

dtm  
Internet-Draft  
Intended status: Informational  
Expires: 24 October 2026

E. J. B. III  
S. Pellegrini  
A. White  
JHU/APL  
22 April 2026

Reliability Considerations for Delay-Tolerant Networks  
draft-birrane-dtn-rel-01

## Abstract

This document provides definitions and concepts related to network reliability as adapted to the Delay-Tolerant Networking (DTN) architecture where there is no presumption of simultaneous end-to-end paths between message sources and destinations.

## About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at  
<https://datatracker.ietf.org/doc/draft-birrane-dtn-rel/>.

Discussion of this document takes place on the Delay/Disruption Tolerant Networking Working Group mailing list (<mailto:dtn@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/dtn/>.  
Subscribe at <https://www.ietf.org/mailman/listinfo/dtn/>.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 24 October 2026.

## Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Motivation for DTN Reliability . . . . .	4
3.1. An Example Mission Network . . . . .	5
3.2. User-Desired Behaviors . . . . .	6
3.2.1. Keep Storage Available . . . . .	7
3.2.2. Delegate Network Monitoring . . . . .	8
3.2.3. Extend Platform Life . . . . .	8
3.2.4. Preserve Data Policies . . . . .	9
4. DTN Architectural View . . . . .	9
4.1. Architectural Layers . . . . .	9
4.2. Application Layer . . . . .	11
4.2.1. Relevant Application Services . . . . .	11
4.2.2. Application Faults . . . . .	12
4.3. The Bundling Layer . . . . .	13
4.3.1. Relevant Bundling Services . . . . .	13
4.3.2. Bundle Layer Faults . . . . .	15
4.4. The Adaptation Layer . . . . .	17
4.4.1. Relevant Adaptation Services . . . . .	18
4.4.2. Adaptation Faults . . . . .	18
4.5. The Underlay Layer . . . . .	19
4.5.1. Relevant Underlay Services . . . . .	20
4.5.2. Underlay Faults . . . . .	20
5. DTN Reliability . . . . .	21
5.1. Reliability Definitions . . . . .	21
5.1.1. Resiliency . . . . .	21
5.1.2. Availability . . . . .	22
5.1.3. Data Durability . . . . .	22
5.2. Reliability Signaling . . . . .	23
5.2.1. Expected Services and user mapping . . . . .	24
5.2.2. Traffic acceptance and rejection . . . . .	24
5.2.3. Delivery acknowledgements . . . . .	25

5.3. Reliability and Layering . . . . .	25
5.3.1. In-Band Exchanges . . . . .	25
5.3.2. Out-Of-Band Exchanges . . . . .	26
5.3.3. Delegation . . . . .	26
6. Custodial Behaviors . . . . .	26
7. Reliability Classes at the Bundle Layer . . . . .	27
7.1. Definitions . . . . .	27
7.1.1. Class 0 Best Effort . . . . .	28
7.1.2. Class 1 Store-Until-Forward . . . . .	28
7.1.3. Class 2 Guaranteed Custodian . . . . .	29
7.1.4. Class 3 Independent Custodians . . . . .	30
7.2. Tracing of Classes to Faults . . . . .	31
8. Security Considerations . . . . .	33
9. IANA Considerations . . . . .	33
10. References . . . . .	33
10.1. Normative References . . . . .	33
10.2. Informative References . . . . .	33
Acknowledgments . . . . .	34
Authors' Addresses . . . . .	34

## 1. Introduction

Delay-Tolerant Networking (DTN) [RFC4838] defines a networking architecture suitable for operating in environments where network communications are challenged by long signal propagation delays, frequent link disruptions, or both. A defining characteristic of this architecture is the inclusion of a bundling layer implemented by the Bundle Protocol ([RFC9171]), whose protocol data units are termed "bundles".

The bundle layer has been designed as an overlay that can be span multiple discontinuous underlay networks and provide additional reliability in the form of provided bundle services. This overlay layer can provide end-to-end reliability services, particularly in cases where challenged networking environments make existing underlay networks unreliable.

While there exist analyses and concepts related to DTN reliability, there is no unifying definition of DTN reliability or how such reliability is or is not enabled through combinations of bundle services (such as store-and-forward operations) and the DTN architecture (such as using reliable underlay networks).

This document defines reliability terms and concepts associated with bundle services and the DTN architecture.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This section defines terminology that either is unique to DTN reliability or is necessary for understanding the concepts defined in this document.

Delay-Tolerant Networking (DTN) :: (1) A networking architecture for tolerating expected and unexpected end-to-end delays and disruptions through the use of a networking layer and associated services built for that purpose. (2) Any network instantiation that conforms to the architecture, protocols, and mechanisms of the Delay-Tolerant Networking Architecture. In particular, any network implementing a bundle layer providing BP data units and BP-related services.

DTN Availability :: The ability to access an ingress into a DTN. The time-varying nature of DTN topology does not require any end-to-end communications for the network to be available to a given user. Any time a user can pass data to the first hop of the DTN, the DTN is considered available for that user.

DTN Data Durability :: The ability to prevent the loss or corruption of data in the network. This includes the preservation of data when otherwise impacted by failures (node loss, misconfiguration, and other impairments).

DTN Reliability :: The ability of the DTN to provide consistent, predictable performance. Performance for a DTN is a function of achieving data deadlines (delivering data within the data lifetime) and applying data policies instead of maintaining data throughput rates.

DTN Resiliency :: The ability of the DTN to recover from interruptions to its main functionality. Since a DTN is, by its nature, delayed and disrupted, resiliency is discussed in terms of DTN availability, durability, and reliability.

## 3. Motivation for DTN Reliability

### 3.1. An Example Mission Network

Resiliency in the DTN architecture can be reasoned about in the context of a sample network. A useful exemplar network is simple and focused without being so trivial as to avoid capturing network challenges. One such simple, non-trivial network is provided in Figure 1.

The example network consists of three Bundle Nodes (A, B, and C) with Bundle Node A representing a bundle source for some sending application (Sender), through an intermediate Bundle Node B, to the bundle destination at Bundle Node C which delivers the bundle to some receiving application (Receiver).

Bundle Nodes are connected by two different underlay networks, with Networks 1 and 2 connecting nodes A and B and Networks 3 and 4 connecting nodes B and C. Networks 1 and 3 are considered reliable (meaning that they never lose their data) whereas networks 2 and 4 are considered unreliable.

Altogether this network is suitable for discussing the end-to-end reliability as a function of characteristics and events present at bundle sources, bundle intermediate nodes, and bundle destinations across both reliable and unreliable networks.

