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

L. M. Contreras
Telefonica
V. Lopez
Nokia
Q. Wu
Huawei

7 May 2026

A YANG Data Model for Network Diagnosis using Scheduled Sequences of OAM Tests

draft-ietf-opsawg-scheduling-oam-tests-06

Abstract

This document defines a YANG data model to support on-demand network diagnosis using 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, intended for use by external management and orchestration systems (including SDN controllers and network orchestrators), 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 November 2026.

Copyright Notice

Table of Contents

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], managing OAM operations is also essential. Relevant data models are still missing to cover specific needs.

The term OAM is used in this document as defined in [RFC6291] and further characterized according to the classification guidelines in [I-D.ietf-opsawg-oam-characterization]. The scope of this document applies primarily to active and hybrid OAM mechanisms, as scheduling tests generally implies the generation of additional OAM traffic. Passive OAM mechanisms are not the focus of this work.

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

* Fault management, which allows network operators 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[RFC0792][RFC4443], VCCV Ping[RFC5085], LSP Ping [RFC4379] or Ethernet Loopback [IEEE-8021Q].

* 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[RFC0792][RFC4443].

* 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[RFC5357], STAMP[RFC8762], Alternative Marking[RFC9341], IOAM (In Situ OAM) [RFC9197], 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 the network incident diagnosis, which can be favored by dynamic invocation of OAM tests.

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

o [RFC8531] "A YANG Data Model for Connection Oriented 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 [RFC6371], TRILL OAM[RFC7174], PBB-TE OAM [IEEE-8021ag], and T-MPLS [ITU-T-G81131] OAM.

o [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)[RFC5880], LBM (Loopback Messaging)[IEEE-8021ag], and VCCV (Virtual Circuit Connectivity Verification)[RFC5085].

o [RFC8533] "A YANG Data Model for Retrieval Methods for the Management of OAM Protocols that Use Connectionless Communications":

provides a YANG data model that can be used to retrieve information related to OAM protocols such as BFD (Bidirectional Forwarding Detection)[RFC5880], LBM (Loopback Messaging)[IEEE-8021ag], and VCCV (Virtual Circuit Connectivity Verification)[RFC5085].

o [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 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.

o OAM unitary test: A set of parameters that define a type of OAM test to be invoked. As an example, it includes the test type, configuration parameters, and target results.

o 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].

This document adopts the OAM characterization defined in [I-D.ietf-opsawg-oam-characterization]:

o Active OAM uses dedicated OAM packets to assess network performance or verify continuity.

o Passive OAM observes existing data traffic without injecting OAM packets.

o Hybrid OAM combines active and passive methods.

The use of the terms in-band and out-of-band is avoided in this document, consistent with [I-D.ietf-opsawg-oam-characterization].

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-test	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. These use cases rely primarily on active or hybrid OAM methods, depending on whether dedicated test packets or augmented data packets are used, following [I-D.ietf-opsawg-oam-characterization].

The following illustrative examples are provided.

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 identify likely root causes 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 locate 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 fulfillment of 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. This help 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. Within 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 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" indicates the state of the OAM unitary test (see the state machine in Figure 2).

Each oam-unitary-test instance defined by this model is conceptually an instance of an active or hybrid OAM operation, since it triggers the generation or coordination of OAM packets. The YANG model allows such differentiation by referencing the underlying test type identity.

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 managed?     boolean
          |  +--rw test-type?   identityref
          |  +--rw root
          +--rw (schedule-class)?
          |  +--:(period)
          |  |  +--rw period
          |  |     +--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
          |  +--:(recurrence)
          |     +--rw recurrence
          |        +--rw recurrence-description?   string
          |        +--rw frequency?                identityref
          |        +--rw interval?                 uint32
          +--rw state?                    identityref
          +--rw version?                  uint16
          +--rw schedule-type?            identityref
          +--ro local-time?               yang:date-and-time
          +--ro last-update?              yang:date-and-time
          +--ro counter?                  yang:counter32
          +--ro last-occurrence?          yang:date-and-time
          +--ro upcoming-occurrence?      yang:date-and-time
          +--ro last-failed-occurrence?   yang:date-and-time
          +--ro failure-counter?          yang:counter32
```

