Delay-Tolerant Networking
Internet-Draft
Intended status: Standards Track
Expires: 4 January 2026

B. Sipos
JHU/APL
J. Deaton
SAIC

3 July 2025

Bundle Protocol (BP) Secure Advertisement and Neighborhood Discovery (SAND)
draft-ietf-dtn-bp-sand-01

Abstract

This document defines the Secure Advertisement and Neighborhood Discovery (SAND) protocol for Bundle Protocol version 7 (BPv7) within a delay-tolerant network (DTN). This protocol defines a general purpose advertisement mechanism with an initial set of data types able to be advertised by participating nodes in a BPv7 network.

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 4 January 2026.

Table of Contents

1. Introduction

Deployments of Bundle Protocol version 7 (BPv7) nodes have required a significant amount of configuration for both the node being enrolled in the BPv7 network as well as the pre-existing (one-hop neighbor) nodes expected to communicate with the new node. The configuration consists of both BP-layer parameters, such as identity and security capabilities, as well as underlying convergence layer (CL) and associated transport parameters.

When nodes are in the same administrative domain, these parameters might be easy to find and the burden is solely about configuring the nodes. But when nodes need to configure across administrative domains simply finding the parameters could be an operational challenge, and if the parameters change keeping them synchronized is yet more complexity. Administrative domains might be crossed at the boundary between organizations (_e.g._, when bridging two BP wide-area networks) but they can also be crossed within a single host or platform where there are nodes from different vendors present which need to interoperate.

Additional considerations for discovery within a BP network are related to the expectation of challenged nature of a delay-tolerant network (DTN) more generally. This means long one-way light-time (OWLT) delays between neighbors, expected time-varying discontinuities between neighbors, and a variety of CL transport types, each with associated parameters, capabilities, and limitations. More detailed descriptions of the challenges of DTNs can be found in "Delay-Tolerant Network Architecture" [RFC4838].

Earlier research into discovery within a BP network led to development of the draft experimental IP Neighbor Discovery (IPND) protocol of [I-D.irtf-dtnrg-ipnd], but that protocol is intimately tied to its use of UDP datagram "beacons" and necessary use of an IP underlay network. It also would require allocation of well-known UDP port number and IP multicast addresses or pre-configuration of those parameters across network nodes, but no such allocations were ever made.

To mitigate the need for manual parameter discovery and configuration, an online neighborhood discovery protocol can be used, and such a protocol is defined in this document. The Secure Advertisement and Neighborhood Discovery (SAND) protocol operates at and above the BP-layer, as shown in Figure 1, which insulates it from strict dependence on any specific CL for its message transport and allows the use of BPSec for message security. The full protocol stack of this document uses the UDPCL of [I-D.ietf-dtn-udpcl] as a zero-configuration default for its any-source multicast (ASM) capabilities but SAND could be, and is expected to be, used over other CLs to include unicast transports which might be informed by lower-layer discovery protocols (see Section 2.2).

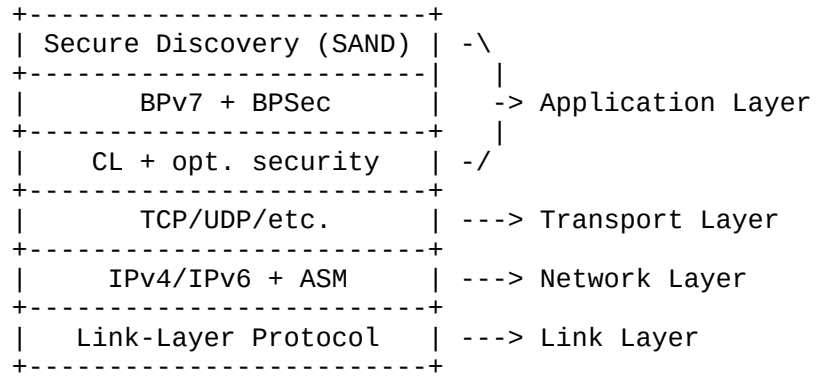

Figure 1: The Locations of SAND and BP above the Internet Protocol Stack

1.1. Scope

This document describes the format of the protocol data units passed between BP nodes for neighborhood discovery and defines behavior at message source and destination nodes.

This document does not address:

* The format of protocol data units of the Bundle Protocol, as those are defined elsewhere in [RFC9171]. This includes the concept of bundle fragmentation or bundle encapsulation.

* Logic for routing bundles along a path toward a bundle’s endpoint. This messaging protocol involves only one-hop singleton and group messaging.

* Policies or mechanisms for using BP extension blocks for purposes not defined in this document. Some networks could require specific extension blocks to be present for valid traffic.

* Policies or mechanisms for issuing Public Key Infrastructure Using X.509 (PKIX) certificates; provisioning, deploying, or accessing certificates and private keys; deploying or accessing certificate revocation lists (CRLs); or configuring security parameters on an individual entity or across a network.

* Uses of Bundle Protocol Security (BPSec) in which authentication of the Source Node ID is not possible (see Section 7.4).

1.2. Use of CDDL

This document defines CBOR structure using the Concise Data Definition Language (CDDL) of [RFC8610]. The entire CDDL structure can be extracted from the XML version of this document using the XPath expression:

’//sourcecode[@type="cddl"]’

The following initial fragment defines the top-level symbols of this document’s CDDL.

```
start = sand-adu-seq / sand-msg
```

1.3. 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.

Terminology used within the SAND protocol includes the following:

Source node: The BP node which is the source of a SAND Bundle containing SAND Messages.

Participating node: A BP node which sources and/or delivers SAND Bundles.

Reachable: A one-way determination of whether a source node can transfer bundles to a destination node (via any number of BP hops using any combination of CLs).

1-hop Reachable: A one-way determination of whether a destination node is reachable via a single BP hop.

1-hop Neighbor: A participating node for which this node is 1-hop reachable. The other node does not need to also be 1-hop reachable from this node to be a neighbor.

Mutual Neighbors: Two nodes which each identify the other as a 1-hop neighbor.

2-hop Neighbor: A participating node which is a 1-hop neighbor of a 1-hop neighbor of this node, but is not itself a 1-hop neighbor of this node.

Neighborhood: The collection of all 1-hop and 2-hop neighbors of a participating node.

2. General Protocol Description

The service of this protocol is the discovery of security credentials and capabilities of peer nodes within a 2-hop neighborhood without needing any pre-configuration on the participating node or on other nodes in the network.

Each participating node uses per-underlayer and per-neighbor timers to determine when to solicit and when to advertise data. Some external events (e.g. network- or link-level discovery) can be used to reset timers so that discovery can be completed more quickly.

The types of data able to be advertised by a node are the following, each associated with a subsection defining its message type and structure. Each type of data can be associated with a desired update time interval to ensure timely synchronization between peers.

Security credentials: Defined in Section 5.2 to contain credentials (_e.g._, PKIX certificates) associated with the node's identities which are used for signing/key-agreement/encryption.

Underlayer interfaces: Defined in Section 5.3 to contain information about what underlayer networks (and endpoints) are available on the node.

Convergence Layer instances: Defined in Section 5.4 to contain CL types and parameters needed to communicate with the node through specific underlayer networks.

Node resource forecast: Contains operating state and other forecasts for the node..

Local (1-hop) topology: Contains 1-hop neighbors seen by the node.

Routing willingness: Contains willingness for the node to route specific traffic, and stub network contents.

Application endpoints: Contains endpoints available on the node.

2.1. Extensibility

Future specifications can use this same messaging and transport mechanism to define additional message types and modes, including types for private or experimental use (see Section 8.3). Future modes could involve multi-hop flooding of bundles to distribute data for link-state style routing algorithms.

2.2. Relationship to other Discovery Protocols

Many of the structural, behavioral, and especially timing definitions in this specification follow the model of MANET messaging [RFC5444] and MANET NHDP [RFC6130] in both terminology and semantics. This is intentional to allow an implementer to understand BP discovery with very similar logic to MANET discovery. Where the NHDP is concerned with IP routers discovering reachable IP routes, the SAND is concerned with BP nodes discovering reachable bundle routes.

A node participating in the SAND protocol is expected to use lower-layer discovery mechanisms as necessary to enroll in a local network, obtain network-layer address(es) and parameters, and possibly

discover network-layer neighbor nodes and routers. This might involve the use of IPv4 Internet Router Discovery Protocol (IRDP) [RFC1256] or IPv6 Secure Neighbor Discovery Protocol (SEND) [RFC3971] [RFC4861] to determine IP neighbors, the Dynamic Host Configuration Protocol (DHCP) [RFC2131] to assign addresses and network-level parameters, or the Dynamic Link Exchange Protocol (DLEP) [RFC8175] to discover connectivity and specific IP neighbor nodes.

The robust and delay-tolerant protocol in this document is also compatible with the DNS-Based Service Discovery (DNS-SD) of BP routers by edge nodes as defined in [I-D.sipos-dtn-edge-zeroconf]. The SAND can be used to enroll an edge router in a network and synchronize routing information across a variety of network and link types, while DNS-SD is used within IP stub underlay networks (or enclaves) at the edges of the BP network.

3. Information Bases

SAND operates by each participating node keeping a persistent store of its enrolled underlayer networks, 1-hop neighbors, symmetric 2-hop neighbors along with attributes for each type of entity. These are used as the basis for outgoing SAND message contents and are updated as part of message reception processing.

3.1. Local Node Information Bases

This category of information is based on a participating node’s knowledge of its own underlayer network (ULN) and PKI configuration. It exists as input to SAND processing and messaging and is unaffected by the results of processing or reception of messages.

The resource information of Table 1 is used to populate Resource Advertisement messages. The information in this table are general to the node as a whole and not for any specific interface, network, or CL.

Name	Description
Validity Interval	This is the full time horizon for resource schedules in this information base.
Operating Schedule	This represents a time-varying operating state of the local node (as either "on" or "off") within the validity interval. An operating schedule which indicates "always on" is a valid default.

Table 1: Local Resource Information

The underlayer information described in Table 2 allows a participating node to define different profiles for different accessible networks (IP or otherwise). As defined in Section 6, when assembling and sending SAND messages much of the data can be filtered-down based on what is accessible via an interface (among other possible additional filtering, see Section 7.1).

Name	Description
Interface ID	This is the operating-system-specific unique identifier for a network interface.
Accessible Network Set	This is the set of IP addresses assigned to and associated IP subnetworks accessible to this node via the interface.
Link MTU	This is the configured or discovered maximum transmission unit (MTU) of the first-hop network link from the interface. Because this is a link MTU it excludes any network packet header overhead and is network-protocol-independent. This also represents a maximum outgoing size and not necessarily the maximum incoming size. This is not necessarily the same as a path MTU between any peers on this network, and a path MTU can be directional.
SAND Timer Configuration	This is the set of timers needed to configure SAND activities, as defined in Table 3.

