Delay/Disruption Tolerant Networking
Internet-Draft
Intended status: Experimental
Expires: 16 September 2026

R. Taylor
Aalyria Technologies

15 March 2026

DTN Peering Protocol
draft-taylor-dtn-dpp-00

Abstract

This document specifies the DTN Peering Protocol (DPP), an inter-domain routing protocol for the Delay-Tolerant Networking (DTN) ecosystem. DPP facilitates the exchange of reachability information between distinct Administrative Domains (ADs), enabling inter-domain routing across the Solar System DTN. DPP separates the control plane from the data plane: a DPP speaker need not be a gateway that forwards bundles, allowing centralized route controllers or orchestration systems to participate in peering on behalf of the gateways they manage.

DPP harmonizes the two DTN addressing schemes -- ipn (integer-based) and dtn (URI-based) -- into a unified routing framework. It leverages DNS for identity verification and supports both reactive routing and scheduled contact windows for deep-space networks.

About This Document

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

The latest revision of this draft can be found at https://ricktaylor.github.io/dpp/draft-taylor-dtn-dpp.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-taylor-dtn-dpp/.

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

Source for this draft and an issue tracker can be found at https://github.com/ricktaylor/dpp.

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

The DTN Peering Protocol (DPP) is designed for the exchange of reachability information between distinct Administrative Domains in Delay-Tolerant Networks. It functions as the inter-domain routing protocol for the Solar System DTN, analogous to BGP's role in the Internet.

1.1. Motivation

Two DPP speakers establish a session to exchange reachability information across administrative boundaries. Common scenarios include:

Inter-Agency Connectivity: NASA's Deep Space Network and ESA's ESTRACK operate independent DTN infrastructures. DPP allows their border gateways to advertise which spacecraft and ground stations are reachable through each network, enabling bundles to flow between agencies.

Commercial-Government Integration: A commercial satellite operator peers with a government science network. DPP enables the commercial network to advertise transit capacity to deep-space assets, while the science network advertises its spacecraft endpoints.

Redundant Path Discovery: A Mars relay satellite is reachable via two ground station complexes. DPP allows both paths to be advertised with appropriate metrics, enabling the network to prefer one path while maintaining the other as backup.

Scheduled Contact Windows: Deep-space links are available only during planned contact windows. DPP's time-variant routing attributes allow DPP speakers to advertise future connectivity, enabling proactive bundle scheduling.

Scalable Contact Graph Routing: Contact Graph Routing (CGR)

algorithms compute optimal bundle forwarding based on predicted contact schedules. However, CGR complexity grows with network size, and a single global contact graph spanning all agencies, spacecraft, and ground stations is neither practical nor desirable -- each organization maintains its own mission planning systems and contact schedules. DPP provides the inter-domain routing layer that connects independent CGR domains. Each Administrative Domain runs its own CGR internally, while DPP speakers advertise summarized reachability (with optional time windows) to peers. This decouples internal scheduling complexity from external routing, allowing the Solar System DTN to scale without requiring a single coordinated contact plan.

Without DPP, each network would require manual configuration of routes to external destinations -- an approach that does not scale as the Solar System DTN grows.

1.2. Design Principles

Transport Agnostic: Defined via Protocol Buffers (gRPC) to run over any reliable transport (TCP/IP, QUIC).

Unified Routing: Harmonizes ipn and dtn schemes into a single routing table (FIB).

DNS-Rooted Identity: Uses standard DNS domains as Administrative Domain identifiers.

Decoupled Trust: Separates "Who I am" (AD Identity) from "What I route" (Allocators/Names).

1.3. Scope

This specification defines the protocol messages, state machine, and operational requirements for DPP implementations. It does not define:

* Internal routing within an Administrative Domain
* Bundle forwarding algorithms (e.g., CGR)
* Transport layer security beyond identity verification
* Policy languages for route filtering

2. Conventions and 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.

2.1. Definitions

Administrative Domain (AD): An autonomous unit of routing control identified by a DNS domain name. Examples include agency networks (e.g., dsn.example.org), commercial providers, or research institutions.