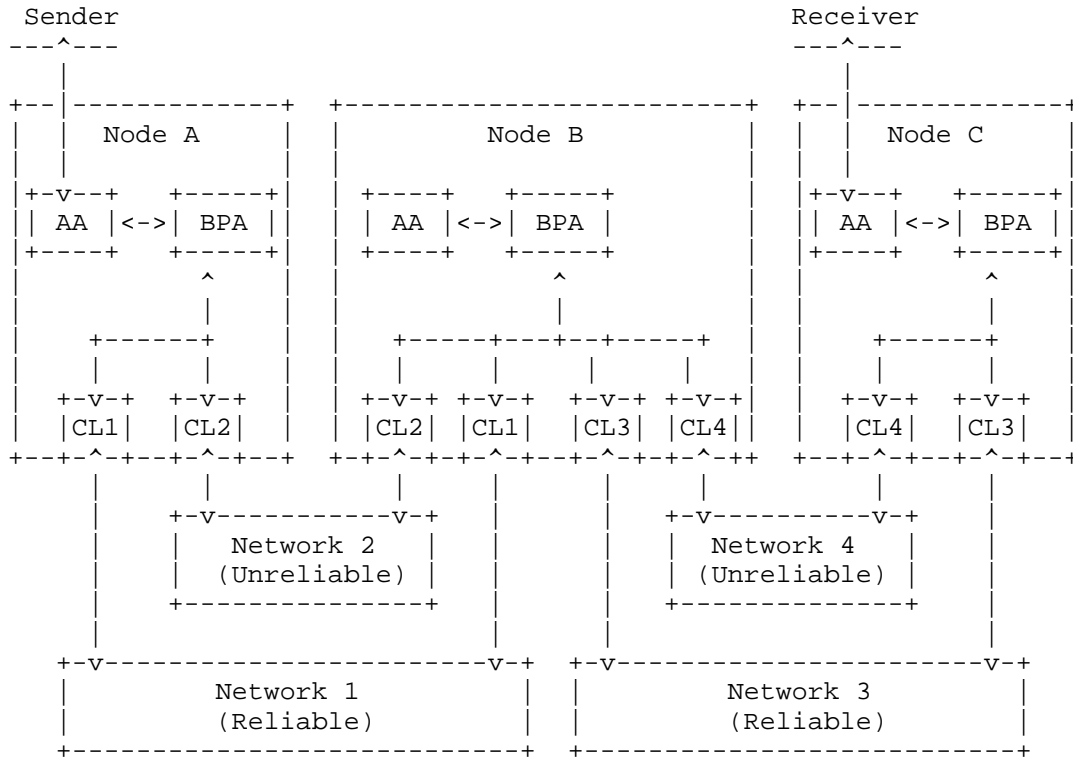


Figure 1: Exemplar DTN

The flow of information provided in Figure 1 between the network Sender and Receiver can be evaluated to identify various places where faults can occur. Each of the processing boxes in this illustration, from the AA associated with the sender to the AA associated with the Receiver can inject faults or errors in the system. This includes boxes associated with the "networks".

### 3.2. User-Desired Behaviors

The primary desire of any network user is the end-to-end exchange of user information across that network in ways compatible with user-expected policies and deadlines. Unchallenged networks that can provide timely end-to-end data exchange imply no other user-desired behaviors than sending and receiving data. In these unchallenged scenarios, user data exists in the network for such a relatively short period of time it precludes user reactions associated with the in-network processing of that data.

However, within the DTN architecture, user data may exist in the network for extended periods of time - and long enough that senders and received may need to take actions prior to end-to-end data delivery (or prior to receiving the acknowledgement of end-to-end data delivery).

This section describes four desirable behaviors, as listed below. Achieving these behaviors requires that users make a presumption about the reliability of the network that must be acted upon before confirming that reliability with acknowledged data delivery. To the extent that these behaviors capture user benefits, meeting them should be seen as the goals of any DTN reliability definition.

1. Keep Storage Available
2. Delegate Network Monitoring
3. Extend Platform Life
4. Preserve Data Policies

#### 3.2.1. Keep Storage Available

Storage is a limited resource for many devices operating in challenged environments. This may be due to the infrequency of network access, the difficulty of deploying devices in locations without access to regular power, and the cost of developing devices to function in extreme environments, among many other reasons.

A reliable DTN is one that allows users to manage their limited storage resources as a function of introducing data into the network, rather than confirming the receipt of data across the network. If a user application were able to rely on the fact that a network accepted data in accordance with some guaranteed reliability service, then the application might be comfortable removing that data from its own storage, even before the data cross the network.

In this way, the delays and disruptions incumbent in operating in challenged environments might not have the same negative impact on user storage as would be the case without reliable delivery.

### 3.2.2. Delegate Network Monitoring

Challenged networks may experience significant topological evolution over time. This evolution might be based on predicted impairments, such as signal propagation delays, occultation, and power duty cycling. However, this evolution might also be based on unplanned occurrences such as device failures, emergency operating conditions, and unexpected changes to the operating environment.

Users of these networks expect that any partitioning or other disconnectivity of the network remain hidden from their primary network interfaces and that data policies remain enforced while data is in either active transit or at rest. Unlike timely networks where node loss can be accounted for through over-provisioning resources and path diversity, a challenged environment might need to buffer data when message endpoints exist in separate network partitions.

The ability for the network to wait on appropriate connectivity without requiring any action from the user is, itself a user-desired behavior. Otherwise, in timely networks, network partitioning is otherwise handled as a network error requiring users to attempt network communications again at a later time.

### 3.2.3. Extend Platform Life

User platforms, especially those harvesting operating energy from their environment, can extend the operational life of their platform and otherwise increase the duty cycles of their platforms, by making more efficient use of their on-board resources. One of the more power-intensive applications on most sensing platforms is a telecommunications system. Keeping telecommunications systems powered and operating to account for delivery acknowledgements, handle retransmissions, and generally to support bi-directional, "chatty" communications protocols can be a significant energy drain on a user platform. For example, the telecommunications system on a deep-space spacecraft can be the single largest energy consumer on that spacecraft.

A reliable DTN can minimize data retransmissions by removing from the user platform the need to remain the retransmission source for any data generated by that node. Similarly, a DTN can increase the overall goodput of a network by reducing the number of individual messages that would need to be communicated based on the ability of DTNs to carry multiple types of secured information in a single network PDU.



By allowing more efficient use of telecommunications systems (and the processing, memory, and storage associated with those systems) a reliable DTN can help platforms make more efficient use of resources, extend overall platform life, and minimize the impact of duty cycles for the platform itself (or any components).

#### 3.2.4. Preserve Data Policies

Policy-based traffic engineering allows users to express complex behaviors associated with their data flows, to include which parts of the network may carry data and what network services are to be applied to that data while in transit. This policy enforcement is usually applied at the interface between a user device and some device representing the "edge" of the network. In cases where data might live within the network long enough for the network to undergo significant topological change, policy expression might need to be examined and applied by more than the network's edge node.