```
          +--ro unitary-test-status?      identityref
```

Figure 1: Tree Structure of OAM Unitary Test

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 by the management and hasn't been applied to the network element.

* "configured": The state where the test is being configured. This state is triggered when the planned test configuration is applied to the network element.

* "ready": The state where the test is ready to be executed. This is state is triggered after the planned test configuration is applied and before the test is executed.

* "on-going": The state where the test is currently running. This state is triggered when the test has been executed but the test results haven't been produced.

* "stop": The state where the test is manually stopped. This state is triggered when the test is manually interrupted.

* "error": The state where an error occurs during the test. This state is triggered when one or more tests haven't been conducted successfully. Implementations may report a more specific error cause using child identities such as "resource-contention" or "priority".

* "success": The final state where the test is completed. This state is triggered when the test has been conducted successfully.

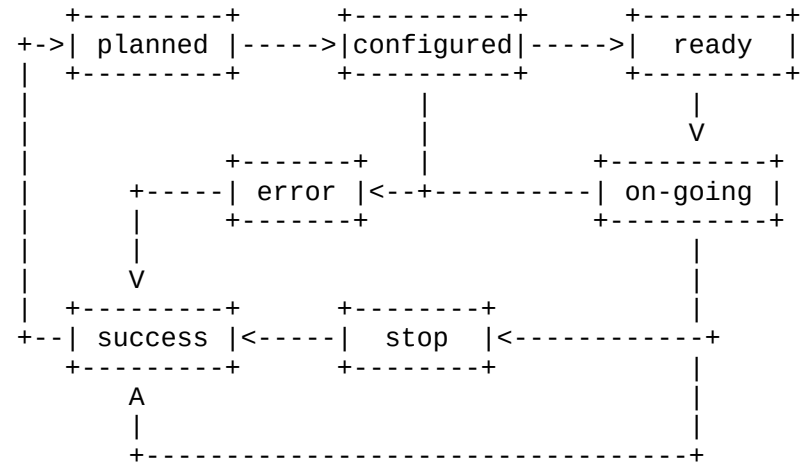

Figure 2: OAM Unitary Test State Machine

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 related to time constraints: "period-of-time" and "recurrence" and one temporal parameter related to ordering: "ordered-by user". Time constraints parameters 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. "ordered-by user" parameter indicates that the user is responsible for the ordering on a collection of OAM unitary tests. "test-sequence-status" shows the state of the OAM test sequence. "state" imported from the "ietf-schedule" module indicates the current state of the schedule.

Note that repetition is specified by "execution-count" parameter and only applies to the recurrence schedule type. If no count is indicated, the test is considered to run indefinitely. In case of the recurrence schedule type, either frequency or interval should be specified. Each execution runs at the scheduled recurrence interval. Since the OAM test sequence model consists of a collection of OAM unitary tests, one or more tests in the sequence might get an error, however error in one or more tests doesn’t prevent the subsequent tests or remaining tests to execute. In addition, any change to the ordering of the OAM test sequence will lead to different reporting output results therefore the user should have full control on the ordering and "ordered-by user" parameters needs to be specified. If two or more tests are to run concurrently, they MUST be run in the order specified by the user.

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 ne-config* [ne-id]
          |  |  +--rw ne-id         rt-types:router-id
          |  |  +--rw managed?      boolean
          |  |  +--rw test-type?    identityref
          |  |  +--rw root
          +--rw (schedule-class)?
          |  +--:(period)
          |  |  +--rw period
          |  |     +--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
          |  +--:(recurrence)
          |     +--rw recurrence
          |        +--rw recurrence-description?   string
          |        +--rw frequency?                identityref
          |        +--rw interval?                 uint32
          |        +--rw execution-count?          uint32
          +--rw state?                    identityref
          +--rw version?                  uint16
          +--rw schedule-type?            identityref
          +--ro local-time?               yang:date-and-time
          +--ro last-update?              yang:date-and-time
          +--ro counter?                  yang:counter32
          +--ro last-occurrence?          yang:date-and-time
          +--ro upcoming-occurrence?      yang:date-and-time
          +--ro last-failed-occurrence?   yang:date-and-time
          +--ro failure-counter?          yang:counter32
          +--ro test-sequence-status?     identityref
```

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 by the management and hasn’t been applied to the network element.

* "configured": The state where the test is being configured. This state is triggered when the planned test configuration is applied to the network element.

