Delay/Disruption Tolerant Networking
Internet-Draft
Intended status: Standards Track
Expires: 21 August 2026

R. Taylor
Aalyria Technologies

17 February 2026

BPv7 Echo Service
draft-taylor-dtn-echo-service-00

Abstract

This document specifies an echo service for Bundle Protocol Version 7 (BPv7) networks. The echo service receives bundles at a well-known endpoint and reflects them back to the originator, making only the minimal changes necessary to route the response. This enables round-trip time measurement and end-to-end connectivity verification in Delay-Tolerant Networks. This document requests IANA allocation of a well-known IPN service number and DTN demux for the echo service.

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/echo-service/draft-taylor-dtn-echo-service.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-taylor-dtn-echo-service/.

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/echo-service.

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/.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions and Terminology 3
3. Echo Service Specification 4
  3.1. Service Endpoint 4
  3.2. Bundle Processing 4
    3.2.1. Primary Block 5
    3.2.2. Payload Block 5
    3.2.3. Extension Blocks 6
    3.2.4. Status Reports 7
    3.2.5. Security Blocks 7
  3.3. Client Considerations 7
    3.3.1. Session Disambiguation 8
    3.3.2. Bundle Integrity 8
    3.3.3. Fragmentation 8
4. Security Considerations 8
  4.1. Amplification Attacks 9
  4.2. Information Disclosure 9
  4.3. Resource Exhaustion 9
5. IANA Considerations 10
  5.1. Well-Known Service Registration 10
6. References 10
  6.1. Normative References 10
  6.2. Informative References 11
Appendix A. Ping Clients 11
  A.1. Round-Trip Time Calculation 11

A.2. Endpoint Selection 12
A.3. Payload Format 12
A.4. Statistics 13
Author's Address 13

1. Introduction

Delay-Tolerant Networks (DTNs) present unique challenges for network diagnostics. Unlike traditional IP networks where ICMP Echo (ping) provides immediate feedback, DTN bundles might traverse store-and-forward paths with significant delays. Nevertheless, the ability to verify end-to-end connectivity and measure round-trip time remains essential for network operators.

This document specifies an echo service for Bundle Protocol Version 7 [RFC9171]. The service operates as a simple reflector: it receives bundles addressed to its well-known endpoint and returns them to the originator, making only the minimal changes necessary to route the response.

Conceptually, the echo service acts as a loopback within the node. Upon receiving a bundle, it clones the bundle and swaps source and destination endpoint identifiers, then submits this response bundle to the local BPA for transmission. The BPA processes the response bundle as it would any outbound bundle, applying standard extension block handling, routing, and forwarding. This design means the echo service itself performs minimal processing, delegating most bundle handling to the BPA.

The echo service is intentionally simple: it performs only bundle reflection with no payload interpretation or special processing. This simplicity ensures maximum interoperability, as any conformant echo service will behave identically regardless of implementation. A standardized echo service enables the development of diagnostic tools such as ping clients that can operate across heterogeneous DTN deployments.

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.

This document uses terminology from the Bundle Protocol Version 7 specification [RFC9171].

Echo Service: A Bundle Protocol service that reflects received bundles back to their originator.

Reflection: The process of creating a response bundle by swapping source and destination, preserving all other bundle content to the extent possible. The echo service makes only the minimum changes necessary to route the bundle back to the originator.

3. Echo Service Specification

3.1. Service Endpoint

An echo service MUST register to receive bundles at a well-known endpoint identifier. Two endpoint schemes are defined:

IPN Scheme: Service number 128 on any node. For example, ipn:2.128 represents the echo service on node number 2. Implementations MAY also support service number 7 for backwards compatibility with existing deployments; service number 7 is in the Private Use range per [RFC9758] and cannot be reserved.

DTN Scheme: The demux echo on any node. For example, dtn://example.dtn/echo represents the echo service on the node with node-name example.dtn.

An implementation MAY support additional endpoints beyond these well-known values, but MUST support at least one of the well-known endpoints defined above.

3.2. Bundle Processing

The echo service operates conceptually as a loopback: upon receiving a bundle, it clones the bundle with source and destination swapped, then submits the response bundle to the local BPA for transmission. The BPA then processes the response bundle as it would any outbound bundle, applying standard extension block handling, routing, and forwarding.