A reliable DTN is one that not only preserved data transmission during periods of delay and disruption; it will also ensure that data policies remain enforced even when the path through the network changes. This includes determining when user data is no longer considered valid in the network and expiring that data when it is considered stale.

### 4. DTN Architectural View

The bundle layer is not the only layer presupposed by the DTN architecture and, therefore, not the only layer that has the opportunity to contribute to the overall reliability of the network. This section provides a brief definition of the layering responsibilities of the DTN architecture, with a focus on the services provided by these layers and the types of faults that can occur at these layers.

#### 4.1. Architectural Layers

Because the DTN architecture requires a bundling layer, the minimal set of other layers in the architecture comprise it and any additional layers required to support that bundle layer. Generally, this leads to an architectural decomposition of at least four layers, as follows.

1. Application Layer: An application layer that uses bundle layer services for the exchange of application (user) data.
2. Bundling Layer: The bundling layer, implemented by BPAs, providing bundle services.

3. Adaptation Layer: An adaptation layer, implemented by Convergence Layer Adapters (CLAs) that adapt bundles for transport over an underlay network.
4. Underlay Layer: An underlay network layer providing transmission of bundles between and amongst BPAs.

A simplified layer view, and its mapping to the logical Bundle Node described in [RFC9171] (Section 3.1), is illustrated in Figure 2.

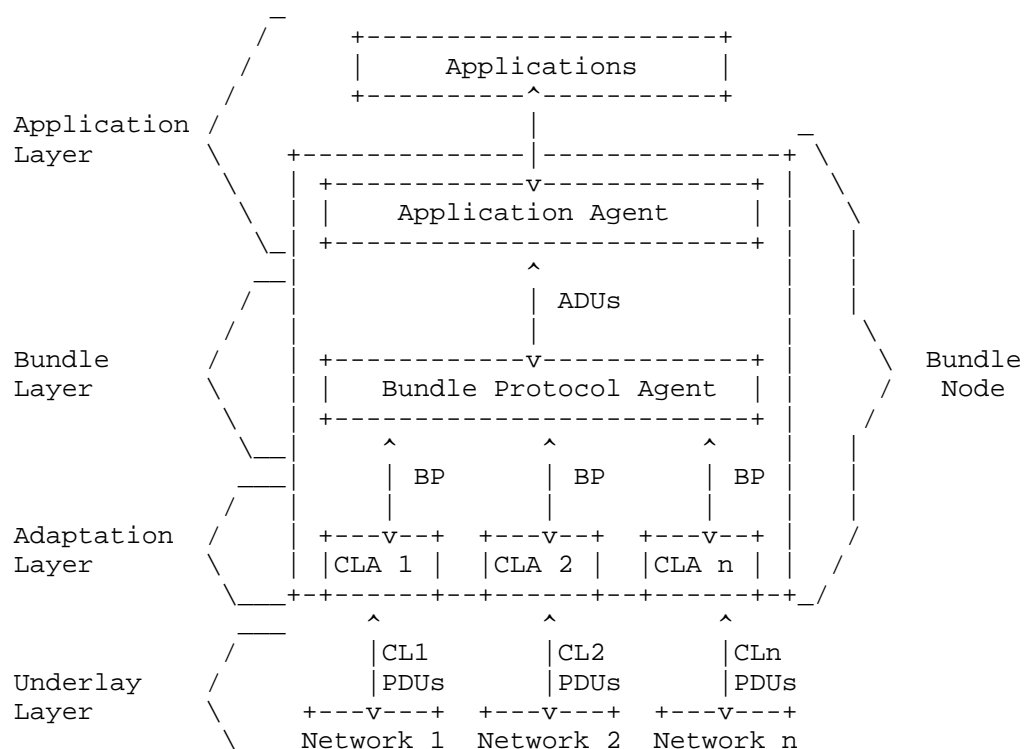


Figure 2: Simplified Bundle Node

The remainder of this section provides an overview of each of these layers, to include

1. A brief description of layer responsibilities
2. Services expected to be available
3. Faults that may be encountered

| TODO: Consider whether metrics by layer is a useful concept to  
| explore here.

## 4.2. Application Layer

The application layer consists of the set of entities service as messaging sources and destinations for traffic in the bundle layer. This can include user applications that directly produce and consume bundles as well as bundle proxy applications that aggregate non-bundle user traffic for the purpose of communicating those data as bundles across the bundling layer.

The design and purpose of applications in this layer are not relevant to the discussion of DTN reliability other than to note that applications must account for longer delays, out of order ADU delivery, and losing expired data. While the DTN architecture mitigates the risks of operating in a challenged networking environment, it does not (and cannot) ensure that (even temporally discontinuous) paths ever exist between a source and destination within some data lifetime.

### 4.2.1. Relevant Application Services

The only relevant service of the application layer, with respect to DTN reliability, is the exchange Application Data Units (ADUs) with the bundling layer.

#### 4.2.1.1. ADU Transmission

Every application in this layer, either directly or through a proxy, must create an ADU that can be communicated to the bundling layer for the purpose of having that ADU placed into one or more bundles and transmitted across the DTN. In doing so, the application provides information necessary for the creation of the bundle, to include information which either directly identifies or indirectly supports the BPA derivation of, the following information.

1. The destination of the ADU and the application expected to receive it.
2. Any special policy designations that must be honored in the creation or processing of bundles carrying all or part of the ADU.
3. The useful lifetime of the data in the network, after which the data can be deleted.

4. Any security information necessary to validate the identify of the application to the BPA.
5. Any acknowledgements needed to confirm to the BPA that ADUs were successfully received.

#### 4.2.1.2. ADU Receipt

Every application in this layer, either directly or through a proxy, must also be able to receive ADUs as assembled and communicated from an underlying BPA. In doing so, the application provides information necessary to help a BPA determine how and when ADUs may be delivered, to include information which either directly identifies or indirectly supports the BPA derivation of, the following information.

1. The identification of the application as a destination for ADUs.
2. Any security information necessary to validate the identify of the application to the BPA.
3. Any acknowledgements needed to confirm to the BPA that ADUs were successfully received.

#### 4.2.2. Application Faults

The logical entity within a bundle node that interacts directly with a BPA is known as the Application Agent (AA). The AA acts as intermediary through which Sender and Receiver applications communicate with the BPA. Faults related to this mechanism include errors with the exchange of Application Data Units (ADUs) and associated status reporting.

