Yang Data Model for OAM and Management of ALTO protocol

draft-zhang-alto-oam-yang

Jingxuan Zhang
Dhruv Dhody
Roland Schott
Kai Gao

Presenter: Mahdi Soleimani

ALTO WG @ IETF 114
Current Status

Main goal: Define a YANG data model for Operations, Administration, and Maintenance (OAM) & Management of ALTO Protocol.


Editor’s copy on GitHub: https://ietf-wg-alto.github.io/draft-ietf-alto-oam-yang/draft-ietf-alto-oam-yang.html

Current Status (Cont.)

Comments/Reviews since IETT 113:

● Received 5 reviews/comments from WG mailing list:
  ○ Many good suggestions on measurement, server setup, logging management, etc.
  ○ Thanks for Qiufang Ma, Qin Wu, and Chongfeng Xie

● 1 discussion with netmod WG is continuing
  ○ Tom Petch provided some intelligent analysis about whether to use IANA maintained YANG modules

● 5 discussions on GitHub:

Achieved Milestones:

● WG document adoption
  ○ Latest version draft-ietf-alto-oam-yang-01 improved YANG model and addressed part of comments from WG

● IETF 114 Hackathon
  ○ Concepts proof in OpenALTO implementation: [https://github.com/openalto/aldo](https://github.com/openalto/aldo)
Data Model Overview

A reference ALTO server architecture:

- **Performance Monitor:** ietf-alto-stats.yang
- **Logging and Fault Manager:** ietf-alto-stats.yang
- **Server Manager:** ietf-alto.yang
- **Information Resource Manager:** ietf-alto.yang
- **Algorithm Plugin:** example-ietf-alto-alg.yang
- **Data Broker**
- **Data Source Listener:** ietf-alto.yang
- **Data Source Listener**
  - Southbound API

**Inside Standard Scope**
- **Out of Standard Scope**
- **Out of ALTO Scope**
List of Remaining Issues/Open Discussions

- **Issues that have been discussed in WG mailing list**
  - Q1: How to handle data types in ALTO related IANA registries?
  - Q2: How to configure server level management for ALTO?
  - Q3: How to define data model for logging and fault management?

- **Issues that have been discussed internally**
  - Q4: How to support ALTO client-side O&M?
  - Q5: How to support ALTO server-to-server communication?

- **Issues that have not been discussed yet**
  - Q6: How to support different data sources using a unified data model?
Q1: Data Types in ALTO Related IANA Registries

ALTO related IANA registries:
https://www.iana.org/assignments/alto-protocol/alto-protocol.xhtml

Created by RFC7285

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Intended Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>routingcost</td>
<td>See Section 6.1.1.1</td>
</tr>
<tr>
<td>priv:</td>
<td>Private use</td>
</tr>
</tbody>
</table>

Table 3: ALTO Cost Metrics

Updated by draft-ietf-alto-path-vector

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Intended Semantics</th>
<th>Security Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ane-path</td>
<td>See Section 6.5.1</td>
<td>See Section 11</td>
</tr>
</tbody>
</table>

Table 1: ALTO Cost Metric Registry

Option 1: Define "enumeration" in "iana-alto-types.yang"

```yaml
typedef cost-metric {
  type enumeration {
    enum routingcost {
      value 1;
      description "Generic measurement";
    }
    enum ane-path {
      value 2;
      description "Array of ANE names";
    }
  }
}
```

Pros:
- Guarantee consistency between different documents / YANG modules

Cons:
- Require IANA to have YANG skills
- Hard to extend new values for experimental drafts
Q1: Data Types in ALTO Related IANA Registries

ALTO related IANA registries:
https://www.iana.org/assignments/alto-protocol/alto-protocol.xhtml

Created by RFC7285

**Table 3: ALTO Cost Metrics**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Intended Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>routingcost</td>
<td>See Section 6.1.1.1</td>
</tr>
<tr>
<td>priv:</td>
<td>Private use</td>
</tr>
</tbody>
</table>

Updated by draft-ietf-alto-path-vector

**Table 1: ALTO Cost Metric Registry**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Intended Semantics</th>
<th>Security Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ane-path</td>
<td>See Section 6.5.1</td>
<td>See Section 11</td>
</tr>
</tbody>
</table>

Option 2 (Current): Define "identity" in "ietf-alto.yang"

Pros:
- Anyone can easily add new values to the data type. (Just define new identities)

Cons:
- Lack of change control (There is no registry to guarantee the consistency. May bring challenges to interop.)
Q2: Server Level Management

ALTO server is based on many underlay protocols. (e.g., HTTP/HTTPS for message, DHCP/DNS for discovery)

Question: whether more configuration parameters in underlay protocol stack should be included (e.g., HTTP Cache-Control, Retry-After, etc.)

