

Operations and Management Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

L. M. Contreras
Telefonica
V. Lopez
Nokia
7 July 2025

A YANG Data Model for Network Diagnosis using Scheduled Sequences of OAM
Tests
draft-ietf-opsawg-scheduling-oam-tests-01

Abstract

This document defines a YANG data model for network diagnosis on-demand relying upon Operations, Administration, and Maintenance (OAM) tests. This document defines both 'oam-unitary-test' and 'oam-test-sequence' YANG modules to manage the lifecycle of network diagnosis procedures, primarily intended for use by an SDN controller or network orchestrator, rather than by individual network nodes.

About This Document

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

The latest revision of this draft can be found at <https://vlopezalvarez.github.io/draft-ietf-opsawg-scheduling-oam-tests/draft-ietf-opsawg-scheduling-oam-tests.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-opsawg-scheduling-oam-tests/>.

Discussion of this document takes place on the Operations and Management Area Working Group Working Group mailing list (<mailto:opsawg@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/opsawg/>. Subscribe at <https://www.ietf.org/mailman/listinfo/opsawg/>.

Source for this draft and an issue tracker can be found at <https://github.com/vlopezalvarez/draft-ietf-opsawg-scheduling-oam-tests>.

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 January 2026.

Copyright Notice

Copyright (c) 2025 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
1.1. Terminology and Notations	5
1.2. Requirements Language	5
1.3. Prefix in Data Node Names	6
2. Network-wide OAM Use Cases	6
2.1. Troubleshooting	6
2.2. Birth Certificate	7
2.3. Proactive Supervision	7
2.4. Performance-based Path Routing	8
3. Modelling the Scheduling of OAM Tests	8
3.1. OAM Unitary Test	8
3.2. OAM Test Sequence	10
4. YANG Data Models for Scheduling OAM Tests	12
4.1. YANG Model for Scheduling OAM Unitary Test	12
4.2. YANG Model for OAM Test Sequence	15
5. Using Device Mode Within OAM Scheduling Models	18
6. Operational Considerations	19
6.1. Conflict Resolution and Reporting Among Scheduled OAM Tasks	19
6.2. Coverage of Input Parameters and Output Results	19

7. Security Considerations	20
8. IANA Considerations	20
9. Implementation Status	20
10. References	21
10.1. Normative References	21
10.2. Informative References	23
Appendix A. Examples	24
A.1. Create a TWAMP OAM test	24
Acknowledgments	27
Authors' Addresses	27

1. Introduction

Operations, Administration, and Maintenance (OAM) tasks are fundamental functions of the network management (see, e.g., [RFC7276]). Given the emergence of data models and their utilization in Service Provider's network management and the need to automate the overall service management lifecycle [RFC8969], the management of OAM operations becomes also essential. Relevant data models are still missing to cover specific needs.

Specifically, OAM functions provide the means to identify and isolate faults, measure and report of performance (see section 4.2, [RFC6632]. For example, [RFC5860] defines the three main areas involved in OAM:

- * Fault management, which allows network operators to quickly identify and isolate faults in the network. Examples of these mechanisms for fault detection and isolation are: continuity check, link trace, and loopback.
- * Performance management enables monitoring network performance and diagnosing performance issues (i.e., degradation). Some of the measurements such as frame delay measurement, frame delay variation measurement, and frame loss measurement.
- * Security management defines mechanisms to protect OAM communications from unauthorized access and tampering.

[RFC7276] presents OAM tools for detecting and isolating failures in networks and for performance monitoring, some examples are:

- * Continuity Check: This function verifies that a path exists between two points in a network and that the path is operational.

- * **Loopback:** This function allows a device to loop back a received packet back to the sender for diagnostic purposes. There are multiple technologies for this function, like IP Ping, VCCV Ping, LSP Ping or Ethernet Loopback.
- * **Link Trace:** This function allows a network operator to trace a path through a network from one device to another. Some technologies following this approach are Y.1731 Linktrace [ITU-T-Y1731] or IP traceroute.
- * **Performance Monitoring:** This function allows a network operator to monitor the performance of a network and to identify and diagnose performance issues. Protocols like TWAMP, or Y.1731 DMM/SLM [ITU-T-Y1731] can obtain performance measurements.

More recently, Incident Management [I-D.ietf-nmop-network-incident-yang] focuses on of incident diagnosis, which can be favored by dynamic invocation of OAM tests.