Beside the clear faults of loss of connection with either the user application or the BPA, the following types of faults can also be encountered:

1. Untrusted Application. Application connections are rejected due to authentication or configuration issues.
2. Malformed Request. The BPA rejects an ADU as a function of policy, implementation limitations, or misconfiguration of metadata associated with the ADU.
3. Offline Application. The AA is unable to deliver an ADU received from a BPA to an application because the application is not online.

### 4.3. The Bundling Layer

The bundle layer is a unique and defining element of the DTN architecture. This section describes those bundle services that might increase the overall reliability of the networks where they are implemented.

#### 4.3.1. Relevant Bundling Services

The set of services expected to exist in the bundling layer are provided in detail in [RFC9171]. However, certain capabilities of the BPv7 bundle (as a PDU) and the behaviors of a BPA in processing that PDU, enable reliability mechanisms. These services are store-and-forward behaviors, time-variant routing, and PDU extensibility and augmentation.

##### 4.3.1.1. In-Network Store-and-Forward

The store-and-forward behavior implemented in the bundle layer allows for the option to hold bundles at Bundle Processing Agents (BPAs) in the network. This behavior allows bundles to be communicated across various underlay networks between BPAs and stored at BPAs pending forwarding opportunities or transmission acknowledgements.

Storage, in this scenario, includes the ability to hold bundles through a reboot or other fault management that might occur at the BPA or any device hosting a BPA. The amount of storage provided by any given BPA is expected to vary as is the access to storage for any given bundle. Generally, how bundles are stored by a BPA is expected to be governed by network management.

Store-and-forward behaviors increase the reliability of a network in cases where data would otherwise be dropped. The most likely causes of dropped data are instances where a BPA uses an underlay network to carry a bundle to its next BPA and the underlay network drops the data. It is expected that underlay networks will drop data if the underlay network is either disconnected or unreliable. In these cases, were the BPA to store the bundle, the bundle could later be retransmitted over the same underlay or over a different underlay. In extremely challenged networks that never have simultaneous, end-to-end paths the only option for data delivery is store-and-forward.

#### 4.3.1.2. Time-Variant Routing

There are multiple reasons why a preferred path for a bundle might change over time, including preserving platform resources, increasing operating efficiency, and reacting to dynamic reachability [RFC9675]. In cases where these changes are unplanned, route computation at the bundle layer would neither have knowledge of the instantaneous topology of the network nor understand how the topology might change over time. To the extent that BPAs detect and react to changes in topology over time, the bundle layer will need to reason about (and recalculate) routes over the lifetime of the bundle.

Time-variant routing behaviors increase the reliability of a network in cases where different paths exhibit different reliability. For example, a time-variant route might be calculated that avoids the use of a currently-available-but-unreliable underlay network connection to wait for a later-available-and-reliable underlay network connection. Alternatively, routes could be selected that mitigate forecasted congestion at BPAs or otherwise select routes that preserve some policy-mandated behavior (such as BPA storage availability).

#### 4.3.1.3. PDU Extensibility and Augmentation

BPv7 bundles carry varied information in various bundle block types. Networks may define private-use blocks for annotating bundles resident in their administrative domain and, generally, new standardized block types may be defined through normative standards body actions. In addition to defining new block types, BPAs may add, remove, or update certain extension blocks in a bundle while the bundle is being processed by the BPA. This block processing may happen at BPAs acting as the bundle source, the bundle destination, or any bundle waypoint in the network.

Extending the information in a bundle through the definition of extension blocks increases the reliability of the bundle in the network because it allows for ways to signal (in-band) reliability information for both the bundle itself and the overall bundle layer. Augmenting existing bundles in the network allows BPAs to react to changes in bundle handling in ways that increase the likelihood for eventual bundle delivery - such as changes to policy, network status, or other processing information that might not have been known at the time of bundle creation.

The ability of the bundle layer to secure individual blocks individually, as outlined in [RFC9172], also provides additional reliability in the sense that malicious or misconfigured data could be detected and removed from the bundle layer to avoid congestion or other resource contention for properly credentialed network traffic.

#### 4.3.2. Bundle Layer Faults

The BPA is responsible for creating bundles from a given ADU, delivering ADUs from a received bundle, and otherwise processing bundles that are transiting through the BPA itself. The BPA is the source of any bundle processing related faults as this is the sole logical entity tasked with bundle processing.

Faults in this area can be organized into creation, processing, and delivery faults.

##### 4.3.2.1. Bundle Creation

Faults related to bundle creation impact the AA requesting bundle creation. This includes both application-based senders as well as the creation of administrative bundles from the administrative element with the AA. BPA faults preventing the creation of a bundle include the following:

1. Malformed bundle. Bundle would violate some configured limits or policies and cannot be created.
2. Incomplete Information. The AA provided insufficient information to create the bundle, such as malformed destinations.
3. Untrusted AA. The AA is not validated as able to request bundle creation.
4. Unsupported Services. The BPA is unable to provide the services necessary for processing the bundle.

##### 4.3.2.2. Bundle Processing

Processing, in this context, refers to all actions taken by a BPA at a source BPA, some intermediate BPA, or destination BPA.

This processing happens after creation of the bundle at a source, after reception of a bundle from the adaptation layer, before sending the bundle to a next hop using the adaptation layer, and before delivering the ADU of a bundle to the AA at the destination node.

Categories of faults related to bundle processing include the following.

1. Resource Exhaustion. Bundles might be kept at a node longer than expected due to lack of available, appropriate next hops.
2. Bundle Mangling. Due to policy or other mis-configuration, bundles may have extension blocks or other modifications that make them difficult or impossible to process downstream. This includes errors related to incorrect fragmentation or application of security.
3. Platform Error Loss. Bundles might be removed from the node unexpectedly due to implementation error or technical fault on the BPA itself - to include the loss of the BPA or the loss of the processing node.
4. Bundle Duplication. Implementation errors in a BPA might result in the creation of bundles that violate rules for uniqueness in the primary block. This can happen, for example, on systems where clocks are being manipulated or where sequence counters/nonces are reset.
5. Wrong CLA. A BPA might transmit/forward a bundle using an inappropriate CLA, as a function of mis-configuration or other error on the BPA.
6. Unsupported Service Loss. A special case of bundle loss where elements of a bundle cannot be processed by a BPA and the BPA drops the bundle due to bundle processing flags.
7. Extension Processing Loss. The bundle contains extension blocks that are in violation of BPA policy, or otherwise cause extension blocks or the bundle itself to be removed from the network.

#### 4.3.2.3. Bundle Delivery

Delivery, in this context, applies to any bundle processing that occurs at the destination BPA of a bundle. Destination processing represents a slightly different circumstance than other bundle processing because it involves resources and status exchange local to the node that is also running the AA.

In addition to the regular faults that can occur as part of bundle processing, the following types of faults are unique to the delivery process.



