

ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: 8 August 2026

J. Zhang
Tongji University
D. Dhody
Huawei
K. Gao
Sichuan University
R. Schott
Deutsche Telekom
Q. Ma
Huawei
4 February 2026

YANG Data Models for the Application-Layer Traffic Optimization (ALTO)
Protocol
draft-ietf-alto-oam-yang-18

Abstract

This document defines a YANG data model for Operations, Administration, and Maintenance (OAM) & Management of the Application-Layer Traffic Optimization (ALTO) Protocol. The operator of an ALTO server can use this data model to (1) set up the ALTO server, (2) configure server discovery, (3) create, update and remove ALTO information resources, (4) manage the access control of each ALTO information resource, and (5) collect statistical data from the ALTO server. The application provider can also use this data model to configure ALTO clients to communicate with known ALTO servers.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the ALTO Working Group mailing list (alto@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/alto/>.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-alto/draft-alto-oam-yang>.

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 <https://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 8 August 2026.

Copyright Notice

Copyright (c) 2026 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	4
3. Acronyms and Abbreviations	4
3.1. Tree Diagrams	5
3.2. Prefixes in Data Node Names	5
3.3. Placeholders in Reference Statements	6
4. Design Scope and Requirements	7
4.1. Scope of Data Models for ALTO O&M	7
4.2. Basic Requirements	8
4.3. Additional Requirements for Extensibility	9
5. Design of ALTO O&M Data Model	9
5.1. Overview of ALTO O&M Data Model	9
5.2. Data Model for ALTO Client Operation and Management	9
5.3. Data Model for Server-level Operation and Management	10
5.3.1. Data Model for ALTO Server Setup	11
5.3.2. Data Model for Logging Management	12
5.3.3. Data Model for ALTO-related Management	13
5.3.4. Data Model for Security Management	13
5.4. Data Model for ALTO Server Configuration Management	13
5.4.1. Data Source Configuration Management	13

5.4.2.	ALTO Information Resources Configuration Management	14
5.4.3.	ALTO Information Resource Access Control Management	17
6.	Design of ALTO O&M Statistics Data Model	19
6.1.	Model for ALTO Server Failure Monitoring	20
6.2.	Model for ALTO-specific Performance Monitoring	20
7.	ALTO OAM YANG Modules	21
7.1.	The "ietf-alto" YANG Module	21
7.2.	The "ietf-alto-stats" YANG Module	52
8.	Security Considerations	60
9.	IANA Considerations	62
10.	References	63
10.1.	Normative References	63
10.2.	Informative References	66
Appendix A.	Examples of Extending the ALTO O&M Data Model	67
A.1.	An Example Module for Extended Server Discovery Manners	67
A.2.	An Example Module for Extended Client Authentication Approaches	70
A.3.	Example Module for Extended Data Sources	72
A.4.	An Example Module for Information Resource Creation Algorithm	75
A.5.	Example Usage	78
Appendix B.	A Sample ALTO Server Architecture to Implement ALTO O&M YANG Modules	82
Acknowledgements	84
Authors' Addresses	84

1. Introduction

This document defines a YANG data model for the Operations, Administration, and Maintenance (OAM) & Management of Application-Layer Traffic Optimization (ALTO) Protocol. The basic purpose of this YANG data model is discussed in Section 16 of [RFC7285].

The operator of an ALTO server can use this data model to:

- * set up the ALTO server,
- * configure server discovery,
- * create, update and remove ALTO information resources,
- * manage the access control of each ALTO information resource,
- * collect statistical data of the ALTO server.

The application provider can also use this data model to configure ALTO clients to communicate with known ALTO servers.

Section 4.1 describes what is and is not in scope. Section 4.2 and Section 4.3 define more concrete requirements for the data model.

The basic structure of this YANG data model is guided by Section 16 of [RFC7285] and [RFC7971]. Although the scope of the YANG data model in this document mainly focuses on the support of the base ALTO protocol [RFC7285] and the existing ALTO standard extensions: [RFC8189], [RFC8895], [RFC8896], [RFC9240], [RFC9241], [RFC9274], [RFC9275], and [RFC9439].

The detailed design of the data model is illustrated in Section 5 and Section 6. Some examples of how to extend this data model for specific ALTO server implementations are shown in Appendix A.

The YANG data models in this document conform to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here. When the words appear in lower case, they are to be interpreted with their natural language meanings.

3. Acronyms and Abbreviations

This document uses the following acronyms:

ALTO: Application-Layer Traffic Optimization

CPU: Central Processing Unit

DNS: Domain Name System

HTTP: Hypertext Transfer Protocol

IRR: Internet Routing Registry

OAM: Operations, Administration, and Maintenance (Section 3 of [RFC6291])

O&M: OAM and Management (Section 3 of [RFC6291])

OAuth: Open Authorization

PID: Provider-defined Identifier in ALTO

TCP: Transmission Control Protocol

TLS: Transport Layer Security

URI: Uniform Resource Identifier

3.1. Tree Diagrams

The meaning of the symbols in the tree diagrams is defined in [RFC8340].

3.2. Prefixes in Data Node Names

The complete name of a data node or data model object includes a prefix, which indicates the YANG module in which the name is defined. In this document, the prefix is omitted when the YANG module is clear from the context; otherwise, the prefix is included. The prefixes indicating the corresponding YANG modules are shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC9911]
inet	ietf-inet-types	[RFC9911]
ds	ietf-datastores	[RFC8342]
yp	ietf-yang-push	[RFC8641]
ts	ietf-truststore	[RFC9641]
tcp	ietf-tcp-server	[RFC9643]
tls	ietf-tls-server	[RFC9645]
http	ietf-http-server	[I-D.ietf-netconf-http-client-server]
ncc	ietf-netconf-client	[I-D.ietf-netconf-netconf-client-server]
rcc	ietf-restconf-client	[I-D.ietf-netconf-restconf-client-server]

Table 1: Prefixes and corresponding YANG modules

3.3. Placeholders in Reference Statements

Note to the RFC Editor: This section is to be removed prior to publication.

This document contains placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Please apply the following replacements:

- * GGGG --> the assigned RFC number for [I-D.ietf-netconf-http-client-server]
- * HHHH --> the assigned RFC number for [I-D.ietf-netconf-netconf-client-server]
- * IIII --> the assigned RFC number for [I-D.ietf-netconf-restconf-client-server]
- * XXXX --> the assigned RFC number for this draft

4. Design Scope and Requirements

4.1. Scope of Data Models for ALTO O&M

The following items are in the scope of the data models specified in this document:

- * Deploying an ALTO server/client.
- * Operating and managing an ALTO server/client.
- * Configuring functionality/capability configuration of ALTO services.
- * Monitoring ALTO-related performance metrics.

This document does not normatively define any data model related to a specific implementation, including:

- * Data structures for how to store/deliver ALTO information resources (e.g., database schema to store a network map).
- * Data structures for how to store information collected from data sources. (e.g., database schema to store topology collected from an Interface to the Routing System (I2RS) client [RFC7921])

Likewise, the specification does not cover considerations that are specific to the ALTO Transport Information Publication Service (TIPS) [RFC9569]. Specifically, considerations related to HTTP/3 are out of scope.

For convenience, examples of how related extensions can be defined are provided in the Appendices.

4.2. Basic Requirements

Based on recommendations in [RFC7285] and [RFC7971], the data models provided by this document satisfy basic requirements listed in Table 2.

Requirement	Reference
R1: The data model should support configuration for ALTO server setup.	Section 16.1 of [RFC7285]
R2: The data model should provide logging management.	Section 16.2.1 of [RFC7285]
R3: The data model should provide ALTO-related management information.	Section 16.2.2 of [RFC7285]
R4: The data model should support configuration for security policy management.	Section 16.2.6 of [RFC7285]
R5-1: The data model should support basic configuration to receive data from different data sources.	Section 16.2.4 of [RFC7285], Section 3.2 of [RFC7971]
R5-2: The data model should support configuration for information resource generation algorithms.	Section 16.2.4 of [RFC7285]
R5-3: The data model should support configuration for access control at information resource level.	Section 16.2.4 of [RFC7285]
R6: The data model should provide metrics for server failures.	Section 16.2.3 of [RFC7285], Section 3.3 of [RFC7971]
R7: The data model should provide performance monitoring for ALTO-specific metrics.	Section 16.2.5 of [RFC7285], Section 3.4 of [RFC7971]

Table 2: Basic Requirements of Data Model for ALTO O&M.

4.3. Additional Requirements for Extensibility

R8: As the ALTO protocol is extensible, the data models for ALTO O&M should allow for augmentation to support potential future extensions.

5. Design of ALTO O&M Data Model

5.1. Overview of ALTO O&M Data Model

The "ietf-alto" module is designed to fit all the requirements listed in Section 4.

As shown in Figure 1, the top-level container "alto" in the "ietf-alto" module contains a list "alto-client" and a single "alto-server" container.

The "alto-client" list defines a list of configurations for other applications to bootstrap an ALTO client. These data nodes can also be used by data sources and information resource creation algorithms that are configured by an ALTO server instance.

The container "alto-server" contains both configuration and operational data of an administrated ALTO server instance.

```

module: ietf-alto
  +-rw alto!
    +-rw alto-client* [client-id] {alto-client}?
    | ...
    +-rw alto-server {alto-server}?
    +...
    +-rw auth-client* [client-id]
    | ...
    +-rw role* [role-name]
    | +-rw role-name    role-name
    | +-rw client*      client-ref
    +-rw data-source* [source-id]
    | ...
    +-rw resource* [resource-id]
    ...

```

Figure 1: IETF ALTO Tree Structure

5.2. Data Model for ALTO Client Operation and Management

As shown in Figure 2, the "alto-client" contains a list of client-side configurations. Each "alto-client" entry contains the following data nodes:

"client-id": A unique identifier that can be referenced by other applications.

"server-discovery-client": A container that is used to configure how this ALTO client discovers an ALTO server.

```

module: ietf-alto
  +--rw alto!
    +--rw alto-client* [client-id] {alto-client}?
      |   +--rw client-id          string
      |   +--rw server-discovery
      |   +---u alto-server-discovery-client
      ...

```

Figure 2: IETF ALTO Client Subtree Structure

5.3. Data Model for Server-level Operation and Management

The ALTO server instance contains a set of data nodes for server-level operation and management for ALTO that are shown in Figure 3. This structure satisfies R1 - R4 in Section 4.2.

```

module: ietf-alto
  +--rw alto!
    ...
    +--rw alto-server {alto-server}?
      +--rw listen
      |   +---u alto-server-listen-stack
      +--rw server-discovery
      |   +---u alto-server-discovery
      +--rw logging-system
      |   +---u alto-logging-system
      +--rw cost-type* [cost-type-name]
      |   +--rw cost-type-name    cost-type-name
      |   +--rw cost-mode         identityref
      |   +--rw cost-metric       identityref
      |   +--rw description?      string
      |   +--rw cost-context {performance-metrics}?
      |       +--rw cost-source    identityref
      |       +--rw parameters
      |       +--rw (parameters)?
      +--rw meta* [meta-key]
      |   +--rw meta-key          meta-key
      |   +--rw meta-value        binary
      ...

```

Figure 3: IETF ALTO Server Level Subtree Structure

5.3.1. Data Model for ALTO Server Setup

To satisfy R1 in Section 4.2, the ALTO server instance contains the basic data nodes for the server setup that are detailed in the following subsections.

5.3.1.1. ALTO Server Listen Stack

The container "listen" contains all the data nodes for the whole server listen stack across TCP, TLS, HTTP and application layers (Figure 4).

```

grouping alto-server:
  +-- base-uri?   inet:uri
grouping alto-server-listen-stack:
  +-- (transport)
    +--:(http) {http-listen}?
    |   +-- http
    |   |   +-- tcp-server-parameters
    |   |   |   +---u tcp:tcp-server-grouping
    |   |   +-- http-server-parameters
    |   |   |   +---u http:http-server-grouping
    |   |   +-- alto-server-parameters
    |   |   |   +---u alto-server
    |   +--:(https)
    |   |   +-- https
    |   |   |   +-- tcp-server-parameters
    |   |   |   |   +---u tcp:tcp-server-grouping
    |   |   |   +-- tls-server-parameters
    |   |   |   |   +---u tls:tls-server-grouping
    |   |   |   +-- http-server-parameters
    |   |   |   |   +---u http:http-server-grouping
    |   |   |   +-- alto-server-parameters
    |   |   |   |   +---u alto-server

```

Figure 4: IETF ALTO Server Groupings Structure

The "transport" choice node enables which protocol layers to be configured. By default, two cases are defined for different deployment scenarios:

- * The "http" case is provided to support scenarios where the TLS-termination is handled by other external components, e.g., reverse proxies or ingress controllers.
- * The "https" case is provided to support scenarios where the whole HTTPS server listen stack including TLS is handled by the ALTO server itself.

