkpoints issued by the block sender in a session.	Test	This requirement is verified when a test shows that a series of checkpoint serial numbers increment by one for each subsequent checkpoint.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-004	Retransmitted Checkpoints	When a checkpoint segment is retransmitted, its serial number shall be the same as initially transmitted.	Retransmitted serial numbers should not be changed from the original serial number since they are related to the same checkpoint.	Test	This requirement is verified when a test shows that the serial number for a retransmitted checkpoint is the same as the serial number of the original checkpoint.
LTP-005	Report Serial Number	When the checkpoint is queued for transmission in response to the reception of a Report Segment (RS), its value shall be the report serial number value of the RS that caused the data segment to be queued for transmission.	This is required according to section 6.13 of RFC 5326. A non-zero value identifies it as a response to the reception.	Test	This requirement is verified when a test shows that the reported serial number of a received RS matches the RS that caused the data segment to be queued for transmission.
LTP-006	Report Segment	Any subsequent Report Segment issued by the receiver shall have the serial number value found by incrementing the last report serial number by 1.	It must be implemented per RFC 5326 section 3.2.2 and helps ensure interoperability with other DTNs.	Test	This requirement is verified when a test shows that a series of Report Segments have a report serial number that increments by 1.
LTP-007	Retransmitted Report Segment	When a Report Segment is retransmitted, its serial number shall be the same as initially transmitted.	It must be implemented per RFC 5326 section 3.2.2 and helps ensure interoperability with other DTNs.	Test	This requirement is verified when a test shows that a retransmitted Report Segment contains its original serial number.
LTP-008	Report Serial Number Greater Than Zero	The report serial number shall be greater than zero.	This must be implemented per RFC 5326 section 3.2.2 and helps ensure interoperability with other DTNs.	Inspection	This requirement is verified when a code inspection shows that the report serial number disallows zero.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-009	Reception Claims Length Limits	The sum of an LTP Reception Claim's length and offset shall not exceed the difference between the upper and lower bounds of the report segment.	This needs to be implemented per RFC 5326 section 3.2.2. The upper and lower bounds define report segment location and the reception claim is a portion of the segment.	Test	This requirement is verified when a test shows that the sum of an LTP Reception Claim's length and offset is between the upper and lower bounds of the report segment.
LTP-010	Reception Claims Offset	An LTP Reception Claim's offset shall be greater than the sum of the offset and the length of the prior claim.	This needs to be implemented per RFC 5326 section 3.2.2. The LTP Reception Claim should not overlap with a prior claim.	Test	This requirement is verified when a test shows that an LTP Reception Claim's offset is greater than the offset's sum and the prior claim's length.
LTP-011	Reception Claims Upper Bound	An LTP Reception Claim's upper bound shall be greater than or equal to the offset, length, and lower bound sum.	This needs to be implemented per RFC 5326 section 3.2.2. It defines the LTP Reception Claim’s location in a report segment.	Test	This requirement is verified when a test shows that an LTP Reception Claim's upper bound is greater than or equal to the sum of the offset, length, and lower bound.
LTP-012	Session Management Segments	An LTP Session Management Segment shall consist of a cancel segment or cancel acknowledgment.	This needs to be implemented per RFC 5326 section 3.2.4. Section 3.2.4 defines the contents of a Session Management Segment.	Test	This requirement is verified when a test shows that an LTP Session Management Segment contains either a cancel segment or cancel acknowledgment.
LTP-013	End of Block	The last data segment in a block shall be marked as the EOB (end of block).	This needs to be implemented per RFC 5326 section 4.1. This determines where the end of the block is located.	Test	This requirement is verified when a test shows that the last data segment in a block is marked as the EOB (end of block).

