Internet
Internet-Draft
Intended status: Standards Track
Expires: 8 July 2026

Y. Qu
Futurewei Technologies
A. Lindem
Arrcus, Inc.
E. Kinzie
D. Fedyk
LabN Consulting, L.L.C.
M. Blanchet
Viagenie

4 January 2026

YANG Data Model for Scheduled Attributes
draft-ietf-tvr-schedule-yang-06

Abstract

The YANG model in this document includes three modules, and can be used to manage network resources and topologies with scheduled attributes, such as predictable link loss and link connectivity as a function of time. The intent is to have this information be utilized by Time-Variant Routing systems.

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 July 2026.

Copyright Notice

Table of Contents

1. Overview 2
1.1. Requirements Language 3
2. Tree Diagrams 3
3. Design of the model 3
3.1. Schedule Definitions 3
3.2. TVR Node YANG Module 4
3.3. TVR Topology YANG Module 4
4. TVR YANG Trees 5
4.1. TVR Node YANG Tree 5
4.2. TVR Topology YANG Tree 6
5. TVR Schedule YANG Modules 8
5.1. TVR Schedule YANG Module 8
5.2. TVR Schedule Node Module 10
5.3. TVR Network Topology Module 14
6. Operational Considerations 18
7. Security Considerations 19
8. IANA Considerations 19
9. Acknowledgements 20
10. Normative References 20
11. Informative References 21
Appendix A. Example: Add a scheduled cost to OSPF interface . . 22
Appendix B. Example: TVR Schedule Node Configuration 23
Appendix C. Example: TVR Network Topology Configuration 25
Authors' Addresses 27

1. Overview

YANG [RFC7950] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g., ReST) and encodings other than XML (e.g., JSON) are being defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

There are use cases where changes to the routing topology are expected as part of network operations. In some network scenarios, it's possible to predict the times at which one router will be able to establish a link with another router. Links can be predictably

lost and re-established, and neighbors may change as a function of time. For examples of such networks and scenarios, please reference TVR (Time-Variant Routing) Use Cases [RFC9657].

The YANG model in this document can be used to manage network resources and topologies with scheduled attributes. There are three YANG modules in this document.

Module ietf-tvr-schedule.yang contains the schedule YANG definitions. Module ietf-tvr-topology.yang defines a network topology with time-variant availability. Module ietf-tvr-node.yang is to be used to manage scheduled attributes of a single node.

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

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] [RFC8174].

2. Tree Diagrams

This document uses the graphical representation of data models defined in [RFC8340].

3. Design of the model

3.1. Schedule Definitions

Module ietf-tvr-schedule.yang contains schedule definitions that can be used by other modules.

The grouping "tvr-schedule" consists of a list of schedules, and each schedule is either a single time period or recurring time periods. For each schedule instance, it utilizes the groupings "period-of-time" and "recurrence-utc" as defined in [I-D.ietf-netmod-schedule-yang].

Users can supply attributes or containers that augment the schedule. The "tvr-node" has a container "scheduled-attributes" that uses the "tvr-schedule".

When an attribute’s schedule ends, the "value-default" MAY be used, when present. Alternatively, values can remain as set until another schedule alters values.

The following figure provides an illustration of two attributes and their scheduled value changes. The attributes A1 and A2 take on different values at different times. The attribute A1 will take on the value v1 from the time t0 until t1, the value v2 from t1 until t2, and v3 from t2 until t3. The schedule of A1 may contain a list of 3 time periods with different values. The attribute A2 will take on the value vv1 from time t0 until t1, default value va2 from t1 until t2, and vv2 from t1 until t3. The schedule of A2 may contain a list of two time periods, and the default value is used between scheduled values.

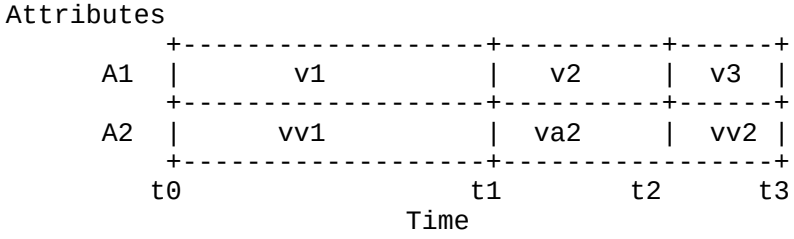

Figure 1: Time Varying Properties

3.2. TVR Node YANG Module

Module ietf-tvr-node.yang is a device model and designed to manage a single node with scheduled attributes.