* "ready": The state where the test is ready to be executed. This is state is triggered after the planned test configuration is applied and before the test is executed.

* "on-going": The state where the test is currently running. This state is triggered when the test has been executed but the test results haven’t been produced.

* "stop": The state where the test is manually stopped. This state is triggered when the test is manually interrupted.

* "success": The final state where all unitary tests are completed. This state is triggered when all tests have been conducted successfully.

* "failure": The state when one or more tests in the sequence got an error.

* "error": The state where an error occurs during the test. This state is triggered when one or more tests haven’t been conducted successfully. Implementations may report a more specific error cause using child identities such as "resource-contention" or "priority".

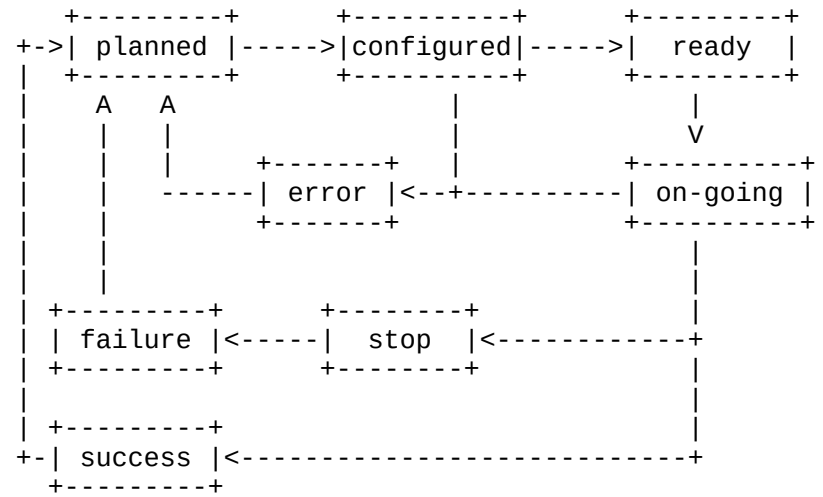

Figure 4: OAM test sequence state machine

4. YANG Data Models for Scheduling OAM Tests

4.1. YANG Model for Scheduling OAM Unitary Test

```
<CODE BEGINS>
 file ietf-oam-unitary-test@2026-01-13.yang
module ietf-oam-unitary-test {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-oam-unitary-test";
  prefix "oamut";
```

```
// 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";
}

import ietf-yang-schema-mount {
   prefix yangmnt;
   reference
     "RFC 8528: YANG Schema Mount";
}

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>
   Author:   Qin Wu
             <bill.wu@huawei.com>";
description
  "This module defines the 'ietf-oam-unitary-test' YANG model for
   activation of network diagnosis procedures.

  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.

       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 (RFC 2119) (RFC 8174) when, and only when,
   they appear in all capitals, as shown here.";

// 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 "2026-01-13" {
  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
}

/* Identities */
       identity unitary-test-status {
        description
        "Base identity for unitary-test-status.";
       }
       identity planned {
        base unitary-test-status;
        description
        "Identity for planned.";
       }
       identity configured {
        base unitary-test-status;
        description
        "Identity for configured.";
       }
       identity ready {
        base unitary-test-status;
        description
        "Identity for ready.";
       }
       identity on-going {
        base unitary-test-status;
        description
        "Identity for on-going.";
       }
       identity stop {
        base unitary-test-status;
        description
        "Identity for stop.";
       }
       identity success {
        base unitary-test-status;
        description
        "Identity for success.";
       }

       identity error {
        base unitary-test-status;
        description
        "Identity for error.";
       }

       identity resource-contention {
        base error;
        description
        "Identity for resource contention.";
       }

       identity priority {
        base error;
        description
        "Identity for priority.";
       }

identity basic-test-type {
     description
       "Base identity of basic test type.";
}

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.";
      }

       leaf managed {
         type boolean;
         default "true";
         description
           "True if the host can access oam unitary test
            using the root mount point.  This value
            may not be modifiable in all implementations.";
       }
      leaf test-type {
           type identityref {
             base basic-test-type;
           }
           description
             "Choose the type of test.";
         }
       container root {
         description
           "Container for mount point.";
         yangmnt:mount-point "root" {
           description
             "Root for models supported per oam unitary test.
                         This mount point may or may not be inline based on
              the server implementation.

              When the associated 'managed' leaf is 'false', any
              operation that attempts to access information below
              the root SHALL fail with an error-tag of
              'access-denied' and an error-app-tag of
              'oamut-not-managed'.";
         }
       }
    }
  }

  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:schedule-status;
      leaf unitary-test-status {
```

```
            type identityref {
                  base unitary-test-status;
            }
                config false;
          description
            "Status of the test.";
        }
    choice schedule-class {
       description
            "Choice based on the type of the time range.";
     container period {
           description
             "The OAM Test takes effect based on a precise period of
                  time.";
             uses schedule:period-of-time;
     }
     container recurrence {
           description
             "The OAM test takes effect based on a recurrence rule.";
           uses schedule:recurrence-basic;
      }
    }
   }
  }
  }
  <CODE ENDS>
```