5.3.1.2. ALTO Server Discovery Setup

In practice, for a large-scale network consisting of multiple administrative domains, the information about the network may be partitioned and distributed over multiple ALTO servers. That may require discovery and communication among different ALTO servers.

The "ietf-alto" module provides the configuration for how an ALTO server can be discovered by another ALTO server or client on demand (Figure 5). However, it does not contain any configuration for the communication among ALTO servers because the related solution has not become a standard. Future documents may extend it to fully support multi-domain scenarios.

```

grouping alto-server-discovery:
  +- (method)?
  +--:(reverse-dns) {xdom-disc}?
    +- rdns-naptr-records
      +- static-prefix*          inet:ip-prefix
      +- dynamic-prefix-source*  data-source-ref

```

Figure 5: IETF ALTO Server Discovery Grouping Structure

The "server-discovery" node provides configuration for the discovery of ALTO servers using a variety of mechanisms. The initial version of the "ietf-alto" module only defines the "reverse-dns" case that is used to configure DNS NAPTR records for ALTO server discovery as suggested by [RFC7286] and [RFC8686]. It configures a set of endpoints that can be served by this ALTO server. The node contains two leaf lists. The "static" list contains a list of manually configured endpoints. The "dynamic" list points to a list of data sources to retrieve the endpoints dynamically. As suggested by [RFC7286] and [RFC8686], the IP prefixes of the endpoints configured by both "static" and "dynamic" lists will be translated into DNS NAPTR resource records for server discovery. The "server-discovery-manner" choice can be augmented by the future modules to support other mechanisms.

5.3.2. Data Model for Logging Management

To satisfy R2 in Section 4.2, the ALTO server instance contains the the logging data nodes shown in Figure 6.

The "logging-system" data node provides configuration to select a logging system to capture log messages generated by an ALTO server.

By default, "syslog" is the only supported logging system. When selecting "syslog", the related configuration is delegated to the configuration file of the syslog [RFC5424] server.

```
grouping alto-logging-system:
  +-- (logging-system)?
    +--:(syslog)
      +-- syslog-params
        +-- config-file?   inet:uri
```

Figure 6: IETF ALTO Logging System Grouping Structure

A specific server implementation can extend the "logging-system" node to add other logging systems.

5.3.3. Data Model for ALTO-related Management

To satisfy R3 in Section 4.2, the data model contains the following ALTO-related management information (Figure 3):

- * The "cost-type" list is the registry for the cost types that can be used in the ALTO server.
- * The "meta" list contains the customized meta data of the ALTO server. It is populated into the meta field of the default Information Resource Directory (IRD).

5.3.4. Data Model for Security Management

To satisfy R4 in Section 4.2, the data model leverages HTTP and TLS to provide basic security management for an ALTO server. All the related configurations are covered by the server listen stack.

5.4. Data Model for ALTO Server Configuration Management

5.4.1. Data Source Configuration Management

To satisfy R5-1 in Section 4.2, the ALTO server instance contains a list of "data-source" entries to subscribe the data sources from which ALTO information resources are derived (Section 16.2.4 of [RFC7285]).

```

module: ietf-alto
  +-rw alto!
  ...
  +-rw alto-server {alto-server}?
    ...
    +-rw data-source* [source-id]
      | +-rw source-id           source-id
      | +-rw source-type        identityref
      | +-rw source-params
    ...

```

Figure 7: IETF ALTO Server Data Source Subtree Structure

As shown in Figure 7, a "data-source" list entry includes:

- * A unique "source-id" for resource creation algorithms to reference.
- * The "source-type" attribute to declare the type of the data source.
- * The "source-params" to specify where and how to query the data.

The "data-source/source-params" node can be augmented for different types of data sources. Note that the purpose of this node is not to fully set up the communication mechanisms for specific data sources, but to maintain how data sources are configured and expose them to the ALTO server.

This data model only includes a basic structure for an ALTO server to correctly interact with a data source. The implementation-specific parameters of any certain data source can be augmented in another module. An example is included in Appendix A.3.

5.4.2. ALTO Information Resources Configuration Management

To satisfy R5-2 and R-3 in Section 4.2, the ALTO server instance contains a list of "resource" entries (Figure 8). Each "resource" entry contains the data nodes of an ALTO information resource (See Section 8.1 of [RFC7285]). The operator of the ALTO server can use this model to create, update, and remove the ALTO information resources.

Each "resource" entry provides data nodes defining how to create or update an ALTO information resource. Adding a new "resource" entry notifies the ALTO server to create a new ALTO information resource. Updating an existing "resource" entry notifies the ALTO server to update the generation parameters (e.g., capabilities and the creation

algorithm) of an existing ALTO information resource. Removing an existing "resource" entry will remove the corresponding ALTO information resource.

```

module: ietf-alto
+--rw alto!
  ...
  +--rw alto-server {alto-server}?
    ...
    +--rw resource* [resource-id]
      +--rw resource-id          resource-id
      +--rw resource-type        identityref
      +--rw description?         string
      +--rw accepted-role*       role-ref
      +--rw dependency*          resource-ref
      +--rw alto-ird-params
      |   +--rw delegation      inet:uri
      +--rw alto-networkmap-params
      |   +--rw is-default?     boolean
      |   +--rw filtered?       boolean
      |   +---u algorithm
      +--rw alto-costmap-params
      |   +--rw filtered?       boolean
      |   +---u filter-costmap-cap
      |   +---u algorithm
      +--rw alto-endpointcost-params
      |   +---u filter-costmap-cap
      |   +---u algorithm
      +--rw alto-endpointprop-params
      |   +--rw prop-type*      endpoint-property
      |   +---u algorithm
      +--rw alto-propmap-params {propmap}?
      |   +---u algorithm
      +--rw alto-cdni-params {cdni}?
      |   +---u algorithm
      +--rw alto-update-params {incr-update}?
      |   +---u algorithm
      +--rw resource-limits
      |   +--rw notify-res-mem-limit?      uint64
      |   +--rw notify-upd-stream-limit?   uint64
      |   {incr-update}?
  notifications:
    +---n alto-resource-event {alto-server}?
      +---ro resource-id          resource-ref
      +---ro notify-res-mem-threshold?   uint64
      +---ro notify-upd-stream-threshold? uint64 {incr-update}?

```

```

grouping filter-costmap-cap:
  +-- cost-type-name*          cost-type-ref
  +-- cost-constraints?        boolean
  +-- max-cost-types?          uint32 {multi-cost}?
  +-- testable-cost-type-name* cost-type-ref {multi-cost}?
  +-- calendar-attributes {cost-calendar}?
    +-- cost-type-name*        cost-type-ref
    +-- time-interval-size      decimal64
    +-- number-of-intervals     uint32
grouping algorithm:
  +-- (algorithm)

```

Figure 8: IETF ALTO Resource Subtree Structure

A "resource" list entry MUST include a unique "resource-id" and a "resource-type".

It may also include an "accepted-role" node containing a list of "role-name"s that is used by role-based access control for this ALTO information resource. See Section 5.4.3 for details of information resource access control.

For some "resource-type", the "resource" entry may also include a "dependency" node containing the "resource-id" of the dependent ALTO information resources (Section 9.1.5 of [RFC7285]).

For each type of ALTO information resource, the "resource" entry may also need type-specific parameters. These type-specific parameters can be split into two categories:

1. One category of the type-specific parameters is common for the same type of ALTO information resource. They declare the Capabilities attribute of the ALTO information resource (Section 9.1.3 of [RFC7285]).
2. The other category of the type-specific parameters is algorithm-specific. The developer of the ALTO server can implement their own creation algorithms and augment the "algorithm" node to declare algorithm-specific input parameters.

Except for the "ird" resource, all the other types of "resource" entries have an augmented "algorithm" node. The augmented "algorithm" node can reference data sources subscribed by the "data-source" entries (See Section 5.4.1). An example of extending the "algorithm" node for a specific type of "resource" is included in Appendix A.4.

The developer does not have to customize the creation algorithm of the "ird" resource. The default "ird" resource will be created automatically based on all the added "resource" entries. The delegated "ird" resource will be created as a static ALTO information resource (Section 9.2.4 of [RFC7285]).

Each "resource" entry may also set thresholds of memory usage and active update streams (if "incr-update" feature is enabled). Table 3 describes limits that, once exceeded, will trigger notifications to be generated:

Notification Threshold	Description
notify-res-mem-limit	Used to notify high memory utilization of the resource configured to an ALTO server instance. When exceeded, an alto-resource-event will be generated.
notify-upd-stream-limit	Used to notify a high number of active update streams that are serviced by an update resource configured to an ALTO server instance. When exceeded, an alto-resource-event will be generated.

Table 3: Notification Thresholds.

5.4.3. ALTO Information Resource Access Control Management

To satisfy R-3 in Section 4.2 and as per Section 15.5.2 of [RFC7285], the "ietf-alto" module also defines authentication and authorization related configuration to employ access control at the information resource level. The ALTO server returns the IRD to the ALTO client based on its authentication information.

The information resource access control is supported using the structure shown in Figure 9.

```

module: ietf-alto
+--rw alto!
...
+--rw alto-server {alto-server}?
...
+--rw auth-client* [client-id]
|
|   +--rw client-id                               string
|   +--rw (authentication)?
|       +--:(http)
|           +--rw http-auth-client
|               {http-listen,http:client-auth-supported,
|                http:local-users-supported}?
|           +--rw user-id      http-user-id-ref
|       +--:(https)
|           +--rw https-auth-client
|               {http:client-auth-supported,
|                http:local-users-supported}?
|           +--rw user-id?     https-user-id-ref
|       +--rw tls-auth-client
|           {http:client-auth-supported}?
|       +--rw ca-cert {tls:client-auth-x509-cert}?
|           | +---u inline-or-truststore-ca-cert-ref
|       +--rw ee-cert {tls:client-auth-x509-cert}?
|           | +---u inline-or-truststore-ee-cert-ref
|       +--rw raw-public-key
|           | {tls:client-auth-raw-public-key}?
|           | +---u inline-or-truststore-public-key-ref
|       +--rw tls12-psks?      empty
|           | {tls:client-auth-tls12-psk}?
|       +--rw tls13-epsks?     empty
|           | {tls:client-auth-tls13-epk}?
+--rw role* [role-name]
|   +--rw role-name      role-name
|   +--rw client*        client-ref
...

```

Figure 9: IETF ALTO Client Authentication Subtree Structure

The structure shown in Figure 9 can be used to configure the role-based access control:

- * "auth-client" declares a list of ALTO clients that can be authenticated by the internal or external authorization server. This basic model only includes authentication approach directly provided by the HTTP or TLS server, but the operators or future documents can augment the "authentication" choice for different authentication mechanisms.

- * "role" defines a list of roles for access control. Each role contains a list of authenticated ALTO clients. Each client can be assigned to multiple roles. The "role-name" can be referenced by the "accepted-role" list of a "resource". If an authenticated ALTO client is included in any roles with access permission to a resource, the client is granted access to that resource.

6. Design of ALTO O&M Statistics Data Model

The "ietf-alto-stats" module augments the "ietf-alto" module to include statistics at the ALTO server and information resource level (Figure 10).

```

module: ietf-alto-stats

augment /alto:alto/alto:alto-server:
  +--rw server-level-monitor-config
  |   +--rw time-window-size?   uint32
  +--ro server-level-stats
  |   +---u server-level-stats
augment /alto:alto/alto:alto-server/alto:resource:
  +--ro resource-level-stats
  |   +---u resource-level-stats

grouping server-level-stats:
  +-- discontinuity-time?      yang:timestamp
  +-- last-report-time?       yang:timestamp
  +-- num-total-req?           yang:counter64
  +-- num-total-succ?          yang:counter64
  +-- num-total-fail?          yang:counter64
  +-- num-total-last-req?      yang:gauge64
  +-- num-total-last-succ?     yang:gauge64
  +-- num-total-last-fail?     yang:gauge64
grouping network-map-stats:
  +-- num-map-pid?             yang:gauge64
grouping prop-map-stats:
  +-- num-map-entry?           yang:gauge64
grouping cdni-stats:
  +-- num-base-obj?            yang:gauge64
grouping upd-stream-stats:
  +-- num-upd-stream?          yang:gauge64
  +-- num-upd-msg-total?       yang:gauge64
  +-- num-upd-msg-max?         yang:gauge64
  +-- num-upd-msg-min?         yang:gauge64
  +-- num-upd-msg-avg?         yang:gauge64
  +-- num-upd-msg-total-last?  yang:gauge64
  +-- num-upd-msg-max-last?    yang:gauge64
  +-- num-upd-msg-min-last?    yang:gauge64

```

```

    +-- num-upd-msg-avg-last?      yang:gauge64
grouping resource-level-stats:
  +-- discontinuity-time?         yang:timestamp
  +-- last-report-time?          yang:timestamp
  +-- num-res-upd?                yang:counter64
  +-- res-mem-size?              uint64
  +-- res-enc-size?              uint64
  +-- num-res-req?                yang:counter64
  +-- num-res-succ?              yang:counter64
  +-- num-res-fail?              yang:counter64
  +-- num-res-last-req?          yang:gauge64
  +-- num-res-last-succ?         yang:gauge64
  +-- num-res-last-fail?         yang:gauge64
  +-- network-map-stats
  | +---u network-map-stats
  +-- endpoint-prop-stats
  | +---u prop-map-stats
  +-- property-map-stats
  | +---u prop-map-stats
  +-- cdni-stats
  | +---u cdni-stats
  +-- upd-stream-stats

```

Figure 10: IETF ALTO Statistics Structure

6.1. Model for ALTO Server Failure Monitoring

To satisfy R6 in Section 4.2, the "ietf-alto-stats" module contains statistics that indicate server failures (Figure 10).