Each node has scheduled power at the node level. At the interface level, a node has a list of interfaces, and each interface has its own schedule for availability/power up and down, scheduled bandwidth, and scheduled neighbors.

3.3. TVR Topology YANG Module

Module ietf-tvr-topology.yang describes a network topology with a time-variant availability schedule.

The module has a list of nodes, identified by a unique "node-id". Each node has a list of links. Links are modeled as unidirectional. Link availability is described from the viewpoint of a particular source node (the transmitting node) and beginning at a particular time. Each link in the list contains the range of times during which it is available.

The "source-link-id" is a string and used to identify a link as viewed from the source-node. Bandwidth and delay are predicted link attributes. Delay is the link propagation time and does not include any queuing delays. "destination-node" of a link may follow a schedule as well.

4. TVR YANG Trees

4.1. TVR Node YANG Tree

The following figure shows the tree diagram of the TVR Node scheduling.

```
module: ietf-tvr-node
  +--rw node-schedule
     +--rw node-id?               inet:uri
     +--rw node-power-schedule
     |  +--rw power-default?   boolean
     |  +--rw schedule* [schedule-id]
     |     +--rw schedule-id                     uint32
     |     +--rw (schedule-type)?
     |     |  +--:(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-first
     |     |     |  +--rw start-time-utc?   yang:date-and-time
     |     |     |  +--rw duration?         uint32
     |     |     +--rw (recurrence-end)?
     |     |     |  +--:(until)
     |     |     |  |  +--rw utc-until?          yang:date-and-time
     |     |     |  +--:(count)
     |     |     |     +--rw count?             uint32
     |     |     +--rw recurrence-description?   string
     |     |     +--rw frequency?                identityref
     |     |     +--rw interval?                 uint32
     |     +--rw power-state?                    boolean
     +--rw interface-schedule
        +--rw interface* [name]
           +--rw name                  union
           +--rw default-available?    boolean
           +--rw default-bandwidth?    yang:gauge64
```

```
            +--rw attribute-schedule
               +--rw schedule* [schedule-id]
                  +--rw schedule-id                      uint32
                  +--rw (schedule-type)?
                  |  +--:(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-first
                  |     |  +--rw start-time-utc?   yang:date-and-time
                  |     |  +--rw duration?         uint32
                  |     +--rw (recurrence-end)?
                  |     |  +--:(until)
                  |     |  |  +--rw utc-until?
                  |     |  |          yang:date-and-time
                  |     |  +--:(count)
                  |     |     +--rw count?              uint32
                  |     +--rw recurrence-description?   string
                  |     +--rw frequency?                identityref
                  |     +--rw interval?                 uint32
                  +--rw scheduled-attributes
                     +--rw available?   boolean
                     +--rw bandwidth?   yang:gauge64
                     +--rw neighbor?    inet:uri
```