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-014	Data Segment	An LTP data segment shall only contain either red-part data, green-part data, or both.	This needs to be implemented per RFC 5326 section 4.1. Red and green data segments pertain to the reliability applied to the data segment. Data can be sent reliably, unreliably or a combination of both.	Test	This requirement is verified when a test shows that an LTP data segment contains either a red-part, a green-part, or both.
LTP-015	End of Red-Part	The Last Data Segment for Red-part data shall be marked as the EORP (end of red-part) segment.	This needs to be implemented per RFC 5326 section 4.1. This is needed to determine where the end of red-part data occurs.	Test	This requirement is verified when a test shows that the last data segment for a red-part data is marked as the EORP (end of red-part) segment.
LTP-016	Maximum Transmission Unit	Data shall be subdivided into data segments within a user-specified maximum transmission unit size.	This needs to be implemented per RFC 5326 section 4.1. Data segments must fit into the framing size of the underlying network layers. This will be specific to the user's particular network.	Test	This requirement is verified when a test shows that data segments are within the specified maximum transmission unit size.
LTP-017	Requirements from the Operating Environment	LTP shall be run directly over a data-link layer protocol.	LTP is meant to provide additional reliability on top of an existing data link layer.	Inspection	This requirement is verified when a code inspection shows the LTP implementation utilizes a lower-level data-link layer protocol.
LTP-018	LTP Link Status	The LTP Engine shall detect the status of an LTP destination.	This needs to be implemented per RFC 5326 section 5. LTP Engine must detect whether the link was brought up or shut down.	Test	This requirement is verified when a test shows that the LTP Engine detects if the link to the destination is active.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-019	One-way Light Time	The LTP Engine shall read the current distance from the configuration file.	This needs to be implemented per RFC 5326 section 5. This is used to calculate timeout intervals.	Test	This requirement is verified when a test shows that the one-way light time parameter is read correctly from the configuration file.
LTP-020	Local Data-link Layer Protocols	The content of each local data-link layer protocol frame shall contain an integer number of LTP segments.	This needs to be implemented per RFC 5326 section 5. Fractions of LTP segments would lack the needed context and could become invalid if received out of order.	Test	This requirement is verified when a test shows that the content of each local data-link layer protocol frame is an integral number of LTP segments.
LTP-021	Invalid Segments	The LTP Engine shall discard invalid segments.	This needs to be implemented per RFC 5326 section 6. LTP segments that do not conform to the specification are discarded.	Test	This requirement is verified when a test shows that the LTP Engine discards invalid segments.
LTP-022	UNREACH Reason Code	The LTP Engine shall send a Cancel by block Receiver (CR) with reason-code UNREACH if the invalid data segment contains red-part data.	This needs to be implemented per RFC 5326 section 6. Red-part data must be transmitted reliably and the LTP engine must signal if an error has occurred.	Test	This requirement is verified when a test shows that the LTP Engine sends a CR with reason-code UNREACH when an invalid data segment contains red-part data.
LTP-023	Retransmit Checkpoint	The expiration of a countdown timer associated with a Checkpoint (CP) segment shall invoke the Cancel Session procedure for the session associated with this segment.	This needs to be implemented per RFC 5326 section 6.7. As countdowns expire, they must end the corresponding session via the cancel session procedure and retransmission.	Test	This requirement is verified when a test shows that the expiration of a countdown timer associated with a CP segment triggers the Cancel Session procedure.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-024	Retransmit RS	When the number of times any affected RS segment has been queued for transmission exceeds the report retransmission limit established for the local LTP engine, the "Cancel Session" procedure shall be invoked.	This needs to be implemented per RFC 5326 section 6.8. The retransmission limit prevents excessive attempts to retransmit data.	Test	This requirement is verified when a test shows that the "cancel session" procedure is invoked when the retransmission attempts of RS segment(s) exceed the report retransmission limit.
LTP-025	Signify Red-Part Reception	Upon the arrival of a CP segment, when the EORP for this session has been received, and all data in the red-part of the block being transmitted in this session have been received, the LTP engine shall send a red-part reception notice to the specified client service.	This needs to be implemented per RFC 5326 section 6.9. The LTP engine needs to notify the sender when all red-part has been received.	Test	This requirement is verified when a test shows that the LTP engine sends a red-part reception notice to the specified client service upon the arrival of a CP segment at the end-of-red, and all red-part data in the block being transmitted in this session has been received.
LTP-026	Signify Green-Part Segment Arrival	Upon the arrival of a data segment whose content is a portion of the green-part of a block, the LTP engine shall send a green-part segment arrival notice to the specified client service.	This needs to be implemented per RFC 5326 section 6.10. The LTP engine needs to notify the sender when any green-part arrives.	Test	This requirement is verified when a test shows that upon the arrival of a data segment whose content is a portion of the green-part of a block, the LTP engine sends a green-part segment arrival notice to the specified client service.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-027	Send Reception Report	When the number of reception problems detected for this session exceeds a limit established for the local LTP engine, the "Cancel Session" procedure shall be invoked.	This needs to be implemented per RFC 5326 section 6.11. Resources must be freed if the canceling session exceeds the reception problems limit.	Test	This requirement is verified when a test shows that the cancel session procedure is invoked when the number of reception problems detected for this session exceeds the limit established for the local LTP engine.
LTP-028	Signify Transmission Completion Notice	A transmission-session completion notice shall be sent to the local client service associated with the session when these conditions have been met: 1. Data in the block is known to have been transmitted. 2. The entire red-part of the block is known to have been successfully received.	This needs to be implemented per RFC 5326 section 6.12. Need to notify sender when entire transmission has been received, and all data is accounted for.	Test	This requirement is verified when a test shows that a transmission-session completion notice is sent to the local client service associated with the session when these conditions have been met: 1. Data in the block is known to have been transmitted. 2. The entire red-part of the block is known to have been successfully received.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-029	Reason Code RLEXC	When the number of transmission problems for this session exceeds a limit established for the local LTP engine, a CS with reason-code Retransmission limit exceeded (RLEXC) shall be appended to the transmission queue specified in the transmission request that started this session, and a transmission-session cancellation notice is sent back to the client service that requested the transmission.	There should only be a finite number of retransmission attempts. See section RFC 5326 section 7.5 for the transmission session cancellation notice.	Test	This requirement is verified when a test shows that the number of transmission problems for a session exceeds a limit established for the local LTP engine. Also, a CS with reason code RLEXC shall be appended to the transmission queue specified in the transmission request that started the session, and a transmission-session cancellation notice is sent back to the client service that requested the transmission.
LTP-030	Stop RS Timer	The countdown timer associated with the original RS segment (identified by the report serial number of the Report-Acknowledgment (RA) segment) shall be deleted upon an RA reception.	This needs to be implemented per RFC 5326 section 6.14. Resources should be freed by ending countdown timers associated with RS segments.	Inspection	This requirement is verified when a code inspection shows that the countdown timer associated with the original RS segment (identified by the report serial number of the RA segment) is deleted upon an RA reception.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-031	CS Start Cancel Timer	Upon the arrival of a link state cue indicating the de-queuing (for transmission) of a CS segment, a countdown timer for the expected arrival time of the Cancel Acknowledgment to block Sender (CAS) segment shall be started.	This needs to be implemented per RFC 5326 section 6.15. Timers are needed to diagnose connection conditions.	Inspection	This requirement is verified when a code inspection shows that upon the arrival of a link state cue indicating the de-queuing (for transmission) of a CS segment, a countdown timer for the expected arrival time of the CAS segment is started.
LTP-032	CR Start Cancel Timer	Upon the arrival of a link state cue indicating the de-queuing (for transmission) of a CR segment, a countdown timer for the expected arrival time of the Cancel-Acknowledgment segment to block Receiver (CAR) segment shall be started.	This needs to be implemented per RFC 5326 section 6.15. Timers are needed to diagnose connection conditions.	Inspection	This requirement is verified when a code inspection shows that upon the arrival of a link state cue indicating the de-queuing (for transmission) of a CR segment, a countdown timer for the expected arrival time of the CAR segment is started.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-033	CS Acknowledge Cancellation	When a CS segment has a transmission queue-set bound for the sender, a CAS (cancel acknowledgment to block sender) segment shall be appended to the queue of internal operations traffic bound for the sender.	This needs to be implemented per RFC 5326 section 6.17. The LTP engine needs to acknowledge when a cancel session segment is received.	Inspection	This requirement is verified when a code inspection shows that if a CS segment has a transmission queue-set bound for the sender, a CAS (cancel acknowledgment to block sender) segment is appended to the queue of internal operations traffic bound for the sender.
LTP-034	CR Acknowledge Cancellation	When a CR segment has a transmission queue-set bound for the sender, a CAR (cancel acknowledgment to block receiver) segment shall be appended to the queue of internal operations traffic bound for the receiver.	This needs to be implemented per RFC 5326 section 6.17. The LTP engine needs to acknowledge when a cancel session segment is received.	Inspection	This requirement is verified when a code inspection shows that if a CR segment has a transmission queue-set bound for the sender, a CAR (cancel acknowledgment to block receiver) segment is appended to the queue of internal operations traffic bound for the receiver.
LTP-035	CAS Stop Cancel Timer	Upon reception of a CAS segment, the timer associated with the CS segment shall be deleted.	This needs to be implemented per RFC 5326 section 6.18. Resources must be freed by ending countdown timers associated with CS segments.	Inspection	This requirement is verified when a code inspection shows that the timer associated with the CS segment is deleted upon a CAS segment reception.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-036	CAR Stop Cancel Timer	Upon reception of a CAR segment, the timer associated with the CR segment shall be deleted.	This needs to be implemented per RFC 5326 section 6.18. Resources must be freed by ending countdown timers associated with CR segments.	Inspection	This requirement is verified when a code inspection shows that upon reception of a CAR segment, the timer associated with the CR segment is deleted.
LTP-037	Cancel Session	When a session is canceled, the LTP engine shall delete all queued segments from outbound traffic queues.	This needs to be implemented per RFC 5326 section 6.19. Resources associated with a canceled session must be freed.	Inspection	This requirement is verified when a code inspection shows that the LTP engine deletes all queued segments from outbound traffic queues when a session has been canceled.
LTP-038	Countdown Timers	When a session is canceled, the LTP engine shall delete all countdown timers currently associated with the session.	This needs to be implemented per RFC 5326 section 6.19. Resources associated with a canceled session must be freed.	Inspection	This requirement is verified when a code inspection shows that the LTP engine deletes all countdown timers currently associated with a session that has been canceled.
LTP-039	Cancel Session Buffer	When the local LTP engine is the sender, the remaining data retransmission buffer space allocated to a canceled session shall be released.	This needs to be implemented per RFC 5326 section 6.19. Resources associated with a canceled session must be freed.	Inspection	This requirement is verified when a code inspection shows that if the local LTP engine is the sender, the remaining data retransmission buffer space allocated to a canceled session is released.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-040	Close Session	The remaining countdown timers associated with a closed session shall be deleted.	This needs to be implemented per RFC 5326 section 6.20. Resources associated with a closed session must be freed.	Inspection	This requirement is verified when a code inspection shows that the remaining countdown timers associated with a closed session are deleted.
LTP-041	Handling Miscolored Segments	When miscolored data blocks are received, the LTP engine shall discard them.	This needs to be implemented per RFC 5326 section 6.21. Miscolored data blocks need to be removed from the system. This procedure is triggered by the arrival of either (a) a red-part data segment whose block offset begins at an offset higher than the block offset of any green-part data segment previously received for the same session or (b) a green-part data segment whose block offset is lower than the block offset of any red-part data segment previously received for the same session. The arrival of a segment matching either of the above checks violates the protocol requirement of having all red-part data as the block prefix and all green-part data as the block suffix.	Test	This requirement is verified when a test shows that miscolored data blocks are discarded.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
LTP-042	Cancel Session for Miscolored Data	The local LTP engine shall cancel the session when miscolored data blocks are received.	This needs to be implemented per RFC 5326 section 6.21. The LTP engine should prevent further miscolored blocks from entering the system.	Test	This requirement is verified when a test shows that the LTP session is canceled when miscolored data has been received.
LTP-043	UDP Port Number for LTP	The UDP port number shall be user-configurable.	This needs to be implemented per RFC 5326 section 10.1.	Test	This requirement is verified when a test shows that the LTP port is configured based on the configuration file.

3.4 Application Requirements

3.4.1 BPGen Application Requirements

The BPGen is an application that generates bundles of any specified size, and it is intended to be used with its receiving application called BPSink. The primary use of the BPGen application is to generate bundles for testing and benchmarking. The following requirements capture the expected functionality for the application.

Table 3-9 BPGen Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPGEN-001	BPGen Bundle Size	The BPGen application shall generate bundles with the size specified.	The bundle size is configurable in support of benchmark testing.	Test	This requirement is verified when a test shows the transmitted bundle payload size matches the user-specified size.
HDTN BPGEN-002	BPGen Aggregate Custody Signals	The BPGen application shall allow aggregate custody signal custody transfer usage.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use aggregate custody signals when a Bundle Protocol Version 6 is configured, and bundles are transmitted successfully.
HDTN BPGEN-003	BPGen Bidirectional Communication	The BPGen application shall support bidirectional communication when receiving custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use custody signals when a Bundle Protocol Version 6 is configured, and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPGEN-004	BPGen Bundle Priority	The BPGen application shall have the option to specify the bundle priority from the predefined range for BP version 6.	Bundle priority is a part of RFC 5050 and QoS support.	Test	This requirement is verified when a test shows that the bundle priority field of a transmitted bundle matches the specified priority.
HDTN BPGEN-005	BPGen Statistics	The BPGen application shall keep statistics of bundle count and order.	Statistics are needed for debugging purposes.	Test	This requirement is verified when a test shows that bundle count and bundle order statistics are generated correctly.
HDTN BPGEN-006	BPGen Rate	The BPGen application shall transfer bundles at a specified rate.	Convergence layers such as LTP and UDP must have a rate set, or bundles will be dropped.	Test	This requirement is verified when a test shows that the bundles per second generated (approximately/are within 10%) match the specified rate.
HDTN BPGEN-007	BPGen Duration	The BPGen application shall send bundles for a specified duration.	The user should be able to set a time for the application to stop sending data.	Test	This requirement is verified when a test shows that the application stops sending bundles after the specified duration.
HDTN BPGEN-008	BPGen Destination	The BPGen application shall send bundle data to a specified Endpoint ID (EID).	The application needs a destination Endpoint ID to send bundles to.	Test	This requirement is verified when a test shows that the transmitted bundles' bundle destination EID field matches the specified EID.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPGEN-009	BPGen Source	The BPGen application shall send bundle data with a specified source Endpoint ID (EID).	The application needs to specify a source Endpoint ID.	Test	This requirement is verified when a test shows that the transmitted bundles' bundle source EID field matches the specified EID.
HDTN BPGEN-010	BPGen Timeout	The BPGen application shall allow the user to specify a maximum timeout.	A timeout should be specified to determine if bundles are being received.	Test	This requirement is verified when a test shows an error message is printed when bundles are not received within the specified timeout.
HDTN BPGEN-011	BPGen Timeout Display	The BPGen application shall display when a bundle timeout has been exceeded.	The user should be notified if bundles are not being received.	Test	This requirement is verified when a test shows an error message is printed when bundles are not received within the specified timeout.
HDTN BPGEN-012	BPGen Bundle Lifetime	The BPGen application shall allow the user to set a maximum bundle lifetime.	A bundle lifetime is needed to determine how long bundles should be kept in storage.	Test	This requirement is verified when a test shows the transmitted bundles' bundle time to live field matches the specified bundle lifetime.
HDTN BPGEN-013	BPGen CLA Rate	The BPGen application shall allow the user to specify a Convergence-layer adapter (CLA) rate.	The application will configure with the provided value if the convergence layer supports the rate configuration.	Demonstration	This requirement is verified when a demonstration shows that bundles are transmitted according to the specified rate. This only pertains to UDP and LTP.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPGEN-014	BPGen LTP Convergence Layer	The BPGen application shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use an LTP convergence layer and bundles are transmitted successfully.
HDTN BPGEN-015	BPGen TCP Convergence Layer	The BPGen application shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use a TCP v4 convergence layer, and bundles are received successfully.
HDTN BPGEN-016	BPGen UDP Convergence Layer	The BPGen application shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use a UDP convergence layer and bundles are received successfully.
HDTN BPGEN-017	BPGen STCP Convergence Layer	The BPGen application shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use an STCP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPGEN-018	BPGen Bundle Protocol Version 6	The BPGen application shall support Bundle Protocol version 6.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use a Bundle Protocol v6 convergence layer and bundles are received successfully.
HDTN BPGEN-019	BPGen Bundle Protocol Version 7	The BPGen application shall support Bundle Protocol version 7.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use a Bundle Protocol v7 convergence layer and bundles are received successfully.
HDTN BPGEN-020	BPGen BPSec Support	The BPGen application shall support BPSec (RFC9172/RFC9173).	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPGen application is configured to use BpSec and bundles are received successfully.
HDTN BPGEN-021	BPGen Custodian Service ID	The BPGen application shall have the option to specify the Custodian Service ID for the node.	The user can specify the custodian service ID to identify who will receive a response.	Test	This requirement is verified when a test shows bundles generated by BPGen contain the custodian service ID.