Table 2: Local Interface Information Columns

The items in Table 3 represents the set of timer configuration needed to operate a participating node. As an information model, details such as specific units or encoding forms are left as an implementation matter. Because SAND uses the DTN time epoch and encoded form, SAND timer configuration SHOULD have a resolution down to at least one millisecond.

Name	Scope	Description
Minimum Time Interval	default and per-message-type	This represents the shortest time interval between sending messages of the same type on a particular interface or to a specific singleton destination.
Maximum Time Interval	default and per-message-type	This represents the longest time interval between sending messages of the same type on a particular interface or to a specific singleton destination. This is used as a timeout for Periodic Update messaging. The Maximum Time Interval MUST be longer than the Minimum Time Interval by some factor.
Validity Duration	default and per-message-type	This is embedded in messages optionally and used for SAND Bundle lifetimes. The Validity Duration MUST be longer than the Maximum Time Interval by some factor.

Table 3: SAND Timer Configuration

The Identity information of Table 4 is a logical table used both as a source for sending Credential Advertisement messages as well as for deriving BPSec policy used to send signed payloads and/or receive encrypted payloads.

Name	Description
Thumbnail	This is the x5t or c5t thumbnail of the encoded certificate, used as a selector.
Key Usage	This is the extracted Key Usage value, from Section 4.2.1.3 of [RFC5280], used as a selector.
Validity Time Interval	This is the extracted Validity interval, from Section 4.1.2.5 of [RFC5280], used as a selector.
Encoded Certificate	This is the DER-encoded X509 or encoded C509 certificate contents.

Table 4: Local Identity Information Columns

The trust anchor information of Table 5 is a logical table used for validating received peer certificates and for deriving BPSec policy used to receive signed payloads.

Name	Description
Subject Key Identifier	This is the extracted Subject Key Identifier, from Section 4.2.1.2 of [RFC5280], used as a selector.
Validity Time Interval	This is the extracted Validity interval, from Section 4.1.2.5 of [RFC5280], used as a selector.
Encoded Certificate	This is the DER-encoded X509 or encoded C509 certificate contents.

Table 5: Local Trust Anchor Information Columns

The convergence layer information of Table 6 is a logical table separated from the network information of Table 2 because many BP node deployments are expected to have CL instances that are bound to "any endpoint" addresses and can operate across multiple networks. Even in cases where a CL establishes persistent sessions which might be bound to a specific endpoint address or network, the CL instance as a whole can operate simultaneous sessions across many networks.

When used as a source for sending Convergence Layer Advertisement messages the advertised CL List is expected to be, but not required to be, filtered-down based on the interface/network on which the message will be sent. Besides being filtered-out for a specific network, a CL entry SHALL NOT be represented differently across different interfaces.
// TBD How to signal removal of a CL instance consistently across
// interfaces?

Name	Description
CL Type	This is the type of CL being represented, which need not be unique when there are multiple instances of a CL operating on a single node (with different parameters presumably).
Bind IP Addresses	This is the set of IP addresses to which the CL instance is bound (for either listening/receiving or connecting/sending). This includes both IPv4 and IPv6 addresses, and can include the "any endpoint" IPv4 _and_ IPv6 addresses (0.0.0.0 and :: respectively).
Bind Port Number	This is the specific transport-layer port number to which the CL instance is bound. This includes the default port number for each CL type.
Transport Security	This indicates whether transport security is required, prohibited, or neither (meaning it can be opportunistic or conditional) by the CL instance.
Roles	This indicates the logical roles which this CL instance is able to perform among "active" or "passive" options. The definition of an active role is CL-specific, but is expected to involve initiating outgoing conversations/connections/sessions, while a passive role is expected to involve listening for incoming ones. A single CL instance can be capable of both roles.
Type-Specific Parameters...	Each CL type (see Section 5.4) able to be represented by SAND can have a set of parameters specific to that type.

Table 6: Local Convergence Layer Information Columns

3.2. Neighbor Information Bases

An information base for 1-hop neighbor existence and intrinsic properties is managed separately from other information bases which represent relationships between nodes. Neighbor information can be received from any number of interfaces and is aggregated together into these information bases. In some cases the original received interface is kept and in others it is discarded in order to have a single record representing the neighbor node.

The neighbor node information of Table 7 is a logical table of immediate neighbors of this node. Multiple sources of information are aggregated together into this table.

Name	Description
Node ID	This is the SAND Singleton EID for the node.
MPR Selection	// TBD
Operating Schedule	This represents a time-varying operating state of the node (as either "on" or "off") as reported in Resource Advertisement messages.

Table 7: Neighbor Node Information Columns

An information base for 1-hop neighbor reachability in Table 8 is a logical table relating 1-hop neighbor nodes from Table 7 to specific local interfaces from Table 2 on which the node is reachable or on which messages have been received. Due to having multiple-network connectivity, it is possible to have multiple records identifying the same 1-hop Neighbor but each will have their own set of path metrics for a specific network.

Name	Description
Node ID	This is a cross-reference to the unique identifier from Table 7.
Interface ID	This is a cross-reference to the unique identifier from Table 2, the local network interface which has seen messages from the node.

Latest Timestamps	This is the latest bundle creation timestamp (Section 4.2.7 of [RFC9171]) for each SAND Message Type (Section 8.3) received from the neighbor on this interface, which is used to filter-out old, out-of-order messages in Section 4.5.
DNS Name Set	This is the set of DNS Names assigned to the neighbor node and accessible on the associated network.
IP Address Set	This is the set of IP addresses assigned to the neighbor node and accessible on the associated network.
Link MTU	This is the configured or discovered MTU of the first-hop link for the neighbor on the associated network. Similar to the local interface Link MTU, the actual Path MTU to and from this peer might be reduced from any one-hop Link MTU and might be directional.
Reachability	An indication of whether this neighbor has been only received from (HEARD), or if this node is present in that neighbor’s own 1-hop neighbor list (SYMMETRIC), or if no messages have been received after some time (LOST).
Path Metrics	This is the set of network-level metrics for expected path delay, maximum data rate, and bit error rate in each direction.
Timeout	This is the absolute local-clock time when this record becomes invalid.

Table 8: Neighbor Reachability Information Columns

Name	Description
Node ID	This is the SAND Singleton EID for the node.
Interface ID	This is a cross-reference to the unique identifier from Table 2, the local network interface which has seen messages from the node.
CL Type	This is the type of CL being represented, which need not be unique when there are multiple instances of a CL operating on a single node-and-underlayer.
CL Parameters	These are the transport and network parameters (see Table 6), as reported in Convergence Layer Advertisement messages.

Table 9: Neighbor CL Information Columns

3.3. Network Information Bases

This category of information is about an individual node, or pairs of nodes, independent of the location of the node in the network topology relative to this node.

An information base for 2-hop neighbors is limited to only those which have symmetric reachability between that node and one of the 1-hop neighbors from Table 7. This information includes simplified path metrics between the 1-hop and 2-hop neighbors. Due to having multiple-network connectivity, it is possible to have multiple records identifying the same 2-hop Neighbor but each will have their own set of path metrics for a specific network.

Name	Description
Node ID	This is the SAND Singleton EID for the node.
Latest Timestamps	This is the latest bundle creation timestamp (Section 4.2.7 of [RFC9171]) for each SAND Message Type (Section 8.3) received from the node, which is used to filter-out old, out-of-order messages in Section 4.5.

Table 10: Peer Node Information Columns

// TBD

Name	Description
Left Node ID	This is a cross-reference to a unique identifier from Table 10.
Right Node ID	This is a cross-reference to a unique identifier from Table 10.
Path Metrics	This is the set of network-level path metrics between the left and right node.

Table 11: Peer Reachability Information Columns

The Peer Certificate Information of Table 12 is used as way to store and cache certificates received via Credential Advertisement messages and validated in a time-independent way.

This means that certificates SHALL only be considered for caching by a node unless they have been part of a chain validated in accordance with the procedures of Section 6 of [RFC5280], up to a root CA from the Trust Anchor information of Table 5, while ignoring validity times. In addition to the base validation, all end-entity certificates SHALL only be considered for caching by a node if it conforms to the certificate profile of Section 4 of [I-D.ietf-dtn-bpsec-cose]. The Peer Certificate Information SHALL be de-duplicated from the Trust Anchor information of Table 5 by ignoring root CA certificates.

Name	Description
Node ID	This is the SAND Singleton EID for the node.
Thumbnail	This is x5t or c5t thumbnail of the encoded certificate, used as a selector.
Subject Key Identifier	This is the extracted Subject Key Identifier, from Section 4.2.1.2 of [RFC5280], used as a selector.
Key Usage	This is the extracted Key Usage value, from Section 4.2.1.3 of [RFC5280], used as a selector.
Validity Time Interval	This is the extracted Validity interval, from Section 4.1.2.5 of [RFC5280], used as a selector.
Encoded Certificate	This is the DER-encoded X509 or encoded C509 certificate contents.

Table 12: Peer Certificate Information

4. Message Transport

The SAND relies on BPv7 for end-to-end transport, one or more CL for one-hop transport, and BPSec for message security (both end-to-end and one-hop).

4.1. SAND Endpoints

Within BPv7, the SAND uses two types of well-known endpoint identifier (EID) used as source and/or destination for bundles transported between SAND participants.

SAND Singleton EID: This identifies the SAND application on the participating node and is used as the Source EID for SAND Bundles from the node. The SAND Singleton EID uses either the DTN or IPN scheme with a well-known service part as registered in // TBD and Section 8.2 respectively.

SAND Group EID: This allows participating nodes to receive SAND Bundles without any pre-configuration. The SAND Group EID uses the interplanetary multipoint communication (IMC) scheme with a well-known group number // TBA1 and service number // TBA2 as registered in Section 8.1.

Beyond its necessary use as a bundle EID, the SAND Singleton EID also serves as a unique identifier for the participating node and a unique and stable correlator for the SAND information bases (Section 3).

4.2. SAND Bundle

For the remainder of this document a bundle with a source matching the SAND Singleton EID will be referred to as a SAND Bundle. A SAND Bundle will have a destination of either the SAND Group EID or another SAND Singleton EID. This is illustrated by the following EID Pattern of [I-D.ietf-dtn-eid-pattern].

imc:TBA1.TBA2|ipn:*.*.TBA3|dtn://**/TBA4

A SAND Bundle has the following basic characteristics:

* The primary block of a SAND Bundle SHALL NOT be marked with the administrative flag, as the destination is not an administrative endpoint.

* A SAND Bundle SHALL contain a Hop Count extension block Section 4.4.3 of [RFC9171] to control the scope of the message. A message set intended only for 1-hop neighbors uses a Hop Limit of 1. That doesn’t prohibit a single outgoing message from being conveyed over multiple CLs (which is distinct from a single CL with multicast behavior).