More specifically, "num-total-*" and "num-total-last-*" provide server-level failure counters; "num-res-*" and "num-res-last-*" provide information resource-level failure counters.

6.2. Model for ALTO-specific Performance Monitoring

To satisfy R7 in Section 4.2, the "ietf-alto-stats" module also contains statistics for ALTO-specific performance metrics (Figure 10).

More specifically, this data model contains the following measurement information of "system and service performance" suggested by [RFC7285] and [RFC7971]:

- * Requests and responses for each information resource
- * CPU and memory utilization

- * ALTO map updates
- * Number of PIDs
- * ALTO map sizes

Besides the above measurement information suggested by [RFC7285] and [RFC7971], the "ietf-alto-stats" module also contains useful measurement information for other ALTO extensions:

- * "num-map-entry" and "num-base-obj" provide measurement for number of generic ALTO entities (for [RFC9240] and [RFC9241])
- * "num-upd-stream" and "num-upd-msg-*" provide statistics for update streams and messages (for [RFC8189])

The "ietf-alto-stats" module only focuses on the performance metrics that can be directly measured at the ALTO server. The following metrics for "measurement of the impact" suggested by [RFC7971] are not contained in this data model:

- * Total amount and distribution of traffic
- * Application performance

7. ALTO OAM YANG Modules

7.1. The "ietf-alto" YANG Module

```
<CODE BEGINS> file "ietf-alto@2026-02-04.yang"
module ietf-alto {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-alto";
  prefix alto;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 9911: Common YANG Data Types, Section 4";
  }
  import ietf-truststore {
    prefix ts;
    reference
      "RFC 9641: A YANG Data Model for a Truststore";
  }
  import ietf-tcp-server {
    prefix tcp;
    reference
```

```
    "RFC 9643: YANG Groupings for TCP Clients and TCP Servers";
}
import ietf-tls-server {
    prefix tls;
    reference
        "RFC 9645: YANG Groupings for TLS Clients and TLS Servers";
}
import ietf-http-server {
    prefix http;
    reference
        "RFC GGGG: YANG Groupings for HTTP Clients and HTTP Servers";
}

organization
    "IETF ALTO Working Group";
contact
    "WG Web:  <https://datatracker.ietf.org/wg/alto/about/>
    WG List:  <alto@ietf.org>";
description
    "This YANG module defines a set of configured and operational
    parameters of an administrated ALTO server instance.

    Copyright (c) 2026 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Revised BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";

revision 2026-02-04 {
    description
        "Initial Version.";
    reference
        "RFC XXXX: YANG Data Models for the Application-Layer Traffic
        Optimization (ALTO) Protocol";
}

// Features

feature alto-client {
    description
```

```
    "Indicates that the implementation embeds an ALTO client
    instance.";
  reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
    Protocol";
}

feature alto-server {
  description
    "Indicates that the implementation embeds an ALTO server
    instance.";
  reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
    Protocol";
}

feature http-listen {
  description
    "The 'http-listen' feature is only used for test deployment.
    This feature shouldn't be used in the production
    deployment.";
  reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
    Protocol, Section 8.3.5 ";
}

feature xdom-disc {
  description
    "Indicates support of cross-domain server discovery.";
  reference
    "RFC 8686: Application-Layer Traffic Optimization (ALTO)
    Cross-Domain Server Discovery";
}

feature multi-cost {
  description
    "Indicates support of multi-cost extension.";
  reference
    "RFC 8189: Multi-Cost Application-Layer Traffic Optimization
    (ALTO)";
}

feature incr-update {
  description
    "Indicates support of incremental update extension.";
  reference
    "RFC 8895: Application-Layer Traffic Optimization (ALTO)
    Incremental Updates Using Server-Sent Events
```

```
        (SSE)";
    }

    feature cost-calendar {
        description
            "Indicates support of cost calendar extension.";
        reference
            "RFC 8896: Application-Layer Traffic Optimization (ALTO) Cost
            Calendar";
    }

    feature propmap {
        description
            "Indicates support of entity property map extension.";
        reference
            "RFC 9240: An ALTO Extension: Entity Property Maps";
    }

    feature cdni {
        description
            "Indicates support of CDNI extension.";
        reference
            "RFC 9241: Content Delivery Network Interconnection (CDNI)
            Request Routing: CDNI Footprint and Capabilities
            Advertisement using ALTO";
    }

    feature path-vector {
        description
            "Indicates support of path vector extension.";
        reference
            "RFC 9275: An Extension for Application-Layer Traffic
            Optimization (ALTO): Path Vector";
    }

    feature performance-metrics {
        description
            "Indicates support of performance metrics extension.";
        reference
            "RFC 9439: ALTO Performance Cost Metrics";
    }

    // Base identities

    identity resource-type {
        description
            "Base identity for type of information resource.";
        reference
```

```
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
      Protocol, Section 8.1 ";
  }

  identity source-type {
    description
      "Base identity for type of data source. A data source
       indicates the origin from which the ALTO information
       resources are derived.";
  }

  identity cost-metric {
    description
      "The cost metric indicates what the cost represents.";
    reference
      "RFC 7285: Application-Layer Traffic Optimization (ALTO)
       Protocol, Section 6.1.1";
  }

  identity cost-mode {
    description
      "The cost mode indicates how costs should be interpreted.
       Specifically, the cost mode attribute indicates whether
       indicated costs should be interpreted as numerical
       values or ordinal rankings.";
    reference
      "RFC 7285: Application-Layer Traffic Optimization (ALTO)
       Protocol, Section 6.1.2
       RFC 9274: A Cost Mode Registry for the Application-Layer
       Traffic Optimization (ALTO) Protocol";
  }

  identity cost-source {
    description
      "The cost source indicates the high-level type of the
       data source.";
    reference
      "RFC 9439: ALTO Performance Cost Metrics, Section 3.1";
  }

  // Identities for ALTO information resources

  identity ird {
    base resource-type;
    description
      "Identity for information resource directory.";
    reference
      "RFC 7285: Application-Layer Traffic Optimization (ALTO)
```

```
        Protocol, Section 9";
    }

    identity network-map {
        base resource-type;
        description
            "Identity for network map.";
        reference
            "RFC 7285: Application-Layer Traffic Optimization (ALTO)
             Protocol, Section 5";
    }

    identity cost-map {
        base resource-type;
        description
            "Identity for cost map.";
        reference
            "RFC 7285: Application-Layer Traffic Optimization (ALTO)
             Protocol, Section 6";
    }

    identity endpoint-prop {
        base resource-type;
        description
            "Identity for endpoint property service.";
        reference
            "RFC 7285: Application-Layer Traffic Optimization (ALTO)
             Protocol, Section 11.4.1";
    }

    identity endpoint-cost {
        base resource-type;
        description
            "Identity for endpoint cost service.";
        reference
            "RFC 7285: Application-Layer Traffic Optimization (ALTO)
             Protocol, Section 11.5.1";
    }

    identity property-map {
        base resource-type;
        description
            "Identity for property map.";
        reference
            "RFC 9240: An ALTO Extension: Entity Property Maps";
    }

    identity cdni {
```

```
    base resource-type;
    description
        "Identity for Content Delivery Network Interconnection (CDNI)
        advertisement service.";
    reference
        "RFC 9241: Content Delivery Network Interconnection (CDNI)
        Request Routing: CDNI Footprint and Capabilities
        Advertisement using ALTO";
}

identity update {
    base resource-type;
    description
        "Identity for update stream service.";
    reference
        "RFC 8895: Application-Layer Traffic Optimization (ALTO)
        Incremental Updates Using Server-Sent Events
        (SSE)";
}

// Identities for cost mode

identity numerical {
    base cost-mode;
    description
        "This mode indicates that it is safe to perform numerical
        operations";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
        Protocol, Section 6.1.2.1";
}

identity ordinal {
    base cost-mode;
    description
        "This mode indicates that the cost values in a cost map
        represent ranking";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
        Protocol, Section 6.1.2.2";
}

identity array {
    if-feature "path-vector";
    base cost-mode;
    description
        "This mode indicates that every cost value in the response
        body of a (Filtered) Cost Map or an Endpoint Cost Service is
```

```
        interpreted as a JSON array.";
    reference
        "RFC 9275: An Extension for Application-Layer Traffic
          Optimization (ALTO): Path Vector, Section 6.5.2";
}

// Identities for cost metrics

identity routingcost {
    base cost-metric;
    description
        "This metric conveys a generic measure for the cost of
         routing traffic from a source to a destination.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
          Protocol, Section 6.1.1.1";
}

identity ane-path {
    if-feature "path-vector";
    base cost-metric;
    description
        "This metric indicates that the value of such a cost type
         conveys an array of Abstract Network Element (ANE) names,
         where each ANE name uniquely represents an ANE traversed by
         traffic from a source to a destination.";
    reference
        "RFC 9275: An Extension for Application-Layer Traffic
          Optimization (ALTO): Path Vector, Section 6.5.1";
}

identity delay-ow {
    if-feature "performance-metrics";
    base cost-metric;
    description
        "One-way delay.";
    reference
        "RFC 9439: ALTO Performance Cost Metrics, Section 4.1";
}

identity delay-rt {
    if-feature "performance-metrics";
    base cost-metric;
    description
        "Round-trip delay.";
    reference
        "RFC 9439: ALTO Performance Cost Metrics, Section 4.2";
}
```

```
identity delay-variation {
  if-feature "performance-metrics";
  base cost-metric;
  description
    "Delay variation.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 4.3";
}

identity lossrate {
  if-feature "performance-metrics";
  base cost-metric;
  description
    "Loss rate.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 4.4";
}

identity hopcount {
  if-feature "performance-metrics";
  base cost-metric;
  description
    "Hop count.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 4.5";
}

identity tput {
  if-feature "performance-metrics";
  base cost-metric;
  description
    "TCP throughput.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 5.1";
}

identity bw-residual {
  if-feature "performance-metrics";
  base cost-metric;
  description
    "Residual bandwidth.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 5.2";
}

identity bw-available {
  if-feature "performance-metrics";
  base cost-metric;
```

```
    description
      "Available bandwidth.";
    reference
      "RFC 9439: ALTO Performance Cost Metrics, Section 5.3";
  }

// Identities for cost sources

identity nominal {
  if-feature "performance-metrics";
  base cost-source;
  description
    "The 'nominal' category indicates that the metric value is
     statically configured by the underlying devices.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 3.1";
}

identity sla {
  if-feature "performance-metrics";
  base cost-source;
  description
    "The 'sla' category indicates that the metric value is
     derived from some commitment which this document refers to
     as service-level agreement (SLA).";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 3.1";
}

identity estimation {
  if-feature "performance-metrics";
  base cost-source;
  description
    "The 'estimation' category indicates that the metric value is
     computed through an estimation process.";
  reference
    "RFC 9439: ALTO Performance Cost Metrics, Section 3.1";
}

// Typedefs

typedef resource-id {
  type string {
    length "1..64";
    pattern '[0-9a-zA-Z\-\:@_]*';
  }
  description
    "Type for a resource ID that are used to reference an ALTO
```

```
        information resource.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
          Protocol, Section 9.1.1";
}

typedef cost-type-name {
    type string {
        length "1..max";
    }
    description
        "Type for the name of a single CostType that can be
         referenced by other ALTO information resources.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
          Protocol, Section 9.2.2";
}

typedef meta-key {
    type string {
        length "1..max";
    }
    description
        "Type for a custom meta key of an ALTO server.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
          Protocol, Section 8.4.1";
}

typedef endpoint-property {
    type union {
        type resource-specific-endpoint-property;
        type global-endpoint-property;
    }
    description
        "Type for an endpoint property.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization (ALTO)
          Protocol, Section 10.8";
}

typedef resource-specific-endpoint-property {
    type string {
        length "1..97";
        pattern '[0-9a-zA-Z\-\:@_]*\.[0-9a-zA-Z\-\:@_]*';
    }
    description
        "Type for a resource-specific endpoint property.";
```

```
    reference
      "RFC 7285: Application-Layer Traffic Optimization (ALTO)
        Protocol, Section 10.8.1";
  }

  typedef global-endpoint-property {
    type string {
      length "1..32";
      pattern '[0-9a-zA-Z\-\:]*';
    }
    description
      "Type for a global endpoint property.";
    reference
      "RFC 7285: Application-Layer Traffic Optimization (ALTO)
        Protocol, Section 10.8.2";
  }

  typedef source-id {
    type string {
      length "1..max";
    }
    description
      "Type for a data source ID that are used to reference a data
        source.";
  }

  typedef role-name {
    type string {
      length "1..max";
    }
    description
      "Type for a name of a role for role-based access control.";
  }

  // Typedefs for referencing purposes

  typedef cost-type-ref {
    type leafref {
      path "/alto:alto/alto:alto-server/alto:cost-type"
        + "/alto:cost-type-name";
    }
    description
      "Type to reference a cost type name.";
  }

  typedef data-source-ref {
    type leafref {
      path "/alto:alto/alto:alto-server/alto:data-source"
```

```
        + "/alto:source-id";
    }
    description
        "Type to reference a data source identifier.";
}

typedef http-user-id-ref {
    type leafref {
        path "/alto:alto/alto:alto-server/alto:listen"
            + "/alto:http/alto:http-server-parameters"
            + "/alto:client-authentication/alto:users"
            + "/alto:user/alto:user-id";
    }
    description
        "Type to reference an HTTP client user id.";
}

typedef https-user-id-ref {
    type leafref {
        path "/alto:alto/alto:alto-server/alto:listen"
            + "/alto:https/alto:tls-server-parameters"
            + "/alto:client-authentication/alto:users"
            + "/alto:user/alto:user-id";
    }
    description
        "Type to reference an HTTPS client user id.";
}

typedef inline-ca-cert-ref {
    type leafref {
        path "/alto:alto/alto:alto-server/alto:listen"
            + "/alto:https/alto:tls-server-parameters"
            + "/alto:client-authentication/alto:ca-certs"
            + "/alto:inline-definition/alto:certificate"
            + "/alto:name";
    }
    description
        "Type to reference a TLS CA certificate.";
}

typedef inline-ee-cert-ref {
    type leafref {
        path "/alto:alto/alto:alto-server/alto:listen"
            + "/alto:https/alto:tls-server-parameters"
            + "/alto:client-authentication/alto:ee-certs"
            + "/alto:inline-definition/alto:certificate"
            + "/alto:name";
    }
}
```

```
    description
      "Type to reference a TLS EE certificate.";
  }

typedef inline-raw-public-key-ref {
  type leafref {
    path "/alto:alto/alto:alto-server/alto:listen"
      + "/alto:https/alto:tls-server-parameters"
      + "/alto:client-authentication/alto:raw-public-keys"
      + "/alto:inline-definition/alto:public-key"
      + "/alto:name";
  }
  description
    "Type to reference a raw public key.";
}

typedef resource-ref {
  type leafref {
    path "/alto:alto/alto:alto-server/alto:resource"
      + "/alto:resource-id";
  }
  description
    "Type to reference a resource identifier.";
}

typedef role-ref {
  type leafref {
    path "/alto:alto/alto:alto-server/alto:role"
      + "/alto:role-name";
  }
  description
    "Type to reference a role.";
}

typedef client-ref {
  type leafref {
    path "/alto:alto/alto:alto-server/alto:auth-client"
      + "/alto:client-id";
  }
  description
    "Type to reference an authenticated client.";
}

// Groupings

grouping filter-costmap-cap {
  description
    "This grouping defines a data model for
```

```
    FilteredCostMapCapabilities.";
reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
      Protocol, Section 11.3.2.4";
leaf-list cost-type-name {
    type cost-type-ref;
    min-elements 1;
    description
        "Supported cost types.";
}
leaf cost-constraints {
    type boolean;
    default false;
    description
        "If set to true, then the ALTO server allows cost
        constraints to be included in requests to the
        corresponding URI.";
}
leaf max-cost-types {
    if-feature "multi-cost";
    type uint32;
    default "0";
    description
        "If present with value N greater than 0, this resource
        understands the multi-cost extensions and can return a
        multi-cost map with any combination of N or
        fewer cost types in the 'cost-type-names' list.";
}
leaf-list testable-cost-type-name {
    if-feature "multi-cost";
    type cost-type-ref;
    description
        "If present, the resource allows constraint tests, but only
        on the cost type names in this array.";
}
container calendar-attributes {
    if-feature "cost-calendar";
    description
        "Configuration for CalendarAttributes.";
    reference
        "RFC 8896: Application-Layer Traffic Optimization (ALTO)
        Cost Calendar, Section 4.1";
    leaf-list cost-type-name {
        type cost-type-ref;
        min-elements 1;
        description
            "An array of one or more elements indicating the cost
            type names in the IRD entry to which the values of
```

```
        'time-interval-size' and 'number-of-intervals' apply.";
    }
    leaf time-interval-size {
        type decimal64 {
            fraction-digits 4;
        }
        units "seconds";
        mandatory true;
        description
            "The duration of an ALTO Calendar time interval.";
    }
    leaf number-of-intervals {
        type uint32 {
            range "1..max";
        }
        mandatory true;
        description
            "A strictly positive integer (greater or equal to 1) that
            indicates the number of values of the Cost Calendar
            array.";
    }
}

grouping algorithm {
    description
        "This grouping defines the base data model for information
        resource creation algorithm.";
    choice algorithm {
        mandatory true;
        description
            "Information resource creation algorithm to be augmented.";
    }
    reference
        "RFC XXXX: YANG Data Models for the Application-Layer Traffic
        Optimization (ALTO) Protocol, Section 5.4.2";
}

grouping alto-server {
    description
        "A reusable grouping for configuring an ALTO server without
        any consideration for how underlying transport sessions are
        established.";
    leaf base-uri {
        type inet:uri;
        description
            "The base URI for the ALTO server.";
    }
}
```

```
}

grouping alto-server-listen-stack {
  description
    "A reuseable grouping for configuring an ALTO server
    'listen' protocol stack for a single connection.";
  choice transport {
    mandatory true;
    description
      "Selects between available transports.";
    case http {
      if-feature "http-listen";
      container http {
        description
          "Configures ALTO server stack assuming that
          TLS-termination is handled externally.";
        container tcp-server-parameters {
          description
            "A wrapper around the TCP server parameters
            to avoid name collisions.";
          uses tcp:tcp-server-grouping {
            refine "local-bind/local-port" {
              default "80";
              description
                "The ALTO server will listen on the IANA-
                assigned well-known port value for 'http' (80)
                if no value is specified.";
            }
          }
        }
      }
    }
    container http-server-parameters {
      description
        "A wrapper around the HTTP server parameters
        to avoid name collisions.";
      uses http:http-server-grouping;
    }
    container alto-server-parameters {
      description
        "A wrapper around the ALTO server parameters
        to avoid name collisions.";
      uses alto-server;
    }
  }
}

case https {
  container https {
    description
      "Configures ALTO server stack assuming that
```

```

        TLS-termination is handled internally.";
    container tcp-server-parameters {
        description
            "A wrapper around the TCP server parameters
            to avoid name collisions.";
        uses tcp:tcp-server-grouping {
            refine "local-bind/local-port" {
                default "443";
                description
                    "The ALTO server will listen on the IANA-
                    assigned well-known port value for 'https'
                    (443) if no value is specified.";
            }
        }
    }
    container tls-server-parameters {
        description
            "A wrapper around the TLS server parameters
            to avoid name collisions.";
        uses tls:tls-server-grouping;
    }
    container http-server-parameters {
        description
            "A wrapper around the HTTP server parameters
            to avoid name collisions.";
        uses http:http-server-grouping;
    }
    container alto-server-parameters {
        description
            "A wrapper around the ALTO server parameters
            to avoid name collisions.";
        uses alto-server;
    }
}
}
}
}

grouping alto-server-discovery {
    description
        "Grouping for the configuration of how to set up server
        discovery for clients or other ALTO servers to discover the
        URI of this ALTO server.";
    choice method {
        description
            "Selects among available server discovery methods.";
        case reverse-dns {
            if-feature "xdm-disc";
        }
    }
}