This model means the echo service itself performs minimal processing - it modifies the primary block and preserves the payload, then delegates all other bundle processing to the BPA. The block-specific behavior described below reflects this division of responsibility.

The echo service MUST NOT generate bundles other than reflections of received bundles. Other services at the node SHOULD NOT generate bundles using the echo service endpoint as source.

3.2.1. Primary Block

The echo service modifies the primary block to route the response back to the originator. Since the primary block is modified, its CRC MUST be recalculated before transmission. The CRC type MUST NOT be changed.

The following specifies the handling of each primary block field:

Source and Destination: The echo service MUST set the destination to the received bundle's source, and MUST set the source to the echo service's endpoint (the received bundle's destination).

Creation Timestamp and Lifetime: The echo service MUST preserve the creation timestamp and lifetime unchanged. Preserving these fields allows the originating client to control the maximum round-trip time for the bundle.

Bundle Processing Control Flags: The echo service MUST preserve the bundle processing control flags unchanged.

Report-to EID: The echo service MUST preserve the report-to EID unchanged. If the original sender requested status reports, this ensures reports about the response bundle are delivered to the same endpoint.

Note: When the creation timestamp is preserved, a theoretical collision is possible if two bundles from different originators arrive with identical creation timestamps. Both reflections would have the same source and creation timestamp, making them indistinguishable. This is extremely rare in practice and represents an acceptable trade-off; diagnostic tools that use the echo service are typically designed to tolerate occasional packet loss.

3.2.2. Payload Block

The payload block is central to the echo service's purpose: it carries the data that the client expects to receive back unchanged. The echo service MUST preserve the payload block exactly as received. This preservation allows clients to verify round-trip integrity by comparing the reflected payload against what was originally sent.

3.2.3. Extension Blocks

The echo service preserves extension blocks in the cloned bundle, retaining each block's block type code, block processing control flags, CRC type, and block-type-specific data. Block numbers SHOULD NOT be changed, as other blocks (such as BPSec blocks) may reference them. CRCs MUST be recalculated for blocks whose content changes during BPA processing.

Since the echo service submits the response bundle to the BPA for transmission, the BPA applies standard [RFC9171] extension block processing, updating blocks as appropriate for an outbound bundle.

For unrecognized blocks, the BPA applies block processing control flags per [RFC9171], Section 4.2.4, respecting the "Delete bundle if block can't be processed" flag (bit 2) and discarding unrecognized blocks that lack this flag.

The following subsections describe the expected behavior for commonly used extension blocks.

3.2.3.1. Hop Count Block

If the received bundle contains a Hop Count Block (block type 10), the echo service preserves it in the response bundle. When the BPA processes the outbound response bundle, it increments the hop count as specified in [RFC9171]. If the hop count exceeds the hop limit, the BPA will delete the bundle rather than forward it. Clients need to set hop limits with this additional increment in mind.

3.2.3.2. Previous Node Block

If the received bundle contains a Previous Node Block (block type 6), the echo service preserves it; the BPA updates it during normal forwarding processing.

3.2.3.3. Bundle Age Block

If the received bundle contains a Bundle Age Block (block type 7), the echo service SHOULD preserve it unchanged. The BPA updates the bundle age during normal forwarding processing.

3.2.3.4. Other Extension Blocks

For any other extension blocks, the echo service preserves them in the cloned bundle, and the BPA applies standard block processing.

3.2.4. Status Reports

When the received bundle has status report request flags set, the echo service node generates multiple status reports for the bundle's journey:

1. *Received*: Generated when the BPA receives the bundle at the echo service node

2. *Delivered*: Generated when the BPA delivers the bundle to the echo service

These reports are sent to the report-to endpoint specified in the original bundle and describe the original bundle's arrival at its destination.

The response bundle, having preserved the status report request flags, will generate its own separate status reports:

1. *Forwarded*: Generated when the BPA forwards the response bundle from the echo service node

2. *Received* and *Forwarded*: Generated at each intermediate node on the return path

3. *Delivered*: Generated when the response bundle reaches the originating client

Since the response bundle also preserves the report-to endpoint (Section 3.2.1), these reports are sent to the same destination. Clients can distinguish original bundle reports from response bundle reports by examining the bundle identifier in each status report.

