Delay-Tolerant Networking
Internet-Draft
Intended status: Informational
Expires: May 3, 2017

E. Birrane
Johns Hopkins Applied Physics Laboratory

October 30, 2016

Asynchronous Management Architecture
draft-birrane-dtn-ama-04

Abstract

This document describes the motivation, desirable properties, system model, roles/responsibilities, and component models associated with an asynchronous management architecture (AMA) suitable for providing application-level network management services in a challenged networking environment. Challenged networks are those that require fault protection, configuration, and performance reporting while unable to provide human-in-the-loop operations centers with synchronous feedback in the context of administrative sessions. In such a context, networks must exhibit behavior that is both determinable and autonomous while maintaining compatibility with existing network management protocols and operational concepts.

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 http://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 May 3, 2017.

Copyright Notice

Copyright (c) 2016 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 (http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

This document presents an Asynchronous Management Architecture (AMA) providing application-layer network management services over links where delivery delays prevent timely communications between a network operator and a managed device. These delays may be caused by long signal propagations or frequent link disruptions (such as described in [RFC4838]) or by non-environmental factors such as unavailability of network operators, administrative delays, or delays caused by quality-of-service prioritizations and service-level agreements.

1.1. Purpose

This document describes the motivation, rationale, desirable properties, and roles/responsibilities associated with an asynchronous management architecture (AMA) suitable for providing network management services in a challenged networking environment. These descriptions should be of sufficient specificity that an implementing Asynchronous Management Protocol (AMP) in conformance with this architecture will operate successfully in a challenged networking environment.

An AMA is necessary as the assumptions inherent to the architecture and design of synchronous management tools and techniques are not valid in challenged network scenarios. In these scenarios, synchronous approaches either patiently wait for periods of bi-directional connectivity or require the investment of significant time and resources to evolve a challenged network into a well-connected, low-latency network. In some cases such evolution is merely a costly way to over-resource a network. In other cases, such evolution is impossible given physical limitations imposed by signal propagation delays, power, transmission technologies, and other phenomena. Asynchronous management of asynchronous networks enables large-scale deployments, distributed technical capabilities, and reduced deployment and operations costs.

1.2. Scope

It is assumed that any challenged network where network management would be usefully applied supports basic services (where necessary) such as naming, addressing, integrity, confidentiality,

authentication, fragmentation, and traditional network/session layer functions. Therefore, these items are outside of the scope of the AMA and not covered in this document.

While likely that a challenged network will eventually interface with an unchallenged network, this document does not address the concept of network management compatibility with synchronous approaches. An AMP in conformance with this architecture should examine compatibility with existing approaches as part of supporting nodes acting as gateways between network types.

1.3. Requirements Language

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 [RFC2119].

1.4. ORganization

The remainder of this document is organizaed into seven sections that, together, describe an AMA suitable for enterprise management of asynchronous networks: terminology, motivation, service definitions, desirable properties, roles/responsibilities, system model, and logical component model. The description of each section is as follows.

- Motivation - This section provides an overall motivation for this work as providing a novel and useful alternative to current network management approaches. Specifically, this section describes common network functions and how synchronous mechanisms fail to provide these functions in an asynchronous environment.

- Service Definitions - This section defines asynchronous network management services in terms of terminology, scope, and impact.

- Desirable Properties - This section identifies the properties to which an AMP should adhere to effectively implement service definitions in an asynchonous environment. These properties guide the subsequent definition of the system and logical models that comprise the AMA.

- Roles and Responsibilities - This section identifies the roles of logical Actors in the AMA and their associated responsibilities. It provides the terminology and context for discussing how network management services interact.

- System Model - This section describes data flows amongst various defined Actor roles. These flows capture how the AMA system works

to provide asynchronous network management services in accordance with defined desirable properties.

- Logical Component Model - This section describes those logical functions that must exist in any instantiation of an AMP.

2. Terminology

This section identifies those terms critical to understanding the proper operation of the AMA. Whenever possible, these terms align in both word selection and meaning with their analogs from other management protocols.

- Actor - A software service running on either managed or managing devices for the purpose of implementing management protocols between such devices. Actors may implement the "Manager" role, "Agent" role, or both.

- Agent Role (or Agent) - The role associated with a managed device, responsible for reporting performance data, enforcing administrative policies, and accepting/performing actions. Agents exchange information with Managers operating either on the same device or on a remote managing device.

- Asynchronous Management Protocol (AMP) - An application-layer protocol used to manage the Data, Controls, and other items necessary for configuration, monitoring, and administration of applications and protocols on a node in a challenged network.

- Application Data Model (ADM) - The set of predefined data definitions, reports, literals, operations, and controls given to an Actor to manage a particular application or protocol. Actors support multiple ADMs, one for each application/protocol being managed.

- Externally Defined Data (EDD) - Information made available to an Agent by a managed device, but not computed directly by the Agent. EDD definitions form the "lingua franca" for data within the AMA and are defined by ADMs.

- Variable (VAR) - Information that is computed by an Agent, typically as a function of EDDs and/or other Variables. A VAR is a strongly-typed value. When a VAR is specified in an ADM, its type and default value are immutable. When a VAR is defined outside of an ADM, the type and default value may be changed. If an ADM wishes to define an item whose type and value are both immutable, that is no longer considered a Variable and should be represented as a Literal.