```

```
description
  "Configures DNS NAPTR records for cross-domain ALTO server
  discovery using reverse DNS lookup.";
reference
  "RFC 8686: Application-Layer Traffic Optimization (ALTO)
  Cross-Domain Server Discovery.";
container rdns-naptr-records {
  description
    "Configuration parameters for DNS NAPTR records.";
  leaf-list static-prefix {
    type inet:ip-prefix;
    description
      "Specifies a list of static IP prefixes.";
  }
  leaf-list dynamic-prefix-source {
    type data-source-ref;
    description
      "Dynamic IP prefixes collected from data sources.";
  }
}
}
}
}

grouping alto-server-discovery-client {
  description
    "Grouping for configuration of how a client can discover
    an ALTO server.";
  choice method {
    description
      "Selects among available server discovery methods.";
    case reverse-dns {
      if-feature "xdom-disc";
      description
        "Uses reverse DNS lookup to discover an ALTO server.";
      reference
        "RFC 8686: Application-Layer Traffic Optimization (ALTO)
        Cross-Domain Server Discovery.";
      container rdns-params {
        description
          "Defines a set of parameters for reverse DNS
          lookups.";
        leaf-list dns-server {
          type inet:host;
          description
            "Provides a DNS server list for reverse DNS lookup.";
        }
      }
    }
  }
}
```

```
    }  
  }  
}  
  
grouping alto-logging-system {  
  description  
    "Grouping for configuration of logging system used by the  
    ALTO server.";  
  choice logging-system {  
    description  
      "Selects among available logging systems.";  
    case syslog {  
      description  
        "Specifies syslog as the logging system.";  
      container syslog-params {  
        description  
          "Provides a set of syslog parameters.";  
        leaf config-file {  
          type inet:uri {  
            pattern 'file:.*';  
          }  
          description  
            "Indicates the file location of the syslog  
            configuration.";  
        }  
      }  
    }  
  }  
}  
  
grouping inline-or-truststore-ca-cert-ref {  
  description  
    "Grouping for the reference of a CA certificate to  
    authenticate the TLS client.";  
  choice inline-or-truststore {  
    description  
      "Selects between inline and truststore";  
    case inline {  
      if-feature "ts:inline-definitions-supported";  
      leaf inline {  
        type inline-ca-cert-ref;  
        description  
          "Reference to an inline CA certificate configured by  
          the TLS server.";  
      }  
      description  
        "Reference of an inline CA certificate to authenticate  
        the TLS client.";  
    }  
  }  
}
```

```
    }
    case central-truststore {
      if-feature "ts:central-truststore-supported";
      if-feature "ts:certificates";
      uses ts:central-certificate-ref-grouping;
      description
        "Reference of a CA certificate in the truststore to
        authenticate the TLS client.";
    }
  }
}

grouping inline-or-truststore-ee-cert-ref {
  description
    "Grouping for the reference of a EE certificate to
    authenticate the TLS client.";
  choice inline-or-truststore {
    description
      "Selects between inline and truststore";
    case inline {
      if-feature "ts:inline-definitions-supported";
      leaf inline {
        type inline-ee-cert-ref;
        description
          "Reference to an inline EE certificate configured by
          the TLS server.";
      }
    }
    description
      "Reference of an inline EE certificate to authenticate
      the TLS client.";
  }
  case central-truststore {
    if-feature "ts:central-truststore-supported";
    if-feature "ts:certificates";
    uses ts:central-certificate-ref-grouping;
    description
      "Reference of a EE certificate in the truststore to
      authenticate the TLS client.";
  }
}

grouping inline-or-truststore-public-key-ref {
  description
    "Grouping for the reference of a raw public key to
    authenticate the TLS client.";
  choice inline-or-truststore {
    description
```

```
    "Selects between inline and truststore";
  case inline {
    if-feature "ts:inline-definitions-supported";
    leaf inline {
      type inline-raw-public-key-ref;
      description
        "Reference to an inline public key configured by the
        TLS server.";
    }
    description
      "Reference of an inline public key to authenticate the
      TLS client.";
  }
  case central-truststore {
    if-feature "ts:central-truststore-supported";
    if-feature "ts:public-keys";
    uses ts:central-public-key-ref-grouping;
    description
      "Reference of a raw public key in the truststore to
      authenticate the TLS client.";
  }
}
}

// Top-level container

container alto {
  presence "The ALTO service is enabled";
  description
    "Indicates a set of parameters for both ALTO clients and
    servers. A single device can implement either alto-client or
    alto-server. No need to implement both.";
  list alto-client {
    if-feature alto-client;
    key "client-id";
    description
      "The ALTO client configuration.";
    leaf client-id {
      type string;
      description
        "A unique identifier of a client that can be referenced
        by a data source or a resource creation algorithm to
        communicate with other ALTO servers.";
    }
  }
  container server-discovery {
    description
      "Specifies a set of parameters for ALTO server discovery.";
    uses alto-server-discovery-client;
  }
}
```

```
    }  
  }  
  container alto-server {  
    if-feature alto-server;  
    description  
      "The ALTO server instance configuration.";  
    container listen {  
      description  
        "Configure the ALTO server to listen for ALTO clients.";  
      uses alto-server-listen-stack;  
    }  
    container server-discovery {  
      description  
        "Configures how the ALTO server to be discovered by  
        others.";  
      uses alto-server-discovery;  
    }  
    container logging-system {  
      description  
        "Configure logging system to capture log messages  
        generated by the ALTO server.";  
      uses alto-logging-system;  
    }  
    list cost-type {  
      key "cost-type-name";  
      description  
        "Mapping between name and referenced cost type.";  
      leaf cost-type-name {  
        type cost-type-name;  
        description  
          "The name to reference a cost type.";  
      }  
      leaf cost-mode {  
        type identityref {  
          base cost-mode;  
        }  
        mandatory true;  
        description  
          "The referenced cost mode.";  
      }  
      leaf cost-metric {  
        type identityref {  
          base cost-metric;  
        }  
        mandatory true;  
        description  
          "The referenced cost metric.";  
      }  
    }  
  }
```

```
leaf description {
  type string;
  description
    "A human-readable description for the 'cost-mode' and
    'cost-metric'.";
}
container cost-context {
  if-feature "performance-metrics";
  description
    "Context of how the metric is obtained.";
  leaf cost-source {
    type identityref {
      base cost-source;
    }
    mandatory true;
    description
      "The referenced cost source.";
  }
  container parameters {
    description
      "Additional computation parameters for the cost
      source.";
    choice parameters {
      description
        "Cases of parameters to be augmented.";
    }
  }
}
}
list meta {
  key "meta-key";
  description
    "Mapping of custom meta information";
  reference
    "RFC 7285: Application-Layer Traffic Optimization
    (ALTO) Protocol, Section 8.4.1";
  leaf meta-key {
    type meta-key;
    description
      "Custom meta key";
  }
  leaf meta-value {
    type binary;
    mandatory true;
    description
      "Custom meta value encoded with the base64 encoding
      schema. The encoded value must be a valid JSON
      value.";
```

```
    }
  }
  list auth-client {
    key "client-id";
    description
      "List of authenticated ALTO clients.";
    leaf client-id {
      type string;
      description
        "Identifier to reference an ALTO client.";
    }
    choice authentication {
      description
        "Choice of authentication methods to identify this
        ALTO client. If no authentication method is
        configured, the client must be ignored.";
      case http {
        description
          "The client is authenticated by the HTTP server.";
        container http-auth-client {
          if-feature "http-listen";
          if-feature "http:client-auth-supported";
          if-feature "http:local-users-supported";
          description
            "Parameters of the authenticated HTTP client.";
          leaf user-id {
            type http-user-id-ref;
            mandatory true;
            description
              "Reference of the user-id for the authenticated
              HTTP client.";
          }
        }
      }
      case https {
        description
          "The client is authenticated by the HTTP server.";
        container https-auth-client {
          if-feature "http:client-auth-supported";
          if-feature "http:local-users-supported";
          description
            "Parameters to identify an authenticated HTTPS
            client.";
          leaf user-id {
            type https-user-id-ref;
            description
              "Reference of the user-id for the authenticated
              HTTPS client.";
          }
        }
      }
    }
  }
}
```

```

    }
  }
  container tls-auth-client {
    if-feature "tls:client-auth-supported";
    description
      "Parameters to identify an authenticated TLS
      client.";
    container ca-cert {
      if-feature "tls:client-auth-x509-cert";
      description
        "Reference of the CA certificate to authenticate
        the TLS client.";
      uses inline-or-truststore-ca-cert-ref;
    }
    container ee-cert {
      if-feature "tls:client-auth-x509-cert";
      description
        "Reference of the EE certificate to authenticate
        the TLS client.";
      uses inline-or-truststore-ee-cert-ref;
    }
    container raw-public-key {
      if-feature "tls:client-auth-raw-public-key";
      description
        "Reference of the raw public key to authenticate
        the TLS client.";
      uses inline-or-truststore-public-key-ref;
    }
    leaf tls12-psks {
      if-feature "tls:client-auth-tls12-psk";
      type empty;
      description
        "Indicates that the client is authenticated by
        the TLS server using the configured PSKs
        (pre-shared or pairwise-symmetric keys).";
    }
    leaf tls13-epsks {
      if-feature "tls:client-auth-tls13-epsk";
      type empty;
      description
        "Indicates that the client is authenticated by
        the TLS 1.3 server using the configured external
        PSKs (pre-shared keys).";
    }
  }
}
}
}
}