DPP Speaker: An entity that participates in the DPP protocol, maintaining peering sessions with other DPP speakers and exchanging route information on behalf of an Administrative Domain. A DPP speaker may be a gateway, a centralized route controller, or any other system authorized to advertise and receive routes for its AD.

Gateway: A border node within an Administrative Domain that forwards bundles to other ADs. A gateway may or may not also be a DPP speaker. When a DPP speaker is not itself a gateway, it uses the gateway_eid attribute (see Section 6.7) to direct traffic to the appropriate gateway.

EID Pattern: A pattern matching one or more Bundle Protocol [RFC9171] Endpoint Identifiers, as defined in [I-D.ietf-dtn-eid-pattern].

Forwarding Information Base (FIB): The consolidated routing table used by a gateway to forward bundles.

Metric: A numeric value expressing route preference, analogous to BGP’s Multi-Exit Discriminator (MED).

AD_PATH: An ordered list of Administrative Domains that a route advertisement has traversed, used for loop detection.

2.2. On the Use of DNS

Readers familiar with deep-space networking may question the reliance on DNS, given that DNS is not deployed on spacecraft and requires low-latency connectivity. This concern, while understandable, misses the architectural context:

DPP speakers are ground infrastructure. DPP operates at administrative boundaries. DPP speakers are ground stations, mission control centers, network operations facilities, or centralized route controllers -- all of which have reliable Internet connectivity and can perform DNS lookups in milliseconds. Gateways, by contrast, are the forwarding nodes that handle bundle traffic and may be located anywhere -- including on spacecraft or relay satellites -- since they do not need DNS access or direct participation in DPP.

DNS is used for authentication, not routing. DNS lookups occur only during session establishment (the handshake phase) to retrieve the peer's public key for identity verification. Once a session is established, no further DNS queries are required. Route advertisements and withdrawals flow over the established session without DNS involvement.

Spacecraft do not run DPP. Spacecraft run the Bundle Protocol and may use CGR or static routing for internal forwarding decisions. They receive forwarding instructions from their ground-based mission operations, which participate in DPP on their behalf. The spacecraft need not know about Administrative Domains or DNS. However, spacecraft and relay satellites MAY serve as gateways -- forwarding bundles between Administrative Domains -- without participating in DPP directly. A ground-based DPP speaker advertises routes on their behalf using the gateway_eid attribute to identify the in-space forwarding node.

This architecture mirrors the Internet: BGP routers require IP connectivity to establish sessions, but the end hosts they route traffic for need not understand BGP. Similarly, DPP speakers require DNS access, but the spacecraft and sensors they route bundles for need not understand DPP.

3. Architecture Overview

3.1. Administrative Domains

An Administrative Domain is an autonomous unit of routing control.

Identity: An AD is identified solely by a DNS domain name (e.g., dsn.example.org).

Independence: An AD Identity is distinct from the resources it advertises. An AD named dsn.example.org may advertise ipn:100 and dtn://rover1.example.org.

3.2. DPP Speakers and Gateways

DPP speakers and gateways are defined in Section 2.1. A DPP speaker and a gateway MAY be the same node, but they need not be. For example, a centralized route controller may run DPP sessions on behalf of multiple gateways within its AD, using the gateway_eid attribute to direct traffic to the appropriate forwarding node. This separation of the control plane (DPP) from the data plane (bundle forwarding) allows flexible deployment topologies.

An Administrative Domain MAY deploy multiple DPP speakers, each maintaining independent peering sessions with external ADs. Because loop detection operates on the AD identity (the local_ad_id in the AD_PATH) rather than on individual speaker identity, multiple speakers within the same AD do not introduce routing loops (see Section 5.3).

4. Addressing and Metrics

DPP introduces a Harmonized Specificity Score to allow integer-based ipn patterns and string-based dtn patterns to coexist in a single routing decision process.

4.1. EID Pattern Constraints (Monotonic Specificity)

To ensure efficient FIB lookups and O(1) specificity scoring, DPP restricts the generic EID patterns defined in [I-D.ietf-dtn-eid-pattern] to a Strict Monotonic Subset.

A pattern exhibits monotonic specificity if all wildcard components are confined to the logical "leaves" of the naming hierarchy. A specific child component cannot exist under a wildcard parent.