- Controls (CTRLs) - Operations that may be undertaken by an Actor to change the behavior, configuration, or state of an application or protocol managed by an AMP. Similar to Externally Defined Data, Controls are defined solely in ADMs and their definition is immutable.

- Literals (LIT) - Constants, enumerations, and other immutable definitions.

- Macros - A named, ordered collection of Controls. When a Macro is defined in an ADM, that definition is immutable. When a Macro is defined outside of an ADM, that definition may be changed.

- Manager - A role associated with a managing device responsible for configuring the behavior of, and receiving information from, Agents. Managers interact with one or more Agents located on the same device and/or on remote devices in the network.

- Operator (OP) - The enumeration and specification of a mathematical function used to calculate computed data definitions and construct expressions to calculate state. Operators are specified in Application Data Models (ADMs) and their definition is immutable.

- Report Entry (RPTE) - A named, typed, ordered collection of data values gathered by one or more Agents and provided to one or more Managers. Report entries populate report templates with values.

- Report Template (RPTT) - An ordered collection of data identifiers. When defined in an ADM the report template definition is immutable. When defined outside of an ADM, the template definition may change.

- Rule - A unit of autonomous specification that provides a stimulus-response relationship between time or state on an Agent and the Controls to be run as a result of that time or state.

3. Motivation

Challenged networks, to include networks challenged by administrative or policy delays, cannot guarantee capabilities required to enable synchronous management techniques. These capabilities include high-rate, highly-available data, round-trip data exchange, and operators "in-the-loop". The inability of current approaches to provide network management services in a challenged network motivates the need for a new network management architecture focused on asynchronous, open-loop, autonomous control of network components.

3.1. Challenged Networks

A growing variety of link-challenged networks support packetization to increase data communications reliability without otherwise guaranteeing a simultaneous end-to-end path. Examples of such networks include Mobile Ad-Hoc Networks (MANets), Vehicular Ad-Hoc Networks (VANets), Space-Terrestrial Internetworks (STINTs), and heterogeneous networking overlays. Links in such networks are often unavailable due to attenuations, propagation delays, occultation, and other limitations imposed by energy and mass considerations. Data communications in such networks rely on store-and-forward and other queueing strategies to wait for the connectivity necessary to usefully advance a packet along its route.

Similarly, there also exist well-resourced networks that incur high message delivery delays due to non-environmental limitations. For example, networks whose operations centers are understaffed or where data volume and management requirements exceed the real-time cognitive load of operators or the associated operations console software support. Also, networks where policy restricts user access to existing bandwidth creates situations functionally similar to link disruption and delay.

Independent of the reason, when a node experiences an inability to communicate it must rely on autonomous mechanisms to ensure its safe operation and ability to usefully re-join the network at a later time. In cases of sparsely-populated networks, there may never be a practical concept of "the connected network" as most nodes may be disconnected most of the time. In such environments, defining a network in terms of instantaneous connectivity becomes impractical or impossible.

Specifically, challenged networks exhibit the following properties that may violate assumptions built into current approaches to synchronous network management.

- Links may be uni-directional.
- Bi-directional links may have asymmetric data rates.
- No end-to-end path is guaranteed to exist at any given time between any two nodes.
- Round-trip communications between any two nodes within any given time window may be impossible.

3.2. Current Management Approaches

Network management tools in unchallenged networks provide mechanisms for communicating locally-collected data from Agents to Managers, typically using a "pull" mechanism where data must be explicitly requested by a Manager in order to be transmitted by an Agent.

A near ubiquitous method for management in unchallenged networks today is the Simple Network Management Protocol (SNMP) [RFC3416]. SNMP utilizes a request/response model to set and retrieve data values such as host identifiers, link utilizations, error rates, and counters between application software on Agents and Managers. Data may be directly sampled or consolidated into representative statistics. Additionally, SNMP supports a model for asynchronous notification messages, called traps, based on predefined triggering events. Thus, Managers can query Agents for status information, send new configurations, and be informed when specific events have occurred. Traps and queryable data are defined in one or more Managed Information Bases (MIBs) which define the information for a particular data standard, protocol, device, or application.

In challenged networks, the request/response method of data collection is neither efficient nor, at times, possible as it relies on sessions, round-trip latency, message retransmission, and ordered delivery. Adaptive modifications to SNMP to support challenged networks would alter the basic function of the protocol (data models, control flows, and syntax) so as to be functionally incompatible with existing SNMP installations.

The Network Configuration Protocol (NETCONF) provides device-level configuration capabilities [RFC6241] to replace vendor-specific command line interface (CLI) configuration software. The XML-based protocol provides a remote procedure call (RPC) syntax such that any exposed functionality on an Agent can be exercised via a software application interface. NETCONF places no specific functional requirements or constraints on the capabilities of the Agent, which makes it a very flexible tool for configuring a homogeneous network of devices. However, NETCONF does place specific constraints on any underlying transport protocol: namely, a long-lived, reliable, low-latency sequenced data delivery session. This is a fundamental requirement given the RPC-nature of the operating concept, and it is unsustainable in a challenged network.

3.3. Limitations of Current Approaches

Management approaches that rely on timely data exchange, such as those that rely on negotiated sessions or other synchronized acknowledgment, do not function in challenged network environments.