4.2. TVR Topology YANG Tree

The following figure shows the tree diagram of the TVR Topology scheduling.

```
module: ietf-tvr-topology
  +--rw topology-schedule
     +--rw node* [node-id]
     |  +--rw node-id      inet:uri
     |  +--rw available
     |     +--rw default-node-available?   boolean
     |     +--rw schedule* [schedule-id]
     |        +--rw schedule-id                     uint32
     |        +--rw (schedule-type)?
```

```
        |        |  +--:(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-first
        |        |     |  +--rw start-time-utc?   yang:date-and-time
        |        |     |  +--rw duration?         uint32
        |        |     +--rw (recurrence-end)?
        |        |     |  +--:(until)
        |        |     |  |  +--rw utc-until?
        |        |     |  |          yang:date-and-time
        |        |     |  +--:(count)
        |        |     |     +--rw count?              uint32
        |        |     +--rw recurrence-description?   string
        |        |     +--rw frequency?                identityref
        |        |     +--rw interval?                 uint32
        |        +--rw node-available?                 boolean
        +--rw link* [source-node source-link-id]
           +--rw source-node       inet:uri
           +--rw source-link-id    string
           +--rw available
              +--rw schedule* [schedule-id]
              |  +--rw schedule-id                     uint32
              |  +--rw (schedule-type)?
              |  |  +--:(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-first
              |  |     |  +--rw start-time-utc?   yang:date-and-time
              |  |     |  +--rw duration?         uint32
              |  |     +--rw (recurrence-end)?
```

```
              |  |     |  +--:(until)
              |  |     |  |  +--rw utc-until?
              |  |     |  |          yang:date-and-time
              |  |     |  +--:(count)
              |  |     |     +--rw count?              uint32
              |  |     +--rw recurrence-description?   string
              |  |     +--rw frequency?                identityref
              |  |     +--rw interval?                 uint32
              |  +--rw link-attributes
              |     +--rw link-available?     boolean
              |     +--rw bandwidth?          yang:gauge64
              |     +--rw delay?              uint32
              |     +--rw destination-node?   inet:uri
              +--rw default-link-available?   boolean
              +--rw default-bandwidth?        yang:gauge64
              +--rw default-delay?            uint32
```

5. TVR Schedule YANG Modules

The following RFC is not referenced in the document text but is referenced in the "ietf-tvr-schedule.yang" module and "ietf-tvr-topology.yang" module: [RFC6991].

5.1. TVR Schedule YANG Module

<CODE BEGINS> file "ietf-tvr-schedule@2025-06-30.yang"

```
module ietf-tvr-schedule {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-schedule";
  prefix tvr-schd;

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

  organization
    "IETF TVR - Time Variant Routing Working Group";
  contact
    "WG Web:   <http://datatracker.ietf.org/wg/tvr>
     WG List:  <mailto:tvr@ietf.org>

     Author:   Yingzhen Qu
               <mailto:yingzhen.ietf@gmail.com>
     Author:   Acee Lindem
               <mailto:acee.ietf@gmail.com>
     Author:   Marc Blanchet
```

```
                <mailto:marc.blanchet@viagenie.ca>
        Author:   Eric Kinzie
                <mailto:ekinzie@labn.net>
        Author:   Don Fedyk
                <mailto:dfedyk@labn.net>";

    description
      "The YANG module contains common YANG definitions for
       time-variant schedule.

       This YANG model conforms to the Network Management
       Datastore Architecture (NMDA) as described in RFC 8342.

       Copyright (c) 2025 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.";

    reference
      "RFC XXXX: YANG Data Model for Scheduled Attributes";

    revision 2025-06-30 {
      description
        "Initial Version";
      reference
        "RFC XXXX: YANG Data Model for Scheduled Attributes.";
    }

    grouping tvr-schedule {
      list schedule {
        key schedule-id;
        leaf schedule-id {
          type uint32;
          description
            "Identifies the schedule.";
        }
        choice schedule-type {
          description
            "Choice of schedule type.";
```

```
            case period {
              description
                "A schedule with a single instance.";
              uses schedule:period-of-time;
            }
            case recurrence {
              description
                "A schedule with recurrence. The time is defined in UTC
                 format.";
              uses schedule:recurrence-utc;
            }
          }

          description
            "list of schedules.";
        }
        description
          "A common grouping definition of schedules.";
      }

    }
    <CODE ENDS>
```