* A SAND Bundle which is being forwarded SHALL contain a previous node identification in accordance with Section 4.3 This is a more strict requirement than BPv7 itself because SAND processing handles 1-hop neighbors differently than more distant nodes.

* A SAND Bundle SHALL be secured using BPSec blocks as defined in Section 4.4 in accordance with [RFC9172]. This document does not allow for an insecure use of SAND, although prototype implementations might use insecure transport as an intermediate step to full SAND compliance.

* The payload block of a SAND Bundle SHALL contain a CBOR sequence of items. The sequence SHALL consist of SAND version number followed by one or more bstr items, each containing an encoded SAND Message as defined in Section 5.

```
; The actual ADU is the sequence ~sand-adu-seq, not array enveloped
sand-adu-seq = [
    version: 1,
    1* adu-item
]
adu-item = bstr .cbor sand-msg
```

Each encoded SAND Message SHOULD use CBOR core deterministic encoding requirements from Section 4.2.1 of [RFC8949]. Even if not using deterministic encoding the first item of each SAND Message map SHALL have key zero (the Message Type item). This will cause the Message Type item to be the first one in the encoded message, which will allow a SAND processor to quickly determine if the specific message is of interest and skip over it if not.

Because multiple SAND Messages can be sent in a single bundle to which a Hop Limit applies, all messages in a single bundle need to have the same restriction (or non-restriction) of Hop Limit.

4.3. Previous Node Identification

In order to properly handle an SAND Bundle, the previous-hop node needs to be positively identified. This occurs by using either an authenticated identity from the CL over which the bundle was received, if available, a Previous Node extension block Section 4.4.1 of [RFC9171], if present, or the Source Node ID from the Primary block Section 4.3.1 of [RFC9171].

A SAND Bundle which is forwarded over a CL which includes an authenticated identity SHOULD NOT contain a Previous Node extension block. Otherwise, a SAND Bundle which is forwarded but not sourced on a node SHALL contain a Previous Node extension block to indicate that the node sending it is not its source.

4.4. Bundle Security

All SAND Bundles SHALL contain a Block Integrity Block (BIB) which targets the payload block. If that BIB does not include the primary block as additional authenticated data (AAD) then the BIB SHALL also target the primary block. The BIB MAY target any other blocks in the SAND Bundle.

The BIB targeting the payload block SHALL have a Security Source identifying the same node as the bundle Source EID. Due to node and network security policy, the Security Source EID MAY be different than the bundle Source EID. For example, a bundle source of ipn:974848.10.3 might have an associated Security Source of ipn:974848.10.0 but both identify the same IPN node.

Any SAND Bundles which contain a Previous Node block SHALL also contain a BIB which targets that Previous Node block. If that BIB does not include the primary block as additional authenticated data (AAD) then the BIB SHALL also target the primary block. The BIB MAY target any other blocks in the SAND Bundle. Similar to the payload, any BIB targeting the Previous Node block SHALL have a Security Source identifying the same node as the Previous Node block.

Any BIB used by SAND SHALL authenticate the bundle source EID and provide proof-of-possession (PoP) of the private key bound to the bundle source EID via PKIX certificate. This could be done using a cryptographic signature as available in the COSE Context of [I-D.ietf-dtn-bpsec-cose] because the primary block Creation Timestamp functions as a unique nonce for PoP.

A SAND Bundle MAY contain a Block Confidentiality Block (BCB) which targets the payload block when being transported over an insecure CL to a known set of recipients. If the BCB acceptors are not using group keys or known individual-recipient keys, the SAND Bundle SHOULD NOT be transported over a multicast CL.

When BPSec blocks can contain either certificate contents or thumbprints, the use of thumbprints is RECOMMENDED along with the use of Credential Advertisement messages to convey full credentials between nodes. To avoid the bootstrapping issue described in Section 7.3, the requirements of that section need to be met by a participating node.

4.5. Superseding Messages

Like MANET discovery and routing protocols, all of the message types defined in this document contain the full set of data of a particular type from a source node. The processing of any one message does not rely on incremental changes caused by the message or processing of any preceding-in-time messages of the same type. This also makes SAND Message processing idempotent and immune to duplicate reception, which is an expected property of BPv7 transport.

Because of this, the reception of a message sent earlier than the last-received message of the same type from the same source can be completely ignored. This logic applies per-message-type so a single SAND Bundle can contain some messages which are superseded along with others which are not. This comparison logic below along with the BPv7 requirement of timestamp uniqueness provide a strict ordering of all bundles from a source.

After receiving and processing each SAND Message, a node SHALL record the Reference Time from the message (using the bundle Creation Timestamp as alternative) along with the bundle source and message type. After receiving but before fully processing each SAND Message, a node SHALL look up the latest processed Reference Time based on the bundle source and message type. If the received message is identical to or earlier than the latest processed timestamp it SHALL be ignored by the application. The timestamp comparison SHALL be based on ordering of the DTN Time followed by the Sequence Number. Ignoring a superseded message SHALL NOT be considered a failure of processing the message, its containing ADU, or its containing bundle.

4.6. Default Convergence Layer

Part of the ability of the SAND to be a _discovery_ protocol is the need for initial authenticated messaging without any pre-configuration of any participating node. This is accomplished by using the UDPCL with an IP multicast destination, either IPv4 or IPv6 or both as needed on each interface.

All SAND-participating nodes SHALL listen for UDPCL packets on default port 4556, defined in Section 6.2 of [I-D.ietf-dtn-udpcl], and by joining IP multicast group(s) defined in Section 6.1 of [I-D.ietf-dtn-udpcl] on all interfaces over which the entity is participating in discovery. Nodes MAY listen for UDPCL packets destined for other (unicast) addresses and/or on other ports as needed.

When sending SAND Bundles, participating nodes SHALL use this default convergence layer in accordance with the modes defined in Section 6, one of which uses the above multicast configuration. Because an IP multicast destination is used, the source node will need to condition certain UDP and IP parameters based on a specific network interface to send from.

To send bundles using the UDPCL on a specific interface:

* An implementation-defined Redundancy Factor SHALL be used based on the specific interface.

* The default UDP port 4556 SHALL be used as its destination.

* The default UDP port 4556 SHOULD be used as its source.

* If a specific destination IP address is given that SHALL be used as its destination. Otherwise, use one or more of the following:

- If the interface has an assigned IPv4 address, a UDPCL transfer SHALL be sent using the IPv4 multicast address for "All BP Nodes" as its destination and that assigned address as its source.

- If the interface has an assigned IPv6 address, a UDPCL transfer SHALL be sent using the IPv6 multicast address for "All BP Nodes" as its destination and that assigned address as its source.

* Unless there is additional configuration available, the link MTU SHALL be assumed to be the path MTU for all nodes on that IP network. The sending node SHALL use CL segmentation as necessary to adapt the SAND Bundle size to the path MTU.

5. Message Structure and Types

A SAND Message is the top-level encoded structure exchanged between nodes. Messages are encoded according to the following requirements and the CDDL in Figure 2.

A SAND Message SHALL consist of a CBOR map containing at least one pair. All keys in the SAND Message map SHALL be CBOR int16 (unsigned or negative) items. This specification follows the pattern of CBOR [RFC8152] to use positive-valued map keys to indicate common parameters and negative-valued map keys to indicate type-specific parameters. This convention also applies to subordinate maps within SAND messages.

```
sand-generic-structure = {
    * label => value,
}
; Generic map label
label = int16
; Generic map value
value = any

; Signed integer that fits in 16-bit two’s complement form
int16 = (-32768 .. 32767) .within int
; Positive part of int16 for common values
comm16 = uint .le 32767
; Negative part of int16 for private values
priv16 = nint .ge -32768
```

Figure 2: SAND Generic Structure CDDL

The message common parameters are listed below and correspond with the CDDL of Figure 3. These are also registered in the IANA registry defined in Section 8.3.

Message Type: This pair uses key 0 and value of int16 identifying the type of message. The registry of message types is IANA-managed and defined in Section 8.3.

Reference Time: This pair uses key 2 and value of dtn-time indicating the absolute time of the start of validity of this message in the DTN time epoch (see Section 4.2.6 of [RFC9171]). If no Reference Time is present, the message SHALL be treated as being valid from the containing bundle's Creation Timestamp. The Reference Time is also used as the epoch for any schedule structure in the same message, defined later in this section.

For nodes with low-fidelity timing needs or having a low-precision clock this value SHOULD be omitted. Otherwise, this value SHALL be present to avoid any difference between message creation time and the BPA-sourced Creation Timestamp.

Validity Duration: This pair uses key 3 and value of time-duration indicating the validity-time of the message contents in milliseconds. If no Validity Duration is present, the message SHALL be treated as being valid through the containing bundle's Lifetime. The Validity Duration SHALL be interpreted as starting at the Reference Time from the same message, if present, or the bundle's creation timestamp.

For nodes with low-fidelity timing needs this value SHOULD be omitted. Otherwise, this value SHALL be sourced from the Validity Duration of Table 3.

Repetition Interval: This pair uses key 4 and value of time-duration indicating the periodic interval of the message type in milliseconds. If no Repetition Interval is present, the message SHALL NOT be assumed to be sent at a fixed periodic interval.

Every SAND Message SHALL contain a Message Type pair. Every SAND Message MAY contain any combination of other pairs with positive keys. The remaining pairs with negative keys SHALL be interpreted according to the Message Type.

```
sand-msg = $sand-msg .within sand-generic-structure

; Generic for messages, where ‘val‘ is the Message Type value
msg-base<val> = (
    0: val .within int16,
)
$$msg-common-grp //= (
    ? ref-time,
    ? msg-validity,
    ? msg-interval,
)
ref-time = (
    2: dtn-time,
)
msg-validity = (
    3: time-duration,
)
msg-interval = (
    4: time-duration,
)

; Duration in DTN units of milliseconds
time-duration = uint
```

Figure 3: SAND Message Structure and Common Parameters

Some of the advertisements defined in this document associate an optional validity _schedule_ with select data. Because the advertisements are expected to be sent by nodes periodically on the order of minutes, the form of this schedule is very simplified and focused only on a short-term time horizon using the Reference Time of the same message as its zero-offset epoch. When any schedule is present within a message the Schedule Reference Time item SHALL be present in the message and used as the schedule epoch time.

The schedule consists of pairs of duration values, with each pair representing an interval of time during which the schedule applies (and the gaps between intervals representing time during which the schedule does not apply).

```
schedule = [1* schedule-interval-pair]
schedule-interval-pair = (
  offset: time-duration,
  length: time-duration .gt 0,
)
```

Figure 4: Common Schedule CDDL

5.1. Data Solicitation

The Data Solicitation message type informs recipients that the sender desires specific types of SAND data from its peers. A peer entering a network SHOULD send a Data Solicitation message after an implementation defined time delay.

The Data Solicitation message SHALL be identified by message type 1. The message parameters are listed below and correspond with the CDDL of Figure 5.