Inspired by RESTCONF model, complete configuration for server-listen-stack is added.

More ALTO specific configuration should be put here, e.g., base uri, server discovery.
Q3: Logging and Fault Management

Current related records:
success/failure counts for requests/responses

server-level:

```yaml
augment /alto:alto-server:
  +--ro num-total-req?  yang:counter32
  +--ro num-total-succ? yang:counter32
  +--ro num-total-fail?  yang:counter32
  +--ro num-total-last-req? yang:counter32
  +--ro num-total-last-succ? yang:counter32
  +--ro num-total-last-fail? yang:counter32
```

resource-level:

```yaml
augment /alto:alto-server/alto:resource:
  +--ro num-res-upd?  yang:counter32
  +--ro res-mem-size? yang:counter32
  +--ro res-enc-size?  yang:counter32
  +--ro num-res-req?  yang:counter32
  +--ro num-res-succ? yang:counter32
  +--ro num-res-fail?  yang:counter32
```

Following records suggested by Qiufang Ma and Chongfeng Xie will be added in the next revision:

- Success/failure records for configuration update events
  - e.g., Add a new information to an existing "update" resource to enable incremental updates.
- Records for types of configuration update events
  - Can YANG model provide this feature itself?
- Status updates of connections to data sources

We are looking forward to getting more suggested useful metrics from experiences of real deployments.
Q4: ALTO Client-Side O&M

So far, only data model for ALTO server O&M is defined. But in practice, ALTO clients also need O&M.

From experience in two IETF Hackathons, the following parameters are suggested for ALTO client O&M:

● Data model for how to access ALTO services
  ○ URI to the IRD of the ALTO server (can also be discovered using ALTO server discovery mechanism [RFC7286] [RFC8686])
  ○ Resource IDs of required ALTO information resources
  ○ Parameters to required ALTO information resources (e.g., cost types, endpoint properties)

● Data model for transport mechanism control
  ○ Data polling (e.g., polling interval)
  ○ Pub/sub (e.g., SSE or HTTP/2)
  ○ On-demand query (some applications may change the requests frequently)

Design option 1:
Add a new top-level container/list for ALTO client.

Design option 2:
Add a new data source type for ALTO server, and consider ALTO client as an ALTO data source listener.
Q5: Server-to-Server Communication

Several drafts have already discussed ALTO use cases in multi-domain settings:

To support ALTO O&M in multi-domain settings, the following features are required:
- ALTO server need to be a data source of another ALTO server:
  - Design option 1: collect data using ALTO protocol directly. (may need extension to ALTO)
  - Design option 2: Use other southbound protocol to expose database of ALTO server.
    - No standard so far; need to decide whether to be a standard.
- Cross-domain path discovery:
  - Non-existing ALTO services can provide information for cross-domain path discovery.
  - Require some mechanism to look up ingress/egress points of inter-domain routes in each administrative domain.
Q6: Model for Data Sources

Data model for data source listener configuration:
- Southbound protocol stack parameters
  - Protocol version, URI, authentication, etc.
- Query parameters
  - Query expression (PromQL)
  - Xpath (RESTCONF)
- Data collection mechanism parameters
  - Periodically polling
  - Pub/sub
  - On demand

Example of data source for YANG data store:
- local: access data store from memory.
- remote: access data store using RESTCONF.

Data Source Listener: ietf-alto.yang

Algorithm Plugin: example-ietf-alto-alg.yang

When Algorithm Plugin accesses data from ALTO data broker, there should be a mapping between data structure in Data Broker and data collected from Data Source Listener.
Next Step

● The authors will:
  ○ Make decisions to Q1-Q3 as soon and submit a new version.
  ○ Continue to collect feedback to Q4-Q6 from ALTO WG and other related WGs/communities.
  ○ Test YANG model in real implementation.

● Target milestones
  ○ Reach agreement on Q1-Q6 and get the document ready before next IETF.
  ○ Push ALTO O&M YANG Model to OpenALTO implementation and get deployment before Mar 2023.