[RFC8531], [RFC8532], [RFC8533], and [RFC8913] defined YANG models for OAM technologies:

- * [RFC8531] "A YANG Data Model for MPLS-TP OAM": defines a YANG data model for connection-oriented OAM protocols. The main aim of this document is to define a generic YANG data model that can be used to configure, control, and monitor connection-oriented OAM protocols such as MPLS-TP OAM, PBB-TE OAM, and G.7713.1 OAM.
- * [RFC8532] "A YANG Data Model for Connectionless OAM Protocols": provides a generic YANG data model that can be used to configure, control, and monitor connectionless OAM protocols such as BFD (Bidirectional Forwarding Detection), LBM (Loopback Messaging), and VCCV (Virtual Circuit Connectivity Verification).
- * [RFC8533] "A YANG Data Model for BFD, LBM, and VCCV OAM Protocols": provides a YANG data model that can be used to retrieve information related to OAM protocols such as Bidirectional Forwarding Detection (BFD), Loopback Messaging (LBM), and Virtual Circuit Connectivity Verification (VCCV).
- * [RFC8913] "A YANG Data Model for Two-Way Active Measurement Protocol (TWAMP)": specifies a YANG data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP).

These OAM related YANG data models defined the parameters required for each of the different tests that are used in network elements today. This work aims to reuse and build upon existing YANG models

for OAM technologies, such as those defined in [RFC8531], [RFC8532], and [RFC8533]. By leveraging these foundational models, this document specifies a YANG data model for scheduling and coordinating sequences of OAM tests, enabling more advanced and automated network diagnosis procedures. In addition to reusing the device-level OAM YANG models from [RFC8531], [RFC8532], and [RFC8533], this document builds upon the generic scheduling framework defined in [I-D.ietf-netmod-schedule-yang]. The ietf-schedule module provides reusable groupings and mechanisms for specifying periods of time, recurrence rules, and scheduling status. These constructs are directly imported and used in the OAM unitary test and OAM test sequence models defined in this document, enabling precise scheduling, repetition, and conflict reporting for OAM tasks in a network-wide context.

The YANG data model resulting from this document will conform to the Network Management Datastore Architecture (NMDA) [RFC8342].

1.1. Terminology and Notations

This document assumes that the reader is familiar with the contents of [RFC7950] "The YANG 1.1 Data Modeling Language".

Following terms are used for the representation of this data model.

- * OAM unitary test: A set of parameters that define a type of OAM test to be invoked. As an example, it includes the type test, configuration parameters, and target results.
- * OAM test sequence: A set of OAM unitary tests that are run based on a set of time constraints, number of repetitions, order, and reporting outputs.

Tree diagrams used in this document follow the notation defined in [RFC8340].

1.2. 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 [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Prefix in Data Node Names

In this document, names of data nodes and other data model objects will be prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in the following table.

Prefix	Yang Module	Reference
oamut	ietf-oam-unitary-tests	RFCXXXX
oamts	ietf-oam-test-sequence	RFCXXXX
yang	ietf-yang-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

RFC Editor Note: Please replace XXXX with the RFC number assigned to this document if the document becomes a RFC. Please remove this note in that case.

2. Network-wide OAM Use Cases

This document covers how to use OAM for network-wide use cases. Following, some illustrative examples are presented.

2.1. Troubleshooting

After the detection of a problem [I-D.ietf-nmop-terminology] in the network, OAM tests are performed to find the root cause for the detected problem. However, a detected problem can be caused by a variety of factors, such as a misconfiguration, hardware failure, or a software bug. OAM tests can help to find the root cause by testing specific components of the network and looking for anomalies or issues. Also, the reliability and efficiency of the tests depend on the nature of the test itself.

There are a variety of OAM tests that can be executed as a function of the target scenario. For example, if the issue is related to a Layer 2 capability, specific tests can be designed and run to check the status of the path via Ethernet Linktrace and later run an Ethernet Loopback to a concrete network element. These tests can be coupled with others to test if any filtering is in place by varying, e.g., some Layer 2 fields or checking the configuration of relevant nodes. If these tests are correct, the operator may want to check the availability of the service (or its delivered performance).

Even though the troubleshooting process may be different depending on the problem detected, there are certain common procedures or logics that can be executed in order to narrow down the cause of the problem and thus help locating candidate root cause.