3.4.2 BPSink Application Requirements

The BPSink application receives and validates the bundles sent from the BPGen application. BPGen will be used to generate testing and benchmarking bundles, and the BPSink application is specifically designed to receive those generated bundles. The following requirements capture the expected functionality for the application.

Table 3-10 BPSink Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSN K-001	BPSink Statistics	The BPSink application shall keep statistics of bundle count and order.	Statistics are needed for debugging purposes.	Test	This requirement is verified when a test shows that bundle count and bundle order statistics are generated correctly.
HDTN BPSN K-002	BPSink Data Received	The BPSink application shall discard the data received after taking the statistics.	BPSink is a test application and is not meant to store data.	Demonstration	This requirement is verified when a demonstration shows that received bundles are discarded after generating associated statistics.
HDTN BPSN K-003	BPSink LTP Convergence Layer	The BPSink application shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use an LTP convergence layer and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSN K-004	BPSink TCP Convergence Layer	The BPSink application shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use a TCP v4 convergence layer and bundles are received successfully.
HDTN BPSN K-005	BPSink UDP Convergence Layer	The BPSink application shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use a UDP convergence layer and bundles are received successfully.
HDTN BPSN K-006	BPSink STCP Convergence Layer	The BPSink application shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use an STCP convergence layer and bundles are received successfully.
HDTN BPSN K-007	BPSink Bundle Protocol Version 6	The BPSink application shall support Bundle Protocol version 6.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use a Bundle Protocol v6 convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSN K-008	BPSink Bundle Protocol Version 7	The BPSink application shall support Bundle Protocol version 7.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use a Bundle Protocol v7 convergence layer and bundles are received successfully.
HDTN BPSN K-009	BPSink BPSec Support	The BPSink application shall support BPSec (RFC9172/RFC9173).	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use BpSec and bundles are received successfully.
HDTN BPSN K-010	BPSink Aggregate Custody Signals	The BPSink application shall allow aggregate custody signal custody transfer usage.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use aggregate custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSN K-011	BPSink Bidirectional Communication	The BPSink application shall support bidirectional communication when receiving custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSink application is configured to use custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.
HDTN BPSN K-012	BPSink Destination	The BPSink application shall receive bundle data with a specified Endpoint ID (EID).	The application needs a destination Endpoint ID to receive bundles.	Test	This requirement is verified when a test shows that the received bundles' bundle destination EID field matches the specified EID.

3.4.3 BPing Application Requirements

The BPing application can confirm the existence of nodes, determine network latency, and verify a round-trip communication path exists between nodes. BPing application generates ping bundles intended to be used with any bundling agent that supports an echo service. The following requirements capture the expected functionality for the application.

Table 3-11 BPing Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPING -001	BPing Creation Timestamp	The BPing application shall create a custom payload bundle with the creation timestamp.	The timestamp is used to calculate bundle life elapsed time.	Test	This requirement is verified when a test shows that a series of bundles contain unique payloads with the bundle’s creation timestamp.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPING -002	BPing Destination Endpoint ID	The BPing application shall send the custom payload bundle to a specified node and service ID.	The node will be the end recipient of the bundle. The service ID should be an ID that correlates to an "ECHO" service, which will generate a response.	Test	This requirement is verified when a test shows that bundles are sent to and received by the specified node and service ID.
HDTN BPING -003	BPing Elapsed Time Calculation	The BPing application shall calculate the time elapsed from received payload bundles with specified service ID.	This calculation is used to know how long it took to receive a response.	Test	This requirement is verified when a test shows the correct time elapsed is calculated to receive a bundle.
HDTN BPING -004	BPing Elapsed Time Display	The BPing application shall display the calculated time elapsed to the user.	This provides the user with information about the network.	Test	This requirement is verified when a test shows that the correct time elapsed to receive a bundle is printed to standard out.
HDTN BPING -005	BPing Send Duration	The BPing application shall send a series of bundles for the specified duration.	The application can be configured to send multiple bundles to calculate the responses' average duration.	Test	This requirement is verified when a test shows that the application stops sending bundles after the specified duration.
HDTN BPING -006	BPing Response Received	The BPing application shall send a new custom payload bundle after receiving a response.	The application will continuously generate a new bundle after receiving a response unless a specific number is provided or the application is stopped.	Test	This requirement is verified when a test shows that a new bundle is sent after receiving a response.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPING -007	BPing LTP Convergence Layer	The BPing application shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use an LTP convergence layer and bundles are transmitted successfully.
HDTN BPING -008	BPing TCP Convergence Layer	The BPing application shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use a TCP v4 convergence layer and bundles are received successfully.
HDTN BPING -009	BPing UDP Convergence Layer	The BPing application shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use a UDP convergence layer and bundles are received successfully.
HDTN BPING -010	BPing STCP Convergence Layer	The BPing application shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use an STCP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPING -011	BPing Bundle Protocol Version 6	The BPing application shall support Bundle Protocol version 6.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use a Bundle Protocol v6 convergence layer and bundles are received successfully.
HDTN BPING -012	BPing Bundle Protocol Version 7	The BPing application shall support Bundle Protocol version 7.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use a Bundle Protocol v7 convergence layer and bundles are received successfully.
HDTN BPING -013	BPing BPSec Support	The BPing application shall support BPSec (RFC9172/RFC9173).	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows the BPing application is configured to use BpSec and bundles are received successfully.
HDTN BPING -014	BPing Custodian Service ID	The BPing application shall have the option to specify the Custodian Service ID for the node.	The user can specify the custodian service ID to identify who will receive a response.	Test	This requirement is verified when a test shows bundles generated by BPing contain the custodian service ID.

3.4.4 BPSendFile Application Requirements

The BPSendFile application sends either a single file or a directory of files (with recursion). It takes those file(s) and breaks them into the maximum specified size bundles. It is intended to be used with BPReceiveFile as the receiving application. The application's code is an example for users who want to write custom applications for HDTN that handle unidirectional bundles. The following requirements capture the expected functionality for the application.

