JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

AMM/ADM/ARI/AMP Updates for DTNMA

IETF 120 DTN WG

Brian Sipos, Justin Ethier, Jenny Cao
JHU/APL

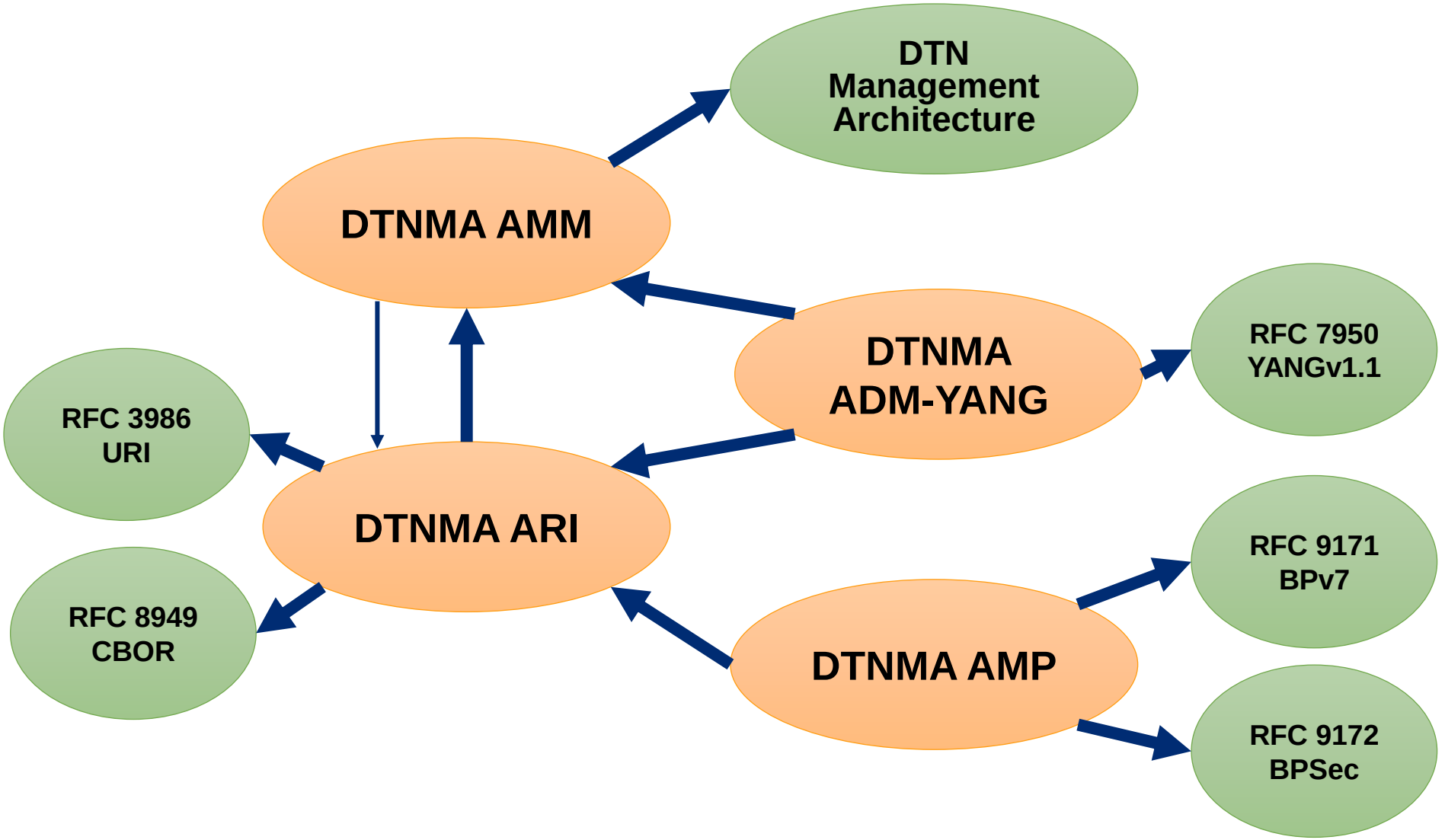
Background

- The DTNMA draft draft-ietf-dtn-dtnma-14 is in the Editor's queue
- The AMM/ADM/ARI documents have been adopted by WG
  - AMM document draft-ietf-dtn-amm-01 has been refined down to a few number of TBDs within specific activities
  - The YANG-encoded ADM module is now a separate document draft-ietf-dtn-adm-yang-01
  - ARI document draft-ietf-dtn-ari-02 is updated with some refinements
- The AMP document has been updated for latest ARI and for BPv7
  - This is still personal draft draft-birrane-dtn-amp-09
  - Significantly smaller; 40 pages down to 12 pages
- Existing implementation of earlier drafts of DTNMA (then called AMP) are part of NASA-AMMOS ANMS and APL DTNMA Tools
- Work has begun on updating implementations to the latest drafts

