~~Internet~~ TVR
Y. Qu
Internet-Draft Futurewei Technologies
Intended status: Standards Track A. Lindem
Expires: 7 January 2025 E. Kinzie
D. Fedyk
LabN Consulting, L.L.C.
M. Blanchet
Viagenie
6 July 2024

A YANG Data Model for Scheduled Attributes in TVR (Time-Variant Routing)

draft-ietf-tvr-schedule-yang-01

Abstract

The YANG model in this document includes three modules~~, and~~ that 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 t~~This information is meant to be utilized by Time-Variant Routing (TVR) systems.

described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. 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 a variety of network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. 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~~were defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

In some network scenarios, ~~it's~~ it is 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~~the reader may refer to TVR (Time-Variant Routing) Use Cases [I-D.ietf-tvr-use-cases].

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~~.~~:

The ~~Module~~ module "ietf-tvr-schedule~~.yang~~" contains ~~the~~ a set of 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.

Commenté [BMI1]: YANG can be used Independent of NETCONF.

Commenté [BMI2]: This was defined since many years now.

Commenté [BMI3]: Not sure this is needed.

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].

Commenté [BMI4]: Please use the latest boilerplate.

2. Tree Diagrams

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

3. Design of the Mmodel

3.1. Schedule Definitions

~~Module~~ The YANG module "ietf-tvr-schedule~~.yang~~" contains a set of reusable schedule definitions ~~that can be used by other modules~~.

The grouping "tvr-schedule" consists of a list of schedules with ~~, 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].

The container "attr-value" is a place holder for a module that uses the "tvr-schedule", where attribute values can be augmented.

When an attribute's schedule ends, the "value-default" SHOULD be used, if present.

Commenté [BMI5]: There is not such attribute. Do you mean use one of the "default-*" (e.g., default-bandwidth)

~~The following figure~~Figure 1 ~~provides an illustration~~ illustrates ~~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 attribute A2 will take on the value vv1 from time t0 until t1 and vv2 from v1 until v3.

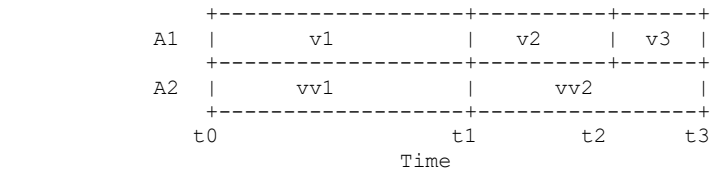

Figure 1: Example of Time Varying Properties

3.2. TVR Node YANG Module

The YANG ~~Module~~ module "ietf-tvr-node~~.yang~~" is a device model ~~and designed~~that is used 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, as well as scheduled bandwidth.

3.3. TVR Topology YANG Module

The ~~Module~~ YANG 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.

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?                yang:dotted-quad
     +--rw node-power-schedule
     |  +--rw power-default?   boolean
     |  +--rw schedule~~s~~* [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 utc-start-time?   yang:date-and-time
     |     |     |  +--rw duration?         uint32
     |     |     +--rw (recurrence-bound)?
     |     |        +--:(until)
```

Commenté [BMI6]: I would merge this section with the previous to help linking the narrative text with the structu

Commenté [BMI7]: Per the YANG guidance, list are singular.

```
|      |      |  |  +--rw utc-until?          yang:date-and-time
|      |      |  +--:(count)
|      |      |     +--rw count?              uint32
|      |      +--rw recurrence-description?   string
|      |      +--rw frequency                 identityref
|      |      +--rw interval?                 uint32
|      +--rw attr-value
|         +--rw power-state?   boolean
+--rw interface-schedule
   +--rw interfaces* [name]
      +--rw name                    union
      +--rw default-available?      boolean
      +--rw default-bandwidth?      yang:gauge64
      +--rw attribute-schedule
         +--rw schedules* [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 utc-start-time?   yang:date-and-time
            |     |  +--rw duration?         uint32
            |     +--rw (recurrence-bound)?
            |     |  +--:(until)
            |     |  |  +--rw utc-until?          yang:date-and-time
            |     |  +--:(count)
            |     |     +--rw count?              uint32
            |     +--rw recurrence-description?   string
            |     +--rw frequency                 identityref
            |     +--rw interval?                 uint32
            +--rw attr-value
               +--rw available?   boolean
               +--rw bandwidth?   yang:gauge64
```

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 nodes* [node-id]
     |  +--rw node-id      inet:uri
     |  +--rw available
     |     +--rw default-node-available?   boolean
     |     +--rw schedulesschedulesp* [schedule-id]
     |        +--rw schedule-id                       uint32
     |        +--rw (schedule-type)?
     |        |  +--:(period)
     |        |  |  +--rw period-description?        string
```

Commenté [BMI8]: Consider adding a sentence to basically say these are described in the common module, not reiterated here.

Commenté [BMI9]: Idem as previous comment

Commenté [BMI10]: Why is this different from the nam format in rfc8343?

Commenté [BMI11]: Weird that a default value is a gau

Commenté [BMI12]: Is there a case where the same schedule/value will be use for many interfaces of the sam node?

If so, I would add a provision for compact configuration by allowing a schedule to define once then referenced by ea relevant interface.

Commenté [BMI13]: Per the YANG authors guidance.

Commenté [BMI14]: Please check this is relevant.

Commenté [BMI15]: You may clarify why 8543 structu not reused here.