4.1.1. IPN Scheme Constraints (Left-to-Right Hierarchy)

The ipn scheme hierarchy is defined as Allocator -> Node.

* Wildcards or ranges are only permitted in the Node number if the Allocator is specific.

* Valid: ipn:100.1 (Specific), ipn:100.* (Monotonic).

* Invalid: ipn:*.1, ipn:[100-200].1 (Specific Node under Wildcard Allocator).

4.1.2. DTN Scheme Constraints (Right-to-Left Hierarchy)

The dtn scheme hierarchy follows standard DNS rules: TLD <- Domain <- Host.

* Wildcards are only permitted in the left-most label (the Host or lowest subdomain).
* Valid: dtn://rover1.example.org (Exact), dtn://*.example.org (Wildcard Host), dtn://rover*.example.org (Partial Host).
* Invalid: dtn://rover1.*.example.org (Specific Child under Wildcard Parent).

4.1.3. Single Wildcard Constraint

To support the simplified specificity metric, a pattern MUST contain at most one wildcard character (*).

Complex globs (e.g., dtn://r*v*r.example.org) MUST NOT be used.

4.2. Specificity Scoring

The specificity score prioritizes Exact Matches over Wildcards, and Longer Prefixes over Shorter Prefixes.

The specificity score is computed locally by the receiver from the EidPattern. It is NOT transmitted in route advertisements, as it can be deterministically derived from the pattern itself.

The score is calculated as a uint32:

Score = (IsExact x 256) + LiteralLength

IsExact (0 or 1): 1 if the pattern contains NO wildcards; 0 otherwise.

Multiplier (256): A weight ensuring any Exact Match outranks any Wildcard Match (assuming max literal length < 256).

LiteralLength: The count of non-wildcard characters (or equivalent bits for integers).

4.2.1. DTN Scheme Scoring

For dtn URIs, LiteralLength is the count of characters in the Authority string excluding the wildcard.

Pattern	Characters	Score
dtn://rover1.example.org	18 (exact)	274
dtn://rover*.example.org	17 (wildcard)	17

Table 1

4.2.2. IPN Scheme Scoring

For ipn EIDs, LiteralLength represents the "Effective Bit Depth" of the pattern. The IPN address space is treated as a 64-bit virtual integer, split logically into a 32-bit Allocator and a 32-bit Node.

LiteralLength = Bits_Allocator + Bits_Node

1. Allocator Part (Bits 63-32):
    * Specific Allocator (e.g., 100): 32 bits.
    * Wildcard Allocator (*): 0 bits.
2. Node Part (Bits 31-0):
    * Specific Node (e.g., .1): 32 bits.
    * Wildcard Node (.*): 0 bits.
    * Range .[min-max]: 32 - ceil(log2(Count)) bits.

IPN Pattern	Classification	IsExact	LiteralLength	Score
ipn:100.1	Specific Node	1	64	320
ipn:100.*	Allocator Wildcard	0	32	32
ipn:100.[10-13]	Node Range (Size 4)	0	62	62
ipn:*	Default Route	0	0	0

Table 2

4.3. The Metric Field

The metric field in route advertisements is analogous to BGP's Multi-Exit Discriminator (MED) [RFC4271]. It allows an Administrative Domain to express a preference for how traffic should enter its network when multiple paths exist.

For example:

Organization B has two DPP speakers peering with Organization A:
- Speaker in Europe (gateway with high latency to Mars relay)
- Speaker in Australia (gateway with low latency to Mars relay)

Organization B advertises routes to ipn:200.* via both speakers:
- Europe speaker: metric = 100
- Australia speaker: metric = 10

Organization A, having learned the same route from both peers, prefers the lower metric and routes traffic via Australia.

Property	Description
Direction	Controls incoming traffic (traffic TO your AD)
Scope	Only compared between routes from the SAME origin AD
Default	Lower metric = more preferred
Use Cases	Capacity balancing, latency optimization, cost management

Table 3

The metric is set by the originating AD and propagated unchanged through intermediate ADs. It provides a policy knob for traffic engineering without requiring coordination between ADs.