Message Type List: This pair uses key -1 and value of an array of Message Type values. The Message Type List SHALL contain at least one item. Each Message Type List item SHALL be unique. The order of items within the array SHALL NOT be treated as significant by the recipient.

Each Data Solicitation message SHALL contain a Message Type List. The Data Solicitation message SHALL NOT be used to request a Data Solicitation type.

```
$sand-msg /= solicit-msg
solicit-msg = {
    msg-base<1>,
    $$msg-common-grp,
    solicit-types,
}
solicit-types = (
    -1: [1* msg-type]
)
msg-type = int16
```

Figure 5: Data Solicitation Message CDDL

5.2. Credential Advertisement

The Credential Advertisement message contains security credentials which identify the source node and contain key material for different security purposes. Each credential is itself verifiable up to a trusted root which is assumed to be configured in receivers of the advertisement.

Credentials in this message are sourced from the Identity Information Base of Table 4. Each credential can contain validity time intervals which have no strict relationship to the validity time of the containing advertisement message or the lifetime of the containing bundle, and do not relate to any SAND-form of schedule. The creator of an Credential Advertisement message MAY filter-in or filter-out credentials based on their validity time.

Each message SHOULD contain credentials valid at the time of creation. Each message MAY contain credentials valid only in the past or future. Those non-present-time credentials could be needed to verify old signatures or to pre-load future rollover keys respectively.

The Credential Advertisement message SHALL be identified by message type 2. The message parameters are listed below and correspond with the CDDL of Figure 6.

X509 Bag: This pair uses key -1 and value type COSE_X509 defined in [RFC9360] to convey PKIX certificates as an unordered "bag". Each bag MAY contain multiple end-entity certificates identifying the source node with different validity time or different extension items. Each bag SHOULD contain intermediate CA certificates up to, but not including, the root CA needed to verify all end-entity certificates.

C509 Bag: This pair uses key -2 and value type COSE_C509 defined in [I-D.ietf-cose-cbor-encoded-cert] to convey C509 certificates as an unordered "bag". Each bag MAY contain multiple end-entity certificates identifying the source node with different validity time or different extension items. Each bag SHOULD contain intermediate CA certificates up to, but not including, the root CA needed to verify all end-entity certificates.

Each Credential Advertisement SHALL contain an at least one end-entity credential identifying the sending node. A Credential Advertisement SHALL NOT contain any end-entity credential that does not identify the sending node.

The Credential Advertisement message is populated in-part using the data identified in Table 4, part of the Local Node Information Base described in Section 3.1.

```
$sand-msg /= cred-msg
cred-msg = {
    msg-base<2>,
    $$msg-common-grp,
    ? cred-x5bag,
    ? cred-c5bag,
}
cred-x5bag = (
    -1: COSE_X509  ; From [RFC 9360]
)
cred-c5bag = (
    -2: COSE_C509  ; From [I-D.ietf-cose-cbor-encoded-cert]
)
```

Figure 6: Credential Advertisement Message CDDL

5.3. Underlayer Advertisement

The Underlayer Advertisement message contains information about the ULN interface(s) on which a SAND Bundle has been sent, providing recipients information about communicating with the source node via the underlayer. When creating Underlayer Advertisement messages, the source node will populate it with parameters specific to the network on which the interface is an access point.

> | The well-known parameters defined in this document are focused on IP-based underlayer networks because this protocol is a product of the IETF. Other ULN technologies can be supported by SAND to advertise other forms of network addresses and/or protocol identifiers by either registering well-known type-specific parameters or using the private use range of type-specific parameters.

Each Underlayer Advertisement message SHALL be transported with a Hop Limit of 1. Only 1-hop neighbors are capable of using underlayer network parameters so there is no need to forward this to any other nodes in the network.

The Underlayer Advertisement message SHALL be identified by message type 8. The message parameters are listed below and correspond with the CDDL of Figure 7.

Validity Schedule: This pair uses key -1 and value type schedule as defined in Figure 4. Each time at which the schedule is valid indicates when the interface or link is expected to be usable. If this parameter is absent the interface SHALL be treated as always valid (within the validity schedule of the node itself, see Section 5.5).

DNS Name List: This pair uses key -2 and value type tstr or array of tstr from DNS names in accordance with [RFC1034]. If this parameter is absent the node SHALL be treated as not having a DNS name on the underlay network.

IP Address List: This pair uses key -3 and value of a single bstr or array of bstr from IPv4 or IPv6 addresses encoded as four-byte or 16-byte sequences respectively (consistent with untagged values of [RFC9164]). If this parameter is absent the node SHALL be treated as having only the IP address from which the containing bundle was sent, if it was received through an IP-based CL.

This address list MAY contain link-local addresses if the sender has an expectation that CLs will be usable over the associated IP endpoint.

Link MTU: This pair uses key -4 and value indicating the link MTU, as seen by the interface of the source node, in units of octets. This value SHOULD adhere to the lower limit of 68 octets for IPv4 [RFC791] or 1280 for IPv6 [RFC8200]. Other ULN technologies will still have an MTU value but with a different lower bound. If this parameter is absent then other means of configuring or estimating link or path MTU are needed.

```
$sand-msg /= uln-msg
uln-msg = {
    msg-base<8>,
    $$msg-common-grp,
    ? uln-schedule,
    ? uln-dns-name-list,
    ? uln-ip-addr-list,
    ? uln-mtu,
}
uln-schedule = (
    -1: schedule,
)

uln-dns-name-list = (
    -2: dns-name-ctr
)
dns-name-ctr = dns-name / [1* dns-name]
; Should agree with actual DNS restrictions in [RFC 1034]
dns-name = tstr .abnf ("subdomain" .det dns-name-syntax)
dns-name-syntax = '
    subdomain = label  *("." label)
    label = letter [[ldh-str] let-dig]
    ldh-str = let-dig-hyp *(let-dig-hyp)
    let-dig-hyp = let-dig / "-"
    let-dig = letter / digit
    letter = %x41-5A / %x61-7A
    digit = %x30-39
'

uln-ip-addr-list = (
    -3: ip-addr-ctr
)
ip-addr-ctr = ip-address / [1* ip-address]
; Agrees with untagged bstr contents from [RFC 9164]
ip-address = ipv4-address / ipv6-address
ipv4-address = bstr .size 4
ipv6-address = bstr .size 16

uln-mtu = (
    -4: mtu-size
)
mtu-size = uint .gt 0
```

Figure 7: Underlayer Advertisement Message CDDL

5.4. Convergence Layer Advertisement

The Convergence Layer Advertisement message indicates the CLA instances available on the source node interface sending the message, including both active and passive roles where applicable, and any parameters necessary for peers to make use of those instances. Each instance can also have an associated validity schedule.

Each Convergence Layer Advertisement message SHALL be transported with a Hop Limit of 1. Only 1-hop neighbors are capable of using CL data so there is no need to forward this to any other nodes in the network.

The Convergence Layer Advertisement message SHALL be identified by message type 3. The message parameters are listed below and correspond with the CDDL of Figure 8.

Convergence Layer List: This pair uses key -1 and value type of an array containing CL Instance items defined later in this section. Each Convergence Layer List SHALL contain at least one item. A Convergence Layer List item MAY have a non-unique CL Type parameter, indicating multiple instances of a particular CL.

Each Convergence Layer Advertisement SHALL contain a Convergence Layer List. Each item of the Convergence Layer List SHOULD be reachable via the interface over which the enveloping message is sent. Advertising CL instances which are not reachable by receiving SAND participants is simply a waste of advertising resources and possibly by resources on other participants trying to determine reachability.

```
$sand-msg /= cl-msg
cl-msg = {
    msg-base<3>,
    $$msg-common-grp,
    cl-list,
}
cl-list = (
    -1: [1* cl-recv]
)
```

Figure 8: CL Advertisement Message CDDL

5.4.1. CL Instance

Because different CLs are likely to have varying parameter sets, each CL is encoded as a CBOR map following the same conventions of SAND Message structure. There are several common CL parameters related to network- and transport-layer: a DNS name or IPv4/IPv6 address used to communicate with the node, and information about transport security policy.

It is also an important distinction that the CL parameterization is about the capability of delivering bundles to the advertising node. It is not about ability of the node to transmit bundles, which may in fact be more broad than its ability to receive. For example, in the situation where a node has an ephemeral IP address and no DNS name that node may not listen with any CL yet, because some CLs are bidirectional, it may have symmetric (BP layer) connectivity to some set of peer nodes. Even in that case there is still value in discovering the presence of the non-listening node because there is the potential for a contact (coming from that node) to allow bundle routes to other nodes "behind" that non-listening node.

The CL common parameters are listed below and correspond with the CDDL of Figure 9. These are also registered in the IANA registry defined in Section 8.4.

CL Type: This pair uses key 0 and value of int16 identifying the type of CL being defined. Possible CL Type values are defined Section 8.4 where, similar to message types, positive values are for well-known CL types and negative values are for private or experimental types.

Bind Address List: This pair uses key 3 and value of a single bstr or array of bstr from IPv4 or IPv6 addresses encoded as four-byte or 16-byte sequences respectively (consistent with untagged values of [RFC9164]). Each address represents a destination to which the CL is bound in order to receive traffic. If this parameter is absent, the CL SHALL bet treated as if it was bound to the "any endpoint" IPv4 _and_ IPv6 addresses (0.0.0.0 and :: respectively). If a node has either IPv4 or IPv6 addresses assigned but is not listening on the associated address family, this list SHALL contain the associated "any destination" bind address on which it is listening.

Bind Port Number: This pair uses key 4 and value of uint indicating the transport-layer port number to which the CL is bound. If this parameter is absent the default (_i.e._, IANA assigned) port number SHALL be used.

Transport Security Required: This pair uses key 5 and value of bool indicating whether transport security is required (when true) or prohibited (when false). If this parameter is absent there is no information about the required policy.

Role: This pair uses key 6 and value of uint containing flags indicating which CL-specific roles the source node can act as.

The role flag at bit 0 indicates that the node can act in an passive role. The role flag at bit 1 indicates that the node can act in an active role. If this parameter is absent it is assumed to be 0b11 (the node can be either role).

The definition of an "active role" is CL-specific but is expected to involve initiating outgoing conversations/connections/sessions rather than listening for incoming ones. Even when acting in the active role only, the CL MAY still be bound to a specific port number.

```
cl-recv = $cl-recv .within sand-generic-structure

; Generic for CLs, where ‘val‘ is the CL Type value
cl-base<val> = (
    0: val .within int16,
)
$$cl-common-grp //= (
    ? cl-bind-addr-list,
    ? cl-bind-port,
    ? cl-transport-sec-require,
    ? cl-role,
)

cl-bind-addr-list = (
    3: ip-addr-ctr,
)
cl-bind-port = (
    4: 1 .. 0xFFFF
)
; A hint about the security need, if any, for this CL
cl-transport-sec-require = (
    5: bool
)
; Indicate whether the entity can operate in CL-defined roles
cl-role = (
    6: uint .bits cl-role-flags
)
cl-role-flags = &(
    passive: 0,
    active: 1,
)
```

Figure 9: CL Structure and Common Parameters CDDL

An Underlayer Advertisement from a node can contain any combination of DNS Name List, IP Address List, and Link MTU items. Because of this individual CL Instances MAY contain additional DNS names and/or IP addresses specific to that instance. Duplication between underlayer DNS Name or IP Address and CL instance values SHOULD be avoided, but has no effect on the interpretation of the values.