Table 3-12 BPSendFile Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSF-001	BPSendFile Maximum Bundle Size	The BPSendFile application shall allow the user to specify the maximum bundle size.	There is no maximum size for a bundle according to BPv6 and BPv7, so the user sets the maximum size limit.	Demonstration	This requirement is verified when a demonstration shows the user can configure the maximum bundle size.
HDTN BPSF-002	BPSendFile Minimize Bundles Transmitted	The BPSendFile application shall minimize the number of bundles transmitted per file.	Bundles should be sized according to the maximum bundle size when possible.	Test	This requirement is verified when a test is performed to ensure that each file is transmitted using the fewest possible number of bundles, considering the maximum bundle size.
HDTN BPSF-003	BPSendFile Transmit Existing Files	The BPSendFile application shall be able to transmit existing files from a user-specified directory.	BPSendFile reads files from a directory into memory and transmits them as bundles.	Demonstration	This requirement is verified when a demonstration shows that an option exists to transmit existing files from a user-specified directory.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSF-004	BPSendFile Transmit Newly Added Files	The BPSendFile shall be able to transmit new files added to an existing directory.	Polling for new files added to the directory is a configurable option.	Demonstration	This requirement is verified when a demonstration shows an option to transmit newly added files from an existing user-specified directory.
HDTN BPSF-005	BPSendFile LTP Convergence Layer	The BPSendFile shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the LTP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendFile is configured to use the LTP convergence layer and bundles are received successfully.
HDTN BPSF-006	BPSendFile TCP Version 4 Convergence Layer	The BPSendFile shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the TCP version 4 convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPSendFile is configured to use the TCP version 4 convergence layer and bundles are received successfully.
HDTN BPSF-007	BPSendFile UDP Convergence Layer	The BPSendFile shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the UDP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendFile is configured to use the UDP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSF-008	BPSendFile STCP Convergence Layer	The BPSendFile shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the STCP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendFile is configured to use the STCP convergence layer and bundles are received successfully.
HDTN BPSF-009	BPSendFile Bundle Protocol Version 6	The BPSendFile shall support Bundle Protocol version 6.	Bundle Protocol version 6 is required for legacy implementations.	Demonstration	This requirement is verified when a demonstration shows BPSendFile is configured to use Bundle Protocol Version 6 and bundles are received successfully.
HDTN BPSF-010	BPSendFile Bundle Protocol Version 7	The BPSendFile shall support Bundle Protocol version 7.	Bundle Protocol version 7 is the latest specification.	Demonstration	This requirement is verified when a demonstration shows BPSendFile is configured to use Bundle Protocol Version 7 and bundles are received successfully.
HDTN BPSF-011	BPSendFile Bundle Protocol Security	The BPSendFile shall support Bundle Protocol Security (BPSec) [RFC 9172/RFC 9173].	BPSec is required for secure communications.	Demonstration	This requirement is verified when a demonstration shows BPSendFile is configured to use Bundle Protocol Security and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSF-012	BPSendFile Aggregate Custody Signals	The BPSendFile application shall allow aggregate custody signal custody transfer usage.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSendFile application is configured to use aggregate custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.
HDTN BPSF-013	BPSendFile Bidirectional Communication	The BPSendFile application shall support bidirectional communication for receiving custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSendFile application is configured to use custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.
HDTN BPSF-014	BPSendFile Bundle Priority	The BPSendFile application shall have the option to specify the bundle priority from the predefined range for BP version 6.	Bundle priority is a part of RFC 5050 and QoS support.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the specified bundle priority.
HDTN BPSF-015	BPSendFile Bundle Lifetime	The BPSendFile application shall have the option to specify the bundle lifetime.	The user can specify the bundle lifetime, which helps define the bundle time to live before it expires.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the bundle lifetime.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSF-016	BPSendFile Convergence Layer Transmission Rate	BPSendFile application shall have the option to set the convergence layer transmission rate.	LTP and UDP must have a rate limit configured since they do not inherently implement flow control.	Demonstration	This requirement is verified when a demonstration shows an option exists to set the convergence layer transmission rate.
HDTN BPSF-017	BPSendFile Recursive Directories Depth	BPSendFile application shall have the option to specify the recursive directory depth.	The user can specify the directory depth to include subdirectories.	Demonstration	This requirement is verified when a demonstration shows an option to specify the recurse directory depth.
HDTN BPSF-018	BPSendFile Source URI Endpoint Identifier	BPSendFile application shall have the option to specify the source Uniform Resource Identifier (URI) Endpoint Identifier.	The user can specify the source URI endpoint identifier to identify the originator of the bundle.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the source URI endpoint identifier.
HDTN BPSF-019	BPSendFile Destination URI Endpoint Identifier	BPSendFile application shall have the option to specify the destination URI Endpoint Identifier.	The user can specify the destination URI endpoint identifier.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the destination URI endpoint identifier.
HDTN BPSF-020	BPSendFile Custodian Service ID	BPSendFile application shall have the option to specify the Custodian Service ID for the node.	The user can specify the custodian service ID to identify who will receive a response.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the custodian service ID.

3.4.5 BPReceiveFile Application Requirements

The BPReceiveFile application receives bundles sent from the BPSendFile application in any order. It reassembles the file fragments, closes the file when all file fragments have been received and writes them to a user-specified directory. The code of this application serves as an example for users who want to write custom applications for HDTN that handle unidirectional bundles. The following requirements capture the expected functionality for the application.

Table 3-13 BPReceiveFile Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRF-001	BPReceiveFile Maximum Bundle Size	The BPReceiveFile application shall allow the user to configure the maximum bundle size.	There is no maximum size for a bundle according to BPv6 and BPv7, so the user sets the maximum size limit.	Demonstration	This requirement is verified when a demonstration shows the user can configure the maximum bundle size.
HDTN BPRF-002	BPReceiveFile Save Directory	The BPReceiveFile application shall have the option to save the receiving files to a user-specified directory.	To save the receiving files from BPSendFile to a user-specified directory.	Demonstration	This requirement is verified when a demonstration shows an option exists to save the receiving files to a user-specified directory and the files are received successfully.
HDTN BPRF-003	BPReceiveFile LTP Convergence Layer	The BPReceiveFile shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the LTP convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use the LTP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRF-004	BPReceiveFile TCP Version 4 Convergence Layer	BPReceiveFile shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the TCP version 4 convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use the TCP version 4 convergence layer and bundles are received successfully.
HDTN BPRF-005	BPReceiveFile UDP Convergence Layer	BPReceiveFile shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the UDP convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use the UDP convergence layer and bundles are received successfully.
HDTN BPRF-006	BPReceiveFile STCP Convergence Layer	BPReceiveFile shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the STCP convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use the STCP convergence layer and bundles are received successfully.
HDTN BPRF-007	BPReceiveFile Bundle Protocol Version 6	BPReceiveFile shall support Bundle Protocol version 6.	Bundle Protocol version 6 is required for legacy implementations.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use Bundle Protocol Version 6 and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRF-008	BPReceiveFile Bundle Protocol Version 7	BPReceiveFile shall support Bundle Protocol version 7.	Bundle Protocol version 7 is the latest specification.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use Bundle Protocol Version 7 and bundles are received successfully.
HDTN BPRF-009	BPReceiveFile Bundle Protocol Security	BPReceiveFile shall support Bundle Protocol Security (BPSec) [RFC 9172/RFC 9173].	BPSec is required for secure communications.	Demonstration	This requirement is verified when a demonstration shows BPReceiveFile is configured to use Bundle Protocol Security and bundles are received successfully.
HDTN BPRF-010	BPReceiveFile Aggregate Custody Signals	The BPReceiveFile application shall allow aggregate custody signal custody transfer usage.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPReceiveFile application is configured to use aggregate custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRF-011	BPReceiveFile Bidirectional Communication	The BPReceiveFile application shall support bidirectional communication for receiving custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPReceiveFile application is configured to use custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.
HDTN BPRF-012	BPReceiveFile Source URI Endpoint Identifier	The BPReceiveFile application shall have the option to specify the source URI Endpoint Identifier.	The user can specify the source URI endpoint identifier.	Demonstration	This requirement is verified when a demonstration shows an option exists to specify the source URI endpoint identifier.

3.4.6 BPSendPacket Application Requirements

The BpSendPacket application receives a data payload over UDP or STCP, extracts, bundles, and sends it over an HDTN-supported convergence layer. This was implemented to provide external applications a convenient method to bundle data and utilize delay-tolerant capabilities to send it. The following requirements capture the expected functionality for the application.