Familiar examples of TCP/IP based management via closed-loop, synchronous messaging does not work when network disruptions increase in frequency and severity. While no protocol delivers data in the absence of a networking link, protocols that eliminate or drastically reduce overhead and end-point coordination require smaller transmission windows and continue to function when confronted with scaling delays and disruptions in the network.

Just as the concept of a loosely-confederated set of nodes changes the definition of a network, it also changes the operational concept of what it means to manage a network. When a network stops being a single entity exhibiting a single behavior, "network management" becomes large-scale "node management". Individual nodes must share the burden of implementing desirable behavior without reliance on a single oracle of configuration or other coordinating function such as an operator-in-the-loop.

4. Service Definitions

This section identifies the services that must exist between Managers and Agents within an AMA. These services include configuration, reporting, parameterized control, and administration.

4.1. Configuration

Configuration services update local Agent information relating to managed applications and protocols. This information may be configured from ADMs, the specification of parameters associated with these models, and as defined by operators in the network.

New configurations received by a node must be validated to ensure that they do not conflict with other configurations at the node, or prevent the node from effectively working with other nodes in its region. With no guarantee of round-trip data exchange, Agents cannot rely on remote Managers to correct erroneous or stale configurations from harming the flow of data through a challenged network.

Examples of configuration service behavior include the following.

- Creating a new datum as a function of other well-known data: C = A + B.
- Creating a new report as a unique, ordered collection of known data: RPT = {A, B, C}.
- Storing pre-defined, parameterized responses to potential future conditions:

IF (X > 3) THEN RUN CMD(PARM).

4.2. Reporting

Reporting services populate pre-defined Report Templates with values collected or computed by an Agent. The resultant Report Entries are sent to one or more Managers by the Agent. The term "reporting" is used in place of the term "monitoring", as monitoring implies a timeliness and regularity that cannot be guaranteed by a challenged network. Report Entries sent by an Agent provide best-effort information to receiving Managers.

Since a Manager is not actively "monitoring" an Agent, the Agent must make its own determination on when to send what Report Entries based on its own local time and state information. Agents should produce Report Entries of varying fidelity and with varying frequency based on thresholds and other information set as part of configuration services.

Examples of reporting service behavior include the following.

- Generate Report Entry R1 every hour (time-based production).
- Generate Report Entry R2 when X > 3 (state-based production).

4.3. Autonomous Parameterized Control

Controls represent a function that can be run by an Agent to affect its behavior or otherwise change its internal state. In this context, a Control may refer to a single function or an ordered set of functions run in sequence (e.g., a macro). The set of Controls understood by an Agent define the functions available to affect the behavior of applications and protocols managed by the Agent.

Since there is no guarantee that a Manager will be in contact with an Agent at any given time, the decisions of whether and when a Control should be run must be made locally and autonomously by the Agent. Two types of automation triggers are identified in the AMA: triggers based on the general state of the Agent and, more specifically, triggers based on an Agent’s notion of time. As such, the autonomous execution of Controls can be viewed as a stimulus-response system, where the stimulus is the positive evaluation of a state or time based predicate and the response is the Control to be executed.

The autonomous nature of Control execution by an Agent implies that the full suite of information necessary to run a Control may not be known by a Manager in advance of running the Control on an Agent. To address this situation, Controls in the AMA MUST support a

parmeterization mechanism so that required data can be provided at the time of execution on the Agent rather than at the time of definition/configuration by the Manager.

Autonomous, parameterized control provides a powerful mechanism for Managers to "manage" an Agent asynchronously during periods of no communication by pre-configuring responses to events that may be encountered by the Agent at a future time.

Examples of potential control service behavior include the following.

- Updating local routing information based on instantaneous link analysis.
- Managing storage on the device to enforce quotas.
- Applying or modifying local security policy.

4.4. Administration

Administration services enforce the potentially complex mapping of configuration, reporting, and control services amongst Agents and Managers in the network. Fine-grained access control specifying which Managers may apply which services to which Agents may be necessary in networks dealing with multiple administrative entities or overlay networks crossing multiple administrative boundaries. Whitelists, blacklists, key-based infrastructures, or other schemes may be used for this purpose.

Examples of administration service behavior include the following.

- Agent A1 only Sends reports for Protocol P1 to Manager M1.
- Agent A2 only accepts a configurations for Application Y from Managers M2 and M3.
- Agent A3 accepts services from any Manager providing the proper authentication token.

Note that the administrative enforcement of access control is different from security services provided by the networking stack carrying AMP messages.

5. Desirable Properties

This section describes those design properties that are desireable when defining an architecture that must operate across challenged links in a network. These properties ensure that network management

capabilities are retained even as delays and disruptions in the network scale. Ultimately, these properties are the driving design principles for the AMA.

5.1. Intelligent Push of Information

Pull management mechanisms require that a Manager send a query to an Agent and then wait for the response to that query. This practice implies a control-session between entities and increases the overall message traffic in the network. Challenged networks cannot guarantee timely roundtrip data-exchange and, in extreme cases, are comprised solely of uni-directional links. Therefore, pull mechanisms must be avoided in favor of push mechanisms.

Push mechanisms, in this context, refer to Agents making their own determinations relating to the information that should be sent to Managers. Such mechanisms do not require round-trip communications as Managers do not request each reporting instance; Managers need only request once, in advance, that information be produced in accordance with a pre-determined schedule or in response to a pre-defined state on the Agent. In this way information is "pushed" from Agents to Managers and the push is "intelligent" because it is based on some internal evaluation performed by the Agent.