4.4. Route Selection Algorithm

When a DPP speaker receives multiple routes to the same destination, it MUST select the Best Path using the following tie-breaking order:

1. Highest Specificity Score: Prefer rover1 over rover*. (Computed locally from the EidPattern.)

2. Shortest AD_PATH: Prefer 1 hop over 3 hops.

3. Lowest Metric: Administrator preference, analogous to BGP MED.

4. Oldest Route: Stability preference.

A DPP speaker MAY implement additional local policy that overrides this order, but such policy is outside the scope of this specification.

5. Trust and Verification Model

DPP leverages DNS as the root of trust for Administrative Domain identities, utilizing a "Check-and-Challenge" mechanism.

5.1. The _dtn_domain SVCB Record

To prove ownership of an AD Identity (e.g., dsn.example.org), the administrator MUST publish one or more SVCB records [RFC9460] at _dtn_domain.<AD-Domain>.

The _dtn_domain prefix is chosen to reserve the broader _dtn namespace for future transport-specific uses (e.g., convergence layer discovery).

Record format:

_dtn_domain.<AD-Domain>. IN SVCB <priority> <target> <params>

Required SvcParams:

Parameter	Key ID	Description
dtn-alg	TBD	Signature algorithm (e.g., ed25519)
dtn-pubkey	TBD	Base64-encoded public key

Table 4

Optional SvcParams:

Parameter	Key ID	Description
alpn	1	Application protocols (e.g., dpp)
port	3	DPP service port
ipv4hint	4	IPv4 address hints
ipv6hint	6	IPv6 address hints

Table 5

Example:

```
_dtn_domain.dsn.example.org.  IN  SVCB  1 . (
    dtn-alg=ed25519
    dtn-pubkey=MCowBQYDK2VwAyEAGb9... )
```

An SVCB record with target "." indicates the service is available at the AD domain itself. Alternatively, the target MAY specify a different hostname:

```
_dtn_domain.dsn.example.org.  IN  SVCB  1 gateway.dsn.example.org. (
    dtn-alg=ed25519
    dtn-pubkey=MCowBQYDK2VwAyEAGb9...
    port=4556 )
```

Implementations MUST support the ed25519 algorithm. Implementations MAY support additional algorithms.

5.1.1. Multiple DPP Speakers per AD

An Administrative Domain that deploys multiple DPP speakers has two options for key management:

Shared key: All speakers share a single key pair. The AD publishes one SVCB record and distributes the private key to each speaker. This is operationally simple but means that compromise of any single speaker exposes the key for the entire AD.

Per-speaker keys: Each speaker has its own key pair. The AD publishes multiple SVCB records at _dtn_domain.<AD-Domain>, each with a distinct dtn-pubkey. During verification, the Responder MUST attempt signature verification against each public key in the record set until one succeeds or all have been tried.

Example with per-speaker keys:

```
; Speaker at Goldstone
_dtn_domain.dsn.example.org.  IN  SVCB  1 gds.dsn.example.org. (
    dtn-alg=ed25519
    dtn-pubkey=MCowBQYDK2VwAyEAGb9...
    port=4556 )

; Speaker at Canberra
_dtn_domain.dsn.example.org.  IN  SVCB  2 cbr.dsn.example.org. (
    dtn-alg=ed25519
    dtn-pubkey=MCowBQYDK2VwAyEAdQ7...
    port=4556 )
```

Both approaches are valid. Per-speaker keys provide better operational security -- a compromised speaker can be revoked by removing its SVCB record without affecting other speakers in the AD. The choice between shared and per-speaker keys is a local operational decision and is transparent to the peering partner.

5.1.2. Alignment with CL Discovery

This approach aligns with [I-D.ek-dtn-ipn-arpa] which uses DNS for convergence layer endpoint discovery. The _dtn_domain prefix is used for AD identity verification, leaving the broader _dtn namespace available for transport-specific service discovery in future specifications.

5.1.3. Alternative Approaches (Working Group Note)

This subsection documents alternative DNS-based identity verification approaches considered during protocol design. It is intended for Working Group discussion and will be removed before publication.