Table 3-14 BPSendPacket Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSP-001	BPSendPacket Induct Configuration	The BPSendPacket application shall allow the user to configure an induct that will be used to receive a data payload from a local UDP client.	The user can specify the induct type to use to provide data payload to the application.	Demonstration	This requirement is verified when a demonstration shows that BPSendPacket is configured to receive data payload from a local UDP client and that the data is received successfully by HDTN.
HDTN BPSP-002	BPSendPacket Maximum Bundle Size	The BPSendPacket application shall allow the user to specify the maximum bundle size.	There is no maximum size for a bundle according to BPv6 and BPv7, so the user sets the maximum size limit.	Demonstration	This requirement is verified when a demonstration shows the user can configure the maximum bundle size.
HDTN BPSP-003	BPSendPacket LTP Convergence Layer	The BPSendPacket shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the LTP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendPacket is configured to use the LTP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSP-004	BPSendPacket TCP Version 4 Convergence Layer	The BPSendPacket shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the TCP version 4 convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendPacket is configured to use the TCP version 4 convergence layer and bundles are received successfully.
HDTN BPSP-005	BPSendPacket UDP Convergence Layer	The BPSendPacket shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the UDP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendPacket is configured to use the UDP convergence layer and bundles are received successfully.
HDTN BPSP-006	BPSendPacket STCP Convergence Layer	The BPSendPacket shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the STCP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendPacket is configured to use the STCP convergence layer and bundles are received successfully.
HDTN BPSP-007	BPSendPacket Bundle Protocol Version 6	The BPSendPacket shall support Bundle Protocol version 6.	Bundle Protocol version 6 is required for legacy implementations.	Demonstration	This requirement is verified when a demonstration shows BPSendPacket is configured to use Bundle Protocol Version 6 and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSP-008	BPSendPacket Bundle Protocol Version 7	The BPSendPacket shall support Bundle Protocol version 7.	Bundle Protocol version 7 is the latest specification.	Demonstration	This requirement is verified when a demonstration shows BPSendPacket is configured to use Bundle Protocol Version 7 and bundles are received successfully.
HDTN BPSP-009	BPSendPacket Bundle Protocol Security	The BPSendPacket shall support Bundle Protocol Security (BPSec) [RFC 9172/RFC 9173].	BPSec is required for secure communications.	Demonstration	This requirement is verified when a demonstration shows that the BPSendPacket is configured to use Bundle Protocol Security and bundles are received successfully.
HDTN BPSP-010	BPSendPacket Aggregate Custody Signals	The BPSendPacket application shall allow custody transfer using aggregate custody signals.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSendPacket application is configured to use aggregate custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSP-011	BPSendPacket Bidirectional Communication	The BPSendPacket application shall support bidirectional communication when receiving custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows the BPSendPacket application is configured to use custody signals when a Bundle Protocol Version 6 is configured and bundles are transmitted successfully.
HDTN BPSP-012	BPSendPacket Bundle Lifetime	The BPSendPacket application shall have the option to specify the bundle lifetime.	The user can specify the bundle lifetime, which helps define the bundle time to live before it expires.	Test	This requirement is verified when a test shows bundles generated by BPSendPacket contain the bundle lifetime.
HDTN BPSP-013	BPSendPacket Convergence Layer Transmission Rate	The BPSendPacket application shall have the option to set the convergence layer transmission rate.	LTP and UDP must have a rate limit configured since they do not inherently implement flow control.	Demonstration	This requirement is verified when a demonstration shows an option exists to set the convergence layer transmission rate.
HDTN BPSP-014	BPSendPacket Source URI Endpoint Identifier	The BPSendPacket application shall have the option to specify the URI Endpoint Identifier source.	The user can specify the source URI endpoint identifier to identify the bundle’s originator.	Test	This requirement is verified when a test shows bundles generated by BPSendPacket contain the source URI endpoint identifier.
HDTN BPSP-015	BPSendPacket Destination URI Endpoint Identifier	The BPSendFile application shall have the option to specify the destination URI Endpoint Identifier.	The user can specify the destination URI endpoint identifier.	Test	This requirement is verified when a test shows bundles generated by BPSendFile contain the destination URI endpoint identifier.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPSP-016	BPSendPacket Custodian Service ID	The BPSendPacket application shall have the option to specify the Custodian Service ID for the node.	The user can specify the custodian service ID to identify who will receive a response.	Test	This requirement is verified when a test shows bundles generated by BPSendPacket contain the custodian service ID.

3.4.7 BPReceivePacket Application Requirements

The BpReceivePacket application receives bundles over a DTN convergence layer supported by HDTN, converts it into a data payload, and sends it via a UDP or STCP network session. This was implemented to provide external applications a convenient method to receive data over delay-tolerant protocols. The following requirements capture the expected functionality for the application.

Table 3-15 BPReceivePacket Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRP-001	BPReceivePacket Outduct UDP	The BPReceivePacket application shall allow users to deliver a data payload to a local application listening on a UDP socket.	HDTN needs to be compatible with flight software like Fprime, and it needs to have a tool that converts bundles to Fprime data and sends them to a UDP socket and to Fprime flight software.	Demonstration	This requirement is verified when a demonstration shows that BPReceivePacket is configured to send data payload to a local application listening on a UDP socket, and the data is received successfully by that application.
HDTN BPRP-002	BPReceivePacket Outduct STCP	The BPReceivePacket application shall allow users to deliver a data payload to a local application listening on an STCP socket.	HDTN needs to be compatible with flight software like Fprime and needs to have a tool that converts bundles to Fprime data and sends them on an STCP socket to Fprime flight software.	Demonstration	This requirement is verified when a demonstration shows that the BPReceivePacket is configured to send data payload to a local application listening on an STCP socket and that the data is received successfully.
HDTN BPRP-003	BPReceivePacket Maximum Bundle Size	The BPReceivePacket application shall allow the user to configure the maximum bundle size.	There is no maximum size for a bundle according to BPv6 and BPv7, so the user sets the maximum size limit.	Demonstration	This requirement is verified when a demonstration shows the user can configure the maximum bundle size.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRP-004	BPReceivePacket LTP Convergence Layer	The BPReceivePacket shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the LTP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPReceivePacket is configured to use the LTP convergence layer and bundles are received successfully.
HDTN BPRP-005	BPReceivePacket TCP Version 4 Convergence Layer	The BPReceivePacket shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the TCP version 4 convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceivePacket is configured to use the TCP version 4 convergence layer and bundles are received successfully.
HDTN BPRP-006	BPReceivePacket UDP Convergence Layer	The BPReceivePacket shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the UDP convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceivePacket is configured to use the UDP convergence layer and bundles are received successfully.
HDTN BPRP-007	BPReceivePacket STCP Convergence Layer	The BPReceivePacket shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the STCP convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPReceivePacket is configured to use the STCP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRP-008	BPReceivePacket Bundle Protocol Security	The BPReceivePacket shall support Bundle Protocol Security (BPSec) [RFC 9172/RFC 9173].	BPSec is required for secure communications.	Demonstration	This requirement is verified when a demonstration shows BPReceivePacket is configured to use Bundle Protocol Security and bundles are received successfully.
HDTN BPRP-009	BPReceivePacket Aggregate Custody Signals	The BPReceivePacket application shall allow usage of aggregate custody signal custody transfer.	Aggregate custody signals provide bundle layer reliability when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows that the BPReceivePacket application uses aggregate custody signals when a Bundle Protocol Version 6 is configured, and bundles are transmitted successfully.
HDTN BPRP-010	BPReceivePacket Bidirectional Communication	The BPReceivePacket application shall support bidirectional communication to receive custody signals.	Bidirectional communication is needed for acknowledgments when using the Bundle Protocol version 6.	Demonstration	This requirement is verified when a demonstration shows that the BPReceivePacket application is configured to use custody signals when a Bundle Protocol Version 6 is configured, and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN BPRP-011	BPReceivePacket Source URI Endpoint Identifier	The BPReceivePacket application shall have the option to specify the source URI Endpoint Identifier.	The user can specify the source URI endpoint identifier.	Demonstration	This requirement is verified when a demonstration shows an option exists to specify the source URI endpoint identifier.

3.4.8 BPSendStream Application Requirements

The BPSendStream application allows the transmission of video and audio data over an intermittent network environment. BPSendStream receives Real-Time Protocol (RTP) packets as input directly from an RTP stream or a file path to an H.264 encoded video. BPSendStream encapsulates the RTP packets into bundles and then transmits the bundles via a DTN Convergence Layer supported by HDTN. The following requirements capture the expected functionality for the application.

Table 3-16 BPSendStream Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM S-001	HDTN Streaming Bundle Creation	The BPSendStream application shall transform RTP into bundles.	An outduct that implements CCSDS 766.3-R-2 will be used.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is provided an RTP stream and bundles are transmitted.
HDTN STRM S-002	HDTN Streaming Bundle Transmit	The BPSendStream application shall transmit bundles to a specified destination.	A recipient for the bundles needs to be specified to be transmitted.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is provided to a recipient and the bundles are transmitted to the specified recipient.
HDTN STRM S-003	HDTN Source Node Identifier Sender	The BPSendStream application shall allow the user to provide a source Endpoint ID.	Allows the application to identify itself as the bundle originator.	Test	This requirement is verified when a test shows bundles generated by the BPSendStream application contain the specified source Endpoint ID.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM S-004	HDTN Streaming Send LTP Convergence Layer	The BPSendStream application shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendStream is configured to use an LTP convergence layer and bundles are transmitted successfully.
HDTN STRM S-005	HDTN Streaming Send TCP Convergence Layer	The BPSendStream application shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a TCP v4 convergence layer and bundles are transmitted successfully.
HDTN STRM S-006	HDTN Streaming Send UDP Convergence Layer	The BPSendStream application shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendStream is configured to use a UDP convergence layer and bundles are transmitted successfully.
HDTN STRM S-007	HDTN Streaming Send STCP Convergence Layer	The BPSendStream application shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that the BPSendStream is configured to use an STCP convergence layer and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM S-008	HDTN Streaming Send Bundle Protocol Version 6	The BPSendStream application shall support Bundle Protocol version 6.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a Bundle Protocol v6 convergence layer and bundles are transmitted successfully.
HDTN STRM S-009	HDTN Streaming Send Bundle Protocol Version 7	The BPSendStream application shall support Bundle Protocol version 7.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a Bundle Protocol v7 convergence layer and bundles are transmitted successfully.
HDTN STRM S-010	HDTN Streaming Send BPSec Support	The BPSendStream application shall support BPSec (RFC9172/RFC9173).	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is configured to use Bundle Protocol Security and bundles are transmitted successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM S-011	HDTN Streaming Stream Circular Buffer Vectors	The BPSendStream application shall have the option to specify the number of circular buffer vectors.	The number of circular buffer vector elements refers to the size of the circular buffer used by the UDP sink to store incoming RTP packets before processing. The user can specify the number of circular buffer vectors.	Inspection	This requirement is verified when a code inspection shows that BPSendStream is configured to use the number of circular buffer vectors specified by the user.
HDTN STRM S-012	HDTN Streaming Max Incoming UDP Packet Size	The BPSendStream application shall have the option to specify the maximum incoming UDP packet size in bytes.	Max size of incoming UDP packets from the RTP stream. The user can specify the maximum incoming UDP packet size in bytes.	Test	This requirement is verified when a test shows that BPSendStream, configured with a maximum incoming UDP packet size, 1) truncates incoming packets that are larger than the specified file and 2) fully receives packets that are equal to or smaller than the configured value.
HDTN STRM S-013	HDTN Streaming RTP Stream Listening Port	The BPSendStream application shall have the option to specify the port that will listen for a RTP stream.	The user can specify the port that will listen for a RTP stream. This is applicable to both UDP and TCP connections.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to listen for an RTP stream on the user-specified port.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM S-014	HDTN Streaming RTP Packets Per Bundle	The BPSendStream application shall have the option to specify the number of RTP packets placed into a bundle.	The user can specify the number of RTP packets placed into a bundle.	Test	This requirement shall be considered verified when a test shows that BPSendStream, configured with a specified packets-per-bundle value, sends bundles containing the number of RTP packets as configured.
HDTN STRM S-015	HDTN Streaming Induct Type	The BPSendStream application shall have the option to specify the induct type.	The user can specify the induct type used. If appsink is specified, an input must be specified for the streaming file. The UDP expects an RTP stream over a UDP socket. The TCP expects an RTP stream over a TCP socket.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use the user-specified induct type.
HDTN STRM S-016	HDTN Streaming File To Stream	The BPSendStream application shall have the option to specify the file path for an H.264 encoded video file to stream.	The user can specify the file path for an H.264 encoded video file to stream.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured by the user to stream the video file specified from the file path.