2.2. Birth Certificate

The aim of a birth certificate process is to validate that all relevant parameters are set appropriately in accordance with the target network service. The birth certificate process is done once the configuration of the network elements is completed, and they are ready for service.

If the birth certificate is successful, it means that the network service is functioning correctly (that is, measured service is matching the expected service) and meets the requirements defined by the operator. The process requires running a set of OAM tasks (e.g., tests) to verify that the service is performing as expected.

The set of OAM tests conducted as part of a birth certificate process depends on the network service that is tested. For example, if the service is a Virtual Private Network (VPN). Two-Way Active Measurement Protocol (TWAMP) Light [RFC5357] will be used, while if the service is an E-LINE, ITU-T Y.1731 Ethernet CFM tests [ITU-T-Y1731] will be executed.

Typically, once the birth certificate process has been completed and the OAM tests have been executed, the test results are stored as part of the documentation process performed by the operator. Many of these tasks take place during pre-deployment phases.

2.3. Proactive Supervision

Some network services require to fulfill strict Service Level Agreements (SLAs). An SLA defines the performance parameters that the service must fulfill in order to meet the requirements of the customer or end user (e.g., IP Connectivity Provisioning Profile (CPP) [RFC7297] and Network Slice Service [RFC9543]).

As part of service fulfillment and assurance (e.g., Section 2.3.3 of [RFC4176]), proactive verification is undertaken to assess whether SLAs are met and implement appropriate adjustment measures when service distortion is observed. Proactive supervision requires running tests both end-to-end, but also on service components to identify early symptoms and resolve issues before they impact the customer or end user, or to prevent or minimize the impact of the end user. Mitigation action may be enforced to alliviate the impact of networks incidents and nullify the impact on services that are delivered via that network.

Proactive testing might be done via OAM tests. These tests can be run periodically at regular intervals depending on the specific SLA requirements and the network operator procedures. These procedures may require documenting the test results for future auditing processes with the customers (eventually, negotiated and agreed with a customer as part of service assurance).

2.4. Performance-based Path Routing

Path Computation Elements (PCEs) are used to compute end-to-end paths in a network [RFC4655]. PCEs are used for Traffic Engineering (TE) purposes (e.g., optimize network performance, reduce congestion, and improve the overall user experience).

There are different algorithms to calculate a path in the network for some of them the PCE requires traffic engineering information. TE information includes data such as link metrics, bandwidth availability, and routing constraints. By using this information, the PCE can compute the optimal path for a particular service [RFC8233], taking into account its constraints and requirements. In addition to TE Metric Extensions in OSPF [RFC7471] or IS-IS [RFC7810], OAM techniques also allow obtaining link metrics like delay and loss which can be used in the PCE algorithms.

3. Modelling the Scheduling of OAM Tests

This document specifies two models: OAM unitary test and OAM test sequence models.

3.1. OAM Unitary Test

The OAM unitary test model encompasses parameters that define a specific type of OAM test to be performed. The YANG model includes a container named "oam-unitary-tests" that serves as a container for activating OAM unitary tests for network diagnosis procedures. Inside the container, there is a list called "oam-unitary-test" representing a list of specific OAM unitary tests. The list key is

defined as "name", which provides a unique name for each test. Each OAM test in the list references a test type with its concrete parameters. The test types are out of the scope of this document. Moreover, each OAM unitary test has two temporal parameters: "period-of-time" and "recurrence". Both are imported from the "ietf-schedule" module from [I-D.ietf-netmod-schedule-yang]. "period-of-time" identifies the period values that contain a precise period of time, while "recurrence" identifies the properties that contain a recurrence rule specification. "unitary-test-status" enumerates the state of the OAM unitary test.

Figure 1 shows the structure of OAM unitary test module:

```

module: ietf-oam-unitary-test
  +-rw oam-unitary-tests
    +-rw oam-unitary-test* [name]
      +-rw name string
      +-rw ne-config* [ne-id]
        | +-rw ne-id rt-types:router-id
        | +-rw (test-type)
      +-rw period-description? string
      +-rw period-start yang:date-and-time
      +-rw time-zone-identifier? sys:timezone-name
      +-rw (period-type)?
        | +--:(explicit)
        | | +-rw period-end? yang:date-and-time
        | | +--:(duration)
        | +-rw duration? duration
      +-rw recurrence-description? string
      +-rw frequency identityref
      +-rw interval? uint32
      +-rw unitary-test-status? enumeration

```

Figure 1: Tree Structure of OAM Unitary Test

(Note: alignment with [I-D.ietf-netmod-schedule-yang] will continue with the progress of that document). The 'unitary-test-status' state machine is shown in Figure 2. The state machine includes the following states:

- * "planned": The initial state where the test is planned.
- * "configured": The state where the test is being configured.
- * "ready": The state where the test is ready to be executed.
- * "on-going": The state where the test is currently running.