1. Reassembly Failure. A deliverable ADU cannot be extracted from a series of bundles each holding ADU fragments. This is separate from issues reassembling a bundle up to the bundling layer from the adaptation layer.
2. ADU Handoff Loss. The BPA is unable to send an ADU to an AA. This can happen if the AA cannot resolve the service/application that is the recipient of the ADU, or if the AA generally rejects the ADU as a matter of policy or configuration.

#### 4.4. The Adaptation Layer

The BPv7 specification describes "Services Required of the Convergence Layer" [RFC9171] (Section 7) in which the "Convergence Layer" is defined only as "underlying protocols" accessed via a "Convergence Layer Adapter" (CLA). This definition can blur understanding between a convergence adapter, a convergence layer stack, and the protocols implemented within some supporting underlay network.

To clarify the differences in responsibilities between the bundling layer and an underlay networking layer, we propose the adaptation layer as a layer devoted to the implementation of convergence layer adapters, protocol stacks, and associated services. A simplified view of these elements is illustrated in Figure 3 and discussed further below.

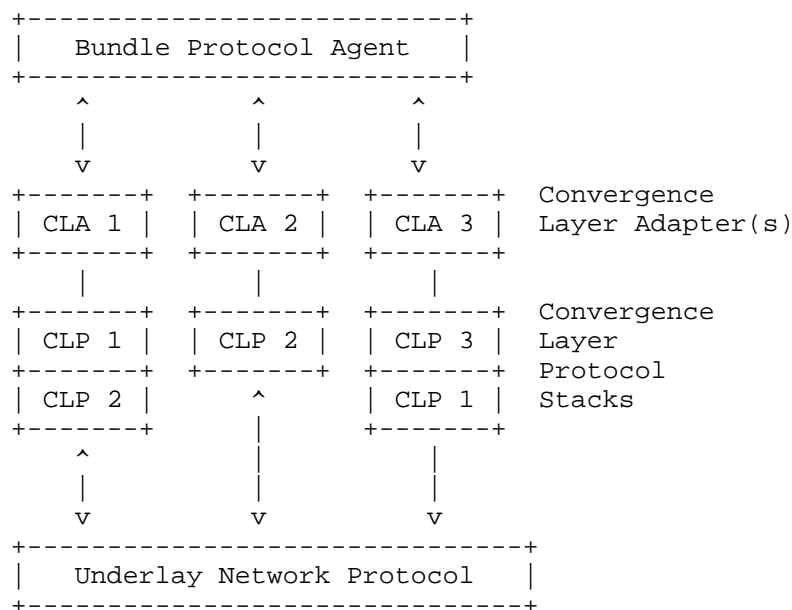


Figure 3: Adapters and Protocol Stacks

A Convergence Layer Protocol (CLP) is one that exists between a BPA and an underlay network and has the responsibility of communicating between the two. Importantly, a CLP is not considered part of the underlay networking stack in the same way that a CLP is not considered part of the bundle protocol or the bundling layer.

CLPs (and the adaptation layer) exist to provide services for the bundling layer in cases where a particular underlay network stack might not provide those services. To that end, it might be the case that multiple CLPs are used together to form a Convergence Layer Protocol Stack (CLPS) and that distinct CLPS's might exist depending on which underlay network is being used and which types of data handling services are requested by the bundling layer.

A Convergence Layer Adapter (CLA) is an application that prepares bundles received from a BPA for transmission over some CLPS. The BPA only ever reasons about its interface to its CLA and the properties of the CLPS behind that CLA. Therefore, from the perspective of the BPA, the depth of the CLPS is less relevant than the overall capabilities and characteristics of the CLPS. For example, a BPA may select a CLPS based on whether the protocols in those stacks implement their own reliability, in-order delivery, and what metrics and management information might be available related to the performance of the protocols in that stack. Based on the "top" protocol in the CLPS, an appropriate CLA can be selected to then communicate bundles to and from that CLPS.

#### 4.4.1. Relevant Adaptation Services

There are three convergence layer services outlined by the BPv7 specification [RFC9171] (Section 7.2): 1. Sending a bundle 1. Notifying the BPA upon the disposition of data-sending procedures 1. Delivering a bundle

#### 4.4.2. Adaptation Faults

Faults that can occur in this layer include the following:

1. Malformed Bundle CLA Rejection. The CLA might refuse to accept a bundle that is malformed.

2. Policy-Induced CLA Rejection. A CLA might refuse to accept a bundle as a function of its own policy, such as if the bundle exceeds maximum sizes, uses EID schemes unknown to the CLA, or includes sources or destinations unknown or unsupported by the CLA. This might also include cases where the BPA and CLA are unable to establish trust/authentication mechanisms between them.
3. CL Processing Loss. The CL stack might fail to encapsulate a bundle as it processes through a CL stack. This may be due to mis-configurations or policy violations of the individual CLPs in the CL stack, malformations due to implementation errors in any implementation of the CLAs in that stack, or general errors or loss of the platform(s) implementing these CLAs and CLPs. This loss may happen both when sending or receiving bundles through the CL.
4. Network Rejection. The CL may fail to transmit the adapted bundle if the underlying network rejects the adapted bundle for any reason.

#### 4.5. The Underlay Layer

Any network capable of transmitting BPv7 PDUs (or portions of BPv7 PDUs carried in CLP PDUs) can act as an underlay network for the purpose of communicating bundles between two or more BPAs. There are no restrictions on the types of underlay networks that can be used, and there is no restriction that such underlay networks only implement specific features or functions. Part of the power of DTN as an overlay across heterogeneous networks lies in its ability to work across vastly dissimilar underlay networks.

For example, all of the following may be considered underlay networks within the DTN architecture.

1. The terrestrial Internet.
2. A single hop between a transmitter and a receiver (such as between a spacecraft and a ground station).
3. A CCSDS constellation network using only framing protocols.

#### 4.5.1. Relevant Underlay Services

Individual underlay networks may support a large number of network services and associated service level agreements. From the perspective of DTN reliability, certain characteristics of the underlay network are more important when selecting which underlay network should be used for transmission and which services from the bundling and adaptation layers may need to be relied upon to cover gaps in underlay network services.

1. The overall security of the underlay.
2. The reliability (to include resiliency and availability) of the network
3. Features such as delivery acknowledgements and in-order delivery

#### 4.5.2. Underlay Faults

Several faults can occur at this layer while trying to communicate between two BPAs. From the point of view of the network, CLPs containing portions of a bundle are simply network traffic. Therefore, the faults associated with the underlay network are faults associated with any of its network traffic and not specific to the fact that the traffic includes bundle information. These faults include the following.

