Delay-Tolerant Networking
Internet-Draft
Intended status: Standards Track
Expires: 29 November 2025

E.J. Birrane
B. Sipos
JHU/APL

28 May 2025

DTNMA Asynchronous Management Protocol (AMP)
draft-ietf-dtn-amp-02

Abstract

This document defines a messaging protocol for the Delay-Tolerant Networking (DTN) Management Architecture (DTNMA) Asynchronous Management Model (AMM) and a transport binding for exchanging those messages over a network. This Asynchronous Management Protocol (AMP) does not require transport-layer sessions, operates over unidirectional links, and seeks to reduce the energy and compute power necessary for performing remote management of resource constrained devices possibly over challenged networks.

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 29 November 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction

Remote management in challenged and resource constrained networks must be accomplished differently than the remote management methods used in low-delay, high-rate, high-availability networks. The Delay-Tolerant Networking (DTN) Management Architecture (DTNMA), as defined in [RFC9675], provides an overview and justification of an alternative to "synchronous" management services such as those provided by SNMP [RFC3411] or NETCONF [RFC6241] (and its derivatives RESTCONF [RFC8040] and CORECONF [I-D.ietf-core-comi]). In particular, the DTNMA defines the need for a flexible, robust, and

efficient autonomy engine to handle decisions when operators cannot be active in the network.

The logical description of that DTNMA Application Management Model (AMM), and its realization in static Application Data Models (ADMs) and dynamic Operational Data Models (ODMs), is in [I-D.ietf-dtn-amm]. The AMM presents an efficient and expressive model for the asynchronous management of a network node, but does not specify any particular message structure or encoding.

The overall function of the DTNMA Asynchronous Management Protocol (AMP) is to deliver Execution-Set (EXECSET) values to a DTNMA Agent and Reporting-Set (RPTSET) values to a DTNMA Manager as described in the AMM Section 2.3 of [I-D.ietf-dtn-amm]. This specification provides an enveloping of those ARIs and, in doing so, AMP defines very few structures of its own.

This document specifies the content of messages which envelope ARI values [I-D.ietf-dtn-ari] as service data units (SDUs), an encoding of those messages as a protocol data units (PDU) in Section 3, and a transport for these PDUs as a Bundle Protocol version 7 (BPv7) [RFC9171] application data unit (ADU) in Section 4.1.

1.1. Scope

The AMP provides data monitoring, administration, and configuration for applications operating above the data link layer of the OSI networking model. While the AMP may be configured to support the management of network layer protocols, it also uses these protocol stacks to encapsulate and communicate its own messages.

It is assumed that the transport(s) used to carry AMP messages provide addressing, confidentiality, integrity, security, fragmentation/reassembly, and other network functions. Therefore, these items are outside of the scope of this document.

This document describes the format of messages used to exchange data models between managing and managed devices in a network. The rationale for this type of exchange is outside of the scope of this document and is covered in [RFC9675]. The description and explanation of the data models exchanged is also outside of the scope of this document and is covered in [I-D.ietf-dtn-amm].

This document does not address specific configurations of AMP-enabled devices or any ADMs or ODMs available on such devices. This also does not discuss the interface, if any, between AMP and other management protocols.

1.2. Use of CDDL

This document defines Concise Binary Object Representation (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, which includes the example CBOR content.

```
start = amp-adu / [id-amp-msg]
```

From the document [I-D.ietf-dtn-ari] the definitions are taken for ari, lit-execset, and lit-rptset.

1.3. Example ARI Line Folding

The URI encoding of [I-D.ietf-dtn-ari] does not allow blank space characters and some of the example ARIs are longer than RFC recommended line lengths, the examples uses the "single backslash" folding strategy of Section 7 of [RFC8792] for line wrapping. The header from that strategy is not used explicitly in this document, so any use of indentation in example ARIs will make use of this folding strategy.

1.4. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD 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.

The terms "Agent", "Application Data Model", "Externally Defined Data", "Variable", "Control", "Literal", "Macro", "Manager", "Report Template", "Report", "Table", "Constant", "Operator", "Time-Based Rule" and "State-Based Rule" are used without modification from the definitions provided in [I-D.ietf-dtn-amm].

2. Constraints and Assumptions

The desirable properties of an asynchronous management protocol, as specified in the DTNMA, are summarized here to represent design constraints on the AMP specification.

Intelligent Push of Information: Nodes in a challenged network

cannot guarantee concurrent, bi-directional communications. Some links between nodes may be strictly unidirectional. In the DTNMA, Agents "push" data to Managers rather than Managers "pulling" data from Agents.

Small Message Sizes: Smaller messages require smaller periods of viable transmission for communication, incur less retransmission cost, and consume fewer resources when persistently stored en route in the network. The AMP minimizes message size wherever practical, to include binary data representations and predefined data definitions and templates.