5.2. TVR Schedule Node Module

```
  <CODE BEGINS> file "ietf-tvr-node@2025-06-30.yang"
  module ietf-tvr-node {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-node";
    prefix tvr-node;

    import ietf-inet-types {
      prefix inet;
      reference
        "RFC 6991: Common YANG Data Types";
    }

    import ietf-yang-types {
      prefix "yang";
      reference
        "RFC 6991: Common YANG Data Types";
    }

    import ietf-tvr-schedule {
      prefix "tvr-schd";
    }

    organization
```

```
      "IETF TVR - Time Variant Routing Working Group";
    contact
      "WG Web:   <http://datatracker.ietf.org/wg/tvr>
       WG List:  <mailto:tvr@ietf.org>

       Author:   Yingzhen Qu
                 <mailto:yingzhen.ietf@gmail.com>
       Author:   Acee Lindem
                 <mailto:acee.ietf@gmail.com>
       Author:   Marc Blanchet
                 <mailto:marc.blanchet@viagenie.ca>
       Author:   Eric Kinzie
                 <mailto:ekinzie@labn.net>
       Author:   Don Fedyk
                 <mailto:dfedyk@labn.net>";

    description
      "The YANG module is to configure and manage node attributes
       with schedules.

       This YANG model conforms to the Network Management
       Datastore Architecture (NMDA) as described in RFC 8342.

       Copyright (c) 2025 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.";

    reference
      "RFC XXXX: YANG Data Model for Scheduled Attributes";

    revision 2025-06-30 {
      description
        "Initial Version";
      reference
        "RFC XXXX: YANG Data Model for Scheduled Attributes.";
    }

    container node-schedule {
```

```
      description
        "This container defines a node's time variant attributes
         with schedules.";

      leaf node-id {
        type inet:uri;
        description
          "Identifier for a node, uniquely identifies a node.";
      }


      container node-power-schedule {
        description
          "Power schedule for the node. The node's power is
           represented by a boolean value with 'true' indicating
           the node is powered on and 'false' indicating the node
           is powered off.";

        leaf power-default {
          type boolean;
          default false;
          description
            "This indicates the default node power for the time
             periods when no specific power value is specified. If
             unspecified, the node is powered down by default.";
        }
        uses tvr-schd:tvr-schedule {
          augment "schedule" {
            description
              "Augment the power state within each period.";
            leaf power-state {
              type boolean;
              description
                "Indicates whether the node is powered on.";
            }
          }
        }
      }

      container interface-schedule {
        description
          "Container for TVR node interface attributes.";

        list interface {
          key "name";
          description
            "List of interface with schedules.";
          leaf name {
```

```
            type union {
              type yang:xpath1.0;
              type string;
            }
            description
              "Name of the interface.
               If used with the ietf-interfaces module, the xpath name
               is to identify the interface.";
          }
          leaf default-available {
            type boolean;
            default false;
            description
              "By default, the link is not available.";
          }
          leaf default-bandwidth {
            type yang:gauge64;
            units "bits/second";
            default "0";
            description
              "The default interface bandwidth in bits
               per second";
          }

          container attribute-schedule {
            description
              "Interface attributes with schedules.";

            uses tvr-schd:tvr-schedule {
              augment "schedule" {
                description
                  "Augment scheduled interface state.";
                container scheduled-attributes {
                  description
                    "Augmentation container for links.";
                  leaf available {
                    type boolean;
                    description
                      "Scheduled interface power state. This is to
                       work with the leaf 'enabled' for the configured
                       state of the interface.";
                  }
                  leaf bandwidth {
                    type yang:gauge64;
                    units "bits/second";
                    description
                      "The scheduled bandwidth in bits per second";
                  }
```

```
                  leaf neighbor {
                    type inet:uri;
                    description
                      "The node expected to be this interface's neighbor
                      during this time.";
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

5.3. TVR Network Topology Module

```
 <CODE BEGINS> file "ietf-tvr-topology@2026-01-04.yang"
 module ietf-tvr-topology {
   yang-version 1.1;
   namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-topology";
   prefix tvr-topo;

   import ietf-inet-types {
     prefix inet;
     reference
       "RFC 6991: Common YANG Data Types";
   }

   import ietf-yang-types {
     prefix "yang";
     reference
       "RFC 6991: Common YANG Data Types";
   }

   import ietf-tvr-schedule {
     prefix "tvr-schd";
   }

   organization
     "IETF Time-Variant Routing Working Group";

   contact
     "WG Web:   <https://datatracker.ietf.org/wg/tvr/>
      WG List:  <mailto:tvr@ietf.org>

      Author:   Eric Kinzie
```

```
                <mailto:ekinzie@labn.net>
       Author:  Don Fedyk
                <mailto:dfedyk@labn.net>
       Author:  Yingzhen Qu
                <mailto:yingzhen.ietf@gmail.com>
       Author:  Acee Lindem
                <mailto:acee.ietf@gmail.com>
       Author:  Marc Blanchet
                <mailto:marc.blanchet@viagenie.ca>";

   description
     "This YANG module contains YANG definitions for describing
     network topology with a time-variant availability schedule.