4.2. YANG Model for OAM Test Sequence

```
  <CODE BEGINS>
   file ietf-oam-test-sequence@2026-01-13.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>
           Author:   Qin Wu
                    <bill.wu@huawei.com>";
    description
      "This module defines the 'oam-test-sequence-test' YANG model for
      management of network diagnosis procedures.

      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.";

// 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 "2026-01-13" {
  description "Initial version";
  reference "RFCXXXX";
  // Update with the correct RFC number when assigned
}

/* Identities */
       identity test-sequence-status {
        description
        "Base identity for test-sequence-status.";
       }
       identity planned {
        base test-sequence-status;
        description
        "Identity for planned.";
       }
       identity configured {
        base test-sequence-status;
        description
        "Identity for configured.";
       }
       identity ready {
        base test-sequence-status;
        description
        "Identity for ready.";
       }
       identity on-going {
        base test-sequence-status;
        description
        "Identity for on-going.";
       }
       identity stop {
        base test-sequence-status;
        description
        "Identity for stop.";
       }
       identity success {
        base test-sequence-status;
        description
        "Identity for success.";
       }
       identity failure {
        base test-sequence-status;
        description
        "Identity for failure";
       }
       identity error {
        base test-sequence-status;
        description
        "Identity for error.";
```

```
       }
       identity resource-contention {
        base error;
        description
        "Identity for resource-contention error cause.";
       }
       identity priority {
        base error;
        description
        "Identity for priority error cause.";
       }
/* 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";
      ordered-by user;
      description "References to the ietf-oam-unitary-tests.";

      uses "oamut:oam-unitary-test";
    }
 uses schedule:schedule-status;
       leaf test-sequence-status {
         type identityref {
               base test-sequence-status;
         }
         config false;
         description
               "Status of the test sequence execution.";
       }
choice schedule-class {
   description
        "Choice based on the type of the time range.";
 container period {
       description
         "The OAM Test takes effect based on a precise period of
              time.";
         uses schedule:period-of-time;
 }
 container recurrence {
       description
         "The OAM test takes effect based on a recurrence rule.";
       uses schedule:recurrence-basic;
       leaf execution-count {
         type uint32;
         description
           "If set, limits how many times the test sequence is
            executed for this recurrence. If the leaf is absent, there
            is no limit: executions follow the recurrence until the
            test is removed from the system.";
       }
    }
   }
```

```
      }
    }
  }
}
<CODE ENDS>
```

5. Using Device Model 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 recreating 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.

The "test-type" leaf and the schema mount are complementary. The "test-type" leaf (identityref to "basic-test-type") explicitly indicates which OAM test type, and thus which YANG module, is mounted at the "root" mount point for that "ne-config" list entry. Each "ne-config" entry therefore pairs a test-type identity with the corresponding mounted module configuration under "root", so that management systems and implementations know which OAM module applies to that node. This document defines the base identity "basic-test-type" and one child identity "twamp" for TWAMP; YANG modules that augment "ietf-oam-unitary-test" may define additional child identities derived from "basic-test-type" for other OAM test types.

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) use the unitary-test-status and test-sequence-status leaves to indicate the current scheduling state of each OAM task. These leaves are of type identityref, allowing extensible reporting. If a conflict is detected (e.g., two tests require exclusive access to the same resource at the same time), the server sets the status to error or to a more specific error-cause identity derived from error: resource-contention for resource conflicts, or priority for prioritization-related conflicts. This error-cause indication allows operators and management systems to distinguish the reason for the failure.