If multiple values are present in Bind Address List for a CL it is an implementation matter to choose which one to attempt first, and whether multiple attempts are made sequentially or simultaneously. See [RFC8305] for detailed discussion of one possible algorithm for handling multiple network addresses for the same service.

5.4.1.1. TCPCLv4

This CL type specifically refers to the TCPCL version 4 of [RFC9174]. This CL type SHALL be identified by code point 1.

If the Port Number parameter is absent, the default TCPCL port 4556 SHALL be used. The Transport Security Required parameter SHALL indicate both the Contact Header USE_TLS flag and the post-negotiation policy enforcement (_i.e._, when the session will be disallowed). The Role parameter SHALL indicate whether the the TCPCL entity on the node can function as either active or passive or both.

The CL-specific parameters are listed below and correspond with the CDDL of Figure 10. These are also registered in the IANA registry defined in Section 8.4.

Message Type Support: This pair uses key -1 and value type of an array of TCPCL message type code points indicating which types the advertising node supports. Well-known code points are managed in the "Bundle Protocol TCP Convergence-Layer Version 4 Message Types" registry of [IANA-BP]. All nodes SHALL include the minimum support defined in [RFC9174] as types 0x01 through 0x07 inclusive.

Session Extension Type Support: This pair uses key -2 and value type of an array of TCPCL session extension type code points indicating which types the advertising node supports. Well-known code points are managed in the "Bundle Protocol TCP Convergence-Layer Version 4 Session Extension Types" registry of [IANA-BP]. There is no required minimum support defined in [RFC9174].

Transfer Extension Type Support: This pair uses key -3 and value type of an array of TCPCL transfer extension type code points indicating which types the advertising node supports. Well-known code points are managed in the "Bundle Protocol TCP Convergence-Layer Version 4 Transfer Extension Types" registry of [IANA-BP]. All nodes SHALL include the minimum support defined in [RFC9174] as type 0x01.

```
$cl-recv /= {
    cl-base<1>,
    $$cl-common-grp,
    ? tcpcl-msg-support,
    ? tcpcl-sessext-support,
    ? tcpcl-xferext-support
}

tcpcl-msg-support = (
    -1: [+ tcpcl-msg-type]
)
tcpcl-msg-type = 0 .. 0xFF

tcpcl-sessext-support = (
    -2: [+ tcpcl-ext-type], ; session extension types from [IANA-BP]
)
tcpcl-xferext-support = (
    -3: [+ tcpcl-ext-type], ; transfer extension types from [IANA-BP]
)
tcpcl-ext-type = 0 .. 0xFFFF
```

Figure 10: TCPCLv4 Parameters CDDL

5.4.1.2. UDPCLv2

This CL type specifically refers to the UDPCL Version 2 of [I-D.ietf-dtn-udpcl]. This CL type SHALL be identified by code point 2.

If the Port Number parameter is absent, the default UDPCL port 4556 SHALL be used. The Transport Security Required parameter SHALL indicate the need for DTLS security when receiving CL messages.

The CL-specific parameters are listed below and correspond with the CDDL of Figure 11. These are also registered in the IANA registry defined in Section 8.4.

Extension Support: This pair uses key -1 and value type of an array of UDPCL extension code points indicating which extensions the advertising node supports. Well-known code points are managed in the "UDPCLv2 Extensions" registry of [IANA-BP]. This information is equivalent to the contents of Section 3.5.1 of [I-D.ietf-dtn-udpcl] without needing to operate the actual CL.

```
$cl-recv /= {
    cl-base<2>,
    $$cl-common-grp,
    ? udpcl-ext-support,
}

udpcl-ext-support = (
    -1: [+ ext-key], ; ext-key from [I-D.ietf-dtn-udpcl]
)
```

Figure 11: UDPCLv2 Parameters CDDL

5.4.1.3. CCSDS LTPCL Over UDP

While there is no IETF specification for transporting BPv7 bundles over the Licklider Transport Protocol (LTP) of [RFC5326], the CCSDS profile of BPv7 includes a specification for this in Appendix B of [CCSDS-BPv7] using Client Service ID value 4. Additionally the LTP-over-UDP binding is defined in Section 3.3 of [RFC7122]. This CL type SHALL be identified by code point 3.

If the Port Number parameter is absent, the default LTP port 1113 SHALL be used. The BPv7 use of LTP does not specify a transport-layer security mechanism.

The CL-specific parameters are listed below and correspond with the CDDL of Figure 12. These are also registered in the IANA registry defined in Section 8.4.

Engine ID: This pair uses key -1 and value type uint to advertise the specific Engine ID used by this LTP entity when sending and correlating LTP segments. Knowing the Engine ID of a peer before initiating or responding to LTP sessions is necessary for some implementations.

Extension Support: This pair uses key -2 and value type of an array of LTP extension tag code points indicating which tags the advertising node supports. Well-known code points are managed in the "LTP Extension Tags" registry of [IANA-LTP]. There is no required minimum support defined in [RFC5326].

```
$cl-recv /= {
    cl-base<2>,
    $$cl-common-grp,
    ? ltp-engine-id,
    ? ltp-ext-support,
}

ltp-engine-id = (
    -1: uint,
)
ltp-ext-support = (
    -2: [+ uint]
)
```

Figure 12: LTPCL Parameters CDDL

5.4.1.4. TCPCLv3

While there is no concrete specification for transporting BPv7 bundles over TCPCL version 3 [RFC7242], this specification makes an allocation to allow a node to advertise that it is using this combination of protocols. This CL type SHALL be identified by code point 32766.

If the Port Number parameter is absent, the default TCPCL port 4556 SHALL be used. The TCPCL version 3 does not specify a transport-layer security mechanism.

```
$cl-recv /= {
    cl-base<32766>,
    $$cl-common-grp,
}
```

Figure 13: TCPCLv3 Parameters CDDL

5.4.1.5. RFC 7122 UDPCL

While there is no concrete specification for transporting BPv7 bundles over the UDPCL as defined in [RFC7122], this specification makes an allocation to allow a node to express that it is using this combination of protocols. This CL type SHALL be identified by code point 32767.

```
$cl-recv /= {
    cl-base<32767>,
    $$cl-common-grp,
}
```

Figure 14: RFC 7122 UDPCL Parameters CDDL

5.5. Resource Advertisement

The Resource Advertisement message is used to indicates the node's resource forecast (operating state and storage) for some near time horizon. This corresponds to a node-scope schedule of Section 2.3.1 of [I-D.ietf-tvr-requirements] and these resource relate to all of the CLs exposed in the Convergence Layer Advertisement message from the same node. Per the definitions in Section 5, each schedule applies within the Validity Duration of the message.

Resource Advertisement messages are populated using the data in Table 1, part of the Local Node Information Base described in Section 3.1.

The Resource Advertisement message SHALL be identified by message type 4. The message parameters are listed below and correspond with the CDDL of Figure 15.

Operating State: This pair uses key -1 and value of a schedule item as defined in Figure 4. Each time at which the schedule is valid indicates when the node is forecast to be operating.

```
$sand-msg /= resource-msg
resource-msg = {
    msg-base<4>,
    $$msg-common-grp,
    ? operating-state,
}
operating-state = (
    -1: schedule,
)
; More TBD
```

Figure 15: Resource Advertisement CDDL

5.6. Local Topology Advertisement

The Local Topology Advertisement message allows a participating node to enumerate the 1-hop neighbors with which the source node can communicate (via some unspecified CL or combined aggregate of CLs). Each neighbor is identified by it's SAND Singleton EID which is a unique across a BP network.

Each 1-hop neighbor (peer) is associated with a specific status and a set of communication metrics similar to the behavior of MANET NHDP [RFC6130]. The source of the metrics are not specified by this

document, but might come from estimating based on SAND traffic exchanged with the peer. In addition, some of the data comprising the Local Topology Advertisement message is sourced from the Neighbor Information Base, such as Reachability and Path Metrics as discussed in Table 8.

The Local Topology Advertisement message SHALL be identified by message type 5. The message parameters are listed below and correspond with the CDDL of Figure 16.

Neighbor List: This pair uses key -1 and value of an array of Neighbor Node maps indicating the SAND Singleton EID and routing-related Routing Metrics for each 1-hop neighbor of the source node. Each Neighbor List SHALL contain at least one item. Each Neighbor List item SHALL have a unique Node ID parameter.

Each Local Topology Advertisement SHALL contain a Neighbor List.

```
$sand-msg /= localtopo-msg
localtopo-msg = {
    msg-base<5>,
    $$msg-common-grp,
    locotopo-nbr-list,
}
locotopo-nbr-list = (
    -1: [1* locotopo-nbr]
)
```

Figure 16: Local Topology Advertisement CDDL

5.6.1. Neighbor Node

Each item of the Neighbor List represents a combination of a 1-hop neighbor node, its direct parameters, and routing metrics associated with traffic from and to that node.

The common neighbor parameters are listed below and correspond with the CDDL of Figure 17. These are also registered in an IANA registry defined in Section 8.5. Neighbor parameters with negative keys are reserved for private or experimental use.

Node ID: This pair uses key 0 and value of eid from [RFC9171] representing the unique SAND Singleton EID for the neighbor node.

MPR Selection: This pair uses key 1 and value of // TBD representing the choice of this node as an MPR for the source node.

Reachability: This pair uses key 2 and value type uint containing an enumerated value indicating the status of communication with the neighbor. The value is one of the following:

HEARD (1): This means a message has been received from the peer but this node has not yet appeared in the local topology advertised by that peer.

SYMMETRIC (2): This means that this node is present in the local topology advertised by the peer, so at least one message has been received in both directions between the nodes.

LOST (3): This means that no message has been received from the peer within an implementation-defined timeout interval.

