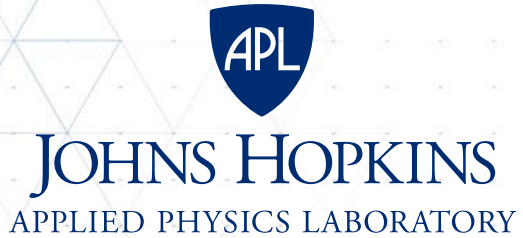


11100 Johns Hopkins Road
Laurel, MD 20723-6099

IETF 117
DTN Management Architecture

DTNMA Review Updates

Sarah Heiner
Johns Hopkins University, Applied Physics Laboratory (JHU/APL)
Sarah.Heiner@jhuapl.edu

Overview

- Current Version 06
- Review Feedback
- Document Updates
- Autonomy Engine
- Rule-Based Management
  - Structure
  - Evaluation

Workgroup: Delay-Tolerant Networking
Internet-Draft: draft-ietf-dtn-dtnma-06
Published: 8 July 2023
Intended Status: Informational
Expires: 9 January 2024
Authors:
E.J. Birrane
Johns Hopkins Applied Physics Laboratory
S.E. Heiner
Johns Hopkins Applied Physics Laboratory
E. Annis
Johns Hopkins Applied Physics Laboratory

DTN Management Architecture

Abstract

The Delay-Tolerant Networking (DTN) architecture describes a type of challenged network in which communications may be significantly affected by long signal propagation delays, frequent link disruptions, or both. The unique characteristics of this environment require a unique approach to network management that supports asynchronous transport, autonomous local control, and a small footprint (in both resources and dependencies) so as to deploy on constrained devices.

This document describes a DTN management architecture (DTNMA) suitable for managing devices in any challenged environment but, in particular, those communicating using the DTN Bundle Protocol (BP). Operating over BP requires an architecture that neither presumes synchronized transport behavior nor relies on query-response mechanisms. Implementations compliant with this DTNMA should expect to successfully operate in extremely challenging conditions, such as over uni-directional links and other places where BP is the preferred transport.

Current Network Management Approaches

Additional references, background, and considerations

- Updated discussion of existing network management approaches
  - SNMP considerations
  - Operational State modeling (configured state vs. applied configuration state)

- Clarified discussion of SNMP event notifications
  - Used to decrease reaction time
  - Emphasizes the need for autonomy in the DTNMA

- Added background from several references:
  - RFC 8342 Network Management Datastore Architecture (NMDA)
  - RFC 2982 Distributed Management Expression MIB
  - RFC 3165 Definitions of Managed Objects for the Delegation of Management Scripts
  - RFC 3410 Introduction and Applicability Statements for Internet Standard Management Framework

DTNMA Document Scope

Move management/data model considerations

- **DTNMA is an architecture document**
- **Separation from DTNMA Application Data Model**
  - **Application Management Model**
  - **Operational Data Models**
- **No underlying transport protocol requirement**
- **Designed to run in every environment BP may be used**

Workgroup:	Delay-Tolerant Networking
Internet-Draft:	draft-birrane-dtn-adm-04
Published:	10 July 2023
Intended Status:	Standards Track
Expires:	11 January 2024