```

```
list role {
  key "role-name";
  description
    "List of roles for access control.";
  leaf role-name {
    type role-name;
    description
      "Name of a role for access control.";
  }
  leaf-list client {
    type client-ref;
    description
      "List of authenticated ALTO clients assigned to the
      role.";
  }
}
list data-source {
  key "source-id";
  description
    "List of subscribed data sources.";
  leaf source-id {
    type source-id;
    description
      "Data source id that can be referenced by information
      resource creation algorithms.";
  }
  leaf source-type {
    type identityref {
      base source-type;
    }
    mandatory true;
    description
      "Identify the type of the data source.";
  }
  container source-params {
    description
      "Data source specific configuration.";
    reference
      "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol,
      Section 5.4.1";
  }
}
list resource {
  key "resource-id";
  description
    "ALTO information resources to be defined";
  leaf resource-id {
```

```
    type resource-id;
    description
      "resource-id to be defined.";
  }
  leaf resource-type {
    type identityref {
      base resource-type;
    }
    mandatory true;
    description
      "identityref to be defined.";
  }
  leaf description {
    type string;
    description
      "The optional description for this information
      resource.";
  }
  leaf-list accepted-role {
    type role-ref;
    description
      "Roles allowed to access this information resource.";
  }
  leaf-list dependency {
    type resource-ref;
    description
      "A list of dependent information resources.";
  }
  container alto-ird-params {
    when 'derived-from-or-self(..resource-type, '
      + '"alto:ird")';
    description
      "IRD-specific configuration.";
    leaf delegation {
      type inet:uri;
      mandatory true;
      description
        "Upstream IRD to be delegated.";
    }
  }
  container alto-networkmap-params {
    when 'derived-from-or-self(..resource-type, '
      + '"alto:network-map")';
    description
      "Filtered Network Map specific configuration.";
    reference
      "RFC 7285: Application-Layer Traffic Optimization
      (ALTO) Protocol, Sections 11.2.1 and
```

```
        11.3.1";
    leaf is-default {
        type boolean;
        description
            "Sets whether this is the default network map.";
    }
    leaf filtered {
        type boolean;
        default false;
        description
            "Configures whether filtered network map is
            supported.";
    }
    uses algorithm;
}
container alto-costmap-params {
    when 'derived-from-or-self(..,resource-type,'
        + '"alto:cost-map")';
    description
        "Filtered Cost Map specific configuration.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization
        (ALTO) Protocol, Sections 11.2.2 and
        11.3.2";
    leaf filtered {
        type boolean;
        description
            "Configures whether filtered cost map is supported.";
    }
    uses filter-costmap-cap;
    uses algorithm;
}
container alto-endpointcost-params {
    when 'derived-from-or-self(..,resource-type,'
        + '"alto:endpoint-cost")';
    description
        "Endpoint Cost Service specific configuration.";
    reference
        "RFC 7285: Application-Layer Traffic Optimization
        (ALTO) Protocol, Section 11.5";
    uses filter-costmap-cap;
    uses algorithm;
}
container alto-endpointprop-params {
    when 'derived-from-or-self(..,resource-type,'
        + '"alto:endpoint-prop")';
    description
        "Endpoint Cost Service specific configuration.";
```

```
reference
  "RFC 7285: Application-Layer Traffic Optimization
    (ALTO) Protocol, Section 11.5";
leaf-list prop-type {
  type endpoint-property;
  min-elements 1;
  description
    "Supported endpoint properties.";
}
uses algorithm;
}
container alto-propmap-params {
  when 'derived-from-or-self(..../resource-type,'
    + '"alto:property-map")';
  if-feature "propmap";
  description
    "(Filtered) Entity Property Map specific
      configuration.";
  reference
    "RFC 9240: An ALTO Extension: Entity Property Maps";
  uses algorithm;
}
container alto-cdni-params {
  when 'derived-from-or-self(..../resource-type,'
    + '"alto:cdni")';
  if-feature "cdni";
  description
    "CDNi specific configuration.";
  reference
    "RFC 9241: Content Delivery Network Interconnection
      (CDNI) Request Routing: CDNI Footprint and
      Capabilities Advertisement using ALTO";
  uses algorithm;
}
container alto-update-params {
  when 'derived-from-or-self(..../resource-type,'
    + '"alto:update")';
  if-feature "incr-update";
  description
    "Incremental Updates specific configuration.";
  reference
    "RFC 8895: Application-Layer Traffic Optimization
      (ALTO) Incremental Updates Using Server-Sent
      Events (SSE)";
  uses algorithm;
}
container resource-limits {
  description
```

```

    "Sets resource limits.";
    leaf notify-res-mem-limit {
        type uint64;
        units "bytes";
        description
            "Notification of resource memory usage.

            Notification must be generated when the defined
            threshold is reached.";
    }
    leaf notify-upd-stream-limit {
        when 'derived-from-or-self ../../resource-type, '
            + '"alto:update"';
        if-feature "incr-update";
        type uint64;
        description
            "Notification of number of active update streams.

            Notification must be generated when the defined
            threshold is reached.";
    }
}
}
}
}

// Notifications

notification alto-resource-event {
    if-feature alto-server;
    description
        "Notifications must be generated when notify-res-mem-limit
        and/or notify-upd-stream-limit thresholds are reached.";
    leaf resource-id {
        type resource-ref;
        mandatory true;
        description
            "Resource identifier.";
    }
    leaf notify-res-mem-threshold {
        type uint64;
        units "bytes";
        description
            "The notify-res-mem-limit threshold has been fired.";
    }
    leaf notify-upd-stream-threshold {
        if-feature "incr-update";
        type uint64;
    }
}

```

```

        description
            "The notify-upd-stream-limit threshold has been fired.";
    }
}
}
<CODE ENDS>

```

7.2. The "ietf-alto-stats" YANG Module

```

<CODE BEGINS> file "ietf-alto-stats@2026-02-04.yang"
module iETF-alto-stats {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-alto-stats";
    prefix alto-stats;

    import iETF-yang-types {
        prefix yang;
        reference
            "RFC 9911: Common YANG Data Types, Section 3";
    }
    import iETF-alto {
        prefix alto;
        revision-date 2026-02-04;
        reference
            "RFC XXXX: YANG Data Models for the Application-Layer
            Traffic Optimization (ALTO) Protocol";
    }

    organization
        "IETF ALTO Working Group";
    contact
        "WG Web:  <https://datatracker.ietf.org/wg/alto/about/>
        WG List:  <alto@ietf.org>";
    description
        "This YANG module defines a set of statistics of an ALTO
        server instance.

        Copyright (c) 2026 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject to
        the license terms contained in, the Revised BSD License set
        forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX

```

```
(https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
for full legal notices.";

revision 2026-02-04 {
  description
    "Initial Version.";
  reference
    "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
}

// Groupings

grouping server-level-stats {
  description
    "This grouping defines statistics for server-level
    monitoring.";
  leaf discontinuity-time {
    type yang:timestamp;
    description
      "The time on the most recent occasion at which the ALTO
      server suffered a discontinuity. This must be initialized
      when the ALTO server is configured or rebooted.";
  }
  leaf last-report-time {
    type yang:timestamp;
    description
      "The time on the most recent occasion at which the
      statistics were reported.";
  }
  leaf num-total-req {
    type yang:counter64;
    description
      "The total number of ALTO requests received by the ALTO
      server.";
  }
  leaf num-total-succ {
    type yang:counter64;
    description
      "The total number of successful responses sent by the ALTO
      server.";
  }
  leaf num-total-fail {
    type yang:counter64;
    description
      "The total number of failed responses sent by the ALTO
      server.";
  }
}
```

```
leaf num-total-last-req {
  type yang:gauge64;
  description
    "The total number of ALTO requests received by the ALTO
    server within the last time window. The duration of the
    time window is configured by time-window-size parameter.";
}
leaf num-total-last-succ {
  type yang:gauge64;
  description
    "The total number of successful responses sent by the ALTO
    server within the last time window. The duration of the
    time window is configured by time-window-size parameter.";
}
leaf num-total-last-fail {
  type yang:gauge64;
  description
    "The total number of failed responses sent by the ALTO
    server within the last time window. The duration of the
    time window is configured by time-window-size parameter.";
}
}

grouping network-map-stats {
  description
    "This grouping defines resource-specific statistics for the
    network map service only.";
  leaf num-map-pid {
    type yang:gauge64;
    description
      "Number of PIDs contained in the network map.";
  }
  reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)
    Protocol, Section 5";
}

grouping prop-map-stats {
  description
    "This grouping defines resource-specific statistics for the
    endpoint property or property map service only.";
  leaf num-map-entry {
    type yang:gauge64;
    description
      "Number of ALTO entities contained in the property map.";
  }
  reference
    "RFC 7285: Application-Layer Traffic Optimization (ALTO)";
}
```

```

        Protocol, Section 11.4.1
        RFC 9240: An ALTO Extension: Entity Property Maps";
    }

    grouping cdni-stats {
        description
            "This grouping defines resource-specific statistics for the
            CDNI advertisement service only.";
        leaf num-base-obj {
            type yang:gauge64;
            description
                "Number of base CDNI advertisement objects contained in the
                CDNI resource.";
        }
        reference
            "RFC 9241: Content Delivery Network Interconnection (CDNI)
            Request Routing: CDNI Footprint and Capabilities
            Advertisement using ALTO";
    }

    grouping upd-stream-stats {
        description
            "This grouping defines resource-specific statistics for the
            update stream service only.";
        leaf num-upd-stream {
            type yang:gauge64;
            description
                "Number of active update streams connected to the update
                stream service.";
        }
        leaf num-upd-msg-total {
            type yang:gauge64;
            description
                "Total number of update messages sent to all the active
                update streams.";
        }
        leaf num-upd-msg-max {
            type yang:gauge64;
            description
                "The maximum value over the total number of update messages
                sent to each active update stream. Assume there are 3
                active update streams A, B, and C with 4, 3, and 2 update
                messages sent to them respectively, the value of this
                metric is 4. After a while, if there is no new update
                message sent to any update stream, but the update stream A
                is closed, then the value of this metric is updated to
                3.";
        }
    }
}
```

```
leaf num-upd-msg-min {
  type yang:gauge64;
  description
    "The minimum value over the total number of update messages
    sent to each active update stream. The procedure is similar
    to num-msg-max.";
}
leaf num-upd-msg-avg {
  type yang:gauge64;
  description
    "The average value over the total number of update messages
    sent to each active update stream. The procedure is similar
    to num-msg-max.";
}
leaf num-upd-msg-total-last {
  type yang:gauge64;
  description
    "Total number of update messages sent to all the active
    update streams within the last time window. The duration
    of the time window is configured by time-window-size
    parameter.";
}
leaf num-upd-msg-max-last {
  type yang:gauge64;
  description
    "The maximum value over the number of update messages sent
    to each active update stream within the last time window.
    The procedure is similar to num-msg-max, but only count
    the update messages within the last time window. The
    duration of the time window is configured by
    time-window-size parameter.";
}
leaf num-upd-msg-min-last {
  type yang:gauge64;
  description
    "The minimal value over the number of update messages sent
    to each active update stream within the last time window.
    The procedure is similar to num-msg-max, but only count
    the update messages within the last time window. The
    duration of the time window is configured by
    time-window-size parameter.";
}
leaf num-upd-msg-avg-last {
  type yang:gauge64;
  description
    "The average value over the number of update messages sent
    to each active update stream within the last time window.
    The procedure is similar to num-msg-max, but only count
```

```
        the update messages within the last time window. The
        duration of the time window is configured by
        time-window-size parameter.";
    }
    reference
        "RFC 8895: Application-Layer Traffic Optimization (ALTO)
        Incremental Updates Using Server-Sent Events
        (SSE)";
}

grouping resource-level-stats {
    description
        "This grouping defines statistics for resource-level
        monitoring.";
    leaf discontinuity-time {
        type yang:timestamp;
        description
            "The time on the most recent occasion at which the ALTO
            service providing the information resource suffered a
            discontinuity. This must be initialized when the ALTO
            information resource is configured or the ALTO server is
            rebooted.";
    }
    leaf last-report-time {
        type yang:timestamp;
        description
            "The time on the most recent occasion at which the
            statistics are reported.";
    }
    leaf num-res-upd {
        type yang:counter64;
        description
            "The number of version updates since the information
            resource was created.";
    }
    leaf res-mem-size {
        type uint64;
        units "bytes";
        description
            "Memory size utilized by the information resource.";
    }
    leaf res-enc-size {
        type uint64;
        units "bytes";
        description
            "Size of JSON encoded data of the information resource.";
    }
    leaf num-res-req {
```

```
    type yang:counter64;
    description
      "The total number of ALTO requests to this information
       resource.";
  }
  leaf num-res-succ {
    type yang:counter64;
    description
      "The total number of successful responses for requests to
       this information resource.";
  }
  leaf num-res-fail {
    type yang:counter64;
    description
      "The total number of failed responses for requests to this
       information resource.";
  }
  leaf num-res-last-req {
    type yang:gauge64;
    description
      "The number of ALTO requests to this information resource
       within the last time window. The duration of the time
       window is configured by time-window-size parameter.";
  }
  leaf num-res-last-succ {
    type yang:gauge64;
    description
      "The number of successful responses for requests to this
       information resource within the last time window. The
       duration of the time window is configured by
       time-window-size parameter.";
  }
  leaf num-res-last-fail {
    type yang:gauge64;
    description
      "The number of failed responses for requests to this
       information resource within the last time window. The
       duration of the time window is configured by
       time-window-size parameter.";
  }
  container network-map-stats {
    when 'derived-from-or-self(.../alto:resource-type,'
      + '"alto:network-map")';
    description
      "Resource-specific statistics for network map
       service only.";
    uses network-map-stats;
  }
}
```

```
container endpoint-prop-stats {
  when 'derived-from-or-self(..../alto:resource-type,'
    + '"alto:endpoint-prop")';
  description
    "Resource-specific statistics for endpoint property
    service only.";
  uses prop-map-stats;
}
container property-map-stats {
  when 'derived-from-or-self(..../alto:resource-type,'
    + '"alto:property-map")';
  description
    "Resource-specific statistics for entity property map
    service only.";
  uses prop-map-stats;
}
container cdni-stats {
  when 'derived-from-or-self(..../alto:resource-type,'
    + '"alto:cdni")';
  description
    "Resource-specific statistics for CDNI advertisement
    service only.";
  uses cdni-stats;
}
container upd-stream-stats {
  when 'derived-from-or-self(..../alto:resource-type,'
    + '"alto:update")';
  description
    "Resource-specific statistics for update stream service
    only.";
  uses upd-stream-stats;
}
}

// Augment modules to add statistics

augment "/alto:alto/alto:alto-server" {
  description
    "Augmenting statistics and configuration parameters for
    server-level monitoring.";
  container server-level-monitor-config {
    description
      "Configuration parameters for server-level monitoring.";
    leaf time-window-size {
      type uint32;
      units "seconds";
      default "300";
      description
```

```
        "Duration of the time window within that the statistics
          are reported.";
    }
}
container server-level-stats {
  config false;
  description
    "Top-level statistics for the whole ALTO server.";
  uses server-level-stats;
}
}

augment "/alto:alto/alto:alto-server/alto:resource" {
  description
    "Augmenting statistics and configuration parameters for
      resource-level monitoring.";
  container resource-level-stats {
    config false;
    description
      "Common statistics for each information resource.";
    uses resource-level-stats;
  }
}
}
<CODE ENDS>
```

8. Security Considerations

The "ietf-alto" and "ietf-alto-stats" YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in these two YANG modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/ vulnerability:

`"/alto/alto-client/server-discovery"`: This subtree specifies a set of parameters for an ALTO client to discover ALTO servers. Unauthorized access to it could cause intruders to modify the ALTO discovery parameters (e.g., "dns-server") in order to expose an ALTO client to fake ALTO servers. Likewise, this data node can be manipulated to prevent an ALTO client from discovering a reachable ALTO server.

`"/alto/alto-server/auth-client"`: This list specifies all the authenticated ALTO clients on an ALTO server. Unauthorized write access to this list can allow intruders to modify the entries so as to add a client that have not been authenticated yet or delete a client that has already been authenticated. Likewise, this data node can be manipulated to prevent access of legitimate ALTO clients.

`"/alto/alto-server/role"`: This list specifies roles which authenticated ALTO clients were assigned to for access control. Unauthorized write access to this list allow intruders to modify the entries so as to permit access that should not be permitted, or deny access that should be permitted.

Some of the readable data nodes in the "ietf-alto" YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/alto/alto-server/logging-system"`: This subtree provides configuration to select a logging system to capture log messages generated by an ALTO server. Unauthorized read access of this node can allow intruders to access logging information, which could be used to craft an attack the server.

The "ietf-alto" supports an HTTP listen mode to cover cases where the ALTO server stack does not handle the TLS termination itself, but is handled by a separate component. Special care should be considered when such mode is enabled. Note that the default listen mode is "https".

Also, please be aware that these modules include choice nodes that can be augmented by other extended modules. The augmented data nodes may be considered sensitive or vulnerable in some network environments. For instance, an augmented case of the "source-params" choice in "data-source" may include authentication information about how to access a data source including private network information. The "yang-datastore" case in Appendix A.3 is such an example. The "restconf" and "netconf" nodes in it may reveal the access to a private YANG datastore. Thus, those extended modules may have the NACM extension "default-deny-all" set.

These modules use groupings defined in other RFCs that define data nodes that do set the NACM "default-deny-all" and "default-deny-write" extensions. Specifically, the following data nodes reuse groupings with their security considerations:

`"/alto/alto-server/listen/http/http-server-parameters":` This subtree reuses the "http-server-grouping" grouping defined in [I-D.ietf-netconf-http-client-server]. The security considerations of [I-D.ietf-netconf-http-client-server] have been applied to it. Specifically, the "server-name" and "client-authentication" nodes in it may be considered sensitive or vulnerable. For this reason, the NACM extension "default-deny-write" has been applied to them.

`"/alto/alto-server/listen/https/http-server-parameters":` This subtree reuses the "http-server-grouping" grouping defined in [I-D.ietf-netconf-http-client-server]. The security considerations of [I-D.ietf-netconf-http-client-server] have been applied to it. Specifically, the "server-name" and "client-authentication" nodes in it may be considered sensitive or vulnerable. For this reason, the NACM extension "default-deny-write" has been applied to them.

`"/alto/alto-server/listen/https/tls-server-parameters":` This subtree reuses the "tls-server-grouping" grouping defined in [RFC9645]. The security considerations of [RFC9645] have been applied to it. Specifically, all the "key" and "private-key" data nodes in it have the NACM extension "default-deny-all" set, thus preventing unrestricted read-access to the cleartext key values. Also, all writable data nodes in it may be considered sensitive or vulnerable. For this reason, the NACM extension "default-deny-write" has been applied to them.

9. IANA Considerations

This document registers the following URIs in the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-alto
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-alto-stats
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

This document registers the following two YANG modules in the "YANG Module Names" registry [RFC6020]:

Name: ietf-alto
Namespace: urn:ietf:params:xml:ns:yang:ietf-alto
Prefix: alto
Maintained by IANA: N
Reference: [RFC XXXX]

Name: ietf-alto-stats
Namespace: urn:ietf:params:xml:ns:yang:ietf-alto-stats
Prefix: alto-stats
Maintained by IANA: N
Reference: [RFC XXXX]

10. References

10.1. Normative References

- [I-D.ietf-netconf-http-client-server]
Watsen, K., "YANG Groupings for HTTP Clients and HTTP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-http-client-server-32, 30 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-http-client-server-32>>.
- [I-D.ietf-netconf-netconf-client-server]
Watsen, K., "A YANG Data Model for NETCONF Clients and Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-netconf-client-server-41, 4 December 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-netconf-client-server-41>>.
- [I-D.ietf-netconf-restconf-client-server]
Watsen, K., "A YANG Data Model for RESTCONF Clients and Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-restconf-client-server-45, 4 December 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-restconf-client-server-45>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/rfc/rfc3688>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <<https://www.rfc-editor.org/rfc/rfc5424>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/rfc/rfc6020>>.
- [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, <<https://www.rfc-editor.org/rfc/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/rfc/rfc6242>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/rfc/rfc7285>>.
- [RFC7286] Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and H. Song, "Application-Layer Traffic Optimization (ALTO) Server Discovery", RFC 7286, DOI 10.17487/RFC7286, November 2014, <<https://www.rfc-editor.org/rfc/rfc7286>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/rfc/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

- [RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <<https://www.rfc-editor.org/rfc/rfc8189>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/rfc/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/rfc/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.
- [RFC8686] Kiesel, S. and M. Stiernerling, "Application-Layer Traffic Optimization (ALTO) Cross-Domain Server Discovery", RFC 8686, DOI 10.17487/RFC8686, February 2020, <<https://www.rfc-editor.org/rfc/rfc8686>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/rfc/rfc8895>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/rfc/rfc8896>>.
- [RFC9240] Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "An Extension for Application-Layer Traffic Optimization (ALTO): Entity Property Maps", RFC 9240, DOI 10.17487/RFC9240, July 2022, <<https://www.rfc-editor.org/rfc/rfc9240>>.
- [RFC9241] Seedorf, J., Yang, Y., Ma, K., Peterson, J., and J. Zhang, "Content Delivery Network Interconnection (CDNI) Footprint and Capabilities Advertisement Using Application-Layer Traffic Optimization (ALTO)", RFC 9241, DOI 10.17487/RFC9241, July 2022, <<https://www.rfc-editor.org/rfc/rfc9241>>.