- * "stop": The state where the test is manually stopped.
- * "error": The state where an error occurs during the test.
- * "finished": The final state where the test is completed.

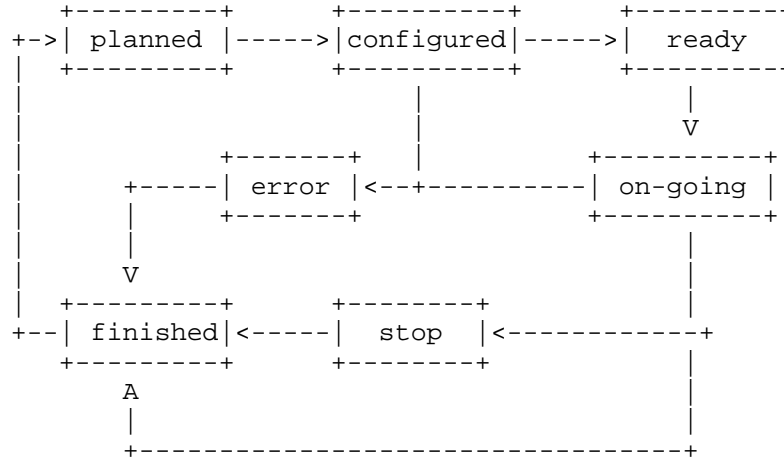


Figure 2: OAM Unitary Test State Machine

3.2. OAM Test Sequence

The OAM test sequence model consists of a collection of OAM unitary tests that are executed based on specified time constraints, repetitions, ordering, and reporting outputs. These sequences provide a structured approach to running multiple OAM tests in a coordinated manner.

Each OAM test sequence references an OAM unitary test type with its concrete parameters. Each OAM test sequence has two temporal parameters: "period-of-time" and "recurrence". Both are imported from the "ietf-schedule" module from [I-D.ietf-netmod-schedule-yang]. "period-of-time" identifies the period values that contain a precise period of time, while "recurrence" identifies the properties that contain a recurrence rule specification. "unitary-test-status" enumerates the state of the OAM test. Finally, "test-sequence-status" shows the state of the OAM test sequence.

Figure 3 shows the structure of OAM test sequence module:

```

module: ietf-oam-test-sequence
  +--rw oam-test-sequence
    +--rw test-sequence* [name]
      +--rw name string
      +--rw test-ref* [name]
        | +--rw name string
        | +--rw (test-type)
        | +--rw numexecutions? uint32
      +--rw period-description? string
      +--rw period-start yang:date-and-time
      +--rw time-zone-identifier? sys:timezone-name
      +--rw (period-type)?
        | +--:(explicit)
        | | +--rw period-end? yang:date-and-time
        | +--:(duration)
        | | +--rw duration? duration
      +--rw recurrence-first
        | +--rw utc-start-time? yang:date-and-time
        | +--rw duration? uint32
      +--rw (recurrence-bound)?
        | +--:(until)
        | | +--rw utc-until? yang:date-and-time
        | +--:(count)
        | | +--rw count? uint32
      +--rw recurrence-description? string
      +--rw frequency identityref
      +--rw interval? uint32
      +--rw test-sequence-status? enumeration

```

Figure 3: OAM test sequence

The 'test-sequence-status' state machine is shown in Figure 4. The state machine includes the following states:

- * "planned": The initial state where the test is planned.
- * "configured": The state where the test is being configured.
- * "ready": The state where the test is ready to be executed.
- * "on-going": The state where the test is currently running.
- * "stop": The state where the test is manually stopped.
- * "success": The state when all unitary tests were successful.
- * "failure": The state when one or more tests in the sequence got an error.

* "error": The state where an error occurs during the test.