5.2. Minimize Message Size Not Node Processing

Protocol designers must balance message size versus message processing time at sending and receiving nodes. Verbose representations of data simplify node processing whereas compact representations require additional activities to generate/parse the compacted message. There is no asynchronous management advantage to minimizing node processing time in a challenged network. However, there is a significant advantage to smaller message sizes in such networks. Compact messages require smaller periods of viable transmission for communication, incur less re-transmission cost, and consume less resources when persistently stored en-route in the network. AMPs should minimize PDUs whenever practical, to include packing and unpacking binary data, variable-length fields, and pre-configured data definitions.

5.3. Absolute Data Identification

Elements within the management system must be uniquely identifiable so that they can be individually manipulated. Identification schemes that are relative to system configuration make data exchange between Agents and Managers difficult as system configurations may change faster than nodes can communicate.

Consider the following SNMP technique for approximating an associative array lookup. A manager wishing to do an associative lookup for some key K1 will (1) query a list of array keys from the agent, (2) find the key that matched K1 and infer the index of K1 from the returned key list, and (3) query the discovered index on the agent to retrieve the desired data.

Ignoring the inefficiency of two pull requests, this mechanism fails when the Agent changes its key-index mapping between the first and second query. Rather than construting an artificial mapping from K1 to an index, an AMP must provide an absolute mechanism to lookup the value K1 without an abstraction between the Agent and Manager.

5.4. Custom Data Definition

Custom definition of new data from existing data (such as through data fusion, averaging, sampling, or other mechanisms) provides the ability to communicate desired information in as compact a form as possible. Specifically, an Agent should not be required to transmit a large data set for a Manager that only wishes to calculate a smaller, inferred data set. The Agent should calculate the smaller data set on its own and transmit that instead. Since the identification of custom data sets is likely to occur in the context of a specific network deployment, AMPs must provide a mechanism for their definition.

5.5. Autonomous Operation

AMA network functions must be achievable using only knowledge local to the Agent. Rather than directly controlling an Agent, a Manager configures the autonomy engine of the Agent to take its own action under the appropriate conditions in accordance with the Agent’s notion of local state and time.

6. Roles and Responsibilities

By definition, Agents reside on managed devices and Managers reside on managing devices. This section describes how these roles participate in the network management functions outlined in the prior section.

6.1. Agent Responsibilities

Application Data Model (ADM) Support
: Agents MUST collect all data, execute all Controls, populate all Report Templates and run operations required by each ADM which the Agent claims to support. Agents MUST report

supported ADMs so that Managers in a network understands what information is understood by what Agent.

Local Data Collection
: Agents MUST collect from local firmware (or other on-board mechanisms) and report all Externally Defined Data defined in all ADMs for which they have been configured.

Autonomous Control
: Agents MUST determine, without Manager intervention, whether a configured Control should be invoked. Agents MUST periodically evaluate the conditions associated with configured Controls and invoke those Controls based on local state. Agents MAY also invoke Controls on other devices for which they act as proxy.

User Data Definition
: Agents MUST provide mechanisms for operators in the network to use configuration services to create customized Variables, Report Templates, Macros and other information in the context of a specific network or network use-case. Agents MUST allow for the creation, listing, and removal of such definitions in accordance with whatever security models are deployed within the particular network.

  Where applicable, Agents MUST verify the validity of these definitions when they are configured and respond in a way consistent with the logging/error-handling policies of the Agent and the network.

Autonomous Reporting
: Agents MUST determine, without real-time Manager intervention, whether and when to populate and transmit a given Report Entry targeted to one or more Managers in the network.

Consolidate Messages
: Agents SHOULD produce as few messages as possible when sending information. For example, rather than sending multiple Report Entry messages to a Manager, an Agent SHOULD prefer to send a single message containing multiple Report Entries.

Regional Proxy
: Agents MAY perform any of their responsibilities on behalf of other network nodes that, themselves, do not have an Agent. In such a configuration, the Agent acts as a proxy for these other network nodes.

6.2. Manager Responsibilities

Agent/ADM Mapping
: Managers MUST understand what ADMs are supported by the various Agents with which they communicate. Managers should not attempt to request, invoke, or refer to ADM information for ADMs unsupported by an agent.

Data Collection
: Managers MUST receive information from Agents by asynchronously configuring the production of data reports and then waiting for, and collecting, responses from Agents over time. Managers MAY try to detect conditions where Agent information has not been received within operationally relevant timespans and react in accordance with network policy.

Custom Definitions
: Managers should provide the ability to define custom definitions. Any custom definitions MUST be transmitted to appropriate Agents and these definitions MUST be remembered to interpret the reporting of these custom values from Agents in the future.

Data Translation
: Managers should provide some interface to other network management protocols, such as the SNMP. Managers MAY accomplish this by accumulating a repository of push-data from high-latency parts of the network from which data may be pulled by low-latency parts of the network.

Data Fusion
: Managers MAY support the fusion of data from multiple Agents with the purpose of transmitting fused data results to other Managers within the network. Managers MAY receive fused reports from other Managers pursuant to appropriate security and administrative configurations.

7. System Model

This section describes the notional data flows and control flows that illustrate how Managers and Agents within an AMA cooperate to perform network management services.