Alternative A: TXT Record

A simpler approach using TXT records, following the pattern established by DKIM [RFC6376] and DMARC [RFC7489]:

```
_dtn_id.dsn.example.org.  IN  TXT  (
    "v=dtn1; a=ed25519; k=MCowBQYDK2VwAyEAGb9..." )
```

Pros	Cons
Simple, widely deployed	Less structured, requires parsing
Proven pattern (DKIM, DMARC, ACME)	Cannot combine identity with service discovery
No IANA registration needed for record format	Separate records needed for CL endpoints

Table 6

Alternative B: Namespaced TXT Record

Using a _dtn namespace with sub-labels for different purposes:

```
_dtn._id.dsn.example.org.  IN  TXT  (
    "v=1 a=ed25519 k=MCowBQYDK2VwAyEAGb9..." )
_dtn._bundle._tcp.dsn.example.org.  IN  SRV  (
    1 1 4556 gateway.dsn.example.org. )
```

Pros	Cons
Separates identity from service discovery	Multiple lookups required
Uses established SRV pattern for CL	More complex DNS configuration
Compatible with existing resolver behavior	Namespace fragmentation

Table 7

Rationale for SVCB:

The authors selected SVCB (Alternative C) because:

1. Unified record: Single DNS lookup provides both identity verification and optional service discovery parameters.

2. Extensibility: SVCB’s typed parameter model allows future extensions without format changes.

3. Alignment: Consistent with the direction of [I-D.ek-dtn-ipn-arpa] and modern DNS service discovery patterns.

4. Structured data: Native support for typed parameters avoids ad-hoc parsing of TXT record values.

The authors invite feedback on this design choice.

5.2. Handshake and Verification Flow

When two DPP speakers peer, they MUST perform the following cryptographic handshake:

1. Hello: The Initiator MUST send a Hello message containing its local_ad_id.

2. Lookup: The Responder MUST query DNS for SVCB records at _dtn_domain.<local_ad_id> to fetch the Initiator's Public Key(s) (from the dtn-pubkey parameter of each record). If the DNS lookup fails or no valid records are found, the Responder MUST reject the session.

3. Challenge: The Responder MUST generate a cryptographically random nonce (minimum 16 bytes) and send a HelloChallenge message.

4. Response: The Initiator MUST sign the nonce with its Private Key and send a HelloResponse message.

5. Verification: The Responder MUST verify the signature against the Public Key(s) from DNS. If multiple SVCB records were returned, the Responder MUST attempt verification against each dtn-pubkey until one succeeds. If any key produces a valid signature, the session is ESTABLISHED. If no key produces a valid signature, the Responder MUST reject the session and SHOULD send a Notification with an appropriate error code.

5.3. Loop Detection

Routing loops are prevented using the AD_PATH attribute (similar to BGP's AS_PATH [RFC4271]). Loop detection operates on Administrative Domain identity, not on individual DPP speaker identity. This means an AD MAY deploy multiple speakers without risk of routing loops, as all speakers within an AD share the same local_ad_id.

Mechanism: Every route advertisement MUST include the ordered list of ADs it has traversed.

Propagation: When a DPP speaker re-advertises a route to a peer, it

MUST prepend its own local_ad_id to the AD_PATH.

Detection: If a DPP speaker receives a route where AD_PATH contains its own local_ad_id, the route MUST be silently discarded.

6. Route Attributes and Advertisements

DPP uses an extensible attribute model inspired by BGP path attributes [RFC4271]. This allows new route properties to be defined in future protocol versions without breaking existing implementations.