     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.";

   revision 2026-01-04 {
     description
       "Initial revision";
     reference
       "RFC XXXX: YANG Data Model for Scheduled Attributes";
   }

   container topology-schedule {
     description
       "Network topology schedules.";
     list node {
       key "node-id";
       description
         "List of nodes with schedules.";
       leaf node-id {
```

```
          type inet:uri;
          description
            "Identifier for a node; uniquely identifies a node. This
             may be the same as the node-id defined in the ietf-network
             module defined in RFC 8345.";
        }
        container available {
          description
            "The time at which this node becomes available.";

          leaf default-node-available {
            type boolean;
            default false;
            description
              "By default, the node is powered off.";
          }

          uses tvr-schd:tvr-schedule {
            augment "schedule" {
              description
                "Augment scheduled node availability.";
              leaf node-available {
                type boolean;
                description
                  "Node availability.";
              }
            }
          }
        }
      }

      list link {
        key "source-node source-link-id";
        description
          "List of links.";
        leaf source-node {
          type inet:uri;
          description
            "A name refers to the source node of the link.";
        }
        leaf source-link-id {
          type string;
          description
            "A name refers to the link of the source node.";
        }
        container available {
          description
            "The time at which this link becomes available.";
```

```
        uses tvr-schd:tvr-schedule {
          augment "schedule" {
            description
              "Augment scheduled values.";
            container link-attributes {
              description "Augmentation container for links.";
              leaf link-available {
                type boolean;
                description
                  "The predicted link availability.";
              }
              leaf bandwidth {
                type yang:gauge64;
                units "bits/second";
                description
                  "The predicted link capacity. If the value measured
                   by the system is less than this value, the system
                   value is used. If the value measured by the system
                   is greater than this value the predicted value
                   SHOULD be used.";
              }
              leaf delay {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "The predicted one-way delay or latency in
                  microseconds. If the value measured by the system is
                  less than this value the predicted value SHOULD be
                  used.";
              }
              leaf destination-node {
                type inet:uri;
                description
                  "A name refers to the destination node of the link.";
              }
            }
          }
        }
        leaf default-link-available {
          type boolean;
          default "false";
          description
            "The default link availability.  During times when the
             schedule does not specify an availability, this value
             is used.";
        }
        leaf default-bandwidth {
```

```
            type yang:gauge64;
            units "bits/second";
            default "0";
            description
              "The default link capacity specified in a
               generic format.";
          }
          leaf default-delay {
            type uint32 {
              range "0..16777215";
            }
            description
              "The default link capacity specified in a
               generic format.";
          }
        }
      }
    }
  }
 <CODE ENDS>
```

6. Operational Considerations

The TVR Schedule YANG modules provide a structured mechanism to represent time-variant attributes of network elements and topologies. The following aspects should be considered by operators deploying the modules.

Operators should be aware that these scheduled attributes represent planned state changes, not instantaneous configuration changes or unplanned breakages.

There should be synchronized clocks between controllers and devices. The time sources (e.g., NTP, PTP) should be accurate to avoid unintended time/schedule offsets. Inconsistent time bases can lead to network misbehavior, such as routing loops or network instability.

The modules are to support TVR use cases where topology and attributes vary as a function of time. How control plane and data plane are impacted by the schedules should be considered by the operators.

The TVR schedule modules are meant to work together with other models that manage node or interface attributes. Operators should consider the conflicting configurations between the scheduled and non-scheduled settings and how to handle the conflicts.

7. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in ietf-tvr-node.yang module and ietf-tvr-topology.yang 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. There are the subtrees and data nodes and their sensitivity/ vulnerability:

/node-schedule/node-power-schedule
/node-schedule/interface-schedule
/topology-schedule/nodes/available
/topology-schedule/links/available
Modifications to these scheduled attributes may result in a denial of service.

Some of the readable data nodes in the ietf-tvr-node.yang module and ietf-tvr-topolgy.yang module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

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

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

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
   name: ietf-tvr-schedule
   namespace: urn:ietf:params:xml:ns:yang:ietf-tvr-schedule
   prefix: tvr-schd
   reference: RFC XXXX

   name: ietf-tvr-node
   namespace: urn:ietf:params:xml:ns:yang:ietf-tvr-node
   prefix: tvr-node
   reference: RFC XXXX

   name: ietf-tvr-topology
   namespace: urn:ietf:params:xml:ns:yang:ietf-tvr-topology
   prefix: tvr-topo
   reference: RFC XXXX
```

9. Acknowledgements

The YANG model was developed using the suite of YANG tools written and maintained by numerous authors.