7.1. Control and Data Flows

The AMA identifies three significant data flows: control flows from Managers to Agents, reports flows from Agents to Managers, and fusion reports from Managers to other Managers. These data flows are illustrated in Figure 1.

AMA Control and Data Flows

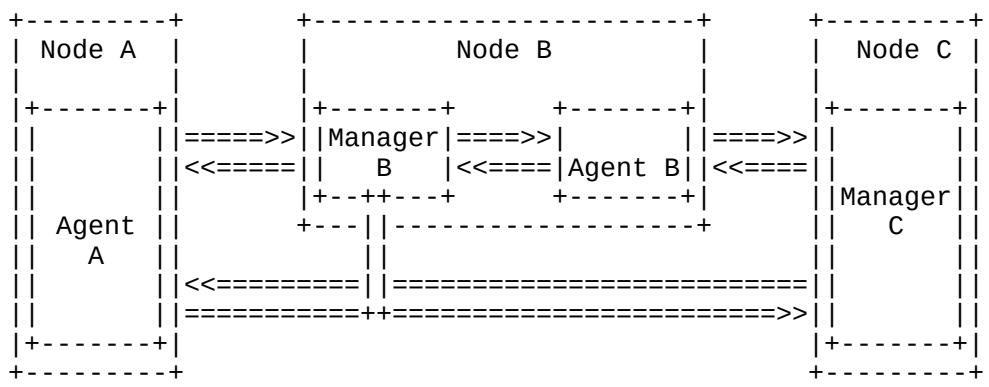


Figure 1

In this data flow, the Agent on node A receives Controls from Managers on nodes B and C, and replies with Report Entries back to these Managers. Similarly, the Agent on node B interacts with the local Manager on node B and the remote Manager on node C. Finally, the Manager on node B may fuse Report Entries received from Agents at nodes A and B and send these fused Report Entries back to the Manager on node C.
From this figure it is clear that there exist many-to-many relationships amongst Managers, amongst Agents, and between Agents and Managers. Note that Agents and Managers are roles, not necessarily differing software applications. Node A may represent a single software application fulfilling only the Agent role, whereas node B may have a single software application fulfilling both the Agent and Manager roles. The specifics of how these roles are realized is an implementation matter.

7.2. Control Flow by Role

This section describes three common configurations of Agents and Managers and the flow of messages between them. These configurations involve local and remote management and data fusion.

7.2.1. Notation

The notation outlined in Table 1 describes the types of control messages exchanged between Agents and Managers.

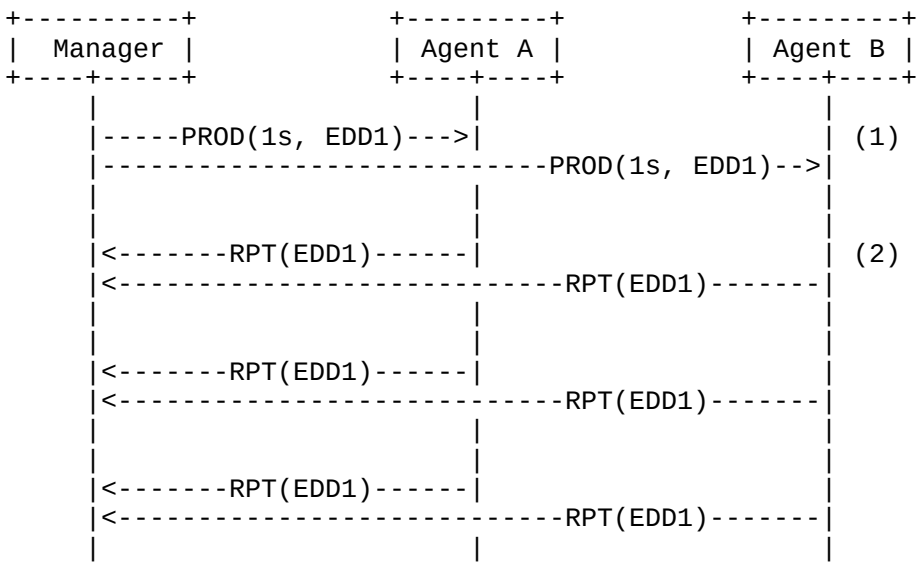
Term	Definition	Example
EDD#	EDD definition, from ADM.	EDD1
V#	Custom data definition.	V1 = EDD1 + V0.
DEF([ACL], ID,EXPR)	Define id from expression. Allow managers in access control list (ACL) to request this id.	DEF([*], V1, EDD1 + EDD2)
PROD(P,ID)	Produce ID according to predicate P. P may be a time period (1s) or an expression (EDD1 > 10).	PROD(1s, EDD1)
RPT(ID)	A report identified by ID.	RPT(EDD1)

Table 1: Terminology

7.2.2. Serialized Management

This is a nominal configuration of network management where a Manager interacts with a set of Agents. The control flows for this are outlined in Figure 2.

Serialized Management Control Flow



In a simple network, a Manager interacts with multiple Agents.