6.1. Advertisement Structure

A RouteAdvertisement follows the BGP UPDATE model:

Patterns: One or more EidPattern destinations (equivalent to NLRI)

Path attributes: Properties shared by all destinations (ad_path, metric)

Extensible attributes: Optional properties (time window, link characteristics, next-hop)

6.2. Path Attributes

Field	Description
patterns	EID patterns for destination matching
ad_path	List of AD domains traversed (for loop detection and path length)
metric	Administrator preference (lower = preferred, analogous to BGP MED)

Table 8

6.3. Extensible Attributes

Optional attributes shared by all destinations:

Attribute	Description
gateway_eid	Bundle forwarding endpoint (see Section 6.7)
valid_from	When routes become active (if omitted: immediate)
valid_until	When routes expire (if omitted: until withdrawn)
bandwidth_bps	Link rate in bits per second
max_bundle_size	Maximum bundle size in bytes

Table 9

6.4. Forward Compatibility

Implementations MUST handle unknown attributes as follows:

1. If transitive = true: Preserve and propagate the attribute unchanged.

2. If transitive = false: Silently discard the attribute.

This allows intermediate DPP speakers to forward routes with attributes they do not understand, enabling incremental deployment of new attribute types.

6.5. Time-Variant Routing

The valid_from and valid_until attributes enable scheduled contacts for deep-space networks ([CCSDS-SABRE]). Routes without these attributes are valid immediately and remain valid until explicitly withdrawn.

A peer MAY send multiple advertisements with non-overlapping time windows to describe a series of scheduled contacts.

6.6. Route Withdrawal

A RouteWithdrawal cancels previously advertised routes. For scheduled routes, the valid_from field identifies which advertisement is being withdrawn. If no matching route exists, the withdrawal is silently ignored.

6.7. Gateway EID Resolution

The gateway_eid attribute specifies the endpoint where bundles matching the advertised patterns should be forwarded. This enables separation of the control plane (DPP session) from the data plane (bundle forwarding).

Resolution rules:

1. If a gateway_eid attribute is present, use it as the forwarding destination.

2. If gateway_eid is absent, derive the gateway EID from the peer's AD identity (local_ad_id from Hello) as follows:

   * If the AD identity is already a valid EID (e.g., dtn://gateway.example.org), use it directly.

   * If the AD identity is a DNS domain name, convert it to a DTN EID:

     <dns_name> -> dtn://<dns_name>/

     For example: dsn.example.org -> dtn://dsn.example.org/

   * Alternatively, implementations MAY query DNS to discover an IPN EID mapping for the domain. See [I-D.ek-dtn-ipn-arpa] for the ongoing work on IPN URI scheme DNS resolution.

This design allows flexibility in deployment while providing sensible defaults:

DNS-based AD: An AD with identity dsn.example.org automatically maps to dtn://dsn.example.org/ as the default gateway. No explicit gateway_eid is required.

Explicit EID AD: A DPP speaker with AD identity dtn://gateway.dsn.example.org uses that EID directly.

Separated control plane: A DPP speaker that is not itself a gateway (e.g., a centralized route controller) can advertise routes with explicit gateway_eid pointing to the forwarding nodes within the AD, enabling load distribution or failover.

Load distribution: Multiple routes to the same destination can be advertised with different gateway_eid values and metrics.

Examples:

```
# AD identity is DNS name - gateway_eid derived automatically
AD Identity: dsn.example.org
Default Gateway EID: dtn://dsn.example.org/

RouteAdvertisement {
  patterns: [ipn:100.*]
  ad_path: ["dsn.example.org"]
  metric: 10
  // No gateway_eid needed - defaults to dtn://dsn.example.org/
}

# Explicit gateway_eid for control/data plane separation
AD Identity: dsn.example.org

RouteAdvertisement {
  patterns: [ipn:100.*]
  ad_path: ["dsn.example.org"]
  metric: 10
  attributes: [
    gateway_eid: "dtn://ingress1.dsn.example.org"
  ]
}
```

7. Protocol Definition

The protocol uses a single bi-directional gRPC stream for the entire lifecycle of the session.

7.1. Service Definition

```
syntax = "proto3";
package dtn.peering.v1;

import "google/protobuf/timestamp.proto";

service DtnPeering {
  // Single stream for Handshake, Auth, Routes, and Keepalives
  rpc Peer(stream PeerMessage) returns (stream PeerMessage);
}
```

7.2. Message Envelope

```
message PeerMessage {
  uint64 sequence_number = 1;
  oneof payload {
    Hello hello = 10;
    HelloChallenge challenge = 11;
    HelloResponse response = 12;
    KeepAlive keep_alive = 20;
    RouteUpdate update = 21;
    Notification notification = 30;
  }
}
```