Routing Metrics List: This pair uses key 3 and value type of an array of Routing Metrics related to the neighbor node. Each Routing Metrics List SHALL contain at least one item. Each Routing Metrics List item SHALL have a unique combination of Routing Type, Direction, and Validity Schedule parameters.

```
locotopo-nbr = locotopo-nbr-base .within sand-generic-structure

locotopo-nbr-base = {
    ; mandatory items
    nbr-nodeid,
    nbr-comm-status,
    nbr-metrics-list,
    ; optional items
    $$nbr-common-grp,
    * priv16 => any,
}

nbr-nodeid = (
    0: eid ; From [RFC 9171]
)

nbr-comm-status = (
    1: &(
        HEARD: 1,
        SYMMETRIC: 2,
        LOST: 3,
    )
)

nbr-metrics-list = (
    2: [1* nbr-metrics]
)
```

Figure 17: Peer Structure and Parameters CDDL

5.6.2. Routing Metrics

Each Routing Metrics map is associated with a specific routing type and a set of metrics for BP and underlayer traffic from and to that node. Some of the Routing Metrics parameters are common across all algorithms and some are inputs to a specific routing algorithm.

It is expected that two nodes which each see the other as a 1-hop neighbor will provide opposite and similar metrics between each other. If the mutual neighbor nodes don't support the same routing algorithms, the total set of metrics will be different. Because there is no specific synchronization between neighbors, even when mutual neighbors advertise the same metric items there is no guarantee or expectation that they will have the same values. It is an implementation detail for how to reconcile routing metrics between mutual neighbors (_e.g._ by averaging between neighbors' advertisements) when needed for input to routing algorithms.

The common routing metrics are listed below and correspond with the CDDL of Figure 18. These are also registered in an IANA registry defined in Section 8.5.

Routing Type: This pair uses key 0 and value type int16 identifying a specific routing algorithm. The registry of SAND routing types is IANA-managed and defined in Section 8.5.

Direction: This pair uses key 1 and value type uint containing an enumerated value indicating the link direction associated with the metrics in the item. The value is one of the following:

TRANSMIT (1): This means the metrics are associated with traffic from this node to the parent peer.

RECEIVE (2): This means the metrics are associated with traffic to this node from the parent peer.

Validity Schedule: This pair uses key 2 and value type schedule as defined in Figure 4. Each time at which the schedule is valid indicates when communication is expected to be available (in the associated Direction).

This is different than the resource schedule of the node itself, and represents availability of the shared network between the source node and this peer.

```
nbr-metrics = nbr-metrics-base .within sand-generic-structure

nbr-metrics-base = {
    ; mandatory items
    metrics-direction,
    ; optional items
    $$nbr-metrics-common-grp,
    * priv16 => any,
}

metrics-direction = (
    1: &(
        TRANSMIT: 1,
        RECEIVE: 2,
    )
)

nbr-metrics-common-grp //= (
    ? nbr-schedule
)

nbr-schedule = (
    3: schedule
)
```

Figure 18: Routing Metrics Structure and Parameters CDDL

5.6.2.1. SABR/CGR

This routing type captures metrics needed for input to the Schedule-Aware Bundle Routing (SABR) algorithm defined in [CCSDS-SABR].

> These parameters function, in a limited form, as a way to represent a short-time-horizon contact plan between the source node and the neighbor node. These metrics are not expected to be used for defining or distributing long-term plans which greatly exceed the Validity Duration of the containing SAND message.

The SABR routing metrics are listed below and correspond with the CDDL of Figure 19. These are also registered in an IANA registry defined in Section 8.5. Metrics parameters with negative keys are delegated to an algorithm-specific registry.

Maximum Data Rate: This pair uses key -1 and value of unsigned-fraction indicating the expected maximum data rate of traffic in octets per second. This data rate is measured at the underlying link layer, not just the throughput of BP PDUs.

Delay: This pair uses key -2 and value of time-duration indicating the expected one-way light time (OWLT) of traffic in milliseconds. This delay is measured between the two BP nodes so it is more than just the free-space propagation delay, it also includes any expected underlay, CLA, and BPA processing time.

Bit Error Rate: This pair uses key -3 and value of unsigned-fraction indicating the expected bit error rate (BER) of traffic as a ratio. This BER is measured at the underlying link layer and includes errors which are caught by underlayer checksums (_e.g._, where the CL segment/frame is lost).

```
;    ? nbr-rx-delay,
;    ? nbr-tx-delay,
;    ? nbr-rx-datarate,
;    ? nbr-tx-datarate,
;    ? nbr-rx-ber,
;    ? nbr-tx-ber,

nbr-rx-delay = (
    4: time-duration
)
nbr-tx-delay = (
    5: time-duration
)
; Maximum data rate in bytes-per-second
nbr-rx-datarate = (
    6: unsigned-fraction
)
nbr-tx-datarate = (
    7: unsigned-fraction
)
; Estimated BER as a ratio
nbr-rx-ber = (
    8: unsigned-fraction
)
nbr-tx-ber = (
    9: unsigned-fraction
)

; Same structure as tag #4 "decimal fraction" but limited in domain
unsigned-fraction = [
    exp: (-20 .. 20) .within int,
    mantissa: uint,
]
```

Figure 19: SABR Routing Metrics CDDL

For conciseness of encoding, the unsigned-fraction values SHOULD limit the mantissa to less than 8 bits. This limits the precision of encoded values but because these are all rough estimates that should be sufficient for contact planning purposes.

There is no required combination of RX and TX parameters for any peer. Because these might be estimated from traffic or some kind of underlying discovery protocol (_e.g._, DLEP) it is possible to obtain estimates for some subset of these but not all of them.

For both RX and TX Data Rate values, the rate is averaged over the entire valid time, so it is actually average-of-maximum rate. Another way to think of it is that the sum-total valid time duration multiplied by the data rate value will yield a total data volume that is transferable from (or to) the peer within the validity duration.

5.7. Router Advertisement

The Router Advertisement message exposes parameters about the source node’s willingness to route bundles with different categories of destination EIDs.
// Each willingness to associate with an EID pattern?

The Router Advertisement message SHALL be identified by message type 6. The message parameters are listed below and correspond with the CDDL of Figure 20.

Willingness for Singleton: This pair uses key -1 and value of uint representing a willingness to route (see later definition) for bundles with a singleton destination EID. The absence of this pair SHALL be interpreted as a willingness of zero (not willing).

Willingness for Multipoint: This pair uses key -2 and value of uint representing a willingness to route (see later definition) for bundles with a non-singleton destination EID. The absence of this pair SHALL be interpreted as a willingness of zero (not willing).

Attached Networks: This pair uses key -3 and value of bstr embedding an EID Pattern as defined in Section 4 of [I-D.ietf-dtn-eid-pattern]. The value contains the set of endpoints not participating in SAND but for which the source node is willing to route. The value MAY be an any-scheme pattern or contain an any-SSP pattern.

Each willingness value is an integer in the inclusive range from 0 through 6, where 0 indicates the node will never route for that type and the values 1 through 6 indicate an increasing level of willingness. In the absence of additional configuration, a node which is willing to route SHALL have a default willingness of 3 and include the associated message item.

The presence of an Attached Networks pattern allows a participating router to expose node information from a stub network setting "behind" the router. All of these endpoints SHALL be treated as having persistent and reliable connectivity to the router sending the message. It also allows the router to advertise that it is acting as a BP gateway by using the pattern "*:**", but care needs to be taken for which underlayer networks the gateway advertisement is made. Only a stub network should see the gateway advertisement.

```
// More TBD

$sand-msg /= router-msg
router-msg = {
    msg-base<6>,
    $$msg-common-grp,
    ? will-route-singleton,
    ? will-route-multipoint,
    ? routeable-endpoints,
}
will-route-singleton = (
    -1: will-route .default 0
)
will-route-multipoint = (
    -2: will-route .default 0
)
will-route = (0 .. 6) .within uint

routeable-endpoints = (
    -3: embed-eid-pattern, ; From [I-D.ietf-dtn-eid-pattern]
)
```

Figure 20: Router Advertisement CDDL

5.8. Endpoint Advertisement

The Endpoint Advertisement message contains information about the endpoints registered on the sending node at the time of the message formation. This information does not include information about the registration state (active or passive, as defined in Section 3.1 of [RFC9171]). When creating Endpoint Advertisement messages, the source node MAY filter advertised endpoints to prevent visibility of particular endpoints to particular underlayer networks or destination nodes.

The Endpoint Advertisement message SHALL be identified by message type 7. The message parameters are listed below and correspond with the CDDL of Figure 21.

Endpoint List: This pair uses key -1 and value of an array containing endpoint-defn items defined later in this section. Each Endpoint List SHALL contain at least one item. Each Endpoint List item SHALL have a unique EID Pattern parameter. There SHALL NOT be any intersection between EID Pattern parameters of multiple items.

Each Endpoint Advertisement SHALL contain an Endpoint List. Each item of the Endpoint List SHOULD be reachable as a bundle destination on the node sending the message.

```
$sand-msg /= endpoint-msg
endpoint-msg = {
    msg-base<7>,
    $$msg-common-grp,
    endpoint-list,
}
endpoint-list = (
    -1: [1* endpoint-defn]
)
```

Figure 21: Endpoint Advertisement CDDL

5.8.1. Endpoint Definition

Because different endpoints (and their applications) are likely to have varying parameter sets, each endpoint definition is encoded as a CBOR map following the same conventions of SAND Message structure. Because a node is expected to have a possibly large number of endpoints registered with similar advertised parameters, each endpoint definition is organized around an EID Pattern rather than a single EID. There are common endpoint parameters related to security policy.

The common endpoint parameters are listed below and correspond with the CDDL of Figure 22. These are also registered in the IANA registry defined in Section 8.6. Endpoint parameters with negative keys are reserved for private or experimental use.

EID Pattern: This pair uses key 0 and a value of a bstr embedding an EID Pattern as defined in Section 4 of [I-D.ietf-dtn-eid-pattern]. Each of the other items in the parent definition applies to all EIDs matched by this pattern. For singleton endpoints, the node-identifying portion of the pattern SHALL agree with the message source node.

Payload Security Required: This pair uses key 5 and value of uint containing flags indicating which aspects of payload security are required for communicating with this endpoint. If this parameter is absent there is no information about the required policy.

The security flag at bit 0 indicates that the payload SHALL be a target of a BIB that the node can accept. The security flag at bit 1 indicates that the payload SHALL be a target of a BCB that the node can accept. The security flag at bit 1 indicates that the any accepted security block SHALL bind to the primary block as AAD.

```
endpoint-defn = endpoint-base .within sand-generic-structure
endpoint-base = {
    0: embed-eid-pattern, ; From [I-D.ietf-dtn-eid-pattern]
    $$endpoint-common-grp,
    * priv16 => any,
}

$$endpoint-common-grp //= (
    ? endpoint-sec-require,
)

; A hint about the security need, if any, for payloads
; delivered to the associated endpoints
endpoint-sec-require = (
    5: uint .bits endpoint-sec-flags
)
endpoint-sec-flags = &(
    need-bib: 0,
    need-bcb: 1,
    bind-primary: 2,
)
```

Figure 22: Endpoint Definition and Common Parameters CDDL

5.8.1.1. SAND Singleton Endpoint

Each participating node SHOULD register and advertise a singleton endpoint for the SAND application itself. This allows SAND Bundles to be transported with payload confidentiality to specific peer nodes. The endpoint SHOULD use the well-known service number from Section 8.2 when the Node ID uses the IPN scheme.