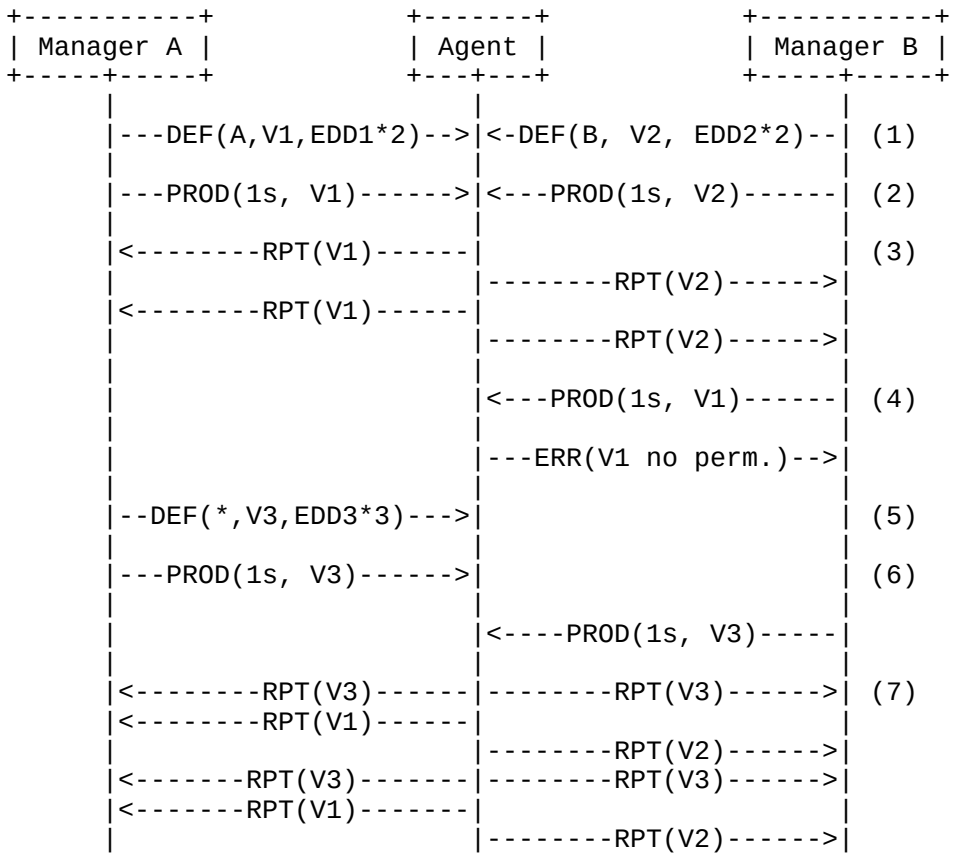
Figure 2

In this figure, the Manager configures Agents A and B to produce EDD1 every second in (1). At some point in the future, upon receiving and configuring this message, Agents A and B then build a Report Entry containing EDD1 and send those reports back to the Manager in (2).

7.2.3. Multiplexed Management

Networks spanning multiple administrative domains may require multiple Managers (for example, one per domain). When a Manager defines custom Report Templates/Variables to an Agent, that definition may be tagged with an access control list (ACL) to limit what other Managers will be privy to this information. Managers in such networks should synchronize with those other Managers granted access to their custom data definitions. When Agents generate messages, they MUST only send messages to Managers according to these ACLs, if present. The control flows in this scenario are outlined in Figure 3.

Multiplexed Management Control Flow



Complex networks require multiple Managers interfacing with Agents.

Figure 3

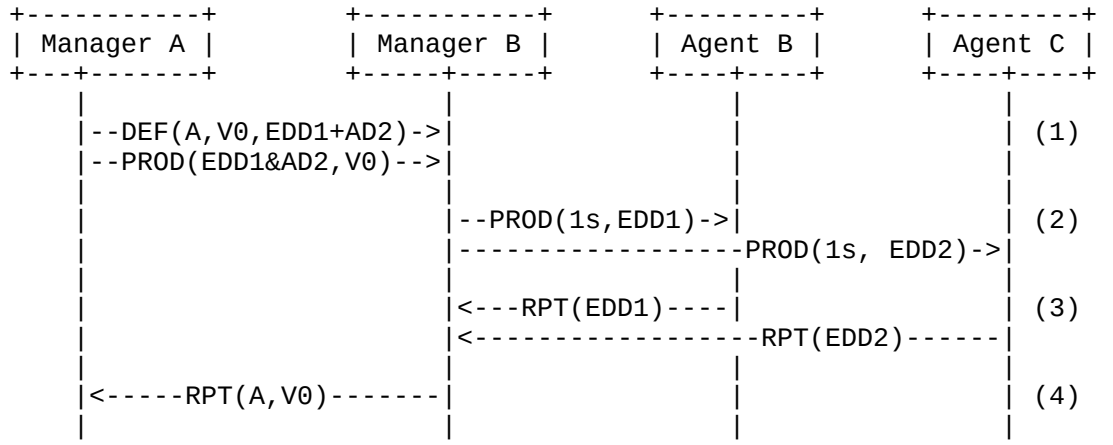
In more complex networks, any Manager may choose to define custom Report Templates and Variables, and Agents may need to accept such definitions from multiple Managers. Variable definitions may include an ACL that describes who may query and otherwise understand these definitions. In (1), Manager A defines V1 only for A while Manager B defines V2 only for B. Managers may, then, request the production of Report Entries containing these definitions, as shown in (2). Agents produce different data for different Managers in accordance with configured production rules, as shown in (3). If a Manager requests the production of a custom definition for which the Manager has no permissions, a response consistent with the configured logging policy on the Agent should be implemented, as shown in (4). Alternatively,

as shown in (5), a Manager may define custom data with no restrictions allowing all other Managers to request and use this definition. This allows all Managers to request the production of Report Entries containing this definition, shown in (6) and have all Managers receive this and other data going forward, as shown in (7).