7.3. Identity and Authentication Messages

```
message Hello {
  string local_ad_id = 1;        // DNS Domain (e.g., "dsn.example.org")
  string speaker_node_id = 2;    // Debug ID (e.g., "dtn://speaker-alpha")
  uint32 hold_time_seconds = 3;
}

message HelloChallenge { bytes nonce = 1; }
message HelloResponse { bytes signature = 1; }
```

7.4. Routing Messages

```
   message RouteUpdate {
     repeated RouteAdvertisement announcements = 1;
     repeated RouteWithdrawal withdrawals = 2;
   }

   // Route advertisement with extensible attributes
   // Next-hop is determined per Gateway EID Resolution
   // (explicit attribute or derived from AD identity)
   // All fields except 'patterns' are path attributes
   // shared by all destinations
   message RouteAdvertisement {
     // Destinations (NLRI equivalent)
     repeated EidPattern patterns = 1;

     // Path attributes (shared by all destinations)
     repeated string ad_path = 2;  // List of AD Domains for loop detection
     uint32 metric = 3;            // Administrator preference (lower = preferred)
     // Note: specificity_score is NOT transmitted;
     // receiver computes it from pattern

     // Extensible attributes
     repeated RouteAttribute attributes = 10;
   }

   message RouteWithdrawal {
     repeated EidPattern patterns = 1;         // Destinations to withdraw
     // For scheduled routes, valid_from identifies which advertisement
     // to withdraw
     google.protobuf.Timestamp valid_from = 2; // Optional: identifies
                                               // scheduled route
   }
```

7.5. Route Attributes

```
// Extensible route attribute (BGP-style path attributes)
// New attribute types can be added in future protocol versions
message RouteAttribute {
  oneof attribute {
    // Gateway for data plane separation
    string gateway_eid = 1;                      // Bundle forwarding
                                                 // endpoint

    // Time window attributes
    google.protobuf.Timestamp valid_from = 2;    // When routes become
                                                 // active
    google.protobuf.Timestamp valid_until = 3;   // When routes expire

    // Link characteristics
    uint64 bandwidth_bps = 4;          // Link rate (bits per second)
    uint32 max_bundle_size = 5;        // Maximum bundle size (bytes)

    // Forward compatibility: unknown attributes are preserved
    // and propagated
    UnknownAttribute unknown = 255;
  }
}

// For forward compatibility with future attribute types
message UnknownAttribute {
  uint32 type_id = 1;      // Identifies the attribute type
  bytes value = 2;         // Opaque value
  bool transitive = 3;     // If true, propagate even if not understood
}
```

7.6. Data Types

```
message EidPattern {
  oneof scheme {
    IpnPattern ipn = 1;
    DtnPattern dtn = 2;
  }
}

message IpnPattern {
  uint32 allocator_id = 1;
  uint32 node_id = 2;
  bool is_wildcard = 3;
}

message DtnPattern {
  string authority_string = 1; // e.g. "rover*.example.org"
  bool is_wildcard = 2;
}
```

7.7. Control Messages

```
message KeepAlive {}

message Notification {
  enum Level { INFO=0; WARNING=1; ERROR=2; }
  Level level = 1;
  string message = 2;
  int32 code = 3;
}
```

8. Operational State Machine

A conformant DPP speaker MUST implement the following connection state machine.

8.1. States

State	Description
CONNECTING	Transport (TCP/TLS) established. Speaker MUST send a Hello message.
HANDSHAKE_SENT	Hello sent. Speaker MUST wait for HelloChallenge.
CHALLENGE_WAIT	Waiting for HelloChallenge. Upon receipt, Speaker MUST sign nonce and send HelloResponse.

RESPONSE_SENT	Signed nonce sent. Speaker MUST wait for verification acknowledgment or RouteUpdate.
ESTABLISHED	Valid signature verified. Speaker MAY send and receive RouteUpdate messages.
FAILED	Error occurred. Speaker MUST send Notification (if possible) and close stream.

Table 10

8.2. Keepalive Requirements

In the ESTABLISHED state, a DPP speaker MUST send KeepAlive messages at intervals not exceeding hold_time_seconds / 3.