- [RFC9274] Boucadair, M. and Q. Wu, "A Cost Mode Registry for the Application-Layer Traffic Optimization (ALTO) Protocol", RFC 9274, DOI 10.17487/RFC9274, July 2022, <<https://www.rfc-editor.org/rfc/rfc9274>>.
- [RFC9275] Gao, K., Lee, Y., Randriamasy, S., Yang, Y., and J. Zhang, "An Extension for Application-Layer Traffic Optimization (ALTO): Path Vector", RFC 9275, DOI 10.17487/RFC9275, September 2022, <<https://www.rfc-editor.org/rfc/rfc9275>>.
- [RFC9439] Wu, Q., Yang, Y., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "Application-Layer Traffic Optimization (ALTO) Performance Cost Metrics", RFC 9439, DOI 10.17487/RFC9439, August 2023, <<https://www.rfc-editor.org/rfc/rfc9439>>.
- [RFC9641] Watsen, K., "A YANG Data Model for a Truststore", RFC 9641, DOI 10.17487/RFC9641, October 2024, <<https://www.rfc-editor.org/rfc/rfc9641>>.
- [RFC9643] Watsen, K. and M. Scharf, "YANG Groupings for TCP Clients and TCP Servers", RFC 9643, DOI 10.17487/RFC9643, October 2024, <<https://www.rfc-editor.org/rfc/rfc9643>>.
- [RFC9645] Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", RFC 9645, DOI 10.17487/RFC9645, October 2024, <<https://www.rfc-editor.org/rfc/rfc9645>>.
- [RFC9911] Schindler, J., Ed., "Common YANG Data Types", RFC 9911, DOI 10.17487/RFC9911, December 2025, <<https://www.rfc-editor.org/rfc/rfc9911>>.

10.2. Informative References

- [RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <<https://www.rfc-editor.org/rfc/rfc6291>>.
- [RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <<https://www.rfc-editor.org/rfc/rfc7921>>.

- [RFC7971] Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations", RFC 7971, DOI 10.17487/RFC7971, October 2016, <<https://www.rfc-editor.org/rfc/rfc7971>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/rfc/rfc8340>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/rfc/rfc8346>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/rfc/rfc8641>>.
- [RFC9569] Gao, K., Schott, R., Yang, Y. R., Delwiche, L., and L. Keller, "The Application-Layer Traffic Optimization (ALTO) Transport Information Publication Service (TIPS)", RFC 9569, DOI 10.17487/RFC9569, September 2024, <<https://www.rfc-editor.org/rfc/rfc9569>>.

Appendix A. Examples of Extending the ALTO O&M Data Model

Developers and operators can also extend the ALTO O&M data model to align with their own implementations. Specifically, the following nodes of the data model can be augmented:

- * The server-discovery-manner choice of the server-discovery.
- * The authentication choice of each auth-client.
- * The data-source node.
- * The algorithm choice of the resource-params of each resource.

A.1. An Example Module for Extended Server Discovery Manners

The base module "ietf-alto" only includes a reverse DNS based server discovery manner. The following example module demonstrates how additional server discovery methods can be augmented into the base data model.

The case internet-routing-registry allows the ALTO server to update the server URI to the attribute of the corresponding aut-num class in IRR.

The case peeringdb allows the ALTO server to update the server URI to the org object of the organization record in PeeringDB.

```
module example-alto-server-discovery {
  yang-version 1.1;

  namespace "https://example.com/ns/example-alto-server-discovery";
  prefix ex-alto-disc;

  import ietf-alto {
    prefix alto;
    reference
      "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
  }

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 9911: Common YANG Data Types";
  }

  organization
    "Example, Inc.";

  contact
    "Example, Inc.
    Customer Service

    E-mail: alto-oam-yang@example.com";

  description
    "This module contains a collection of vendor-specific cases of
    server discovery mechanisms for ALTO.";

  revision 2026-02-04 {
    description
      "Version 1.0";
    reference
      "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
  }

  augment "/alto:alto/alto:alto-server/alto:server-discovery"
```

```
    + "/alto:method" {
description
  "Examples of server discovery mechanisms provided by the ALTO
  server.";
case internet-routing-registry {
  description
    "Update descr attributes of an aut-num class in an Internet
    Routing Registry (IRR) database for ALTO server discovery
    using Routing Policy Specification Language (RPSL).";
  reference
    "RFC 2622: Routing Policy Specification Language (RPSL).";
  container irr-params {
    description
      "Configuration parameters for IRR database.";
    leaf aut-num {
      type inet:as-number;
      description
        "The Autonomous System (AS) number to be updated.";
    }
  }
}
case peeringdb {
  description
    "Update metadata of a network record in PeeringDB database
    for ALTO server discovery using PeeringDB lookup.";
  container peeringdb-params {
    description
      "Configuration parameters for PeeringDB database.";
    leaf org-id {
      type uint32;
      description
        "Specifies an identifier that refers to the org object
        of the organization record in PeeringDB.";
    }
  }
}
}

augment "/alto:alto/alto:alto-client"
  + "/alto:server-discovery/alto:method" {
description
  "Examples of server discovery mechanisms used by an ALTO
  client.";
case internet-routing-registry {
  description
    "Use IRR to discover an ALTO server.";
  reference
    "RFC 2622: Routing Policy Specification Language (RPSL).";
```

```

    container irr-params {
      description
        "Configuration for IRR query using RPSL.";
      leaf whois-server {
        type inet:host;
        description
          "Whois server for an IRR query using RPSL.";
      }
    }
  }
}
case peeringdb {
  description
    "Use PeeringDB to discover an ALTO server.";
  container peeringdb-params {
    description
      "Configuration for PeeringDB queries.";
    leaf peeringdb-endpoint {
      type inet:uri;
      description
        "Endpoint of PeeringDB API server.";
    }
  }
}
}
}
}

```

A.2. An Example Module for Extended Client Authentication Approaches

The base module "ietf-alto" only includes the client authentication approaches directly provided by the HTTP server. However, an implementation may authenticate clients in different ways. For example, an implementation may delegate the authentication to a third-party OAuth 2.0 server. The following example module demonstrates how additional client authentication approaches can enrich the base data model.

In this example, the oauth2 case includes the URI to a third-party OAuth 2.0 based authorization server that the ALTO server can redirect to for the client authentication.

```

module example-alto-auth {
  yang-version 1.1;

  namespace "https://example.com/ns/example-alto-auth";
  prefix ex-alto-auth;

  import ietf-inet-types {
    prefix inet;
  }
}

```

```
    reference
      "RFC 9911: Common YANG Data Types";
  }

import ietf-alto {
  prefix alto;
  reference
    "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
}

organization
  "Example, Inc.";

contact
  "Example, Inc.
  Customer Service

  E-mail: alto-oam-yang@example.com";

description
  "This module contains a collection of vendor-specific cases of
  client authentication approaches for ALTO.";

revision 2026-02-04 {
  description
    "Version 1.0";
  reference
    "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
}

augment "/alto:alto/alto:alto-server/alto:auth-client"
  + "/alto:authentication" {
  description
    "Example of extended ALTO client authentication approaches.";
  case oauth2 {
    description
      "Example of authentication by a third-party OAuth 2.0
      server.";
    container oauth2 {
      description
        "Parameters for authentication by a third-party OAuth 2.0
        server.";
      leaf oauth2-server {
        type inet:uri;
        description
          "The URI to the authorization server.";
      }
    }
  }
}
```

```

    }
  }
}

```

A.3. Example Module for Extended Data Sources

The base module "ietf-alto" does not include any choice cases for specific data sources. The following example module demonstrates how a implementation-specific data source can be augmented into the base data model.

The yang-datastore case is used to import the YANG data from a YANG model-driven datastore. It includes:

- * datastore to indicate which datastore is fetched.
- * target-paths to specify the list of nodes or subtrees in the datastore.
- * protocol to indicate which protocol is used to access the datastore. Either restconf or netconf can be used.

```

module example-alto-data-source {
  yang-version 1.1;

  namespace "https://example.com/ns/example-alto-data-source";
  prefix ex-alto-ds;

  import ietf-alto {
    prefix alto;
    reference
      "RFC XXXX: YANG Data Models for the Application-Layer
       Traffic Optimization (ALTO) Protocol";
  }

  import ietf-datastores {
    prefix ds;
    reference
      "RFC 8342: Network Management Datastore Architecture (NMDA)";
  }

  import ietf-yang-push {
    prefix yp;
    reference
      "RFC 8641: Subscription to YANG Notifications for Datastore
       Updates";
  }
}

```

```
}

import ietf-netconf-client {
  prefix ncc;
  reference
    "RFC HHHH: NETCONF Client and Server Models";
}

import ietf-restconf-client {
  prefix rcc;
  reference
    "RFC IIII: YANG Groupings for RESTCONF Clients and RESTCONF
      Servers";
}

organization
  "Example, Inc.";

contact
  "Example, Inc.
  Customer Service

  E-mail: alto-oam-yang@example.com";

description
  "This module contains a collection of vendor-specific cases of
  data sources for ALTO.";

revision 2026-02-04 {
  description
    "Version 1.0";
  reference
    "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
}

identity yang-datastore {
  base alto:source-type;
  description
    "Identity for data source of YANG-based datastore.";
}

identity protocol-type {
  description
    "Base identity for protocol type.";
}

identity netconf {
```

```
    base protocol-type;
    description
      "Identity for NETCONF protocol.";
  }

  identity restconf {
    base protocol-type;
    description
      "Identity for RESTCONF protocol.";
  }

  augment "/alto:alto/alto:alto-server/alto:data-source"
    + "/alto:source-params" {
    when 'derived-from-or-self(..alto:source-type,'
      + '"yang-datastore")';
    description
      "Example data source for local or remote YANG datastore.";
    container yang-datastore-source-params {
      description
        "YANG datastore specific configuration.";
      leaf datastore {
        type ds:datastore-ref;
        mandatory true;
        description
          "Reference of the datastore from which to get data.";
      }
      list target-paths {
        key name;
        description
          "XML Path Language (XPath) to subscribed YANG datastore
            node or subtree.";
        leaf name {
          type string;
          description
            "Name of the supported XPath or subtree filters.";
        }
        uses yp:selection-filter-types;
      }
      leaf protocol {
        type identityref {
          base protocol-type;
        }
        description
          "Indicates the protocol that is used to access the YANG
            datastore.";
      }
    }
    container restconf {
      uses rcc:restconf-client-app-grouping {
```

```

        when 'derived-from-or-self(..../protocol, "restconf")';
    }
    description
        "Parameters for the RESTCONF endpoint of the YANG
        datastore.";
    }
    container netconf {
        uses ncc:netconf-client-app-grouping {
            when 'derived-from-or-self(..../protocol, "netconf")';
        }
        description
            "Parameters for the NETCONF endpoint of the YANG
            datastore.";
    }
}
}
}
}
}

```

A.4. An Example Module for Information Resource Creation Algorithm

The base module "ietf-alto" does not include any choices cases for information resource creation algorithms. But developers may augment the "ietf-alto" module with definitions for custom creation algorithms for different information resources. The following example module demonstrates the parameters of a network map creation algorithm that translates an IETF Layer 3 unicast topology into a network map.

```
module: example-alto-alg
```

```

augment /alto:alto/alto:alto-server/alto:resource
    /alto:alto-networkmap-params/alto:algorithm:
+--:(l3-unicast-cluster)
+--rw l3-unicast-cluster-algorithm
+--rw l3-unicast-topo
|   +--rw source-datastore      alto:data-source-ref
|   +--rw topo-name?           leafref
+--rw depth?                   uint32

```

This example defines a creation algorithm called l3-unicast-cluster-algorithm for the network map resource. It takes two algorithm-specific parameters:

'l3-unicast-topo': This parameter contains information referring to the target path name of an operational yang-datastore data source node (See Appendix A.3) subscribed in the data-source list (See Section 5.4.1). The referenced target path in the corresponding yang-datastore data source is assumed for an IETF layer 3 unicast

topology defined in [RFC8346]. The algorithm uses the topology data from this data source to compute the ALTO network map resource. 'source-datastore' refers to the 'source-id' of the operational yang-datastore data source node, and 'topo-name' refers to the 'name' of the target path in the source datastore.