An advertisement for the SAND singleton endpoint SHALL contain at least a Payload Security Required value.

6. Messaging Modes

This section outlines the ways in which SAND Messages (Section 5) can be combined into SAND Bundles (Section 4.2) and transported to other SAND-participating nodes. Because SAND messages can be combined in many ways and because the contents of each message can be filtered-out based on the need for data privacy or operational security considerations, these modes are not exhaustive of how SAND messages can be used to advertise to and discover about peers.

6.1. Group Hello

When a node first enrolls in a network, or when a node is informed of a link state change to active, it SHOULD send an Group Hello message set with a Hop Limit of 1 using the Default Convergence Layer. Because this is a group destination, it will be sent as a plaintext payload. This message set consists of the following:

Data Solicitation: The node SHALL include a Data Solicitation message if the time since the last Data Solicitation on that interface has exceeded an implementation-defined threshold.

For a new enrollment, a node SHOULD solicit all of the following: Credential Advertisement, Resource Advertisement, Interface Advertisement, Convergence Layer Advertisement, Local Topology Advertisement. For link state change, a node SHOULD solicit at least Local Topology Advertisement.

Credential Advertisement: For a new enrollment, the node SHALL include an Credential Advertisement message containing certificates which the node considers safe to advertise on that interface and its network. For a link state change, the node SHOULD include an Credential Advertisement message if the time since the last Credential Advertisement on that interface has exceeded an implementation-defined threshold.

Interface Advertisement: The node SHALL include an Interface

Advertisement containing parameters which apply to that interface.

Convergence Layer Advertisement: The node SHALL include a Convergence Layer Advertisement message containing CLs which apply to that interface.

Local Topology Advertisement: The node SHOULD include a Local Topology Advertisement message containing peers which the node considers safe to advertise on that interface and its network.

6.2. Targeted Hello

When a node is informed by some lower-level discovery mechanism that a specific peer is reachable via IP address, it SHOULD send a Targeted Hello message set with a Hop Limit of 1 using the Default Convergence Layer with the peer's IP address as destination. This message set contains the same messages and data as the Group Hello and is also sent as plaintext payload when peer BP identity and security information is not yet available.

6.3. Response to Solicitation

// TBD

6.4. Periodic Update

// TBD

7. Security Considerations

This section separates security considerations into threat categories based on guidance of BCP 72 [RFC3552].

7.1. Threat: Passive Leak of Data

Because this protocol is involved in enrollment of a node into a BP network, the initial group messaging from a participating node necessarily has a plaintext payload.

One avoidance of passive leaking is for the source node to filter-out sensitive data from its initial messages. This could include not disclosing certain IP addresses assigned to interfaces, certain CL instances, or certain 1-hop neighbors from advertisement messages. It could also include not disclosing certificates from CAs or with key purposes which are sensitive. Because the initial group messaging is interface-specific, the filtering-out of data does not need to be symmetric across all interfaces on which the node is participating in SAND.

Another possible mitigation is to avoid group messaging entirely on an interface and rely on lower-layer network peer discovery to identify potential participants and then attempt to use UDPCL with DTLS to establish secure transport with the peer. While more secure from eavesdroppers, this method is more time- and resource-consuming than group messaging. This method also assumes that transport-layer security is even possible while in some environments only BP-layer security is viable.

7.2. Threat: Denial of Service

The behaviors described in this section all amount to a potential denial-of-service to a participating node. The denial-of-service could be limited to an individual node, or could affect all entities on a host or network segment.

Because there is a Data Solicitation mechanism it is possible to attempt an amplification attack by soliciting many types of data, with corresponding large bundle size, using a small request bundle. A mitigation of this kind of attack is to treat solicitation requests in the context of minimum and maximum update intervals. Rather than causing a set of advertisements directly, the solicitation is treated as an update timer reset limited accordingly.

A participating node may, intentionally or not, use singleton or group messaging to overwhelm a link or network, requiring the receiving node to process the data. This kind of attack applies to BP Agents generally and is not specific to SAND messaging. The victim node can block bundles from network peers which are thought to be incorrectly behaving within network.

Because the Default Convergence Layer uses UDP transport, the recommended configurations of this document result in behaviors which conform to the limitations of [RFC8085], specifically Section 4. This protocol uses the "congestion avoidance" strategy by having implementations choose appropriate timer intervals for minimum SAND updates and, when applicable, for UDPCL redundant transmission Section 3.3.1 of [I-D.ietf-dtn-udpcl].

7.3. Identity Bootstrapping

For BP nodes enrolling in a network for the first time, with proper authorization to do so, other participating nodes will not be able to authenticate SAND Bundles per the requirements of Section 4.4 without having associated end-entity certificates available.

A participating node SHOULD have the ability for an application to inspect the payload of a bundle as part of BPSec processing in order to extract necessary certificates from Credential Advertisement messages. If that is not possible, a source node SHOULD include necessary certificates within any BIB needed to satisfy requirements of Section 4.4. Determination of this need is a network administration matter outside the scope of this document.

7.4. Messaging Without Authentication

In environments where PKI is not available for the BP-layer, the SAND could be operated without the requirements of Section 4.4 but doing so is outside the scope of this document. Even in cases where there is network-layer or link-layer security, specifically source authentication with proof-of-possession, having an authorized lower-layer identity does not convey to unlimited BP-layer authorization. Part of the purpose of BP-layer integrity protection is to prevent a misconfigured node from polluting topology information bases of BP routers.

8. IANA Considerations

Registration procedures referred to in this section are defined in [RFC8126].

8.1. Well-Known IMC Group and Service

Within the URI Schemes registry group of [IANA-URI], the registry titled "’imc’ Scheme Well-known Group Numbers for BPv7" has been updated to include the following entry.

Value	Description	Reference
TBA1	SAND Participants	[This specification]

Table 13: ’imc’ Scheme Well-known Group Numbers for BPv7

Within the URI Schemes registry group of [IANA-URI], the registry titled "'imc' Scheme Well-known Service Numbers for BPv7" has been updated to include the following entry.

Value	Description	Reference
TBA2	SAND Messaging	Section 4 of [This specification]

Table 14: 'imc' Scheme Well-known Service Numbers for BPv7

8.2. Well-Known IPN Service

Within the URI Schemes registry group of [IANA-URI], the registry titled "'ipn' Scheme URI Well-known Service Numbers for BPv7" has been updated to include the following entry.

Value	Description	Reference
TBA3	SAND Messaging	Section 4 of [This specification]

Table 15: 'ipn' Scheme URI Well-known Service Numbers for BPv7

8.3. SAND Message Registries

EDITOR NOTE: registries to-be-created upon publication of this specification.

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND Message Common Parameter Keys" and initialize it with the contents of Table 16. The registration procedure is Specification Required.

Specifications of new common parameters need to define the code point (an int16 integer) as well as the CBOR form and meaning of the associated value.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

Code	Name	Reference
-32768 to -32513	Reserved for private and experimental type-specific parameters	[This specification]
-32512 to -1	delegated to the SAND Message Type-Specific Parameter Keys registry	[This specification]
0	Message Type	Section 5 of [This specification]
2	Reference Time	Section 5 of [This specification]
3	Validity Duration	Section 5 of [This specification]
4	Repetition Interval	Section 5 of [This specification]
5 to 32511	unassigned	
32512 to 32767	Reserved for private and experimental common parameters	[This specification]

Table 16: SAND Message Common Parameter Keys

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND Message Types" and initialize it with the contents of Table 17. For positive code points the registration procedure is Specification Required. Negative code points are reserved for use on private networks for functions not published to the IANA.

Specifications of new message types need to define the code point (an int16 integer), as well as what message parameters are required and allowed within the message. Specifications need to define how those CBOR parameters are used by a node to relate the encoded message to the agent's information bases.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

Code	Name	Reference
-32768 to -1	Private/Experimental Use	[This specification]
0	reserved	[This specification]
1	Data Solicitation	Section 5.1 of [This specification]
2	Credential Advertisement	Section 5.2 of [This specification]
8	Underlayer Advertisement	Section 5.3 of [This specification]
3	Convergence Layer Advertisement	Section 5.4 of [This specification]
4	Resource Advertisement	Section 5.5 of [This specification]
5	Local Topology Advertisement	Section 5.6 of [This specification]
6	Router Advertisement	Section 5.7 of [This specification]
7	Endpoint Advertisement	Section 5.8 of [This specification]
9 to 32767	unassigned	

Table 17: SAND Message Types

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND Message Type-Specific Parameter Keys" and initialize it with the contents of Table 18. The registration procedure is Specification Required.

Specifications of new common parameters need to define the associated message type, code point (an int16 integer), and the CBOR form and meaning of the associated value.

Message Type	Code	Name	Reference
Data Solicitation			
1	-1	Message Type List	Section 5.1 of [This specification]
Credential Advertisement			
2	-1	X509 Bag	Section 5.2 of [This specification]
2	-2	C509 Bag	Section 5.2 of [This specification]
Underlayer Advertisement			
8	-1	Validity Schedule	Section 5.3 of [This specification]
8	-2	DNS Name List	Section 5.3 of [This specification]
8	-3	IP Address List	Section 5.3 of [This specification]
8	-4	Link MTU	Section 5.3 of [This specification]
Convergence Layer Advertisement			
3	-1	Convergence Layer List	Section 5.4 of [This specification]
Resource Advertisement			
4	-1	Operating State	Section 5.5 of [This

			specification]
Local Topology Advertisement			
5	-1	Neighbor List	Section 5.6 of [This specification]
Router Advertisement			
6	-1	Willingness TBD	Section 5.7 of [This specification]
6	-3	Attached Networks	Section 5.7 of [This specification]
Endpoint Advertisement			
7	-1	Endpoint List	Section 5.8 of [This specification]

Table 18: SAND Message Type-Specific Parameter Keys

8.4. SAND Convergence Layer Registries

EDITOR NOTE: registries to-be-created upon publication of this specification.

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND CL Common Parameter Keys" and initialize it with the contents of Table 19. The registration procedure is Specification Required.

Specifications of new common parameters need to define the code point (an int16 integer) as well as the CBOR form and meaning of the associated value.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

Code	Name	Reference
-32768 to -32513	reserved for private and experimental type-specific parameters	[This specification]
-32512 to -1	delegated to SAND CL Type-Specific Parameter Keys registry	[This specification]
0	CL Type	Section 5.4.1 of [This specification]
3	Bind Address List	Section 5.4.1 of [This specification]
4	Bind Port Number	Section 5.4.1 of [This specification]
5	Transport Security Required	Section 5.4.1 of [This specification]
6	Role	Section 5.4.1 of [This specification]
7 to 32511	unassigned	
32512 to 32767	reserved for private and experimental common parameters	[This specification]

Table 19: SAND CL Common Parameter Keys

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND CL Types" and initialize it with the contents of Table 20. For positive code points the registration procedure is Specification Required. Negative code points are reserved for use on private networks for functions not published to the IANA.