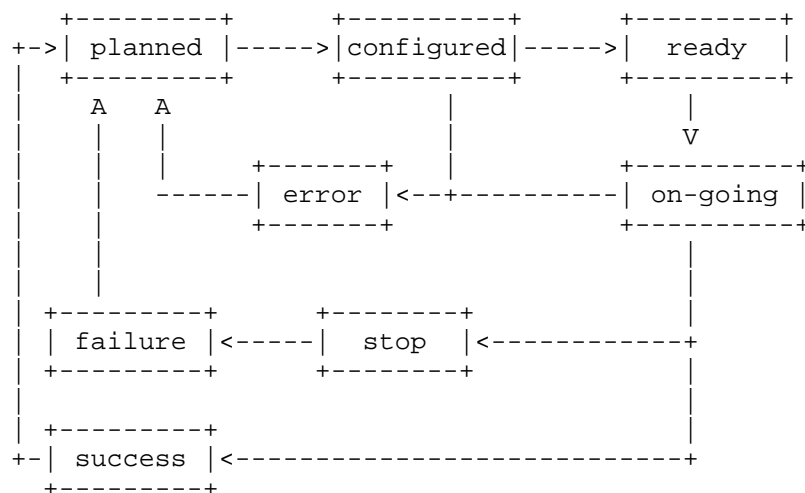


Figure 4: OAM test sequence state machine

4. YANG Data Models for Scheduling OAM Tests

4.1. YANG Model for Scheduling OAM Unitary Test

```

<CODE BEGINS>
file ietf-oam-unitary-test@2024-11-08.yang
module ietf-oam-unitary-test {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-oam-unitary-test";
  prefix "oamut";

  // Import OAM models from RFCs RFC8531, RFC8532 and RFC8533

  // reference ietf-netmod-schedule-yang
  import ietf-schedule {
    prefix schedule;
    reference
      "RFC XXXX: A Common YANG Data Model for Scheduling";
  }

  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
}

```

organization

"IETF OPSAWG (Operations and Management Area Working Group)";

contact

"WG Web: <<https://datatracker.ietf.org/wg/opsawg/>>

WG List: <<mailto:opsawg@ietf.org>>

Author: Luis Miguel Contreras Murillo

<luismiguel.contrerasmurillo@telefonica.com>

Author: Victor Lopez

<victor.lopez@nokia.com>;

description

"This module defines the 'ietf-oam-unitary-test' YANG model for activation of network diagnosis procedures.

Copyright (c) 2024 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.";

// RFC Ed.: update the date below with the date of RFC
// publication and remove this note.
// RFC Ed.: replace XXXX with actual RFC number and remove
// this note.

revision "2024-11-08" {

description

"Initial version";

reference

"RFCXXXX: A YANG Data Model for Network Diagnosis by Scheduling Sequences of OAM Tests";

// Update with the correct RFC number when assigned

}

grouping oam-unitary-test {

description

"Specifies a grouping for OAM unitary test for network diagnosis procedures.";

leaf name {

```
    type string;
    description
        "Defines the name of the test.";
}

list ne-config {
    key ne-id;
    description "List of node configurations required to enable the
        unitary tests.";

    leaf ne-id {
        type rt-types:router-id;
        description
            "A 32-bit number in the dotted-quad format that is used
            to uniquely identify a node within an autonomous system
            the ne-id. This identifier is used for both IPv4 and IPv6.";
    }
    choice test-type {
        mandatory true;
        description
            "Choose the type of test. Pending to add the schema-mount solution";
        // Import OAM models from RFCs RFC8531, RFC8532 and RFC8533
    }
}

container oam-unitary-tests {
    description
        "Container for OAM unitary tests activation for network
        diagnosis procedures.";

    list oam-unitary-test {
        key name;
        description
            "List of OAM unitary tests activation for network diagnosis
            procedures.";

        uses oam-unitary-test;

        uses schedule:period-of-time;

        uses schedule:recurrence;

        leaf unitary-test-status {
            config false;
            type enumeration {
                enum "planned" {
                    description
```

```

        "The test is planned.";
    }
    enum "configure" {
        description
            "The test is configured.";
    }
    enum "ready" {
        description
            "The test status is ready.";
    }
    enum "ongoing" {
        description
            "The test is ongoing.";
    }
    enum "stop" {
        description
            "The test is stopped.";
    }
    enum "finish" {
        description
            "The test is finished.";
    }
    enum "error" {
        description
            "The test has an error.";
    }
    }
    description
        "Status of the test.";
    }
}
}
}
<CODE ENDS>