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 the conflict MUST be clearly reported via the YANG model status leaves.

When a new unitary-test or test-sequence are scheduled, the request for OAM tasks schedule MAY be rejected by the server depending on the server's capability to evaluate the scheduling impact and detect conflicts prior to execution, e.g., the number of schedule conflict exceeds the specific threshold.

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.

6.3. Performance impact of concurrent OAM task scheduling

Concurrent OAM tasks scheduling may cause performance strain on oam test devices due to intensive processing on both the server and the client. Management and orchestration systems need to make sure to have sufficient resource before conducting those multiple concurrent OAM tasks.

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 [RFC8446].

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.

With respect to scheduling, the security considerations in [I-D.ietf-netmod-schedule-yang] also apply.

8. IANA Considerations

8.1. Updates to the IETF XML Registry for New YANG Module

IANA is requested to register the following URI in the "ns" registry within the "IETF XML Registry" group [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-oam-unitary-test
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-oam-test-sequence
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

8.2. Updates to the YANG Module Names Registry for New YANG Module

IANA is requested to register the following YANG module in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group.

```
Name:       ietf-oam-unitary-test
Maintained by IANA? N
Namespace:  urn:ietf:params:xml:ns:yang:ietf-oam-unitary-test
Prefix:     as
Reference:  RFC XXXX

Name:       ietf-oam-test-sequence
Maintained by IANA? N
Namespace:  urn:ietf:params:xml:ns:yang:ietf-oam-test-sequence
Prefix:     as
Reference:  RFC XXXX
```

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-10, 7 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-schedule-yang-10>.

[I-D.ietf-opsawg-oam-characterization]
Pignataro, C., Farrel, A., and T. Mizrahi, "Guidelines for Characterizing the Term "OAM"", Work in Progress, Internet-Draft, draft-ietf-opsawg-oam-characterization-17, 28 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-oam-characterization-17>.

[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>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "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>.

[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>.

[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>.

[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>.

[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-08, 13 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-08>.

[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-23, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>.

[I-D.tt-netmod-yang-config-templates] Wills, R., Ma, Q., and D. Rajaram, "YANG Configuration Templates", Work in Progress, Internet-Draft, draft-tt-netmod-yang-config-templates-02, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-tt-netmod-yang-config-templates-02>.

[IEEE-8021ag] "IEEE Standard for Local and Metropolitan Area Networks Bridges and Bridged Networks Connectivity Fault Management", 2007, <https://standards.ieee.org/ieee/802.1ag/3597/>.

[IEEE-8021Q] "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", October 2012, <https://standards.ieee.org/ieee/802.1Q/6844/>.

[ITU-T-G81131]
"Operation and maintenance mechanism for T-MPLS layer networks", April 2007, <https://www.itu.int/rec/T-REC-G.8113.1-201611-I!Cor1>.

[ITU-T-Y1731]
"OAM Functions and Mechanisms for Ethernet-based Networks", 13 June 2023, <https://www.itu.int/rec/T-REC-Y.1731/en>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/rfc/rfc792>.

[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>.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <https://www.rfc-editor.org/rfc/rfc4379>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/rfc/rfc4443>.

[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>.

[RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <https://www.rfc-editor.org/rfc/rfc5085>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/rfc/rfc5880>.

[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>.

[RFC6371] Busi, I., Ed. and D. Allan, Ed., "Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks", RFC 6371, DOI 10.17487/RFC6371, September 2011, <https://www.rfc-editor.org/rfc/rfc6371>.

[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>.

[RFC7174] Salam, S., Senevirathne, T., Aldrin, S., and D. Eastlake 3rd, "Transparent Interconnection of Lots of Links (TRILL) Operations, Administration, and Maintenance (OAM) Framework", RFC 7174, DOI 10.17487/RFC7174, May 2014,

            <https://www.rfc-editor.org/rfc/rfc7174>.

[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>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple
            Two-Way Active Measurement Protocol", RFC 8762,
            DOI 10.17487/RFC8762, March 2020,
            <https://www.rfc-editor.org/rfc/rfc8762>.

[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>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi,
            Ed., "Data Fields for In Situ Operations, Administration,
            and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197,
            May 2022, <https://www.rfc-editor.org/rfc/rfc9197>.