Authors:	E.J. Birrane	D. Linko	B. Sipos
	JHU/APL	*JHU/APL*	*JHU/APL*

DTNMA Application Data Model

Abstract

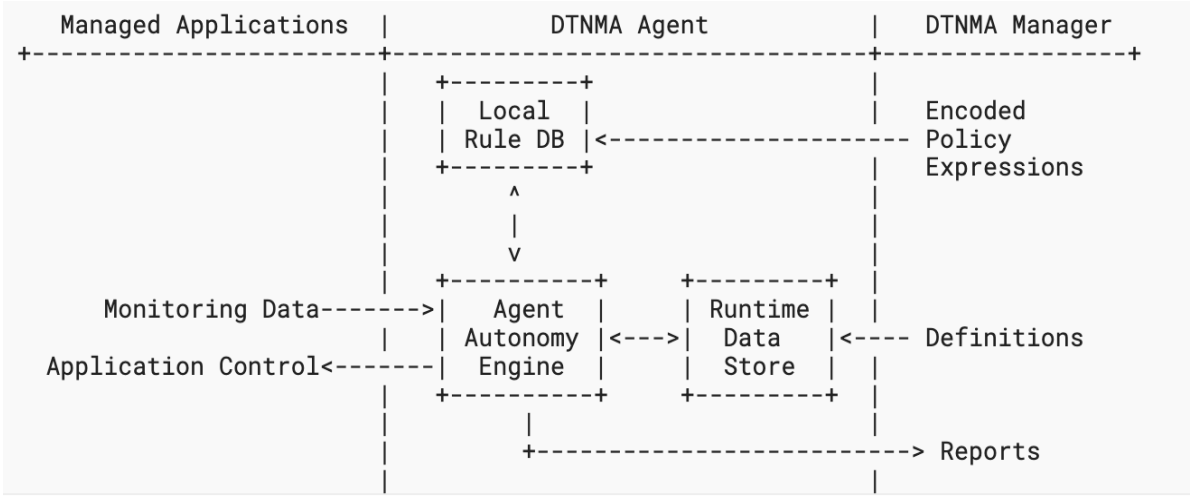
This document defines a data model that captures the information necessary to asynchronously manage applications. This model provides a set of common type definitions, data structures, and a template for publishing standardized representations of model elements.

DTNMA Agent Autonomy Engine

Configuration and Responsibilities

- Acts as a policy execution engine:
  - Interpret and execute rules
  - Monitor managed device state
  - Execute response defined in associated rule

- DTNMA Manager can configure:
  - **Rules Database**
    - Define a new rule
    - Based on policy expressions from managing device
  - **Autonomy Engine**
    - Enable/disable a rule set



DTNMA Autonomy Model

Command-Based Management

Rule-Based Stimulus-Response System

- Feedback: “Command-based” terminology leaves room for interpretation
- Managed devices must issue commands on themselves – as if they are being controlled by a managing device
- Use commands so a DA can receive updates from:
  - Remote management devices (DMs)
  - DA local autonomy engine
- Command execution
  - Control
    - Parameterized, predefined procedure
    - Conceptually *similar* to RPC
    - Do NOT have a concept of a return code
  - Macro
    - Ordered sequence of controls

“The DTNMA autonomy model is a **rule-based model** in which individual rules associate a pre-identified stimulus with a pre-configured response to that stimulus.”

DTNMA Section 9.1 Overview of Logical Autonomy Model

Predicate Autonomy Rules

Rule stimuli support bulk evaluation

- Feedback: consider event-condition-action rules
  - ON <event> IF <condition> THEN <action>
- Pre-identified stimulus
  - One or more predicate logic expressions
  - Based on state of managed device or passage of time
  - **Simple:** Single predicate expression
  - **Complex:** Common condition shared across rules, with specific condition tied to each rule
- Pre-configured response
  - One or more procedures
  - Controls/Macros
- Reduction in total number of predicate evaluations

IF <condition 1> THEN <response 1>

Single Predicate Expression

IF <common condition> THEN

IF <specific condition 1> THEN <response 1>

IF <specific condition 2> THEN <response 2>

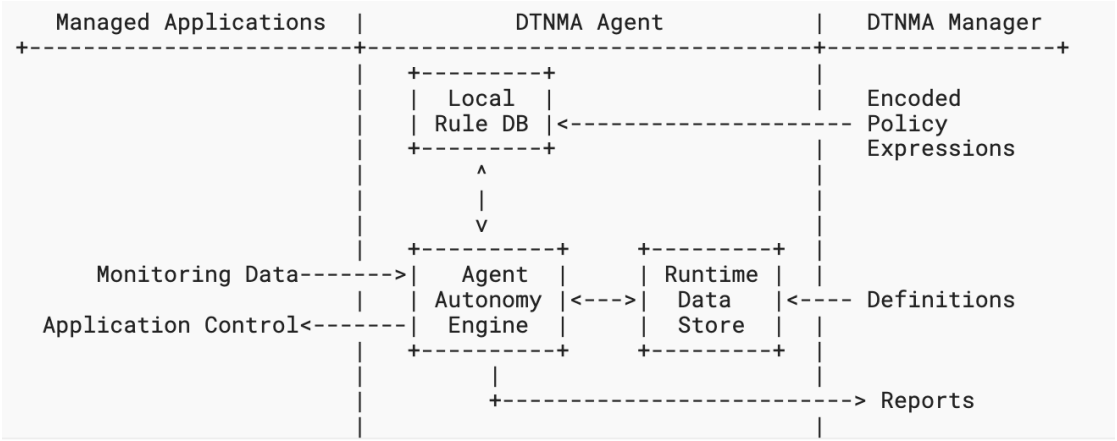
IF <specific condition 3> THEN <response 3>