Note: Status reporting is optional per [RFC9171] and many BPA implementations disable it by default. Clients cannot rely on receiving status reports for correct operation.

3.2.5. Security Blocks

BIB and BCB blocks as defined in [RFC9172] require no special handling; they are processed as extension blocks per Section 3.2.3.

3.3. Client Considerations

While the echo service specification focuses primarily on the reflector, certain requirements apply to clients to ensure correct operation. This section defines those normative requirements.

3.3.1. Session Disambiguation

When multiple clients run concurrently on the same node, each session must be distinguishable so that responses are delivered to the correct client. Multiple concurrent clients on the same node MUST use distinct source endpoint identifiers. Per [RFC9171], each application instance registers with a unique endpoint ID, and the combination of source and destination provides session disambiguation at the bundle layer without requiring any session identifier in the payload.

3.3.2. Bundle Integrity

Clients that wish to sign bundles for integrity verification must account for the fact that the echo service modifies the primary block during reflection. If a client signs bundles with a BIB whose Integrity-Protected Plaintext (IPPT) includes the primary block (integrity scope flags bit 0 set per [RFC9173]), the signature will be computed over the original primary block contents. When the echo service swaps the source and destination, the primary block changes, and verification of such a BIB will fail.

To ensure that integrity verification succeeds on the reflected bundle, clients that sign bundles SHOULD clear the "include primary block" flag (bit 0) in their integrity scope flags. With this flag cleared, the BIB signature covers only the payload and optionally the security headers, all of which remain unchanged through reflection.

3.3.3. Fragmentation

Diagnostic clients SHOULD set the "bundle must not be fragmented" flag in bundles sent to the echo service. Fragmentation complicates round-trip time measurement and payload verification: fragments might take different paths, arrive out of order, or be lost independently. Setting this flag ensures the bundle either traverses the network intact or is dropped, providing cleaner diagnostic results.

If a client needs to test path MTU, it can send bundles of increasing size with fragmentation disabled and observe which sizes succeed. This approach directly reveals the path’s maximum bundle size rather than relying on fragmentation behavior.

4. Security Considerations

This section discusses security issues relevant to the echo service and potential mitigations.

4.1. Amplification Attacks

Like any echo or reflection service, the Bundle Protocol echo service could potentially be used for amplification attacks if the response bundle is larger than the request. This might occur if the echo service adds extension blocks to the response. To mitigate this risk, implementations SHOULD:

* Rate-limit echo responses, particularly from previously unseen source endpoints

* Monitor for unusual traffic patterns that might indicate abuse

* Consider requiring authentication via Bundle Protocol Security [RFC9172] in sensitive deployments

In practice, the amplification potential is limited because the echo service preserves the payload unchanged and is expected to make minimal modifications to the bundle. Any amplification would come only from small differences in bundle overhead.

4.2. Information Disclosure

Echo responses inherently confirm the existence and reachability of the echo service endpoint. Additionally, round-trip time measurements might reveal information about network topology, path characteristics, and store-and-forward delays that could be useful to an adversary mapping a network.

In sensitive environments where this information disclosure is a concern, operators MAY:

* Restrict echo service access to authenticated endpoints using BPSec

* Disable the echo service entirely on nodes where diagnostics are not required

* Deploy the echo service only on designated diagnostic nodes rather than throughout the network

4.3. Resource Exhaustion

An attacker could attempt to exhaust echo service resources by sending a large volume of bundles or bundles with very large payloads. Since the echo service must clone and retransmit each received bundle, this creates both memory and bandwidth pressure. Implementations SHOULD:

* Limit the maximum payload size accepted for reflection

* Implement rate limiting on both connections and bundle processing

* Monitor resource usage and reject or delay bundle processing when under stress

5. IANA Considerations

5.1. Well-Known Service Registration

This document requests IANA to extend the "'ipn' Scheme URI Well-Known Service Numbers for BPv7" registry established by [RFC9758] to include a "DTN Demux" column, and to register the following entry:

Value	DTN Demux	Description	Reference
128	echo	Echo Service	(this document)

Table 1: Echo Service Registration

The IPN service number and DTN demux together define a well-known Bundle Protocol service. For the IPN scheme, the service number is appended to the node number (e.g., ipn:2.128). For the DTN scheme, the demux is appended to the node-name to form the demux component of the URI (e.g., dtn://example.dtn/echo).