[RFC9341] Fioccola, G., Ed., Cociglio, M., Mirsky, G., Mizrahi, T.,
            and T. Zhou, "Alternate-Marking Method", RFC 9341,
            DOI 10.17487/RFC9341, December 2022,
            <https://www.rfc-editor.org/rfc/rfc9341>.

[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. The following example uses the "twamp" identity defined in the ietf-oam-unitary-test module (derived from "basic-test-type") to indicate the test type; the TWAMP device model is mounted at the "root" of each "ne-config" entry. The 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",
      "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",
          "managed": "true",
          "test-type": "twamp",
          "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",
          "managed": "true",
          "test-type": "twamp",
          "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

A.2. Ping OAM Test Template

Ping OAM Test Template can be defined using YANG-based configuration template specified in [I-D.tt-netmod-yang-config-templates] as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<templates xmlns="urn:ietf:params:xml:ns:yang:ietf-config-template">
     <template>
       <id>oam-unitary-test-schedule</id>
       <content>
         <oam-unitary-tests xmlns="urn:example:oam-unitary-tests">
           <oam-unitary-test>
             <name>*ping</name>
             <ne-config>
                   <ne-id>eth*</ne-id>
             </ne-config>
             <period-start>2025-10-01T08:00:00Z</period-start>
              <frequency>hourly</frequency>
           </oam-unitary-test>
         </oam-unitary-tests>
       </content>
     </template>
</templates>
```

Figure 6: Example of OAM Test Template Definition

Template application is indicated using the "apply-templates"

metadata. For example, the following OAM unitary tests configuration may be provided with the container node "oam-unitary-tests" applying the template defined in Figure 6.

As described in [I-D.tt-netmod-yang-config-templates], a template node can be overriden by having its value changed, but it can’t be deleted.

As an example of overriding a node in a template, a client may configure physically present oam unitary tests "lsp-ping", "ip-ping" and "srmpls-ping" inheriting the template defined in Figure 6, but the "ne-id" value of "srmpls-ping" needs to be "203.0.113.4":

```
 <?xml version="1.0" encoding="utf-8"?>
 <oam-unitary-tests xmlns="urn:example:interface"
          xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-config-template"
          ct:apply-templates="oam-unitary-test-schedule">
          <oam-unitary-test>
            <name>lsp-ping</name>
                       <ne-config>
                          <ne-id>eth0</ne-id>
                              <managed>true</managed>
                              ...
                       </ne-config>
          </oam-unitary-test>
          <oam-unitary-test>
            <name>ip-ping</name>
          </oam-unitary-test>
          <oam-unitary-test>
            <name>srmpls-ping</name>
                              <ne-config>
                             <ne-id>203.0.113.4</ne-id>
                              ...
                       </ne-config>
          </oam-unitary-test>
  </oam-unitary-tests>
```

Figure 7: Example of Applying OAM Test Template

And the above OAM Unitary Tests configuration renders the following expanded configuration:

```
<?xml version="1.0" encoding="utf-8"?>
   <oam-unitary-tests xmlns="urn:example:interface"
            xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-config-template">
            <oam-unitary-test>
              <name>lsp-ping</name>
                          <ne-config>
                             <ne-id>eth0</ne-id>
                                 <managed>true</managed>
                                 ...
                          </ne-config>
                          <period-start>2025-10-01T08:00:00Z</period-start>
              <frequency>hourly</frequency>
            </oam-unitary-test>
            <oam-unitary-test>
              <name>ip-ping</name>
                           <ne-config>
                             <ne-id>eth1</ne-id>
                                 <managed>true</managed>
                                 ...
                          </ne-config>
                          <period-start>2025-10-01T08:00:00Z</period-start>
              <frequency>hourly</frequency>
            </oam-unitary-test>
            <oam-unitary-test>
```

```
            <name>srmpls-ping</name>
                              <ne-config>
                           <ne-id>203.0.113.4</ne-id>
                              ...
                        </ne-config>
          </oam-unitary-test>
   </oam-unitary-tests>
```


Acknowledgments

Thanks Joe Clark, Daniel King, Qiufang Ma for valuable review and comments.

Authors' Addresses

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

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

Qin Wu
Huawei
Email: bill.wu@huawei.com