DTNMA Topic Areas

- Application Management Model: What is being managed, structural and behavioral definitions
  - Object types
  - Data values and structure
  - Built-in value types, semantic value types
  - Agent and Manager Activities
- Application Data Models: Static AMM object instances
  - Representation (*e.g.* YANG-syntax module)
  - Base semantic types (*e.g.* AMM module)
- Operational Data Models: Runtime AMM object instances
  - Representation (*e.g.* Agent introspection)
- Data Value Exchange: ARI structure
  - Representation
  - Transport bindings, including security requirements
- Agent State Management
  - Initial “Agent ADM” and “Access Control ADM”

Normative Dependencies



Recent Changes part 1

- Separated AMM document from ADM encoding
- In AMM document:
  - Clarified “built-in type” terminology
  - Added explicit listing of semantic type classes (*e.g.*, type use vs. type union)
  - Added notion of display hint annotation, similar to units annotation
  - Defined specific categories of controls and operators in the Agent ADM
  - Added specific procedures for built-in and semantic type matching and converting
- In ADM-YANG document:
  - Raised up section depths to expand table of contents
  - Improved cross-references to module-level statements
  - More cleanly separated AMM logical definitions from ADM encoding choices
  - More consistently qualified AMM extension keywords
  - Use ARIs as arguments for type-naming statements
- In base ADMs:
  - Added initial display hint IDENT hierarchy (could be moved to separate ADM)
  - Added `tbl_filter` control to Agent ADM for reducing table size

Recent Changes part 2

- In ARI document:
  - Corrected typos in text
  - Fixed some CDDL issues and ABNF issues
  - Changed text form object reference to use URI “authority” part for namespace
  - Added ability to reference just a namespace
  - Removed the concept of an implied built-in type; literals can just be untyped
  - Updated floating point text representation to allow IEEE 754-2008 standard hexadecimal, which is a fast and lossless conversion
  - Removed ability to use base32 representation for BYTESTR
  - Specific character set of base64**url** for BYTESTR to avoid percent encoding
  - Added more encoded examples
  - Added explicit procedure for ID Segment Translation
  - Added appendix for Implementation Guidance related to text processing
- In AMP document:
  - Brought back to life after a long expiry time
  - Greatly simplified the content to transporting and authenticating EXECSET and RPTSET values between DTNMA entities
  - Uses BPv7 for transport and BPSec for source authentication

Open Issues

- Validation with `pyang` enforcing RFC 8407 guidelines
  - Submitted upstream issue #912 with potential patch for ADM modules
- Type Display Hints
  - Similar to the pre-existing notion of “units” text annotation, but for display of the value itself
  - Does this belong in a different ADM than the base `ietf-amm` ?
- Hierarchy in Namespaces
  - Is there value in having a two-level namespace similar to IPN EID updates?
  - This can be handled in a consistent way while still following YANG module naming conventions
  - Use existing “organization” metadata: `ietf-dtnma-agent` ⇔ 1.1
  - Do ODMs also need two-level hierarchy? Or are all ODMs under one parent?
- Access Control Lists (ACL)
  - Discussion needed about the ACL ADM
- ARI Reference resolution
  - ARI References resolving points are still TBD
  - Are runtime stores allowed to contain ARI References, or are must they be resolved before entering the runtime?

Implementation Experience

- Aspects of the new changes have been prototyped using ANMS' ACE (Python library) and DTNMA Tools (C99 library) as a basis for encoding/decoding ARIs and ADMs
- Prototyping has led to improvements in usability for the ADM syntax especially, using ARI as the principal form of cross-reference to AMM objects and type names
- Longer-term plan is to eventually treat these updates as "external" contributions to those open source projects, which is the sustainment mechanism for the associated libraries

Next Steps

- Close any more gaps in the AMM/ADM/ARI documents to make them have complete definitions for all behavior
- Gather more implementation experience of for new ARI and ADM representations
- Prepare some changes / patches for `pyang` (also used by the IETF Datatracker)
- Adopt the simple AMP draft to the DTN WG to complete the document cluster