1. Lost traffic. The network loses traffic for a variety of reasons, to include use of non-reliable architectures, links, or protocols otherwise hidden from the BP layer. This can also include loss of network components necessary for network data exchange.
2. Unexpected traffic delay. Network traffic is delivered, but over a longer time period due to delays caused by retransmissions, security or other round-trip exchanges, or as a function of priority/service priority for the network.
3. Traffic Corruption. Traffic is received but corrupted due to processing while in the network. This includes both intentional and unintentional sources of data corruption. Corrupted traffic might lead to traffic loss when the corruption is detectable and unrecoverable.
4. Improper delivery. At a receiving network node, traffic might be incorrectly delivered to the CL responsible for extracting a bundle which would cause the bundle to never be sent to a BPA.

## 5. DTN Reliability

The most important characteristic of a network is that it can be relied upon to communicate user data within performance, security, timelines, and other user requirements. The overarching networking characteristic capturing this behavior is the reliability of the network. Any operational network needs to be reliable with respect to network-specific and user-specific requirements.

| NOTE: Even best-effort networks can be seen as reliable with  
| respect to user requirements for data delivery. For example,  
| reliability can be asserted by specific data handling at a  
| source node, using physical layers that reduce per-hop data  
| loss, sending duplicates of user data, and other strategies.

Reliability is not an absolute characteristic of a network for two reasons. First, it is defined and applied in the context of network users and their associated data requirements. Therefore, no one set of network behaviors might be "reliable" for all users. Second, as a practical matter, pathological topologies, error conditions, or other issues can be encountered that impede networked data exchange. Therefore, reliability is often considered in the context of some set of supported operating conditions.

One way to describe this nuanced view of reliability is to consider other network characteristics that contribute to the overall ability to exchange user data. These other characteristics include network resiliency, availability, and durability.

The concept of reliability in a DTN needs to be defined in a way that is independent of end-to-end latency.

### 5.1. Reliability Definitions

#### 5.1.1. Resiliency

Network resiliency is the extent to which a network continues to function in a variety of operating conditions. Networks with little resiliency may operate effectively well within nominal operating conditions, but rapidly fail as the environment changes, such as due to component failures, attacks, user congestion, and other impairments.

Inherent in any networking design is an analysis of the potential operating conditions through which a network is expected to function. Not every network is expected to operate in every operating condition, simply as a matter of practicality. Mechanisms for achieving resiliency are numerous, and include designing redundancy

and diversity into the networking architecture, increasing the number and performance of networking devices, and deploying various data, management, and control protocols and associated algorithms.

Defining resiliency as tolerance for operating conditions provides a useful way to assess proposed resiliency mechanisms.

#### 5.1.2. Availability

Network availability is the extent to which the network is able to accept user traffic. Networks with little uptime may only be able to accept user traffic at specific times or under specific conditions. Networks with little availability may be both resilient and reliable during those periods of availability.

Availability can be defined both as an absolute measure or relative to known usage patterns. Absolute availability (often measured through network uptime) is used when user traffic is omnipresent or otherwise when user traffic patterns are unknown but possible at any time. Relative availability is used when it is understood that user traffic will not exist at certain periods, or when the network itself undergoes planned downtime.

| NOTE: This definition of availability intentionally omits the  
| transmission or exchange of user traffic. From a user  
| perspective, the network is available when it accepts user  
| traffic, even if the network itself does not transmit that  
| traffic.

#### 5.1.3. Data Durability

Network data durability is the extent to which information in the network is protected from recoverable errors, such as data corruption. Networks with little data durability may be resilient and available, but if data is corruptible either during transmission or when being processed in the network, it will not be durable.

| NOTE: Data durability includes decisions such as whether to  
| delete in-network user traffic such as occurs during times of  
| congestion, link loss, and other similar events. If user data  
| is deleted in the network as a result of transient network  
| conditions then the data does not have durability.

Durability, as defined here, encompasses concepts of data integrity which may or may not be tied to data security. Where data integrity involves the ability to detect and possibly preserve data from corruption in the network and data security including mechanisms for implementing integrity using cryptographic means.

## 5.2. Reliability Signaling

Network reliability (and associated characteristics) are always considered in the context of network user outcomes. For networks purpose-built for a particular user community there is a direct correlation between network characteristics and user needs. For networks with diverse users, network characteristics are associated with user traffic through network configurations and policies. Therefore, discussions about what mechanisms are needed to implement what levels of reliability (and thus resiliency, availability, and durability) need to occur in the context of what user outcomes need to be preserved.

A common approach to providing these indicators is to build networks whose end-to-end latency are smaller than the timing requirements for providing indicators back to the user. In such cases, the success (or failure) of a given data exchange can be reported with certainty as it has already happened.

This approach cannot be used in networks conforming to the DTN architecture because the expected operating conditions of the network cannot guarantee end-to-end latency smaller than user timing requirements. Instead, DTNs might need to provide user indicators based on processing at the source node itself, and without an understanding of end-to-end performance across the DTN.

This section defines user expectations of indicators and subsequent sections describe how these indicators may be different in the context of DTN deployments.

Expected user outcomes take the form of indicators provided by the user to the network (and confirmed by the network to the user). Some indicators that impact decisions relating to network reliability include the following.

1. Expected services and user mapping
2. Traffic acceptance and rejection
3. Delivery acknowledgements

#### 5.2.1. Expected Services and user mapping

User expectations for services are often captured in various agreements between user communities and network operators. These often take the form of Service Level Agreements (SLAs) where services are pre-negotiated as part of supporting user traffic in a network. In situations where a network is custom-built for a user community, these services may simply be as designed into the network architecture. In situations where networks service multiple users, this may be negotiated dynamically.

Networks need to provide a way to map users to their expected service agreements, and may need to indicate to users that their traffic has been accepted at the expected service level.

#### 5.2.2. Traffic acceptance and rejection

User traffic, when provided to the network, needs to be either accepted or rejected, and users may expect to be informed of both.

User data is usually accepted by the network if it can be associated with a known user, if the network is known to be in an operational state, and if there exists knowledge that the user data can be communicated over the network within various user requirements. Signaling acceptance is often an implementation matter (e.g. through the programming interfaces by which user traffic is presented to the network).

User data is rejected by the network whenever it cannot be accepted. These often occurs for reasons such as failing to authenticate the user traffic or match it to known user service requirements, situations where the network is not operating, or otherwise when there is no known path to the user traffic destination. Similar to acceptance, how rejection is communicated to the user is an implementation matter, but often includes some reason for the rejection.

These indicators are important, as rejection of user traffic from a reliable network requires some action by the user. Therefore, signaling acceptance and rejection in a timely manner is important for many application operational concepts.



