Asynchronous Mgmt Architecture (AMA) & Asynchronous Mgmt Protocol (AMP) Updates

Edward Birrane
Edward.Birrane@jhuapl.edu
443-778-7423
AMA: Overview

From draft-birrane-dtn-ama-03

- **Service Definitions**
  - **Configuration**: Change settings on an Agent.
  - **Reporting**: Receive performance information from an Agent.
  - **Autonomous Parameterized Control**: Change Agent Behavior.
  - **Administration**: Fine-grained access to abilities.

- **Desirable Properties**
  - **Intelligent Information Push**: Can’t rely on others.
  - **Minimize Message Size**: Increase probability of delivery.
  - **Absolute Data Identification**: pre-shared, global naming when possible.
  - **Custom Data Definition**: Only send minimal necessary data sets.
  - **Autonomous Operation**: Decisions local to Agent based on its config.
AMA: System Model

From draft-birrane-dtn-ama-03

- **Agents**
  - Run on Managed Devices
  - Configure/Report on devices
  - Heavy autonomy and parameterized control

- **Manager(s)**
  - Collect/Fuse data from Agents
  - Configure Agent behavior
  - Open-loop control

- **ADM(s)**
  - Well-named Data and Controls
  - Superset of MIB
  - Move to describe them in YANG
  - Preconfiguration reduces msg size
AMA: App. Data Model (ADM)

- **“Atomic” Elements**
  - Solely defined by their ADM.
  - **EDDs**: collected by agents.
  - **Literals**: useful constants.
  - **Ops**: opcodes for math functions.
  - **Ctrls**: opcodes for agent behavior.

- **“Variable” Elements**
  - Defined by ADM or by User
  - ADM definitions are immutable.
  - **Vars**: strong-typed variables, including a type for “expression”.
  - **Macro**: Ordered set ofCtrls.
  - **Rpts**: Ordered sets of data
  - **Rules**: Time or State based autonomy.

An ADM defined 8 types of data for each application/protocol managed in the AMA.
Minor Terminology and Definitions Updates
- Mostly wordsmithing based on feedback.
- Clarify distinction between a Report Template, a Report Entry, and a Report as a collection of entries.
- Some expanded text around parameterization and motivation for the approach.

No significant issues or limitations with the architecture.
- No “structural” changes to the architecture.

Primary focus has been on AMP and associated ADMs.
AMA: TODO

Propose AMA provides the architecture and required functions of a DTN Network Management Protocol.

- Some discussion of terminology
  - Atomic Data vs. Primitive Data vs. Externally Defined Data
  - Computed Data vs Variables
  - Specs out of sync on terminology.
    - Need a refresh across related drafts to synchronize them.

- OPs AD feedback
  - Reviewing RESTful NETCONF and YANG Push models
  - Review to date does not seem to change the AMA.
    - May also not change the AMP, which is considered separately.

- Request AMA be considered by the WG when it is time to address Network Management for DTNs.
AMP being evaluated by space and non-space users. NASA providing an open-source reference implementation in ION.

- **Protocol conformant to the architecture/requirements of AMA.**
  - Implements Agents, Managers, ADM structures.
  - Defines specific data models to implement AMA structures.
  - Defines messages to capture AMA controls/reports/administration.
  - Defines on-the-wire encodings.

- **Data Models**
  - Basic Types: Numeric types, strings, etc…
  - Compound Types: BLOBs, (Typed) Data Collections, Tables, Identifiers, Collections, Expressions, Predicates.

- **Functional Specification**
  - AMP Message Groups: Common headers and trailers.
  - Three messages: RegisterAgent, PerformControl, DataReport.
AMP: Updates (1/2)

From -02 to -03

- **Minor Terminology and Definitions Updates**
  - Wordsmithing based on feedback.
  - Reduced redundancy between AMP and AMA specs.

- **Clarifications**
  - Clarified Report Templates vs Report Entries vs Reports.
  - Clarified State vs Time-based Rules.
  - Corrected AMP Epoch time.
  - Added rationale for design of TDCs.
  - Clarified that OID Nicknames are registered values.
  - Clarified OID Parameterization Approach
  - Clarified definition of Variables and their initializing expression.
AMP: Updates (2/2)

From -02 to -03

- **Additions/Updates**
  - Added Table AMP structure.
  - Added Result Type to Expression structure.
  - Added required levels of Macro nesting.
  - Updated type enumerations.
  - Added allowed numerical promotions
  - Added rules for numeric conversions
  - Updated format of DataReport message.