7.2.4. Data Fusion

In some networks, Agents do not individually transmit their data to a Manager, preferring instead to fuse reporting data with local nodes prior to transmission. This approach reduces the number and size of messages in the network and reduces overall transmission energy expenditure. The AMA supports fusion of NM reports by co-locating Agents and Managers on nodes and offloading fusion activities to the Manager. This process is illustrated in Figure 4.

Data Fusion Control Flow



Data fusion occurs amongst Managers in the network.

Figure 4

In this example, Manager A requires the production of a Variable V0, from node B, as shown in (1). The Manager role understands what data is available from what agents in the subnetwork local to B, understanding that EDD1 is available locally and EDD2 is available remotely. Production messages are produced in (2) and data collected in (3). This allows the Manager at node B to fuse the collected Report Entries into V0 and return it in (4). While a trivial example, the mechanism of associating fusion with the Manager function rather than the Agent function scales with fusion complexity, though it is important to reiterate that Agent and

Manager designations are roles, not individual software components. There may be a single software application running on node B implementing both Manager B and Agent B roles.

8. Logical Data Model

This section identifies the different kinds of information present in an asynchronously-managed network and describes how this information should be communicated in the context of an ADM.

8.1. Data Decomposition

8.1.1. Groups

The AMA supports four basic groups of information: Data, Actions, Literals, and Operators:

Data Data values consist of information collected by an Agent and reported to Managers. This includes definitions from an ADM, derived data values as configured from Managers, and Report Entries which are collections of data elements.

Actions Actions are invoked on Agents and Managers to change behavior in response to some external event (such as local state changes or time). Actions include application-specific functions specified as part of an ADM and macros which are collections of these controls.

Literals Literals are constant numerical values that may be used in the evaluation of expressions and predicates.

Operators Operators are those mathematical functions that operate on series of Data and Literals, such as addition, subtraction, multiplication, and division.

8.1.2. Levels

The AMA defines three levels that describe the origins and multiplicity of data groups within the system. These classifications are atomic, computed, and collection.

Atomic
: The Atomic level contains items computed or defined externally to the AMA and, thus, cannot be changed or otherwise decomposed by Actors within the AMA. These items are described in the context of an ADM and implemented in the context of firmware or software running on an Agent. The

identification of Atomic items MUST be globally unique and should be managed by a registration authority.

Computed
: The Computed level contains items whose definition/value are specified/computed within the scope of an Actor in the AMA. Items at the computed level may be formally specified in an ADM (and therefore have definitions that are not subject to change) or may be defined dynamically on Agents by Managers and therefore have definitions that are subject to change in accordance with configuration services. In either case the definition of a Computed level item may reference other Computed level items and other Atomic level items if such inclusion does not result in a circular reference. When defined in the context of an ADM, a Computed level item MUST be globally unique and should be managed by a registration authority.

Collection
: The Collection level contains items representing groups of other items, including other Collection level items. When a Collection level item definition references another Collection level item, circular references MUST be prevented. When defined in the context of an ADM, a Collection level item MUST be globally unique and should be managed by a registration authority.

8.2. Data Model

Each component of the AMA data model can be identified as a combination of group and level, as illustrated in Table 2. In this table, group/level combinations that are unsupported are listed as N/A. In this context, N/A indicates that the AMA does not require support for groups of data at a particular level for compliance.

	Data	Action	Literals	Operator
Atomic	Externally Defined Data	Control	Literal	Operator
Computed	Variable	Rule	N/A	N/A
Collection	Report Entry	Macro	N/A	N/A

Table 2

The eight elements of the AMA logical data model are described as follows.

8.2.1. EDDs, VARs, and Reporting

Fundamental to any performance reporting function is the ability to measure the state of the Agent. Measurement may be accomplished through direct sampling of hardware, query against in-situ data stores, or other mechanisms that provide the initial quantification of state.

EDDs serve as the "lingua franca" of the management system: the unit of information that cannot be otherwise created. As such, this information serves as the basis for any user-defined (Variable) values in the system.

AMPs MAY consider the concept of the confidence of the EDD as a function of time. For example, to understand at which point a measurement should be considered stale and need to be re-measured before acting on the associated data.

While EDDs provide the full, raw set of information available to Managers and Agents there is a performance optimization to pre-computed re-used combinations of these values. Computing new values as a function of measured values simplifies operator specifications and prevents Agent implementations from continuously re-calculating the same value each time it is used in a given time period.

For example, consider a sensor node which wishes to report a temperature averaged over the past 10 measurements. An Agent may either transmit all 10 measurements to a Manager, or calculate locally the average measurement and transmit the "fused" data. Clearly, the decision to reduce data volume is highly coupled to the nature of the science and the resources of the network. For this reason, the ability to define custom computations per deployment is necessary.

Periodically, or in accordance with local state changes, Agents must collect a series of measured values and computed values and communicate them back to Managers. This ordered collection of value information is noted in this architecture as a Report Entry which populates either a pre-defined or ad-hoc Report Template. In support of hierarchical definitions, Report Entries may, themselves, contain other Report Entries. It would be incumbent on an AMP implementation to guard against circular reference in Report Template definitions.