3.4.9 BPReceiveStream Application Requirements

The BPReceiveStream application receives bundles containing RTP packets via a DTN Convergence Layer, supported by HDTN. BPReceiveStream decapsulates the bundles and outputs the RTP packets, allowing a media player application to reproduce the received audio and video data. The following requirements capture the expected functionality for the application.

Table 3-17 BPReceiveStream Application Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM R-001	HDTN Streaming Bundle Receive	The BPReceiveStream application shall de-encapsulate RTP Frames from received bundles.	The RTP frames need to be populated by the received bundles to provide the RTP stream.	Test	This requirement shall be considered verified when a test shows that BPReceiveStream successfully receives and de-encapsulates RTP Frames from bundles transmitted by another node.
HDTN STRM R-002	HDTN Source Node Identifier Receiver	The BPReceiveStream application shall allow the user to provide a source Endpoint ID.	The Endpoint ID allows the application to identify that it is the intended recipient of a received bundle.	Test	This requirement is verified when a test is performed where BPReceiveStream is provided bundle data with a destination Endpoint ID that matches the specified source Endpoint ID and can receive the data.
HDTN STRM R-003	HDTN Streaming Receive LTP Convergence Layer	The BPReceiveStream application shall support the LTP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPReceiveStream is configured to use an LTP convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM R-004	HDTN Streaming Receive TCP Convergence Layer	The BPReceiveStream application shall support the TCP version 4 convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a TCP v4 convergence layer and bundles are received successfully.
HDTN STRM R-005	HDTN Streaming Receive UDP Convergence Layer	The BPReceiveStream application shall support the UDP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is configured to use a UDP convergence layer and bundles are received successfully.
HDTN STRM R-006	HDTN Streaming Receive STCP Convergence Layer	The BPReceiveStream application shall support the STCP convergence layer.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is configured to use an STCP convergence layer and bundles are received successfully.
HDTN STRM R-007	HDTN Streaming Receive Bundle Protocol Version 6	The BPReceiveStream application shall support Bundle Protocol version 6.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a Bundle Protocol v6 convergence layer and bundles are received successfully.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM R-008	HDTN Streaming Receive Bundle Protocol Version 7	The BPReceiveStream application shall support Bundle Protocol version 7.	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows that BPSendStream is configured to use a Bundle Protocol v7 convergence layer and bundles are received successfully.
HDTN STRM R-009	HDTN Streaming Receive BPSec Support	The BPReceiveStream application shall support BPSec (RFC9172/RFC9173).	The convergence layer requirements are implemented as part of the HDTN development. This application can be configured to use the convergence layer.	Demonstration	This requirement is verified when a demonstration shows BPSendStream is configured to use Bundle Protocol Security and bundles are received successfully.
HDTN STRM R-010	HDTN Streaming Receive Circular Buffer Vectors	The BPReceiveStream application shall have the option to specify the number of circular buffer vectors.	The number of circular buffer vector elements refers to the size of the circular buffer used by the UDP sink to store incoming RTP packets before processing. The user can specify the number of circular buffer vectors.	Inspection	This requirement is verified when a code inspection shows that BPReceiveStream is configured to use the number of circular buffer vectors specified by the user.
HDTN STRM R-011	HDTN Streaming Max Outgoing UDP Packet Size	The BPReceiveStream application shall have the option to specify the maximum outgoing UDP packet size in bytes.	Max size of outgoing UDP packets from the RTP stream. The user can specify the maximum outgoing UDP packet size in bytes.	Test	This requirement is verified when a test shows that the BPReceiveStream is configured with maximum packet size and that sent packets do not exceed the specified size.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM R-012	HDTN Streaming Outgoing RTP Port	The BPReceiveStream application shall have the option to specify the outgoing RTP port.	The user can specify a port to send the RTP packets.	Demonstration	This requirement is verified when a demonstration shows BPReceiveStream is configured to use the outgoing RTP port specified by the user and RTP packets are received on the specified port.
HDTN STRM R-013	HDTN Streaming Outgoing RTP Hostname	The BPReceiveStream application shall have the option to specify the outgoing RTP hostname.	The user can specify a hostname to send the RTP packets.	Demonstration	This requirement is verified when a demonstration shows BPReceiveStream is configured to use the outgoing RTP hostname specified by the user and RTP packets are received using the specified hostname.
HDTN STRM R-014	HDTN Streaming Shared Memory Socket Path	The BPReceiveStream application shall have the option to specify the location of the socket for shared memory sink to GStreamer.	The user can specify the location of the socket for the shared memory sink with GStreamer when the shared memory type is selected as outduct.	Demonstration	This requirement is verified when a demonstration shows that BPReceiveStream is configured to use the user-specified socket location for shared memory sink with GStreamer.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN STRM R-015	HDTN Streaming Outduct Type	The BPReceiveStream application shall have the option to specify the type of outduct	The user can specify the type of outduct to use.	Demonstration	This requirement is verified when a demonstration shows that the BPReceiveStream is configured to use the outduct type specified by the user.

3.5 Routing Requirements

The HDTN routing requirements are built on the concept that a contact plan will determine when neighboring nodes can send and receive data. The requirements do not delve into the specific details of any routing algorithms. The router may compute multi-hop routes or obtain routing information from a precomputed contact plan.

Table 3-18 Routing Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN ROUTING-001	Contact Plan Modification	All route lists shall be recomputed when the contact plan has been modified.	Contact plan changes may invalidate any or all earlier route computations. When the contact plan is updated, new routes are calculated.	Test	This requirement is verified when a test updates the HDTN contact plan and confirms the route list has been recomputed.
HDTN ROUTING-002	Expired Contacts	Expired contacts shall be deleted from the contact graphs.	These contacts have exceeded their defined validity period and are no longer used.	Test	This requirement is verified when a test loads a contact plan with expired contacts into HDTN and confirms the expired contacts have been removed from the contact plan.
HDTN ROUTING-003	Route Computation	The route with the earliest arrival time shall be selected from the list of candidate routes.	Contact Graph Routing (CGR) and Contact Multigraph Routing (CMR) use the earliest arrival time to select the best route.	Test	This requirement is verified when a test loads a contact plan into HDTN and verifies that the route with the earliest arrival time is selected from the list of candidate routes.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN ROUTING-004	CGR Preparation	The list of candidate routes shall be computed from the contact plan.	The contact plan is the schedule of contacts for each node. It is the input to the router. The list of candidate routes is computed for each final destination based on the root contact, final destination, and contact start and stop times. All routes are calculated to select the best.	Test	This requirement is verified when a test loads a contact plan into HDTN and verifies the computed candidate routes match the expected values.
HDTN ROUTING-005	Routing Algorithm Selection	The routing algorithm shall be selected from one of the following: contact graph routing or contact multigraph routing.	HDTN is intended to support multiple routing algorithms, including CGR, CMR, and others. The algorithm is selected by the user.	Inspection	This requirement is verified when an inspection of the code shows that CGR and CMR are both supported as routing algorithms.
HDTN ROUTING-006	Rerouting Around Failed Node	The router shall select the route with the next earliest arrival time if it detects that the current route has failed.	All routes have been precomputed so that the next best route can be used if the current route fails.	Test	This requirement is verified when a test simulates a route failure on the selected route and confirms that HDTN selects a new route.

3.6 HDTN Environment Requirements

The following requirements are regarding the environment in which the HDTN CSCI is meant to operate.