- **Removals**
  - Removed draft design of N of M counts for SRLs.
  - Removed enable/disable from SRL and TRL structures
AMP: TODO

From -02 to -03

- **Upcoming Spec Changes**
  - How best to add N of M and enabled/disabled to SRL/TRLs
  - Change TDC column IDs to be of any type, not just string.
  - Add Access Control Lists (ACLs) and describe behavior.
  - Transition to CBOR for encoding.
  - Add guidance in ADM section on when to define TABLEs versus EDDs vs Controls that return data.
  - Should AMP specify a wire encoding?

- **More Review from Reference Implementations**
  - Continued support of reference implementation efforts
  - At last count there were 4 separate implementation efforts
    - Discussions on 2 additional efforts.
AMA/AMP Related Specifications

- **Core Specs**
  - AMA: draft-birrane-dtn-ama-03
  - AMP: draft-birrane-dtn-amp-03

- **ADMs**
  - AMP Agent ADM: draft-birrane-dtn-adm-agent-02
  - BPSEC ADM: draft-birrane-dtn-adm-bpsec-00
  - BP ADM: draft-birrane-dtn-adm-bp-00
  - YANG profile for ADMs: draft-bsipos-dtn-amp-yang-01

- **Other:**
  - AMP Manager SQL Schema: draft-birrane-dtn-ampmgr-sql-00
Current Status

NASA building out AMP for deployment to ISS and other infusion targets

- Reference implementation in ION open source this year.
  - Supporting AMP protocol messages, Agent, BP, BPSEC ADMs.
- NASA supporting AMA/AMP ongoing work
  - Writing ADMs for BP, BSP, CGR, LTP, and ION.
- Several non-NASA efforts ongoing.
  - AMP is not directly tied to BP or DTN, though it is very helpful for DTN use cases.
- Finalizing AMA and AMP specs for consideration in DTNWG
  - As novel intersection between performance monitoring and safing autonomy
  - Meeting with OPS AD people as they are identified to discuss AMP vs RESTful NETCONF and YANG Push.
Every AMP structure identified and parameterized by a Managed Identifier (MID).

- **Concept**
  - Every AMP structure is identified by a MID.
  - Simplifies processing.
  - Hardware acceleration.

- **Issuer/Tag**
  - MIDs defined in ADM. have no Issuer/Tag.
  - User-defined MIDs must have issuer ID.
  - Tags always optional.

- **Parameters**
  - Captured in the MID itself.

Control MID: Generate Agent ADM Report
0xc304010903021517050182030100020100

Control MID: List VARs known to Agent
0x83040103

Var. MID: User-Defined UINT variable.
0x110103010203
AMA Roles and Responsibilities

AMA Data Flows

Node A

Manager B

Agent A

Node B

Manager C

Agent B

Node C
In a simple network, a Manager interacts with multiple Agents.
AMA Multi-Manager Flow

Multiplexed Management Control Flow

Manager A  |  Agent  |  Manager B

---DEF(A, CD1, AD1*2)---  <-DEF(B, CD2, AD2*2)--- (Step 1)

---PROD(1s, CD1)------  <-PROD(1s, CD2)------ (Step 2)

<--------RPT(CD1)------  <--------RPT(CD2)------ (Step 3)

<--------RPT(CD1)------  <--------RPT(CD2)------

<--------RPT(CD1)------  <--------RPT(CD2)------

<--------RPT(CD1)------  <--------RPT(CD2)------

---DEF(*, CD3, AD3*3)---  ERR(CD1 no perm.)-->

---PROD(1s, CD3)------ (Step 5)

<--------RPT(CD3)------ (Step 6)

<--------RPT(CD3)------  <--------RPT(CD3)------ (Step 7)

<--------RPT(CD1)------

<--------RPT(CD1)------

<--------RPT(CD1)------

<--------RPT(CD1)------

<--------RPT(CD1)------

<--------RPT(CD1)------

<--------RPT(CD1)------
Data fusion occurs amongst Managers in the network.
Compatibility with existing mechanism