8.2.2. Controls and Macros

Just as traditional network management approaches provide well-known identifiers for values, the AMA provides well-known identifiers for Actions. Whereas several low-latency, high-availability approaches in networks can use approaches such as remote procedure calls (RPCs), challenged networks cannot provide a similar function - Managers cannot be in the processing loop of an Agent when the Agent is not in communication with the Manager.

Controls in a system are the combination of a well-known operation that can be taken by an Agent as well as any parameters that are necessary for the proper execution of that function. For specific applications or protocols, a control specification (as a series of opcodes) can be published such that any implementing AMP accepts these opcodes and understands that sending the opcodes to an Agent supporting the application or protocol will properly execute the associated function. Parameters to such functions are provided in real-time by either Managers requesting that a control be run, pre-configured, or auto-populated by the Agent in-situ.

Often, a series of controls must be executed in concert to achieve a particular function, especially when controls represent more primitive operations for a particular application/protocol. In such scenarios, an ordered collection of controls can be specified as a Macro. In support of the hierarchical build-up of functionality, Macros may, themselves, contain other Macros, through it would be incumbent on an AMP implementation to guard against excessive recursion or other resource-intensive nesting.

8.2.3. Rules

Stimulus-response autonomy systems provide a way to pre-configure responses to anticipated events. Such a mapping from responses to events is advantageous in a challenged network for a variety of reasons, as listed below.

- Distributed Operation - The concept of pre-configuration allows the Agent to operate without regular contact with Managers in the system. Configuration opportunities will be sporadic in any challenged network making bootstrapping of the system difficult, but this is a fundamental problem in any network scenario and any autonomy approach.

- Deterministic Behavior - Where the mapping of stimulus to response is stable, the behavior of the Agent to a variety of in-situ state also remains stable. This stable behavior is necessary in

critical operational systems where the actions of a platform must be well understood even in the absence of an operator in the loop.

- Engine-Based Behavior - Several operational systems are unable to deploy "mobile code" based solutions due to network bandwidth, memory or processor loading, or security concerns. The benefit of engine-based approaches is that the configuration inputs to the engine can be flexible without incurring a set of problematic requirements or concerns.

The logical unit of stimulus-response autonomy proposed in the AMA is a Rule of the form:
IF stimulus THEN response
Where the set of such rules, when evaluated in some prioritized sequence, provides the full set of autonomous behavior for an Agent. Stimulus in such a system would either be a function of relative time, absolute time, or some mathematical expression comprising one or more values (measurement values or computed values).

Notably, in such a system, stimuli and responses from multiple applications and protocols may be combined to provide an expressive capability.

8.2.4. Operators and Literals

Computing values or evaluating expressions requires applying mathematical operations to data known to the management system.

Operators in the AMA represent enumerated mathematical operations applied to primitive and computed values in the AMA for the purpose of creating new values. Operations may be simple binary operations such as "A + B" or more complex functions such as sin(A) or avg(A,B,C,D).

Literals represent pre-configured constants in the AMA, such as well-known mathematical numbers (e.g., PI, E), or other useful data such as Epoch times. Literals also represent asserted Primitive Values used in expressions. For example, considering the expression (A = B + 10), A would be a Variable, B would be either a Variable or EDD, + would be an Operator, and 10 would be a Literal.

8.3. Application Data Model

Application data models (ADMs) specify the data associated with a particular application/protocol. The purpose of the ADM is to provide a published interface for the management of an application or protocol independent of the nuances of its software implementation. In this respect, the ADM is conceptually similar to the Managed

Information Base (MIB) used by SNMP, but contains additional information relating to command opcodes and more expressive syntax for automated behavior.

An ADM MUST define all well-known items necessary to manage the specific application or protocol. This includes the definitions of EDDs, Variables, Report Templates, Controls, Macros, Rules, Literals, and Operators.

9. IANA Considerations

At this time, this protocol has no fields registered by IANA.

10. Security Considerations

Security within an AMA MUST exist in two layers: transport layer security and access control.

Transport-layer security addresses the questions of authentication, integrity, and confidentiality associated with the transport of messages between and amongst Managers and Agents in the AMA. This security is applied before any particular Actor in the system receives data and, therefore, is outside of the scope of this document.

Finer grain application security is done via ACLs which are defined via configuration messages and implementation specific.

11. Informative References

[BIRRANE1]
Birrane, E. and R. Cole, "Management of Disruption-Tolerant Networks: A Systems Engineering Approach", 2010.

[BIRRANE2]
Birrane, E., Burleigh, S., and V. Cerf, "Defining Tolerance: Impacts of Delay and Disruption when Managing Challenged Networks", 2011.

[BIRRANE3]
Birrane, E. and H. Kruse, "Delay-Tolerant Network Management: The Definition and Exchange of Infrastructure Information in High Delay Environments", 2011.

[I-D.irtf-dtnrg-dtnmp]
Birrane, E. and V. Ramachandran, "Delay Tolerant Network Management Protocol", draft-irtf-dtnrg-dtnmp-01 (work in progress), December 2014.

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

[RFC3416] Presuhn, R., Ed., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, DOI 10.17487/RFC3416, December 2002, <http://www.rfc-editor.org/info/rfc3416>.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[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, <http://www.rfc-editor.org/info/rfc6241>.

Author’s Address

Edward J. Birrane
Johns Hopkins Applied Physics Laboratory

Email: Edward.Birrane@jhuapl.edu