Table 3-19 HDTN Environment Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-001	HDTN Bundle Size	The HDTN software shall process bundle sizes ranging from 100 B to 4 MB.	This supports a range of data types, from telemetry to scientific.	Demonstration	This requirement is verified when a demonstration shows HDTN is provided a bundle of 100 B and 4 MB and transmitted successfully.
HDTN REQ-002	HDTN Instantaneous Data Rate Minimum	The HDTN Software shall have a minimum 1.25 Gbps instantaneous data rate on a platform consistent with the ISS ILLUMA-T communications platform.	This is the maximum laser modulator rate at the Physical Layer, and we need to meet this rate to ensure that the HDTN software does not cause a bottleneck. This rate will be measured on the ISS laptop (Intel I7 processor) in a laboratory environment.	Demonstration	This requirement is verified when a demonstration shows HDTN can achieve a minimum instantaneous data rate of 1.25 Gbps.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-003	HDTN Minimum link latency	The HDTN Software shall operate across a minimum of 4-second link latency.	This represents the total round trip time from the ISS ILLUMA-T to the ground through the Laser Communications Relay Demonstration (LCRD) network and is considered a minimum link-latency requirement. Future lunar missions may require longer latencies as their mission characteristics are defined.	Demonstration	This requirement is verified when a demonstration shows the network environment has at least a 4-second link latency and data can be routed successfully.
HDTN REQ-004	HDTN Target Operating System – Ubuntu	The HDTN software shall be compatible with the Ubuntu 20.04.2 Long Term Support (LTS) operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for Operating Systems (OSs) not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the Ubuntu OS without encountering a fatal error.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-005	Configuration File Parameter Validation	The HDTN software shall verify that required parameters are provided at startup.	HDTN needs these parameters to operate. Missing parameters could cause undetermined behavior. The parameters are defined in HDTN-SWDD-017 HDTN Data Dictionary.	Demonstration	This requirement is verified when a demonstration shows that the HDTN application logs an error and terminates its execution when a required parameter is not provided at startup.
HDTN REQ-006	HDTN Target Operating System – Red Hat Enterprise Linux 8	The HDTN software shall be compatible with the Red Hat Enterprise Linux (RHEL) 8 operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the RHEL 8 OS without encountering a fatal error.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-007	HDTN Target Operating System – Windows 11	The HDTN software shall be compatible with the Windows 11 (64-bit) operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the Windows 11 (64-bit) OS without encountering a fatal error.
HDTN REQ-008	HDTN Target Operating System – OpenBSD	The HDTN software shall be compatible with the OpenBSD operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the OpenBSD OS without encountering a fatal error.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-009	HDTN Target Operating System – FreeBSD	The HDTN software shall be compatible with the FreeBSD operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the FreeBSD OS without encountering a fatal error.
HDTN REQ-010	HDTN Target Operating System – MacOS	The HDTN software shall be compatible with the MacOS operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the MacOS OS without encountering a fatal error.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN REQ-011	HDTN Target Operating System – Windows Server 2022	The HDTN software shall be compatible with the Windows Server 2022 (64-bit) operating system.	HDTN supports a variety of missions with varying operating systems. Rehosting HDTN for OSs not specified in this document is not scoped in the budget or schedule for HDTN development. Compatibility constraints for software are required to limit developmental scope within the budget made available by project management.	Demonstration	This requirement is verified when a compiled HDTN binary is demonstrated to execute on the Windows Server 2022 (64-bit) OS without encountering a fatal error.

3.7 HDTN Security and Privacy Requirements

HDTN assessed compliance with the Space Systems Protection Standard (NASA STD 1006 W/change 1). (see Appendix C) It was determined that the standard does not apply to HDTN software because it is a delivery service for a mission's communications. The mission adopting HDTN would need to assess this standard for compliance with the specific knowledge of how commanding, data, security, and other environments will be implemented in the mission.

3.8 HDTN Safety Requirements

All potential hazard conditions are expected to be mitigated by the projects adopting HDTN. HDTN software does not create or mitigate hazards and is not safety-critical. Safety criticality has no impact on the requirements or design of the software.

3.9 HDTN Invalid Inputs Requirements

HDTN software does not have individual requirements for handling generically invalid inputs. However, there are requirements within the specifications that specify required formatting and how improperly formatted data is dealt with. The handling of invalid configuration files is left to be defined in the design.

3.10 HDTN Internal Data Requirements

Decisions about HDTN's internal data are left to be defined in the design.

3.11 HDTN Internal Interface Requirements

Decisions about HDTN's internal interfaces are left to be defined in the design.

3.12 HDTN Application Programming Interface (API) Requirements

The following requirements are regarding the API that provides a means for external systems to configure and control the behavior of the HDTN CSCI.

Table 3-20 HDTN API Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN TLM-001	HDTN Getter API Calls	HDTN shall provide a ZeroMQ-Based API to retrieve information on HDTN's configurations and statistics.	The API can retrieve information on HDTN's configurations and statistics.	Demonstration	This requirement is verified when a demonstration is performed where ZeroMQ requests are sent and the requested information is retrieved.
HDTN TLM-002	HDTN Maximum Send Rate for an Outduct API Call	HDTN shall provide a ZeroMQ-Based API to set the maximum send rate in bits per second for a specific outduct in HDTN.	This API Call can set the maximum send rate in bits per second for a specific outduct in HDTN.	Demonstration	This requirement is considered verified when a demonstration is performed where ZeroMQ requests are sent to set the maximum send rate for a specific outduct in HDTN, and a success acknowledgment is received.
HDTN TLM-003	HDTN Upload Contact Plan API Call	HDTN shall provide a ZeroMQ-Based API to upload a contact plan for HDTN.	This API Call can be used to upload a contact plan for HDTN.	Demonstration	This requirement is verified when a demonstration is performed where ZeroMQ requests are sent to upload a contact plan for HDTN, and a success acknowledgment is received.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN TLM-004	HDTN Ping	HDTN shall provide a ZeroMQ-Based API to ping an HDTN node's specific service.	This API Call can ping a specific service of an HDTN node.	Demonstration	This requirement is considered verified when a demonstration is performed where ZeroMQ requests are sent to ping a specific service of an HDTN node and a success acknowledgment is received.
HDTN TLM-005	HDTN Take Link Down	HDTN shall provide a ZeroMQ-Based API to set the link down from the outductVector.	This API Call can be used to set a link down from the outductVector.	Demonstration	This requirement is considered verified when a demonstration is performed where ZeroMQ requests are sent to take a link down and a success acknowledgment is received.
HDTN TLM-006	HDTN Bring Link Up	HDTN shall provide a ZeroMQ-Based API to set the link up from the outductVector.	This API Call can set up a link from the outductVector.	Demonstration	This requirement is considered verified when a demonstration is performed where ZeroMQ requests are sent to bring a link up and a success acknowledgment is received.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTN TLM-007	HDTN Restart API Call	HDTN shall provide a ZeroMQ-Based API to restart HDTN with an optional argument of a new configuration.	This API Call can be used to restart HDTN.	Demonstration	This requirement is considered verified when a demonstration is performed where ZeroMQ requests are sent to restart HDTN with/without a new configuration, and a success acknowledgment is received.

3.13 HDTN Graphical User Interface (GUI) Requirements

The following requirements regarding the GUI allow users to interact with the HDTN CSCI via graphical components.

Table 3-21 HDTN GUI Requirements

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTNGUI-001	Data Rate Display	The HDTN user interface shall display data rates.	The data rates are displayed to assess the network performance over time.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays HDTN data rates.
HDTNGUI-002	Web Browser GUI	The HDTN user interface shall be accessible via a web browser.	A web browser allows system monitoring without having to load dedicated software.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI is accessible via a web browser without loading dedicated software.
HDTNGUI-003	Storage Metrics Display	The HDTN user interface shall display storage metrics listed in Table E-1 on Appendix E.	Storage metrics allow monitoring of the storage usage and capacity.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can display HDTN storage capacity and usage.
HDTNGUI-004	LTP Metrics Display	The HDTN user interface shall display the LTP metrics listed in Table E-2 on Appendix E.	LTP metrics provide insight into the behavior of the LTP convergence layer.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays LTP metrics.
HDTNGUI-005	STCP Metrics Display	The user interface shall display the STCP metrics listed in Table E-3 on Appendix E.	STCP metrics provide insight into the behavior of the STCP convergence layer.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays STCP metrics.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTNGUI-006	TCP Metrics Display	The user interface shall display the TCP metrics listed in Table E-4 on Appendix E.	TCP metrics provide insight into the behavior of the TCP convergence layer.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays TCP metrics.
HDTNGUI-007	UDP Metrics Display	The user interface shall display the UDP metrics listed in Table E-5 on Appendix E.	UDP metrics provide insight into the behavior of the UDP convergence layer.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays UDP metrics.
HDTNGUI-008	System View Display	The user interface shall display a system view showing the HDTN components and how data flows through the system.	The system view provides a way to quickly understand the current state of HDTN and its main components.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays a system view with HDTN components and shows how data flows through the system.
HDTNGUI-009	System View Display Preferences	The user interface shall provide display preference options for the system view.	This allows the system view to adapt to different screen sizes and user preferences, including font size and color theme.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI correctly displays on various screen sizes.
HDTNGUI-010	Ping Via GUI	The user interface shall support sending a ping command.	This lets the user quickly determine whether a connection exists to the given node.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can send a ping to a requested node.
HDTNGUI-011	GUI Outduct Display	The system view shall display metrics when the cursor is over the outduct.	This provides insight into the behavior of the convergence layers.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays outduct metrics when the cursor is over the outduct.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTNGUI-012	GUI Induct Display	The system view shall display metrics when the cursor is over the induct.	This provides insight into the behavior of the convergence layers.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays induct metrics when the cursor is over the induct.
HDTNGUI-013	GUI Storage Display	The system view shall display storage component metrics when the cursor is over the storage module.	This allows storage usage and capacity to be monitored.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI displays storage metrics when the cursor is over the storage module.
HDTNGUI-014	BPSec Policy Rules Configuration	The HDTN user interface shall allow users to configure the BPSec policy rules.	This allows the user to configure the HDTN system without manual editing of JSON files and having to reference key/value documentation.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can input and configure BPSec policy rules.
HDTNGUI-015	BPSec Failure Events Configuration	The HDTN user interface shall allow users to configure the BPSec Failure events actions.	This allows the user to configure the HDTN system without manual editing of JSON files and having to reference key/value documentation.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can input and configure BPSec failure event actions.
HDTNGUI-016	System (HDTN) Configuration	The HDTN user interface shall allow the user to configure the HDTN/System config.	This allows the user to configure the HDTN system without manual editing of JSON files and having to reference key/value documentation.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can input and configure the HDTN System.