```

4.2. YANG Model for OAM Test Sequence

```

<CODE BEGINS>
file ietf-oam-test-sequence@2024-11-08.yang

module ietf-oam-test-sequence {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-oam-test-sequence";
    prefix "oamts";

    import ietf-oam-unitary-test {
        prefix "oamut";
        // Update the reference with the correct RFC number or other
    }
}

```

```
// reference when assigned
// reference "RFCXXXX";
}

// reference ietf-netmod-schedule-yang
import ietf-schedule { prefix "schedule"; }

organization
  "IETF OPSAWG (Operations and Management Area Working Group)";

contact
  "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
  WG List:   <mailto:opsawg@ietf.org>
  Author: Luis Miguel Contreras Murillo
            <luismiguel.contrerasmurillo@telefonica.com>
  Author: Victor Lopez
            <victor.lopez@nokia.com>";
description
  "This module defines the 'oam-test-sequence-test' YANG model for
  management of network diagnosis procedures.

  Copyright (c) 2024 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.";

// RFC Ed.: update the date below with the date of RFC
// publication and remove this note.
// RFC Ed.: replace XXXX with actual RFC number and remove
// this note.

revision "2024-11-08" {
  description "Initial version";
  reference "RFCXXXX";
  // Update with the correct RFC number when assigned
}

// Data model definition
```



```
container oam-test-sequence {
  description
    "Container for executing a sequence of ietf-oam-unitary-tests
    N times.";

  list test-sequence {
    key "name";
    description "List of test sequences.";

    leaf name {
      type string;
      description "Unique name for the test sequence.";
    }

    list test-ref {
      key "name";
      description "References to the ietf-oam-unitary-tests.";

      uses "oamut:oam-unitary-test";

      leaf numexecutions {
        type uint32;
        default 1;
        description
          "Number of times the test sequence should be
          executed.";
      }
    }
  }

  uses schedule:period-of-time;

  uses schedule:recurrence-utc;

  leaf test-sequence-status {
    config false;
    type enumeration {
      enum "planned" {
        description
          "The test sequence is planned.";
      }
      enum "configured" {
        description
          "The test sequence is configured.";
      }
      enum "ready" {
        description
          "The test sequence is ready.";
      }
    }
  }
}
```

```
    enum "ongoing" {
        description
            "The test sequence status is ongoing.";
    }
    enum "stop" {
        description
            "The test sequence is stopped.";
    }
    enum "success" {
        description
            "All tests in the sequence were successful.";
    }
    enum "failure" {
        description
            "One or more tests in the sequence got an error.";
    }
    enum "error" {
        description
            "The test sequence status has an error.";
    }
}

description
    "Status of the test sequence execution.";
}
}
}
}
<CODE ENDS>
```

5. Using Device Mode Within OAM Scheduling Models

This section discusses the issues related to reusing device models already defined in IETF within the context of scheduling OAM tests. There are two main approaches to enable OAM scheduling models:

- * Importing YANG model into the OAM scheduling models. This approach will copy the device model into the OAM unitary test model to enable the configuration and utilization of the desired OAM test. This approach requires to recreate new YANG models for each new test type or variation of the device models.
- * Schema-mount allows mounting a data model at a specified location of another (parent) schema. The main difference with importing the YANG modules is that they don't have to be prepared for mounting; any existing modules such as "ietf-twamp" can be mounted without any modifications.

As an example, we will use [RFC8913], which defines a YANG data model for TWAMP, to illustrate how device models could be used.

6. Operational Considerations

6.1. Conflict Resolution and Reporting Among Scheduled OAM Tasks

When multiple OAM tasks are scheduled to run concurrently or overlap in time, conflicts may arise due to resource contention or operational constraints. This document leverages the scheduling status groupings defined in the common schedule YANG module (see [RFC XXXX: A Common YANG Data Model for Scheduling]) to detect and report such conflicts.

The YANG models defined in this document (both for unitary and sequence tests) uses the unitary-test-status and test-sequence-status grouping to indicate the current scheduling state of each OAM task. If a conflict is detected (e.g., two tests require exclusive access to the same resource at the same time), the unitary-test-status and test-sequence-status leaf will reflect this by reporting a value such as error, reporting the conflict.

Operators and management systems SHOULD monitor the scheduling status of OAM tasks and take appropriate action if a conflict is reported. The resolution of conflicts (e.g., rescheduling, prioritization, or cancellation) is implementation-dependent, but MUST be clearly reported via the YANG model status leaves.