Specifications of new CL types need to define the CL Type value (an int16 integer), as well as the other CL parameters required and allowed. Specifications need to define how those CBOR parameters are used by a node to transfer bundles to the referred-to CL.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

Code	Name	Reference
-32768 to -1	Private/Experimental Use	[This specification]
0	reserved	[This specification]
1	TCPCLv4	Section 5.4.1.1 of [This specification]
2	UDPCLv2	Section 5.4.1.2 of [This specification]
3	CCSDS LTPCL Over UDP	Section 5.4.1.3 of [This specification]
4 to 32765	unassigned	
32766	TCPCLv3	Section 5.4.1.4 of [This specification]
32767	RFC 7122 UDPCL	Section 5.4.1.5 of [This specification]

Table 20: SAND CL Types

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND CL Type-Specific Parameter Keys" and initialize it with the contents of Table 21. The registration procedure is Specification Required.

Specifications of new common parameters need to define the associated CL type, code point (an int16 integer), and the CBOR form and meaning of the associated value.

CL Type	Code	Name	Reference
TCPCLv4			
1	-1	Message Type Support	Section 5.4.1.1 of [This specification]
1	-2	Session Extension Type Support	Section 5.4.1.1 of [This specification]
1	-3	Transfer Extension Type Support	Section 5.4.1.1 of [This specification]
UDPCLv2			
2	-1	Extension Support	Section 5.4.1.2 of [This specification]
CCSDS LTPCL Over UDP			
3	-1	Engine ID	Section 5.4.1.3 of [This specification]
3	-2	Extension Support	Section 5.4.1.3 of [This specification]

Table 21: SAND CL Type-Specific Parameter Keys

8.5. SAND Local Topology Registries

EDITOR NOTE: registries to-be-created upon publication of this specification.

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND Neighbor Parameter Keys" and initialize it with the contents of Table 22. The registration procedure is Specification Required.

Specifications of new peer parameters need to define the code point (an int16 integer) as well as the CBOR form and meaning of the associated value. Specifications need to define how those CBOR parameters are used by a node to relate the encoded message to the node’s information bases.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

// Update this table and others in the section

Code	Name	Reference
-32768 to -1	Private/Experimental Use	[This specification]
0	Node ID	Section 5.6.1 of [This specification]
1	MPR Selection	Section 5.6.1 of [This specification]
2	Reachability	Section 5.6.1 of [This specification]
3	Routing Metrics	Section 5.6.1 of [This specification]

Table 22: SAND Neighbor Parameter Keys

Code	Name	Reference
-32768 to -32513	reserved for private and experimental type-specific parameters	[This specification]
-32512 to -1	delegated to SAND Routing Metrics Type-Specific Parameter Keys registry	[This specification]
0	Routing Type	Section 5.6.2 of [This specification]
1	Direction	Section 5.6.2 of [This specification]
2	Validity Schedule	Section 5.6.2 of [This specification]
3 to 32767	unassigned	

Table 23: SAND Rouging Metrics Parameter Keys

Code	Name	Reference
-32768 to -1	Private/Experimental Use	[This specification]
0	Reserved	[This specification]
1	SABR	Section 5.6.2.1 of [This specification]
2 to 32767	unassigned	

Table 24: SAND Routing Types

Routing Type	Code	Name	Reference
SABR			
1	-1	Maximum Data Rate	Section 5.6.2.1 of [This specification]
1	-2	Delay	Section 5.6.2.1 of [This specification]
1	-3	Bit Error Rate	Section 5.6.2.1 of [This specification]

Table 25: SAND Routing Metrics Type-Specific Parameter Keys

8.6. SAND Endpoint Parameter Keys

EDITOR NOTE: registry to-be-created upon publication of this specification.

IANA will create, under the "Bundle Protocol Secure Advertisement and Neighborhood Discovery (SAND)" registry group [IANA-BPSAND], a registry titled "SAND Endpoint Parameter Keys" and initialize it with the contents of Table 26. The registration procedure is Specification Required.

Specifications of new peer parameters need to define the code point (an int16 integer) as well as the CBOR form and meaning of the associated value. Specifications need to define how those CBOR parameters are used by a node to relate the encoded message to the node’s information bases.

Expert(s) are encouraged to be biased towards approving registrations unless they are abusive, frivolous, or actively harmful (not merely aesthetically displeasing, or architecturally dubious).

Code	Name	Reference
-32768 to -1	Private/Experimental Use	[This specification]
0	EID Pattern	Section 5.8 of [This specification]
5	Payload Security Required	Section 5.8 of [This specification]
6 to 32767	unassigned	

Table 26: SAND Peer Parameter Keys

9. References

9.1. Normative References

[IANA-BP] IANA, "Bundle Protocol", <https://www.iana.org/assignments/bundle/>.

[IANA-BPSAND] IANA, "Bundle Protocol (BP) Secure Advertisement and Neighborhood Discovery (SAND)", <https://www.iana.org/assignments/bp-sand/>.

[IANA-LTP] IANA, "Licklider Transmission Protocol (LTP) Parameters", <https://www.iana.org/assignments/ltp-parameters/>.

[IANA-URI] IANA, "Uniform Resource Identifier (URI) Schemes", <https://www.iana.org/assignments/uri-schemes/>.

[CCSDS-BPv7] Consultative Committee for Space Data Systems, "CCSDS Bundle Protocol Specification", CCSDS 734.20-O-1, April 2025, <https://public.ccsds.org/Pubs/734x20o1.pdf>.

[CCSDS-SABR] Consultative Committee for Space Data Systems, "Schedule-Aware Bundle Routing", CCSDS 734.3-B-1, July 2019, <https://public.ccsds.org/Pubs/734x3b1.pdf>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[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/info/rfc2119>.

[RFC7122] Kruse, H., Jero, S., and S. Ostermann, "Datagram Convergence Layers for the Delay- and Disruption-Tolerant Networking (DTN) Bundle Protocol and Licklider Transmission Protocol (LTP)", RFC 7122, DOI 10.17487/RFC7122, March 2014, <https://www.rfc-editor.org/info/rfc7122>.

[RFC7242] Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol", RFC 7242, DOI 10.17487/RFC7242, June 2014, <https://www.rfc-editor.org/info/rfc7242>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[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/info/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/info/rfc9172>.

[RFC9174] Sipos, B., Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4", RFC 9174, DOI 10.17487/RFC9174, January 2022, <https://www.rfc-editor.org/info/rfc9174>.

[RFC9360] Schaad, J., "CBOR Object Signing and Encryption (COSE): Header Parameters for Carrying and Referencing X.509 Certificates", RFC 9360, DOI 10.17487/RFC9360, February 2023, <https://www.rfc-editor.org/info/rfc9360>.

[I-D.ietf-cose-cbor-encoded-cert] Mattsson, J. P., Selander, G., Raza, S., H旦glund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-14, 23 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-14>.

[I-D.ietf-dtn-bpsec-cose] Sipos, B., "Bundle Protocol Security (BPSec) COSE Context", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-cose-08, 3 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpsec-cose-08>.

[I-D.ietf-dtn-udpcl] Sipos, B. and J. Deaton, "Delay-Tolerant Networking UDP Convergence Layer Protocol Version 2", Work in Progress, Internet-Draft, draft-ietf-dtn-udpcl-01, 12 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-udpcl-01>.

[I-D.ietf-dtn-eid-pattern] Sipos, B., "Bundle Protocol Endpoint ID Patterns", Work in Progress, Internet-Draft, draft-ietf-dtn-eid-pattern-02, 13 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-eid-pattern-02>.

9.2. Informative References

[RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <https://www.rfc-editor.org/info/rfc1256>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[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/info/rfc4838>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5326] Ramadas, M., Burleigh, S., and S. Farrell, "Licklider Transmission Protocol - Specification", RFC 5326, DOI 10.17487/RFC5326, September 2008, <https://www.rfc-editor.org/info/rfc5326>.

[RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, DOI 10.17487/RFC5444, February 2009, <https://www.rfc-editor.org/info/rfc5444>.

[RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, DOI 10.17487/RFC6130, April 2011, <https://www.rfc-editor.org/info/rfc6130>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[RFC8175] Ratliff, S., Jury, S., Satterwhite, D., Taylor, R., and B. Berry, "Dynamic Link Exchange Protocol (DLEP)", RFC 8175, DOI 10.17487/RFC8175, June 2017, <https://www.rfc-editor.org/info/rfc8175>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[RFC9164] Richardson, M. and C. Bormann, "Concise Binary Object Representation (CBOR) Tags for IPv4 and IPv6 Addresses and Prefixes", RFC 9164, DOI 10.17487/RFC9164, December 2021, <https://www.rfc-editor.org/info/rfc9164>.

[I-D.ietf-tvr-requirements] King, D., Contreras, L. M., Sipos, B., and L. Zhang, "TVR (Time-Variant Routing) Requirements", Work in Progress, Internet-Draft, draft-ietf-tvr-requirements-05, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tvr-requirements-05>.

[I-D.irtf-dtnrg-ipnd] Ellard, D., Altmann, R., Gladd, A., in ’t Velt, R., and D. Brown, "DTN IP Neighbor Discovery (IPND)", Work in Progress, Internet-Draft, draft-irtf-dtnrg-ipnd-03, 10 November 2015, <https://datatracker.ietf.org/doc/html/draft-irtf-dtnrg-ipnd-03>.

[I-D.sipos-dtn-edge-zeroconf] Sipos, B., "Lightweight Bundle Protocol Edge Node with Zero-Configuration and Zero-State", Work in Progress, Internet-Draft, draft-sipos-dtn-edge-zeroconf-01, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-sipos-dtn-edge-zeroconf-01>.

[github-dtn-demo-agent]
Sipos, B., "BP SAND Example Implementation",
<https://github.com/BrianSipos/dtn-demo-agent/>.

Acknowledgments

Much pre-draft review was performed to make the document clear and readable by Sarah Heiner of JHU/APL.

Implementation Status

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

[NOTE to the RFC Editor: please remove this section before publication, as well as the reference to [RFC7942] and [github-dtn-demo-agent].]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations can exist.

An example implementation of the this draft of SAND has been created as a GitHub project [github-dtn-demo-agent] and is intended to use as a proof-of-concept and as a possible source of interoperability testing. This example implementation uses D-Bus as the CL-BP Agent interface, so it only runs on hosts which provide the Python "dbus" library.

Authors' Addresses

Brian Sipos
The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Rd.
Laurel, MD 20723
United States of America
Email: brian.sipos+ietf@gmail.com

Joshua Deaton
Science Applications International Corporation
Email: joshua.e.deaton@nasa.gov