Static and Dynamic Identification: All data in the system must be uniquely addressable, to include operator-specified information. AMP provides a compact encoding for identifiers based on the Application Resource Identifier (ARI) of [I-D.ietf-dtn-ari].

Stateless Operation: There is no reliable concept of session establishment or round-trip data exchange in challenged networks. AMP is designed to be stateless and ADM controls are specified to be idempotent when executed. Where helpful, AMP provides mechanisms for ordering of execution within a single AMP protocol data unit, but otherwise degrades gracefully when nodes in the network diverge in their configuration.

Independence from ADMs: Although some portions of the AMP structure share concepts and capabilities of AMM semantic types, the AMP operates independently from any specific ADMs or ODMs which would use the AMP for messaging between entities. This avoids the need for an AMP processor to have information about those specific ADMs or ODMs, similarly to how the ARI syntax processing is independent from specific ADMs or ODMs. The interpreting of ARIs, however, does require the use of specific referenced ADMs and ODMs.

All AMP encodings are self-terminating, based on Concise Binary Object Representation (CBOR). This means that, given an indefinite-length octet stream, each encoding can be unambiguously decoded from the stream without requiring additional information such as a length field separate from the data type definition. CBOR also provides a layer of well-formed data coding separate from valid AMP structure coding.

3. Message Structure and Sequencing

Each AMP message consists of a version number followed by one or more binary form ARI values, as defined in Section 5 of [I-D.ietf-dtn-ari]. All AMP messages conforming to this specification SHALL contain version number 1. Any AMP messages received with an unknown version number SHALL be ignored.

Each of the contained ARIs SHALL be either an EXECSET or a RPTSET. The EXECSET is used to communicate from Manager to Agent and cause execution activities within the Agent as defined in Section 3.1. The RPTSET is used to communicate from Agent to Manager, which includes reports and (specific) execution results from the Agent, as defined in Section 3.2.

Each AMP message has the following CDDL definition representing a CBOR sequence [RFC8742].

```
amp-msg = (
  version: 1,
  + amp-ari
)
amp-ari = (lit-execset / lit-rptset) .within ari
```

3.1. Execution-Set Values

When received by an Agent, an EXECSET value SHALL result in immediate execution activities based on the message contents. Each item in the target list SHALL be executed independently (_i.e._, failures on one item do not affect other items). Each item in the target list MAY be executed in any order or concurrently. This is not the same behavior as execution of a macro (see Section 6.6.3 of [I-D.ietf-dtn-amm]), where execution of items is ordered and a failure of any execution causes subsequent items to not be executed.

When possible, Managers SHOULD coalesce multiple execution targets into a single EXECSET value. This avoids the overhead of processing multiple messages on an Agent to cause multiple executions, but it does require that all or none of the executions are associated with a nonce value.

> | Because execution targets are supposed to be idempotent (see Section 3.4.5 of [I-D.ietf-dtn-amm]) there is no need to differentiate multiple targets with the same object-identity-and-parameters when using the same nonce.

3.2. Reporting-Set Values

When received by a Manager, each report within a RPTSET value SHALL be correlated to its ADM or ODM object used to interpret its source-specific data. Each report in the Report List SHALL be processed independently (_i.e._, failures on one report do not affect other items). Each report in the Report List MAY be processed in any order or concurrently.

When associated to the same nonce value, Agents SHOULD coalesce multiple reports into a single RPTSET value. The coalescing MAY be based on a time interval or an event (e.g. power-saving wake-up). This avoids the overhead of processing multiple RPTSET values on a Manager and improves timestamp compression in the items, but it does require that all or none of the items are associated with the same nonce value.

4. Message Transport

Any transport binding for AMP SHALL provide authentication of the ultimate endpoints of the messages in support of the AMM requirements in Section 2.3 of [I-D.ietf-dtn-amm]. This document defines a specific binding to BPv7 in Section 4.1 and guidance about alternative or proxy bindings in Section 4.2.

4.1. Bundle Protocol Binding

When embedded as block type-specific data (BTSD) within a BPv7 payload block in accordance with [RFC9171], the application data unit SHALL consist of an AMP message (see Section 3) as a CBOR sequence. The payload BTSD has the following CDDL definition, where the bstr value means the field present in the canonical block (_i.e._ the BTSD contains the amp-msg sequence).

```
amp-adu = bstr .cborseq [amp-msg]
```

When Agents and Managers register endpoints on a BPA, they SHOULD use the well-known service numbers defined in Section 5.1. Using well-known identifiers simplifies configuration and troubleshooting but is not necessary for correct AMP operation.