10. Normative References

[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/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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/info/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/info/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/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/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/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/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/info/rfc8446>.

[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-06, 29 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-schedule-yang-06>.

11. Informative References

[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/info/rfc8340>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC9657] Birrane, III, E., Kuhn, N., Qu, Y., Taylor, R., and L. Zhang, "Time-Variant Routing (TVR) Use Cases", RFC 9657, DOI 10.17487/RFC9657, October 2024, <https://www.rfc-editor.org/info/rfc9657>.

Appendix A. Example: Add a scheduled cost to OSPF interface

In OSPF (Open Shortest Path First), the interface cost is a metric used to determine the preference or desirability of a particular link or interface. By default, the OSPF interface cost is calculated based on the bandwidth of the interface, and it is also configurable.

This example demonstrates how an OSPF interface can be extended with a cost that changes with a schedule.

```
module ietf-tvr-ospf-schedule {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-ospf-schedule";
  prefix ospf-schedule;

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing
       Management (NMDA Version)";
  }
  import ietf-ospf {
    prefix "ospf";
    reference
      "RFC 9129: A YANG Data Model for OSPF Protocol";
  }

  import ietf-tvr-schedule {
    prefix "tvr-schd";
  }

  augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface" {
    container scheduled-cost {
      description
        "Augment OSPF interface with a scheduled interface cost.";
      uses tvr-schd:tvr-schedule {
        augment "schedule" {
          leaf cost {
            type uint32;
            description
             "interface cost";
          }
        }
      }
    }
  }
}
```

Appendix B. Example: TVR Schedule Node Configuration

The following is an XML example using the ietf-tvr-node YANG module.

In this example, the node has a node-id of "node:1", and it's scheduled to be powered off starting from 2025-07-26T17:00:00Z to 2025-07-26T18:00:00Z. Interface "eth 1" has a recurrent schedule to be "not available (off)" which starts from 2025-07-20T23:00:00Z, lasts for 7200 seconds (2 hours) each time for 10 times every 2 days.

Note: '\' line wrapping per [RFC8792].

```
<node-schedule xmlns="urn:ietf:params:xml:ns:yang:ietf-tvr-node">
  <node-id>node:1</node-id>
  <node-power-schedule>
    <power-default>true</power-default>
    <schedule>
      <schedule-id>1</schedule-id>
      <period-description>pwr off</period-description>
      <period-start>2025-07-26T17:00:00Z</period-start>
      <time-zone-identifier>UTC</time-zone-identifier>
      <period-end>2025-07-26T18:00:00Z</period-end>
      <power-state>false</power-state>
    </schedule>
  </node-power-schedule>
  <interface-schedule>
    <interface>
      <name>eth 1</name>
      <attribute-schedule>
        <schedule>
          <schedule-id>100</schedule-id>
          <recurrence-first>
            <start-time-utc>2025-07-20T23:00:00Z</start-time-utc>
            <duration>7200</duration>
          </recurrence-first>
          <count>10</count>
          <recurrence-description>daily off</recurrence-description>
          <frequency xmlns:schedule="urn:ietf:params:xml:ns:yang:\
            ietf-schedule">schedule:daily</frequency>
          <interval>2</interval>
          <scheduled-attributes>
            <available>false</available>
          </scheduled-attributes>
        </schedule>
      </attribute-schedule>
    </interface>
  </interface-schedule>
</node-schedule>
```