'depth': This optional parameter sets the depth of the clustering algorithm. For example, if the depth is set to 1, the algorithm will generate PID for every l3-node in the topology.

The creation algorithm can be reactively called once the referenced data source updates. Therefore, the ALTO network map resource can be updated dynamically.

```
module example-alto-alg {
  yang-version 1.1;

  namespace "https://example.com/ns/example-alto-alg";
  prefix ex-alto-alg;

  import ietf-alto {
    prefix alto;
    reference
      "RFC XXXX: YANG Data Models for the Application-Layer
        Traffic Optimization (ALTO) Protocol";
  }

  import ietf-datastores {
    prefix ietf-datastores;
    reference
      "RFC 8342: Network Management Datastore Architecture (NMDA)";
  }

  import example-alto-data-source {
    prefix ex-alto-ds;
  }

  organization
    "Example, Inc.";

  contact
    "Example, Inc.
     Customer Service

     E-mail: alto-oam-yang@example.com";

  description
    "This module contains a collection of vendor-specific cases of
```

```
information resource creation algorithms for ALTO.";

revision 2026-02-04 {
  description
    "Version 1.0";
  reference
    "RFC XXXX: YANG Data Models for the Application-Layer
      Traffic Optimization (ALTO) Protocol";
}

augment "/alto:alto/alto:alto-server/alto:resource"
  + "/alto:alto-networkmap-params/alto:algorithm" {
  description
    "Example of a network map creation algorithm.";
  case l3-unicast-cluster {
    description
      "Example algorithm translating a Layer 3 unicast topology
        of Interface to the Routing System (I2RS) to an ALTO
        network map.";
    container l3-unicast-cluster-algorithm {
      description
        "Parameters for l3-unicast-cluster algorithm.";
    }
    container l3-unicast-topo {
      leaf source-datastore {
        type alto:data-source-ref;
        must 'deref(..)/../alto:source-params'
          + '/ex-alto-ds:yang-datastore-source-params'
          + '/ex-alto-ds:datastore '
          + '= "ietf-datastores:operational"'
          {
            error-message
              "The referenced YANG datastore MUST be
                operational";
          }
        mandatory true;
        description
          "The data source to YANG datastore.";
      }
      leaf topo-name {
        type leafref {
          path '/alto:alto/alto:alto-server/alto:data-source'
            + '[alto:source-id'
              + ' = current()/../source-datastore]'
              + '/alto:source-params'
              + '/ex-alto-ds:yang-datastore-source-params'
              + '/ex-alto-ds:target-paths/ex-alto-ds:name';
        }
        description

```

```

        "The name of the IETF Layer 3 unicast topology.";
    }
    description
        "The data source info to an IETF Layer 3 unicast
        topology.";
    }
    leaf depth {
        type uint32;
        description
            "The depth of the clustering.";
    }
    }
    }
    }
    }

```

A.5. Example Usage

This section presents an example showing how the base data model and all the extended models above are used to set up an ALTO server and configure corresponding components (e.g., data source listener, information resource, and access control).

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```

{
  "ietf-alto:alto": {
    "alto-server": {
      "listen": {
        "https": {
          "tcp-server-parameters": {
            "local-bind": [
              {
                "local-address": "0.0.0.0"
              }
            ]
          },
        },
        "alto-server-parameters": {},
        "http-server-parameters": {
          "server-name": "alto.example.com",
          "client-authentication": {
            "users": {
              "user": [
                {
                  "user-id": "alice",
                  "basic": {
                    "username": "alice",
                    "password": {

```

```

        "hashed-password": "$0$p8ssw0rd"
      }
    }
  ]
}
},
"tls-server-parameters": {
  "server-identity": {
    "raw-private-key": {
      "inline-definition": {
        "hidden-private-key": [null]
      }
    }
  }
},
"server-discovery": {
  "example-alto-server-discovery:irr-params": {
    "aut-num": 64496
  }
},
"auth-client": [
  {
    "client-id": "alice",
    "https-auth-client": {
      "user-id": "alice"
    }
  },
  {
    "client-id": "bob",
    "example-alto-auth:oauth2": {
      "oauth2-server": "https://auth.example.com/login"
    }
  }
],
"role": [
  {
    "role-name": "group0",
    "client": [
      "alice",
      "bob"
    ]
  }
],
"data-source": [

```

```

{
  "source-id": "test-yang-ds",
  "source-type": "example-alto-data-source:yang-datastore",
  "source-params": {
    "example-alto-data-source:yang-datastore-source-params": {
      "datastore": "ietf-datastores:operational",
      "target-paths": [
        {
          "name": "network-topology",
          "datastore-xpath-filter": "/network-topology:\
network-topology/topology[topology-id=bgp-example-ipv4-topology]"
        }
      ],
      "protocol": "restconf",
      "restconf": {
        "listen": {
          "endpoints": {
            "endpoint": [
              {
                "name": "example restconf server",
                "tcp": {
                  "tcp-server-parameters": {
                    "local-bind": [
                      {
                        "local-address": "192.0.2.2"
                      }
                    ]
                  },
                  "http-client-parameters": {
                    "uri": {
                      "scheme": "https",
                      "authority": {
                        "host": "192.0.2.2"
                      }
                    }
                  },
                  "tls-client-parameters": {
                    "client-identity": {
                      "raw-private-key": {
                        "inline-definition": {
                          "hidden-private-key": [null]
                        }
                      }
                    },
                    "server-authentication": {
                      "raw-public-keys": {
                        "inline-definition": {
                          "public-key": [

```

```

        {
          "name": "alto-pubkey",
          "public-key-format": "ietf-\
crypto-types:subject-public-key-info-format",
          "public-key": "\
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsmEuE0u2Q7QB/s0OF8vf+\
f2jw5rNpQt+sWsN9gpOreaOMyAAYyytMF13PZYADO50j3G0eq1jx8R//ZUc+\
G0PN4fpOLDpj4lC5vBb9pnQ8EpqOLJGMe3+\
73fVG2wfZOzi408GQfxrnJjVKUiBqyulMGPW60NjZYvomSuU754JwxZcgWymwHMOtRoD\
KglYTFY2R5Pnq1XKxx8+3BF/PiCZ5dBDFap6oXVEwvwqspQjy4MS7u/hU1Q1sQkvxMM+\
HE6svGfndT9vu5SRhQeYxHQrC5C5rbsyWyg8nVdugmyJDQIYK0pZVTtpi8ucvU4JEtjf\
c/hpxFz28FFszwqg3m71lwIDAQAB"
        }
      ]
    }
  }
}
},
"resource": [
  {
    "resource-id": "default-network-map",
    "resource-type": "network-map",
    "accepted-role": [
      "group0"
    ],
    "alto-networkmap-params": {
      "is-default": true,
      "example-alto-alg:l3-unicast-cluster-algorithm": {
        "l3-unicast-topo": {
          "source-datastore": "test-yang-ds",
          "topo-name": "network-topology"
        },
        "depth": 2
      }
    }
  }
]

```

```
}  
}  
}
```

Note that this example only uses a clear text password for demonstration purpose. In practice, it is NOT RECOMMENDED to use any clear text passwords when using this data model.

Appendix B. A Sample ALTO Server Architecture to Implement ALTO O&M YANG Modules

Figure 11 shows a sample architecture for an ALTO server implementation. It indicates the major server components that an ALTO server usually needs to include and the YANG modules that these server components need to implement.

This section does not intend to impose an internal structure of server implementations, but is provided to exemplify how the various data model components can be used, including having provisions for future augmentations.

The following server components need to implement the 'ietf-alto' module (Section 5):

Server manager: Provides the functionality and configuration of the server-level management, including server listen stack setup, server discovery setup, logging system configuration, global metadata, and server-level security configuration.

Information resource manager: Provides the operation and management for creating, updating, and removing ALTO information resources.

Data source listener: An ALTO server may start multiple data source listeners. Each data source listener defines a communication endpoint that can fetch ALTO-related information from data sources. The information can be either raw network/computation information or pre-processed ALTO-level information.

The following components need to implement the 'ietf-alto-stats' module (Section 6) to provide statistics information:

Performance monitor: Collects ALTO-specific performance metrics at a running ALTO server.

Logging and fault manager: Collects runtime logs and failure events that are generated by an ALTO server and the service of each ALTO information resource.

The following components are also important for an ALTO server, although they are not in the scope of the data models defined in this document:

Data broker: An ALTO server may implement a data broker to store network/computation information collected from data sources or cache some preprocessed data. The service of the ALTO information resource can read them from the data broker to calculate ALTO responses and return to ALTO clients.

Algorithm plugin: The service of each ALTO information resource needs to configure an algorithm to decide how to calculate the ALTO responses. The algorithm plugins implement those algorithms. User-specified YANG modules can be applied to different algorithm plugins by augmenting the ALTO modules (Appendix A).

Generally, the ALTO server components illustrated above have the following interactions with each other:

- * Both the server manager and information resource manager will report statistics data to the performance monitor and the logging and fault manager.
- * The algorithm plugins will register callbacks to the corresponding ALTO information resources upon configuration; Once an ALTO information resource is requested, the registered callback algorithm will be invoked.
- * A data source listener will fetch data from the configured data source using the corresponding data source API in either proactive mode (polling) or reactive mode (subscription/publication).
- * A data source listener will send the preprocessed data to a data broker.
- * An algorithm plugin may read data from an optional data broker to calculate the ALTO information resource.

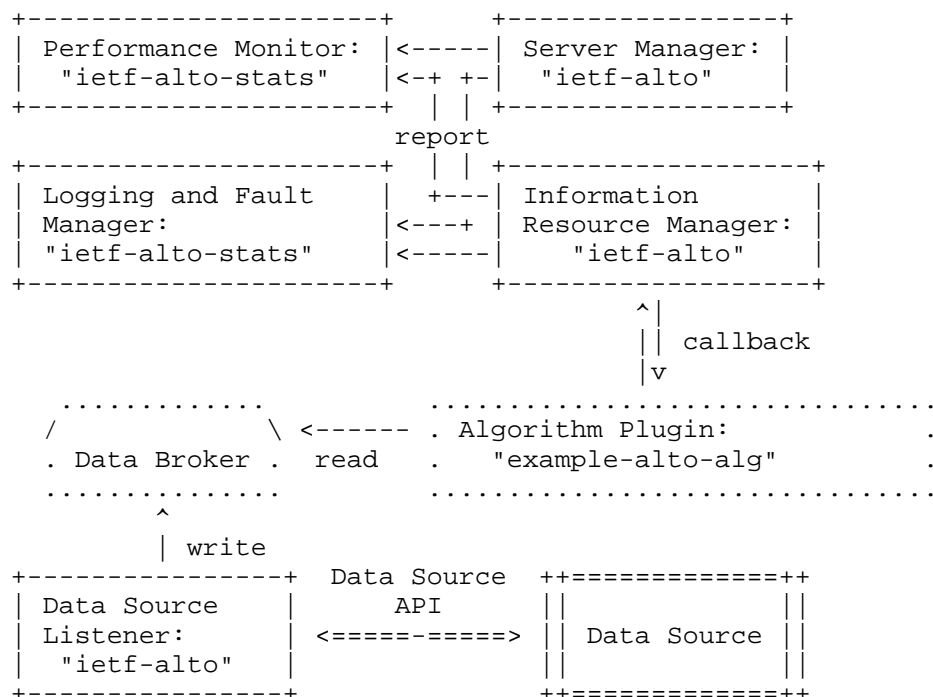


Figure 11: A Sample ALTO Server Architecture and ALTO YANG Modules

Acknowledgements

The authors thank Qin Wu for the help with drafting the initial version of the YANG modules. Big thanks also to ALTO WG chairs (Mohamed Boucadair and Qin Wu) and Adrian Farrel, Dong Guo, Jordi Ros Giralt, Luis M. Contreras, Mahdi Soleimani, Qiao Xiang, Shenshen Chen, and Y. Richard Yang for their thorough reviews, shepherding, and valuable feedback.

Thanks to Dan Romascanu for the opsdire and genart reviews, Andy Bierman for the yangdoctors review, Spencer Dawkins for the tsvar review, Scott Rose for the dnsdire review, and Rich Salz for the secdire review.

Thanks to Martin Duke for the careful AD review.

Authors' Addresses

Jingxuan Jensen Zhang
Tongji University
4800 Cao'An Hwy

Shanghai
201804
China
Email: jingxuan.n.zhang@gmail.com

Dhruv Dhody
Huawei
India
Email: dhruv.ietf@gmail.com

Kai Gao
Sichuan University
No.24 South Section 1, Yihuan Road
Chengdu
Sichuan, 610000
China
Email: kaigao@scu.edu.cn

Roland Schott
Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany
Email: Roland.Schott@telekom.de

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: maqiufang1@huawei.com