When BPv7 is used as transport for AMP, the primary and payload blocks SHALL be authenticated by a BPSec [RFC9172] mechanism traceable to the message source. This can be either block integrity block (BIB) or block confidentiality block (BCB) using an authenticated encryption algorithm, either using an authenticated public key of the source directly or via some security association derived from an authenticated public key or from a security gateway

and delegated for the bundle source. It is an network policy and configuration issue to determine the correct use of BPSec for any particular Manager and Agent.

When processing an AMP ADU, the processing context SHALL include the following:

* The bundle Source Node ID

* An indication of the authenticity of the primary and payload blocks, along with the Security Source Node ID used to authenticate them

4.2. Proxy Transport

In cases where direct transport from the source to the ultimate destination endpoint (either to an Agent or Manager) is not available, it is possible to use an intermediate transport binding which acts as a proxy to reach that ultimate endpoint. Part of the AMM transport information required by Section 3.2 of [I-D.ietf-dtn-amm] is a proxy-authenticated ultimate endpoint identity. When a proxy is used, that ultimate endpoint identity SHALL be included explicitly as part of the proxy interface.

Within some transports, such as the Hypertext Transfer Protocol (HTTP) of [RFC9110], there are application-accessible metadata such as private use headers, URI path segments, or URI query parameters which can be used to convey ultimate endpoint identity along with the encoded AMP message. For other transports, where only an octet string payload is provided, the ultimate endpoint identity needs to be included in that payload directly.

When the identity needs to be conveyed directly with an AMP message it SHALL be present before the message sequence as a binary form ARI value. The identity value SHALL be limited to match the AMM type referenced by "ari://ietf/network-base/typedef/endpoint-or-uri". This type is a a union of constrained IDENT and TEXTSTR types.

```
id-amp-msg = (
    amp-identity,
    amp-msg
)
amp-identity = ari ; Constrained to endpoint-or-uri type
```

It is an implementation matter to determine which identity endpoint objects or URI schemes are supported by the proxy. It is a separate matter to configure a Manager or Agent to use only those identity types for specific proxy instances.

5. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of schema and namespaces related to the Application Resource Identifier (ARI), in accordance with BCP 26 [RFC1155].

5.1. URI Schemes

This document defines entries in the registry "'ipn' Scheme URI Well-known Service Numbers for BPv7" within the "URI Schemes" registry group [IANA-URI] containing the following.

// RFC Editor: The values for TBA1 and TBA2 below should be assigned
// from the range 128-255.

Value	Description	Reference
// TBA1	DTNMA Agent role	[This document]
// TBA2	DTNMA Manager role	[This document]

Table 1: 'ipn' Scheme URI Well-known Service Numbers for BPv7

6. Security Considerations

Security within the AMP exists in two distinct layers as follows.

Transport Security: Transport security addresses the questions of authentication, integrity, and confidentiality associated with the transport of messages between Managers and Agents. This security is applied before any particular entity in the system receives data and, therefore, its specifics are outside of the scope of this document. The BP transport specified in Section 4.1 does require some authentication which covers the AMP payload, but details are network- and implementation-specific.

Access Control: Fine grained object-level security is provided and enforced by Agents via access control lists (ACLs) which are part of an Agent's configuration. An Agent's ACLs could be managed via an ADM using AMP itself, but such details are also outside the scope of this document.

7. References

7.1. Normative References

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

[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>.

[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>.

[RFC8742] Bormann, C., "Concise Binary Object Representation (CBOR) Sequences", RFC 8742, DOI 10.17487/RFC8742, February 2020, <https://www.rfc-editor.org/info/rfc8742>.

[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>.

[I-D.ietf-dtn-amm] III, E. J. B., Sipos, B., and J. Ethier, "DTNMA Application Management Model (AMM) and Data Models", Work in Progress, Internet-Draft, draft-ietf-dtn-amm-04, 27 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-amm-04>.

[I-D.ietf-dtn-ari] Birrane, E. J., Annis, E., and B. Sipos, "DTNMA Application Resource Identifier (ARI)", Work in Progress, Internet-Draft, draft-ietf-dtn-ari-04, 18 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-ari-04>.

7.2. Informative References

[RFC1155] Rose, M. and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets", STD 16, RFC 1155, DOI 10.17487/RFC1155, May 1990, <https://www.rfc-editor.org/info/rfc1155>.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <https://www.rfc-editor.org/info/rfc3411>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC9675] Birrane, III, E., Heiner, S., and E. Annis, "Delay-Tolerant Networking Management Architecture (DTNMA)", RFC 9675, DOI 10.17487/RFC9675, November 2024, <https://www.rfc-editor.org/info/rfc9675>.

[I-D.ietf-core-comi] Veillette, M., Van der Stok, P., Pelov, A., Bierman, A., and C. Bormann, "CoAP Management Interface (CORECONF)", Work in Progress, Internet-Draft, draft-ietf-core-comi-20, 6 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-comi-20>.