### 5.2.3. Delivery acknowledgements

In certain cases positive acknowledgements (ACKs) or negative acknowledgements (NACKs) are expected by users from the network for certain types of traffic. This is separate from application space protocols that exchange request-response messages above the scope of the networking layer.

These acknowledgements are usually reserved for critical information flows.

### 5.3. Reliability and Layering

The reliability of the DTN architecture does not come from a single layer and, therefore, it SHOULD NOT be the case that a single layer account for every possible reliability consideration. The total set of services available across the application, bundling, adaptation, and underlay network layers form a vast set of capabilities from which network engineers, architecture, and operators can select to build networks appropriate for their deployment environments.

This section summarizes how each of these types of reliability mechanism can increase the reliability of the DTN architecture when viewed from the perspective of a given layer.

#### 5.3.1. In-Band Exchanges

In-band reliability information consists of diagnostic information exchanges with other peers at a particular layer in a network. In this viewpoint, in-band refers to the fact that diagnostic information is exchanged within the particular network layer, not that it must all be exchanged by a single protocol operating at that layer.

The content of diagnostic information varies as a function of mechanism, but common types of such information include the following.

1. Acknowledgements (ACKs) indicating delivery of one or more messages.
2. Negative acknowledgements (NAKs) indicating missing messages.
3. Statistics relating to number of bytes of messages delivered.
4. General health of sessions exchanging messages that are encrypted or otherwise opaque.

### 5.3.2. Out-Of-Band Exchanges

Separate from reliability information that comes from other peers within a given layer, information can be communicated across layer "boundaries" in the form of out-of-band information. The types of out-of-band information that can inform reliability mechanisms include the following.

1. Configuration and policy updates from operations and orchestration entities that consider cross-layer and other inputs.
2. Information provided from the management and control planes of protocols operating at other layers. This includes metrics, acknowledgements, and other signaling from different layers in the architecture.
3. Backpressure and other data plane information from other layers as a function of cross-layer data exchange

### 5.3.3. Delegation

In cases where there is implicit trust associated with a given integrated platform, it is possible that no positive reliability signaling is needed beyond (optional) indications that data was received by another layer in the network architecture. For example, it may be the case that if data is accepted by a lower layer (without rejection or negative acknowledgement) that the data is presumed to be handled in accordance with the capabilities of the accepting layer, without additional in-band or out-of-band signaling.

## 6. Custodial Behaviors

Custodial behaviors are the set of actions which permit the transfer of responsibility for retransmitting a message at intermediate nodes or "custodians" in a network. A custodian is the node or hop that acquires the responsibility to ensure data delivery. Simply put, a node source transmits data to a next hop that accepts custody of the data, thereby becoming the new custodian for that data. This means that the original data source may go offline, and data will be retransmitted from the new custodian. By allowing for these behaviors, the reliability of data exchange across a network increases.

There are two primary enabling custodial behaviors: 1. Storing a bundle. 2. Releasing a bundle when release conditions are met.

A custodian must be capable of storing a bundle until the data is expired, the data is delivered, or the custodian is relieved of its custodial duties. A custodian must then also recognize the conditions to be relieved and release or delete the bundle accordingly. For this release to happen, the current custodian must receive a signal of acknowledgement from the new custodian. Depending on the network, the number of signals required for release may vary. For example, if a network is set up to provide multiple, parallel custodians at one time, a custodian may not release bundles from storage until a certain number of new custodians have acknowledged responsibility. This minimum required number of acknowledged custodians is abbreviated as "MAD" below.

Due to the variable levels of reliability that may exist across a given network, networks can be categorized by the different reliability classes defined in the next section.

## 7. Reliability Classes at the Bundle Layer

Because DTNs will vary in their individual complexity and requirements, not all DTNs need the same levels of reliability, and consequently, custodial behaviors. Some data service needs will require end-to-end reliability, more complexity, more resource utilization, or more mission support than others. Missions will need to determine which mission-impacting behaviors need to be realized for which data flows in a mission and decide which custodial services best suit their networks. Allocating these options into different "reliability classes" provides a natural way to express these needs within the service agreement construct already used in terrestrial networks to map user expectations to network mechanisms. These classes are defined below based on three characteristics which stem from the inherent reliability mechanisms in DTN-based protocols (specifically BPv7) and the custodial behaviors listed above.

The following characteristics are used to summarize a reliability class: 1. Minimum required number of acknowledged custodians necessary for data release (MAD). 2. Ability to store a bundle until there is an egress link (SAB). This value is TRUE or FALSE. 3. Ability for the destination node to hold data until the ADU is delivered (HAD). This value is TRUE or FALSE.

### 7.1. Definitions

This section defines four classes of reliability services selectable by missions in the context of a mission-data-specific service agreement, as we would expect them to be applied at the bundle layer. These classes are enumerated (0-3).

#### 7.1.1. Class 0 Best Effort

The class 0 custodial service expressly prohibits the use of store-and-forward or associated custodial behaviors for any traffic associated with this service. Defining a service class for the sole purpose of prohibiting reliability is useful because it allows network nodes to short-circuit a variety of evaluations associated with data storage, buffer management, security processing, time-variant route computations, and other policy-based considerations.

IPN nodes may treat this type of data in the same way as terrestrial routers treat data: data in this class can be delivered, forwarded, or dropped. Even when using BPv7 bundles.

Missions benefit from this class of service in a variety of ways, such as:

1. Data associated with this class of service may be provided at a lower cost to missions.
2. Overall mission throughput may be faster through a network because the network applies less inspection on the data, with less need to process storage.
3. Mission assets will not need to wait for the network to acknowledge that data have been accepted.

Key summary of Class 0:

- \* MAD = 0 (no custodial behaviors)
- \* SAB = FALSE
- \* HAD = FALSE

#### 7.1.2. Class 1 Store-Until-Forward

The class 1 reliability service provides a best-effort, store-and-forward delivery of data similar to that described in BPv7. In this class of service, IPN nodes will accept data and buffer them within some constraints as a function of in-band and out-of-band policy waiting for an appropriate next transmission of the data.

Within this class of services, the MAD variable is 0. In other words, there is no acknowledgement required by a node for the release of a bundle from its storage because there are no custodians. Instead, the term retention in this reliability class implies the persistence of user data at a node if necessary for the transmission of data to its next hop. This means the SAB (storing a bundle) parameter is set to TRUE.

For example, when data is received by a node, a determination is made as to whether the data can be immediately transmitted or not. If data can be immediately transmitted, then there is no need to store it, and it can be sent directly to a transmitter. Alternatively, if it can't be immediately transmitted, the node will need to determine if there is storage available. If so, it can be stored pending a future transmission opportunity. If there is no storage, or if a bundle expires or is otherwise expunged from storage, then the bundle will be lost.