Note: The IPN value 7 has been used by convention in existing implementations, mirroring the well-known UDP port for the Echo Protocol [RFC862]. However, service numbers 1-127 are designated Private Use per [RFC9758]. This document requests service number 128, the lowest value in the Standards Action range. Implementations MAY continue to support service number 7 for backwards compatibility.

6. References

6.1. Normative References

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

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

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

[RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPSec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <https://www.rfc-editor.org/rfc/rfc9172>.

[RFC9173] Birrane, III, E., White, A., and S. Heiner, "Default Security Contexts for Bundle Protocol Security (BPSec)", RFC 9173, DOI 10.17487/RFC9173, January 2022, <https://www.rfc-editor.org/rfc/rfc9173>.

[RFC9758] Taylor, R. and E. Birrane III, "Updates to the 'ipn' URI Scheme", RFC 9758, DOI 10.17487/RFC9758, May 2025, <https://www.rfc-editor.org/rfc/rfc9758>.

6.2. Informative References

[RFC862] Postel, J., "Echo Protocol", STD 20, RFC 862, DOI 10.17487/RFC0862, May 1983, <https://www.rfc-editor.org/rfc/rfc862>.

Appendix A. Ping Clients

This appendix provides non-normative guidance for implementing ping clients that use the echo service. While the echo service specification defines the reflector behavior, effective ping clients require careful attention to timing, session management, and payload design.

A.1. Round-Trip Time Calculation

Accurate round-trip time (RTT) measurement is the primary purpose of most ping implementations. Ping clients should calculate RTT using locally stored timestamps rather than timestamps embedded in the payload:

```
RTT = response_receive_time - request_sent_times[sequence_number]
```

This approach offers several advantages:

* Requires no clock synchronization between nodes
* Works correctly even if the payload is corrupted
* Avoids serialization overhead in the timing path

The client should maintain a map from sequence number to sent timestamp. It should populate the map when each request is transmitted and consult it when each response arrives. Entries should be removed after a configurable timeout to prevent unbounded memory growth.

A.2. Endpoint Selection

As required by Section 3.3.1, multiple concurrent clients on the same node use distinct source endpoint identifiers.

For example, if two concurrent ping sessions on node ipn:1.0 target ipn:2.128, they should use distinct source endpoints such as ipn:1.1001 and ipn:1.1002. The bundle protocol agent will then route responses back to the correct session based on the destination in the reflected bundle.

A.3. Payload Format

Since the echo service preserves bundle content unchanged except for the source and destination swap, the payload format is entirely at the discretion of the ping client. The echo service does not parse or interpret the payload, so implementations may use any encoding that suits their needs.

At minimum, the payload need only contain a sequence number so that the client can match responses to their corresponding requests. For implementations seeking extensibility and interoperability, a CBOR-based format is suggested:

```
[
  sequence,     ; uint: monotonically increasing counter
  options       ; map: optional fields (may be empty)
]
```

Figure 1: Suggested Payload Format (CBOR)

The options map provides room for future extensions and may include:

Key	Name	Description
0	Padding	Padding bytes for path MTU testing; allows clients to test whether bundles of various sizes can traverse the network
1	Timestamp	DTN time ([RFC9171], Section 4.2.6), i.e., milliseconds since the DTN Epoch, for debugging purposes; while not used for RTT calculation (see Appendix A.1), this can help diagnose clock synchronization issues
2	Extension Blocks	List of attached extension block types; enables clients to verify which extension blocks survive the round trip

Table 2: Suggested Payload Options

A.4. Statistics

Ping clients should track and report standard statistics consistent with traditional IP ping:

* Bundles transmitted

* Responses received

* Packet loss percentage

* RTT minimum, average, maximum, and standard deviation

These statistics provide a quick assessment of network health and help identify routing problems, congestion, or intermittent connectivity.

Example output format following ICMP ping conventions:

```
--- ipn:2.128 ping statistics ---
5 bundles transmitted, 4 received, 20% loss
rtt min/avg/max/stddev = 1.234/2.567/4.891/1.203 s
```

Author’s Address

Rick Taylor
Aalyria Technologies
Email: rtaylor@aalyria.com