- **SNMP Uses OIDs as IDs**
  - Global, Managed Tree Structure
    - “Path to data” is concatenation of #s.
    - \( \text{ifSpeed} = 1.3.6.1.2.1.2.2.1.8 \)
  - Supports Binary Encoding (BER)
    - Compress first 2 #s: 1.3 => 43
    - SDNV-encode rest
  - SNMP Identifier: \(<\text{type}> <\text{length}> <\text{value}>\)
    - Type 6 -> OID
    - Length (in this case) = 9 bytes
    - \( \text{ifSpeed} = 0x06092C0601020102020108 \)

- **AMP Uses MIDS (Managed IDs)**
  - MIDS encapsulate OIDs (less \(<\text{type}>\) field)
  - Option to compress OID
  - Makes easy to interoperate with SNMP
- **Full OID**
  - Length + Octets
  - Not interpreted by AMP. Used as a unique bitstream.
  - Encoded in ASN.1 BER for now, assuming SNMP Type 6.

- **Parameterized OID**
  - Full OID followed by AMP Data Collection (DC).
  - DC is a count followed by a series of TLV.
    - *Time, Length, Value*
    - *Type is data type (string, int)*
OID Types (2/2)

- **Compressed OID**
  - AMP supports managed registry of common OID sets.
    - *OIDs can be very long and the portion up to your relative subtree can be reused a lot.*
  - Nickname is an integer that maps to a well-known node in an OID tree.
    - *Relative OID is subtree rooted at that node.*

- **Compressed, Parameterized OID**
  - Compressed OID followed by a Data Collection of Parameters
  - Very similar to a Parameterized OID.
AMP ADMs capture all necessary information for each supported application or protocol..

- **Atomic Data and Controls.**
  - What immutable data definitions are given for any manager/agent supporting a particular application?
  - What common actions can be taken to manage this application?

- **Literals and Operators.**
  - What constants are defined for this application?
  - What special operators can be used to compute new data definitions?

- **Computed Data.**
  - What data definitions are pre-derived from other data definitions?

- **Collections.**
  - What pre-defined collections of data values (reports) and control sequences (macros) have been created?
ADM Example (1)

- Pre-defined, atomic data
  - Definitions from MIBs
    - Global, unique OIDs
    - No tag/issuer fields
    - All data and reports
  - Build blocks for user content
    - Data MIDs can be used in user definitions

- Pre-defined controls
  - Also global, unique OIDs
  - Opcodes, description, params
  - Build blocks for macro commands
    - No ability for user-defined controls outside of these pre-defined functions.

Bundle Protocol ADM

<table>
<thead>
<tr>
<th>Atomic Data</th>
<th>Reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID1 = ExpiredBundleCount</td>
<td>MID5 = MID1, MID2</td>
</tr>
<tr>
<td>MID2 = CustodyAcceptCount</td>
<td>MID6 = MID5, MID3, MID4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Computed Data</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID3 = MID1 + MID2</td>
<td>MID7 = ClearBundleCnt()</td>
</tr>
<tr>
<td>MID4 = AVG(MID3, 10s)</td>
<td>MID8 = ClearAcceptCnt()</td>
</tr>
</tbody>
</table>
ADM Example (2)

A sample ADM for an application implementing a stack.

<table>
<thead>
<tr>
<th>Atomic Controls</th>
<th>Computed Data</th>
<th>Atomic Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>- PUSH(X)</td>
<td>- Average POPs</td>
<td>- Stack Depth</td>
</tr>
<tr>
<td>- POP(X)</td>
<td></td>
<td>- Total Items</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Literals</th>
<th>Data Collections</th>
<th>Control Collections</th>
</tr>
</thead>
<tbody>
<tr>
<td>- MAX_DEPTH = 10</td>
<td>Report 1:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Cur. Stack Depth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Total Items</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Average POPs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EMPTY:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stack Depth &gt; 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POP(X)</td>
<td></td>
</tr>
</tbody>
</table>
Captures all behavior of an AMP Agent
- Keeps AMP functional specification simple
- Items available to AMA/AMP ecosystem because this ADM must be implemented by any deployed AMP agent.

Primitive Values
- Counters, number of AMP types created, active, etc…

Reports
- Full report definitions. Users may customize their own.

Controls
- All functions to create, update, delete, and otherwise manage reports, rules, macros, and other AMA types.

Operators
- Full math function spec
  - +, -, *, /, %, ^, &, |, &&, ||, !, abs(), <, >, <=, >=, !=, ==, >>, <<
Thank you!

Questions?