SW Req ID	Title	Requirement	Rationale	Verification Method	Verification Statement
HDTNGUI-017	Distributed Configuration	The HDTN user interface shall allow users to configure the Distributed HDTN config.	This allows the user to configure the HDTN system without manual editing of JSON files and having to reference key/value documentation.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can input and configure Distributed HDTN.
HDTNGUI-018	Contact Plan Configuration	The HDTN user interface shall allow the user to configure the contact plan	This allows the user to configure the HDTN system without manual editing of JSON files and having to reference key/value documentation	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can input and configure the contact plan.
HDTNGUI-019	Configuration Copy	The HDTN user interface shall allow the user to copy the configurations to the clipboard.	This allows the user to copy HDTN system configurations into existing files.	Demonstration	This requirement is considered verified when a demonstration is performed where the GUI can copy HDTN configurations to the clipboard.

4.0 REQUIREMENTS TRACEABILITY AND VERIFICATION METHODS.

The verification methods are documented in the HDTN-PLAN-022 Software Verification and Validation Plan. The Requirements Traceability Matrix is currently maintained in the HDTN MagicDraw project.

APPENDIX A - DEFINITIONS

The Definitions table contains an alphabetized list of definitions for particular terms used in the document; that is, the terms are used in a sense that differs from or is more specific than the typical usage for such terms.

Table A-1 Definitions

Name	Documentation
MagicDraw®	A visual System Modeling Language (SysML) modeling tool to facilitate the analysis and design of systems and databases.

APPENDIX B - ACRONYMS AND ABBREVIATIONS

The Acronyms and Abbreviations table contains an alphabetized list of the definitions for abbreviations and acronyms used in this document.

Table B-1 Acronyms and Abbreviations

Acronym	Definition
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
API	Application Programming Interface
ASB	Abstract Security Block
ASCII	American Standard Code for Information Interchange
BCB	Block Confidentiality Block
BIB	Block Integrity Block
BP	Bundle Protocol
BPA	Bundle Protocol Agent
BPv6	Bundle Protocol version 6
BPv7	Bundle Protocol version 7
CA	Certificate Authority
CAR	Cancel-Acknowledgment segment to block Receiver
CAS	Cancel Acknowledgment to block Sender
CBHE	Compressed Bundle Header Encoding
CBOR	Concise Binary Object Representation
CCSDS	Consultative Committee for Space Data Systems
CGR	Contact Graph Routing
CL	Convergence Layer
CLA	Convergence-layer adapter
CM	Configuration Management
CMR	Contact Multigraph Routing
CONOPS	Concept of Operations
CP	Checkpoint
CR	Cancel by block Receiver
CS	Cancel by block Sender
CRC	Cyclic Redundancy Check
CSCI	Computer Software Configuration Item
DTN	Delay Tolerant Networking
EID	Endpoint ID
EORP	End of red-part
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
GRC	Glenn Research Center
HDTN	High-Rate Delay Tolerant Networking
HMAC	Hash-based message authentication code
IPN	InterPlaNet networking protocol

Acronym	Definition
LCRD	Laser Communications Relay Demonstration
LTP	Licklider Transmission Protocol
MRU	Maximum Receive Unit
MSB	Most Significant Bit
NASA	National Aeronautics and Space Administration
NPR	NASA Procedural Requirements
RA	Report Acknowledgment
RF	Radio Frequency
RFC	Request for Comments
RLEXC	Retransmission limit exceeded
RS	Report Segment
RTP	Real-Time Protocol
SCaN	Space Communications and Navigation
SDMP	Software Development and Management Plan
SDN	Software-Defined Networking
SDNV	Self-Delimiting Numeric Values
SHA	Secure Hash Algorithm
SHA2	Secure Hash Algorithm 2
SOMD	Space Operations Mission Directorate
SOPS	Security Operations
SRS	Software Requirements Specification
SSL	Secure Socket Layer
SSP	Scheme-Specific Part
STCP	Simple Transmission Control Protocol
STCPCL	Simple Transmission Control Protocol Convergence Layer
SW	Software
TBD	To Be Determined
TBR	To Be Resolved
TCP	Transmission Control Protocol
TCPCL	Transmission Control Protocol Convergence Layer
TLS	Transport Layer Security
UDP	User Datagram Protocol
URI	Uniform Resource Identifier

APPENDIX C - SPACE SYSTEMS PROTECTION STANDARD COMPLIANCE ASSESSMENT

Table C-1 NASA-STD-1006 W/CHANGE 1

Section	Description	Requirement in this Standard	Applicable	Comments
4.1.1	Command Stack Protection	[SSPR 1] Programs/projects shall protect the command stack with encryption that meets or exceeds the Federal Information Processing Standard (FIPS) 140, Security Requirements for Cryptographic Modules, Level 1.	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems.
4.1.2	Backup Command Link Protection	[SSPR 2] If a project uses an encrypted primary command link, any backup command link shall, at minimum, use authentication.	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems.
4.1.3	Command Link Critical Program/Project Information (CPI)	[SSPR 3] The program/project shall protect the confidentiality of command link CPI as NASA sensitive but unclassified (SBU) information to prevent inadvertent disclosure to unauthorized parties per NASA Interim Directive (NID) 1600.55, Sensitive But Unclassified (SBU) Controlled Information, and NPR 2810.1, Security of Information Technology	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems. The mission would provide security aspects for commands and data. HDTN has no plans to transmit SBU data.

4.2.1	Ensure Positioning, Navigation and Timing (PNT) Resilience	[SSPR 4] If project-external PNT services are required, projects shall ensure that systems are resilient to the complete loss of, or temporary interference with, external PNT services.	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems or provide navigational information.
4.3.1	Interference Reporting	[SSPR 5] Projects/Spectrum Managers/Operations Centers shall report unexplained interference to MRPP or to other designated notifying organizations.	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems.
4.3.2	Interference Reporting Training	[SSPR 6] Projects/Spectrum Managers/Operations Centers shall conduct proficiency training for reporting unexplained interference.	No	This does not apply to HDTN, but to the network we are working with. We are just the delivery service for data and do not control systems.

APPENDIX D - TBD/TBR LIST

This appendix provides a list of all TBD/TBR items contained within this document.

Identification of TBD/TBR within the document is as follows.

Where a TBD/TB is included within the text of this document, they **will** be incrementally numbered starting from SRS001 (preceded by a "-", dash) and formatted in ***bold italics***. For example: ***TBD-SRS001*** for TBD or ***TBR-SRS001*** for TBR.

Table D-1 TBD/TBR List.

TBD ID	Description	Task ID	Status	Section
TBD-SRS002	Release date for the HDTN Software Data Dictionary document.	SDF# 316	OPEN	2.1
TBR ID	**Description**	**Task ID**	**Status**	**Section**

APPENDIX E -HDTN GUI DISPLAY METRICS

Table E-1 Storage Metrics

bundles erased from storage	bundles sent to egress from storage	used space byte	free space bytes
data rates in to storage	data rates in and out of storage	percentage of disk used	

Table E-2 LTP Metrics

bundles received	bundle bytes received	report segment timer expired callbacks	report segments unable to be issued
report segments too large	report segments created via split	gaps filled by out of order data segments	delayed fully claimed primary report segments sent
delayed fully claimed secondary report segments sent	delayed partially claimed primary report segments sent	delayed partially claimed secondary report segments sent	cancel segments started
cancel segment send retries	cancel segments failed to send	cancel segments acknowledged	receiver sessions cancelled by sender
stagnant receiver sessions deleted	UDP packets sent, UDP buffer overruns	UDP packets limited by rate	bundles acknowledged
bundle bytes acknowledged	bundles sent	bundle bytes sent	bundles failed to send
physical link status	schedule link status	checkpoints expired	discretionary checkpoints not present
deleted fully claimed pending reports	cancel segments started	cancel segment send retries	cancel segments failed to send
cancel segments acknowledged	pings started	ping retries	pings failed to send
pings acknowledged	sender sessions returned to storage	sender sessions cancelled by receiver	UDP packets sent

Table E-3 STCP Metrics

bundles received	bundles bytes received	STCP bytes received	bundles acknowledged
bundles bytes acknowledged	bundles sent	bundles bytes sent	bundles failed to send
link physical status	schedule link status	STCP bytes sent	number of STCP reconnect attempts

Table E-4 TCP Metrics

total bundles acknowledged	total bundle bytes acknowledged	total bundles sent	total bundle bytes sent
total bundles failed to send	physical link status	total fragments acknowledged	total fragments sent
total bundles received	total bundle bytes received	number of TCP reconnect attempts	

Table E-5 UDP Metrics

bundles received	bundles bytes received	buffer overruns	bundles acknowledged
bundles bytes acknowledged	bundles sent	bundles bytes sent	bundles failed to send
physical link status	schedule link status	packets sent	packets bytes sent
packets dequeued for send	packets bytes dequeued for send	packets limited by rate	