Key Summary of Class 1:

- \* MAD = 0 (no custodial behaviors)
- \* SAB = TRUE
- \* HAD = TRUE or FALSE pending customer requirements

#### 7.1.3. Class 2 Guaranteed Custodian

The class 2 reliability service provides the guarantee that there exists, within the network itself, at least one node configured to serve as a custodian of data. The implication for this class of service is that, unlike the best-effort retention of a Class 1 service, the network will provide storage. The amount and nature of storage provided by the network is expected to be customized to individual mission data flows.

The way in which a network provides this guarantee is expected to be opaque to the network user, but generally there is some serialized order in which one or more nodes will accept custody of data as it works its way through the network. This might mean that there exists only one node in the network that is configured to serve as a custodian, or it might mean that networks may implement progressively updating custodians along a user data path, where each hop in the network is configured to serve as a custodian. Alternatively, it is possible that network service orchestration tools choose common custodians as a function of shared destinations, where the network defines certain nodes as custodians as a function of topological significance, available storage, or other policy inputs. This

implies the data paths can go through any part of the network between custodians as necessary. In all of these cases, class 2 describes a scenario where a custodian must only receive one acknowledgement from a new custodian to release stored data and be relieved of its custodial duties (MAD = 1). It does not assume there are parallel paths and parallel custodians for increased reliability.

Additionally, in each of these example cases, the user source and destination nodes do not require knowledge of how or where custodians are placed on a network.

Key Summary of Class 2:

- \* MAD = 1
- \* SAB = TRUE
- \* HAD = TRUE or FALSE pending customer requirements

#### 7.1.4. Class 3 Independent Custodians

The class 3 reliability service provides a guarantee that the network will assign multiple, parallel custodians within the network so that if there is an issue with a single custodian, data may still be able to be delivered within its lifetime and other policy constraints.

There are a variety of reasons why a particular path or single custodian might not deliver data for a particular mission, including the following.

- \*1. Custodian Loss:\* Space remains a harsh environment, and space-based custodians may be located in spacecraft or other assets that experience catastrophic failure. If a single custodian in the network fails, then all untransmitted data on that custodian is lost.
- \*2. Network Partition:\* The time-variant nature of interplanetary topologies can be unpredictable, especially when nodes in the network experience recoverable errors or other unplanned outages. When this occurs, a custodian may be isolated or disconnected from the rest of the network for a long enough period that the stored data expires and is removed from the node.
- \*3. Configuration Changes:\* Changes to configurations at a custodian (to include misconfigurations) can introduce processing errors, especially as they relate to the security processing of the data. Additionally, policy changes may impose policy-based routing and retention decisions on a node that can result in loss of retained data.

In these cases (and many others), very critical data can be lost if there is only a single custodian for data at a time. To increase the likelihood of successful data transfer, parallel custodians can be required such that a user source node would need to receive confirmation that two or more custodians have actively accepted user data prior to removing those data from the user source (MAD = 2+).

Key Summary of Class 3:

- \* MAD = 2+
- \* SAB = TRUE
- \* HAD = TRUE or FALSE pending customer requirements

## 7.2. Tracing of Classes to Faults

Understanding both the potential network faults and reliability classes allows users to also understand which types of faults may be avoidable or mitigated through particular reliability mechanisms within each class. This is shown in Figure 4 below, which traces the aforementioned network faults with the reliability class(es) with the potential to mitigate those faults.

Type of Fault	Reliability Class	Additional Comments
AA Faults: Untrusted Application	No Class	
AA Faults: Malformed Request	No Class	
AA Faults: Offline Application	Class 2-3	
BPA Faults: Malformed Bundle	No Class	
BPA Faults: Incomplete Information	No Class	
BPA Faults: Untrusted AA	No Class	
BPA Faults: Unsupported Services	No Class	

BPA Faults: Resource Exhaustion	Maybe Class 1 Class 2-3	Class 1 if node is not full
BPA Faults: Bundle Mangling	Class 2-3	If error not at custodian
BPA Faults: Platform Error Loss	Class 2-3	If error not at custodian
BPA Faults: Bundle Duplication	No Class	
BPA Faults: Wrong CLA	Class 2-3	
BPA Faults: Unsupported Service Loss	Class 2-3	Only if retransmission takes different path
BPA Faults: Extension Processing Loss	Class 2-3	Only if retransmission takes different path
BPA Faults: Reassembly Failure	No Class	
BPA Faults: ADU Handoff Loss	Class 2-3	Only if error or handoff is time dependent.
CLA Faults: Malformed Bundle, CLA Rejection	No Class	
CLA Faults: Policy-Induced CLA Rejection	Class 2-3	If retransmission takes different path
CLA Faults: CL Processing Loss	Class 2-3	Only for certain errors like loss of platform
CLA Faults: Network Rejection	Class 2-3	If retransmission or parallel path avoid rejection
Underlay Faults: Lost Traffic	Class 2-3	



Underlay Faults: Unexpected Traffic Delay	Class 3	Parallel paths may result in faster delivery.
Underlay Faults: Traffic Corruption	Class 2-3	Retransmission or parallel paths may avoid corruption
Underlay Faults: Improper Delivery	Class 2-3	

Figure 4: Layer Faults to Reliability Classes

## 8. Security Considerations

TODO Security

## 9. IANA Considerations

This document has no IANA actions.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

### 10.2. Informative References

- [IPN\_CT] Birrane, E. and S. Pellegrini, "Designing Custody Transfer for Interplanetary Networks", 76th International Astronautical Congress , 2025.
- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <<https://www.rfc-editor.org/rfc/rfc4838>>.

- [RFC9171] Burleigh, S., Fall, K., and E. Birrane, III, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <<https://www.rfc-editor.org/rfc/rfc9171>>.
- [RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPsec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <<https://www.rfc-editor.org/rfc/rfc9172>>.
- [RFC9675] Birrane, III, E., Heiner, S., and E. Annis, "Delay-Tolerant Networking Management Architecture (DTNMA)", RFC 9675, DOI 10.17487/RFC9675, November 2024, <<https://www.rfc-editor.org/rfc/rfc9675>>.

#### Acknowledgments

TODO acknowledge.

#### Authors' Addresses

Edward J. Birrane, III  
JHU/APL  
Email: [Edward.Birrane@jhuapl.edu](mailto:Edward.Birrane@jhuapl.edu)

Sabrina Pellegrini  
JHU/APL  
Email: [Sabrina.Pellegrini@jhuapl.edu](mailto:Sabrina.Pellegrini@jhuapl.edu)

Alex White  
JHU/APL  
Email: [Alex.White@jhuapl.edu](mailto:Alex.White@jhuapl.edu)