Appendix A. Example Messages

The examples in this section use the well-known organization "example" (65535) with a single model "adm-a" (1) for simplicity.

A.1. Execution with a Nonce

An example of an Execution-Set being sent to an Agent has the following ARI text representation (folded according to Section 1.3).

```
ari:/EXECSET/n=1234;(
    //example/adm-a/CTRL/dothing,
    //example/adm-a/CONST/amacro)
```

Assuming some enumeration values for the ADM and objects results in the following transformed ARI.

```
ari:/EXECSET/n=1234;(//65535/1/-3/18,//65535/1/-2/43)
```

This is embedded into an AMP message with the following CBOR sequence.

```
1,
[20, [1234, [65535, 1, -3, 18], [65535, 1, -2, 43]]]
```

Which is encoded to the following binary string.

```
0x018214831904D28419FFFF0122128419FFFF0121182B
```

An example of a corresponding Reporting-Set being sent to a Manager has the following ARI text representation (folded according to Section 1.3).

```
ari:/RPTSET/n=1234;r=/TP/20230102T030405Z;\
    (t=/TD/PT0S;s=//example/adm-a/CTRL/dothing;(null))\
    (t=/TD/PT5S;s=//example/adm-a/CTRL/other;(567))
```

Which results in the following transformed ARI.

```
ari:/RPTSET/n=1234;r=/TP/725943845;\
    (t=/TD/0;s=//65535/1/-3/18;(null))\
    (t=/TD/5;s=//65535/1/-3/6;(567)))
```

This is embedded into an AMP message with the following CBOR sequence.

```
1,
[21, 1234, 725943845,
    [0, [65535, 1, -3, 18], null],
    [5, [65535, 1, -3, 6], 567]]
```

Which is encoded to the following binary string.

0x0185151904D21A2B45062583008419FFFF012212F683058419FFFF012206190237

In addition to the direct control result feedback present in that RPTSET, the execution might cause other reports to be produced. Any of these other reports will be sent within a different RPTSET having a null value nonce.

A.2. Execution without a Nonce

An example of an Execution-Set having the same controls as the above example but with a null nonce has the following ARI text representation (folded according to Section 1.3).

```
ari:/EXECSET/n=null;(
    //example/adm-a/CTRL/dothing,
    //example/adm-a/CONST/amacro)
```

Assuming some enumeration values for the ADM and objects results in the following transformed ARI.

```
ari:/EXECSET/n=null;(//65535/1/-3/18,//65535/1/-2/43)
```

This is embedded into an AMP message with the following CBOR sequence.

```
1,
[20, [null, [65535, 1, -3, 18], [65535, 1, -2, 43]]]
```

Which is encoded to the following binary string.

0x01821483F68419FFFF0122128419FFFF0121182B

Because this EXECSET uses a null nonce, there will be no direct feedback from the Agent about when each target control is executed. Any other side effects, including explicit report generation, will behave the same way as if the nonce was non-null.

A.3. Proxy Transport Example

If the previous example needed to be transported via some proxy to an ultimate BPv7 destination of "ipn:974848.34.128", that destination URI would be used to prefix the AMP message as in the following sequence.

```
"ipn:974848.34.128",
1,
[20, [null, [65535, 1, -3, 18], [65535, 1, -2, 43]]]
```

Which is encoded to the following binary string.

0x7169706E3A3937343834382E33342E31323801821483F68419FFFF0122128419FFFF0121182B

Implementation Notes

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

A reference implementation of an earlier revision of the AMP is available in the 3.6.2 release of the ION open source code base available from the ION-DTN (https://sourceforge.net/projects/ion-dtn/) Sourceforge project.

An extraction of the same AMP Agent and Manager from ION into a stand-alone project is available in the DTNMA Tools (https://github.com/JHUAPL/dtnma-tools) GitHub project. This project also contains an updated Wireshark AMP dissector (https://github.com/JHUAPL/dtnma-tools/tree/main/wireshark) for the corresponding earlier revision of this draft.

Acknowledgments

The following participants contributed technical material, use cases, and useful thoughts on the overall approach to this protocol specification: Jeremy Pierce-Mayer of INSYEN AG contributed the concept of the typed data collection and early type checking in the protocol. David Linko and Evana DiPietro of the Johns Hopkins University Applied Physics Laboratory contributed appreciated review and type checking of various elements of this specification.

Authors' Addresses

Edward J. Birrane, III
The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Rd.
Laurel, MD 20723
United States of America
Phone: +1 443 778 7423
Email: Edward.Birrane@jhuapl.edu

Brian Sipos
The Johns Hopkins University Applied Physics Laboratory
Email: brian.sipos+ietf@gmail.com