The following is the same example using JSON format.

```
{
    "node-schedule": {
        "-xmlns": "urn:ietf:params:xml:ns:yang:ietf-tvr-node",
        "node-id": "node:1",
        "node-power-schedule": {
            "power-default": "true",
            "schedule": {
                "schedule-id": "1",
                "period-description": "pwr off",
                "period-start": "2025-07-26T17:00:00Z",
                "time-zone-identifier": "UTC",
                "period-end": "2025-07-26T18:00:00Z",
                "power-state": "false"
            }
        },
        "interface-schedule": {
            "interface": {
                "name": "eth 1",
                "attribute-schedule": {
                    "schedule": {
                        "schedule-id": "100",
                        "recurrence-first": {
                            "start-time-utc": "2025-07-20T23:00:00Z",
                            "duration": "7200"
                        },
                        "count": "10",
                        "recurrence-description": "daily off",
                        "frequency": "ietf-schedule:daily",
                        "interval": "2",
                        "scheduled-attributes": {
                            "available": "false"
                        }
                    }
                }
            }
        }
    }
}
```

Appendix C. Example: TVR Network Topology Configuration

The following is an XML example using the ietf-tvr-topology YANG module.

In this example, the node with a node-id of "node:1" is scheduled to be powered off start from 2025-07-26T17:00:00Z, for 2 days and four and half hours.

Node "node:2" is scheduled to be off starting from
2025-07-20T23:00:00Z, lasting for 7200 seconds (2 hours) every two
days until 2025-08-20T23:00:00Z.

Note: '\' line wrapping per [RFC8792].

```
<topology-schedule xmlns="urn:ietf:params:xml:ns:yang\
  :ietf-tvr-topology">
  <node>
    <node-id>node:1</node-id>
    <available>
      <default-node-available>true</default-node-available>
      <schedule>
        <schedule-id>101</schedule-id>
        <period-description>node1 schedule</period-description>
        <period-start>2025-07-26T17:00:00Z</period-start>
        <time-zone-identifier>UTC</time-zone-identifier>
        <duration>P2DT04:30:00</duration>
        <node-available>false</node-available>
      </schedule>
    </available>
  </node>
  <node>
    <node-id>node:2</node-id>
    <available>
      <default-node-available>true</default-node-available>
      <schedule>
        <schedule-id>201</schedule-id>
        <recurrence-first>
          <start-time-utc>2025-07-20T23:00:00Z</start-time-utc>
          <duration>7200</duration>
        </recurrence-first>
        <utc-until>2025-08-20T23:00:00Z</utc-until>
        <recurrence-description>daily off</recurrence-description>
        <frequency xmlns:schedule="urn:ietf:params:xml:ns:yang:\
          ietf-schedule">schedule:daily</frequency>
        <interval>2</interval>
        <node-available>false</node-available>
      </schedule>
    </available>
  </node>
</topology-schedule>
```

The following is the same example using JSON format.

```
{
    "topology-schedule": {
        "-xmlns": "urn:ietf:params:xml:ns:yang:ietf-tvr-topology",
        "node": [
            {
                "node-id": "node:1",
                "available": {
                    "default-node-available": "true",
                    "schedule": {
                        "schedule-id": "101",
                        "period-description": "node1 schedule",
                        "period-start": "2025-07-26T17:00:00Z",
                        "time-zone-identifier": "UTC",
                        "duration": "P2DT04:30:00",
                        "node-available": "false"
                    }
                }
            },
            {
                "node-id": "node:2",
                "available": {
                    "default-node-available": "true",
                    "schedule": {
                        "schedule-id": "201",
                        "recurrence-first": {
                            "start-time-utc": "2025-07-20T23:00:00Z",
                            "duration": "7200"
                        },
                        "utc-until": "2025-08-20T23:00:00Z",
                        "recurrence-description": "daily off",
                        "frequency": "ietf-schedule:daily",
                        "interval": "2",
                        "node-available": "false"
                    }
                }
            }
        ]
    }
}
```

Authors' Addresses

Yingzhen Qu
Futurewei Technologies
Email: yingzhen.ietf@gmail.com

Acee Lindem
Arrcus, Inc.
Email: acee.ietf@gmail.com

Eric Kinzie
LabN Consulting, L.L.C.
Email: ekinzie@labn.net

Don Fedyk
LabN Consulting, L.L.C.
Email: dfedyk@labn.net

Marc Blanchet
Viagenie
Email: marc.blanchet@viagenie.ca