When a new unitary-test or test-sequence are scheduled, the request MAY be rejected depending on the server's capability to evaluate the scheduling impact and detect conflicts prior to execution.

6.2. Coverage of Input Parameters and Output Results

The YANG models defined in this document are designed to schedule OAM tests at a network-wide level. The input parameters required to configure and execute specific OAM functions (such as test type, target, and configuration options) are referenced or reused from the existing device-level OAM YANG models (e.g., [RFC8531], [RFC8532], [RFC8533], [RFC8913]). This approach avoids duplication and ensures consistency with established models.

Similarly, the output results of OAM tests—such as test status, performance metrics, and diagnostic information—are expected to be reported using the mechanisms and data nodes defined in those foundational YANG modules. The scheduling models in this document provide references to these results and enable their collection and correlation across multiple tests and devices, but do not redefine the detailed input/output parameters of each OAM function.

In summary, this document focuses on the scheduling, coordination, and status tracking of OAM tests, while relying on existing YANG models for the detailed specification of test parameters and results.

7. Security Considerations

The YANG module targeted in this document defines 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 [RFC5246].

The NETCONF access control model [RFC6536] 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 this YANG module 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) to these data nodes without proper protection can have a negative effect on network operations.

In which refers to the scheduling of the tests, security considerations in [I-D.ietf-netmod-schedule-yang] are also applicable here.

8. IANA Considerations

TBC

9. Implementation Status

This section will be used to track the status of the implementations of the model. It is aimed at being removed if the document becomes RFC.

10. References

10.1. Normative References

- [I-D.ietf-netmod-schedule-yang]
Ma, Q., Wu, Q., Boucadair, M., and D. King, "A Common YANG Data Model for Scheduling", Work in Progress, Internet-Draft, draft-ietf-netmod-schedule-yang-08, 4 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netmod-schedule-yang-08>>.
- [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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/rfc/rfc5246>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarez, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/rfc/rfc5357>>.
- [RFC5860] Vigoureux, M., Ed., Ward, D., Ed., and M. Betts, Ed., "Requirements for Operations, Administration, and Maintenance (OAM) in MPLS Transport Networks", RFC 5860, DOI 10.17487/RFC5860, May 2010, <<https://www.rfc-editor.org/rfc/rfc5860>>.
- [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>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/rfc/rfc6536>>.

- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/rfc/rfc6991>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/rfc/rfc7471>>.
- [RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <<https://www.rfc-editor.org/rfc/rfc7810>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/rfc/rfc7950>>.
- [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>>.
- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/rfc/rfc8233>>.
- [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>>.
- [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>>.
- [RFC8531] Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection-Oriented Operations, Administration, and Maintenance (OAM) Protocols", RFC 8531, DOI 10.17487/RFC8531, April 2019, <<https://www.rfc-editor.org/rfc/rfc8531>>.

- [RFC8532] Kumar, D., Wang, Z., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <<https://www.rfc-editor.org/rfc/rfc8532>>.
- [RFC8533] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "A YANG Data Model for Retrieval Methods for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8533, DOI 10.17487/RFC8533, April 2019, <<https://www.rfc-editor.org/rfc/rfc8533>>.
- [RFC8913] Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, Ed., "Two-Way Active Measurement Protocol (TWAMP) YANG Data Model", RFC 8913, DOI 10.17487/RFC8913, November 2021, <<https://www.rfc-editor.org/rfc/rfc8913>>.

10.2. Informative References

- [I-D.ietf-nmop-network-incident-yang]
Hu, T., Contreras, L. M., Wu, Q., Davis, N., and C. Feng, "A YANG Data Model for Network Incident Management", Work in Progress, Internet-Draft, draft-ietf-nmop-network-incident-yang-05, 6 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-05>>.
- [I-D.ietf-nmop-terminology]
Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-19, 18 June 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-19>>.
- [ITU-T-Y1731]
"OAM Functions and Mechanisms for Ethernet-based Networks", 13 June 2023, <<https://www.itu.int/rec/T-REC-Y.1731/en>>.
- [RFC4176] El Mghazli, Y., Ed., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, DOI 10.17487/RFC4176, October 2005, <<https://www.rfc-editor.org/rfc/rfc4176>>.