8.3. Error Handling

Any error condition (DNS lookup failure, invalid signature, hold timer expiry, transport error) MUST cause the DPP speaker to:

1. Send a Notification message with an appropriate error code (if possible).

2. Close the stream.

3. Log the failure reason (SHOULD).

9. Security Considerations

9.1. Identity Verification

DPP relies on DNS for identity verification. The security of this mechanism depends on:

DNSSEC: Deployments SHOULD use DNSSEC to protect against DNS spoofing attacks. Without DNSSEC, an attacker who can manipulate DNS responses could impersonate an Administrative Domain.

Private Key Protection: The private key corresponding to the public key published in the _dtn_domain SVCB record MUST be protected. Compromise of this key allows impersonation of the Administrative Domain.

9.2. Transport Security

This specification does not mandate transport layer security beyond identity verification. However:

* Implementations SHOULD use TLS 1.3 or later for transport encryption.

* Implementations SHOULD verify the peer's TLS certificate matches the claimed AD identity.

9.3. Route Injection Attacks

A malicious or compromised DPP speaker could advertise routes for destinations it does not legitimately serve. Mitigations include:

Out-of-band verification: Operators SHOULD verify route advertisements through external channels before accepting routes to critical destinations.

Route filtering: Implementations SHOULD support policy-based route filtering to reject unexpected advertisements.

Anomaly detection: Operators SHOULD monitor for unexpected route changes.

9.4. Denial of Service

DPP speakers are potentially vulnerable to denial of service attacks:

Connection exhaustion: Implementations SHOULD limit the number of concurrent sessions and implement rate limiting for new connections.

Route table exhaustion: Implementations SHOULD limit the number of routes accepted from a single peer.

Keepalive flooding: Implementations SHOULD rate-limit incoming messages.

9.5. Privacy Considerations

Route advertisements reveal network topology and reachability information. Operators should consider whether this information is sensitive before establishing peering relationships.

10. IANA Considerations

10.1. SVCB SvcParamKeys

This document requests IANA registration of the following entries in the "Service Parameter Keys (SvcParamKeys)" registry:

Number	Name	Meaning	Reference
TBD	dtn-alg	DTN identity signature algorithm	This document
TBD	dtn-pubkey	DTN identity public key (base64)	This document

Table 11

10.2. Underscore-Prefixed DNS Name

This document requests registration of _dtn_domain in the "Underscored and Globally Scoped DNS Node Names" registry per [RFC8552]:

RR Type	_NODE NAME	Reference
SVCB	_dtn_domain	This document

Table 12

10.3. Future Considerations

Future versions may additionally request:

* Registration of a well-known port for DPP (if not using existing gRPC conventions)

* Establishment of a registry for DPP route attribute types

* Registration of dpp in the ALPN Protocol IDs registry

11. References

11.1. Normative References

[I-D.ietf-dtn-eid-pattern]
Sipos, B., "Bundle Protocol Endpoint ID Patterns", Work in Progress, Internet-Draft, draft-ietf-dtn-eid-pattern-06, 16 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-eid-pattern-06>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/rfc/rfc8552>.

[RFC9171] Burleigh, S., Fall, K., and E. Birrane, III, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <https://www.rfc-editor.org/rfc/rfc9171>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/rfc/rfc9460>.

11.2. Informative References

[CCSDS-SABRE]
Consultative Committee for Space Data Systems, "Schedule-Aware Bundle Routing", n.d., <https://public.ccsds.org/>.

[I-D.ek-dtn-ipn-arpa]
Kline, E., "The ipn.arpa Zone and IPN DNS Operations", Work in Progress, Internet-Draft, draft-ek-dtn-ipn-arpa-00, 4 November 2025, <https://datatracker.ietf.org/doc/html/draft-ek-dtn-ipn-arpa-00>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/rfc/rfc4271>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/rfc/rfc6376>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <https://www.rfc-editor.org/rfc/rfc7489>.

Acknowledgments

This document was developed as part of the Hardy BPv7 DTN Router project.

Author’s Address

Rick Taylor
Aalyria Technologies
Email: rtaylor@aalyria.com