Shared Condition for Bulk Evaluation

(1/2) Handling Large Rule Sets

Potential challenges addressed by DTNMA Autonomy Model characteristics

- **Strong Typing**
  - Avoid implicit data conversion errors
  - Detect misconfigurations
- **Acyclic Dependency**
  - Prevent circular dependencies
  - Permit combined expressions
  - Create and support complex behaviors
- **Fresh Data**
  - Indicate if data value represents current device state
  - Avoid incorrectly inferring operational state

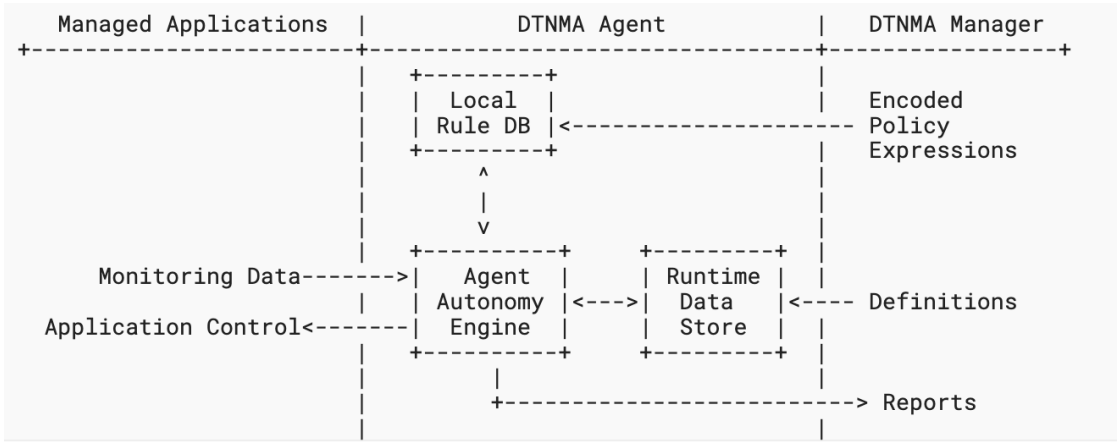


DTNMA Autonomy Model

(2/2) Handling Large Rule Sets

Potential challenges addressed by DTNMA Autonomy Model characteristics

- **Pervasive Parameterization**
  - Flexibility for autonomy model objects
  - Promotes fewer unique objects
  - Allows substitution of local device state in controls/reports
- **Configurable Cardinality**
  - Limit the number of supported objects
  - Based on available resources
- **Control-Based Updates**
  - Change state of managed device via controls
  - Allows all pre-conditions to be checked
  - Enables finer-grained and bulk updates



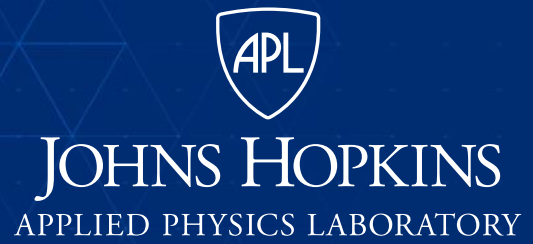
DTNMA Autonomy Model

Conflict Detection and Resolution

Multiple Managers may lead to conflicting rules

- DTNMA supports multiple managers
- Address conflict/stability concerns
- Implementations should include mechanisms to inform the selection of one manager input over others

Mechanism	Explanation	Sample Usage
Analysis of Managed Content	Select Manager based on some managed state data.	Select Manager whose configuration data matches a set of managed elements.
Time	Time considerations may include the time a rule was received by the Agent or defined by a Manager.	The most recent rule takes precedence over other conflicting rules.
Agent Location	Use Agent location in the network to choose Manager input.	Select the Manager in the same administrative domain as the Agent.



11100 Johns Hopkins Road
Laurel, MD 20723-6099

IETF 117
DTN Management Architecture

DTNMA – Requesting Working Group Last Call

Sarah Heiner
Johns Hopkins University, Applied Physics Laboratory (JHU/APL)
Sarah.Heiner@jhuapl.edu