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/rfc/rfc4655>>.
- [RFC6632] Ersue, M., Ed. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, DOI 10.17487/RFC6632, June 2012, <<https://www.rfc-editor.org/rfc/rfc6632>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/rfc/rfc7276>>.
- [RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <<https://www.rfc-editor.org/rfc/rfc7297>>.
- [RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <<https://www.rfc-editor.org/rfc/rfc8969>>.
- [RFC9543] Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", RFC 9543, DOI 10.17487/RFC9543, March 2024, <<https://www.rfc-editor.org/rfc/rfc9543>>.

Appendix A. Examples

This section includes a non-exhaustive list of examples to illustrate the use of the models defined in this document.

A.1. Create a TWAMP OAM test

[RFC8913] defines a YANG model for TWAMP. There is an example for TWAMP. The following example contains the information for the four configurations (Control-Client, Server, Session-Sender and Session-Reflector).

An example of a request message body to create a TWAMP OAM test is shown in Figure 5. Session-Sender and Session-Reflector as expanded for illustrative purposes. The TWAMP Test scheduled in this configuration is a one-hour performance monitoring test that runs

daily at 9 AM UTC. This test session is configured to start on October 17, 2023, at 09:00 UTC and recur at the same time every day. The duration of each test run is one hour, as specified by the ISO 8601 format "PT1H", with the test status marked as "scheduled". The test provides insight into network performance by monitoring the selected parameters, allowing for the detection of any potential degradations in service quality over time.

```
{
  "ietf-oam-unitary-test:oam-unitary-tests": {
    "oam-unitary-test": [
      {
        "name": "TWAMP-Test-scheduled-daily",
        "test-type": "twamp-test",
        "period-description": "TWAMP Test Period",
        "period-start": "2023-10-17T09:00:00Z",
        "time-zone-identifier": "UTC",
        "period-type": {
          "duration": {
            "duration": "PT1H"
          }
        },
        "recurrence-description": "Daily at 9 AM UTC",
        "frequency": "oam-types:daily",
        "interval": 1,
        "unitary-test-status": "scheduled",
        "ne-config": [
          {
            "ne-id": "203.0.113.3",
            "twamp": {
              "session-sender": {
                "admin-state": true,
                "test-session": [
                  {
                    "name": "Test1",
                    "ctrl-connection-name": "RouterA",
                    "fill-mode": "zero",
                    "number-of-packets": 900,
                    "periodic-interval": 1,
                    "sent-packets": 2,
                    "rcv-packets": 2,
                    "last-sent-seq": 1,
                    "last-rcv-seq": 1
                  },
                  {
                    "name": "Test2",
                    "ctrl-connection-name": "RouterA",
                    "fill-mode": "random",
```

```
        "number-of-packets": 900,
        "lambda": 1,
        "max-interval": 2,
        "sent-packets": 21,
        "rcv-packets": 21,
        "last-sent-seq": 20,
        "last-rcv-seq": 20
      }
    ]
  }
},
{
  "ne-id": "203.0.113.4",
  "twamp": {
    "session-reflector": {
      "admin-state": true,
      "test-session": [
        {
          "sid": 1232,
          "sender-ip": "203.0.113.3",
          "sender-udp-port": 54000,
          "reflector-ip": "203.0.113.4",
          "reflector-udp-port": 55000,
          "parent-connection-client-ip": "203.0.113.1",
          "parent-connection-client-tcp-port": 16341,
          "parent-connection-server-ip": "203.0.113.2",
          "parent-connection-server-tcp-port": 862,
          "test-packet-dscp": 32,
          "sent-packets": 2,
          "rcv-packets": 2,
          "last-sent-seq": 1,
          "last-rcv-seq": 1
        },
        {
          "sid": 178943,
          "sender-ip": "203.0.113.1",
          "sender-udp-port": 54001,
          "reflector-ip": "192.0.2.2",
          "reflector-udp-port": 55001,
          "parent-connection-client-ip": "203.0.113.1",
          "parent-connection-client-tcp-port": 16341,
          "parent-connection-server-ip": "203.0.113.2",
          "parent-connection-server-tcp-port": 862,
          "test-packet-dscp": 32,
          "sent-packets": 21,
          "rcv-packets": 21,
          "last-sent-seq": 20,
```

```
    "last-rcv-seq": 20
  }
]
}
]
}
]
}
]
}
]
```

Figure 5: Example of a Message Body to Create a TWAMP OAM test

Acknowledgments

TODO acknowledge.

Authors' Addresses

Luis M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Victor Lopez
Nokia
Email: victor.lopez@nokia.com