Commenté [BMI16]: You may consider if it makes sens include a provision for schedule profiles to factorize schedules that are shared between several nodes/interfa for the sake of compact files.

Commenté [BMI17]: Same as for other lists

Commenté [BMI18]: I know this the type used in RFC8345, but this type is not consistent with what you us in the previous module (yang:dotted-quad).

Is that intentional?

Commenté [BMI19]: Per the YANG naming convention

```
|        |  |  +--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 utc-start-time?   yang:date-and-time
|        |     |  +--rw duration?         uint32
|        |     +--rw (recurrence-bound)?
|        |     |  +--:(until)
|        |     |  |  +--rw utc-until?          yang:date-and-time
|        |     |  +--:(count)
|        |     |     +--rw count?             uint32
|        |     +--rw recurrence-description?   string
|        |     +--rw frequency                 identityref
|        |     +--rw interval?                 uint32
|        +--rw attr-value
|           +--rw node-available?   boolean
+--rw ~~links~~link* [source-node source-link-id]
   +--rw source-node         inet:uri
   +--rw destination-node?   inet:uri
   +--rw source-link-id      string
   +--rw available
      +--rw ~~schedule~~schedules* [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 utc-start-time?   yang:date-and-time
      |  |     |  +--rw duration?         uint32
      |  |     +--rw (recurrence-bound)?
      |  |     |  +--:(until)
      |  |     |  |  +--rw utc-until?          yang:date-and-time
      |  |     |  +--:(count)
      |  |     |     +--rw count?             uint32
      |  |     +--rw recurrence-description?   string
      |  |     +--rw frequency                 identityref
      |  |     +--rw interval?                 uint32
      |  +--rw attr-value
      |     +--rw link-available?   boolean
      |     +--rw bandwidth?        yang:gauge64
      |     +--rw delay?            uint32
      +--rw default-link-available?   boolean
      +--rw default-bandwidth?        yang:gauge64
      +--rw default-delay?            uint32
```

Commenté [BMI20]: Link-id are defined as uri in RFC85 fwiw.

Commenté [BMI21]: Weird for a default

5. TVR Schedule YANG Modules

5.1. TVR Schedule YANG Module

This modules uses groupings defined in [I-D.ietf-netmod-schedule-yang].

```
<CODE BEGINS> file "ietf-tvr-schedule@2024-07-05.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 XXXXCCCC: 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) 2024 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

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

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

  reference
```

Commenté [BMI22]: Please add a note to the RFC edit that this should be replaced with RFC number to be assig to I-D.ietf-netmod-schedule-yang

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

    revision 2024-07-05 {
      description
        "Initial Version";
      reference
        "RFC XXXX: YANG Data Model for Scheduled Attributes.";
    }

    grouping tvr-schedule {
      list schedules 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;
          }
        }

        container attr-value {
          description
           "Attribute value(s). This container should be augmented
            with attributes that apply to the current interval.";
        }
        description
          "List of schedules.";
      }
      description
        "A common grouping definition of TVR schedules.";
    }
  }
  <CODE ENDS>
```

Commenté [BMI23]: Indicate the unicity scope

5.2. TVR Schedule Node Module

This module uses types defined in [RFC6991].

```
  <CODE BEGINS> file "ietf-tvr-node@2024-07-05.yang"
  module ietf-tvr-node {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-node";
    prefix tvr-node;

    import ietf-yang-types {
      prefix "yang";
```

```
  reference
    "RFC 6991: Common YANG Data Types";
}

import ietf-tvr-schedule {
  prefix "tvr-schd";
  reference
    "RFC XXXX: YANG Data Model for Scheduled Attributes.";

}

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) 2024 IETF Trust and the persons identified as
   authors of the code.  All rights reserved.
   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject to
   the license terms contained in, the Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

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

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

revision 2024-07-05 {
  description
    "Initial Version";
  reference
    "RFC XXXX: YANG Data Model for Scheduled Attributes.";
}
```

Commenté [BMI24]: As the module can be accessed o
of the RFC

Commenté [BMI25]: C(configure) is already covered by
"managed' (FCAPS)

a mis en forme : Anglais (États-Unis)

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

      leaf node-id {
        type yang:dotted-quad;
        description
          "A 32-bit number used to identify 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 "schedules/attr-value" {
            description
              "Augments the power state within each period.";
            leaf power-state {
              type boolean;
              description
                "Indicates whether the node is powered on.";
            }
          }
        }
      }

      container interface-schedule {
        description
          "Container forSpecifies scheduled-related attributes of -TVR
node interface attributes.";

        list interfaces interface {
          key "name";
          description
            "List of interfaces 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
```

Commenté [BMI26]: Why this not a uri as in the next module or 8543?

Commenté [BMI27]: Please call out the unicity scope.

Commenté [BMI28]: Redundant with the default statement

Commenté [BMI29]: You may explain why this is not aligned with rfc8343

```
                is to identify the interface.";
            }
            leaf default-available {
              type boolean;
              default false;
              description
                "By default, the link is not availableIndicates the
default value of the link availability.";
            }
            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 "schedules/attr-value" {
                  description
                    "Augments scheduled interface state.";
                  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 secondof the
interface";
                  }
                }
              }
            }
          }
        }
      }
    }
    <CODE ENDS>
```

Commenté [BMI30]: Consider adding a reference statement

Commenté [BMI31]: I'm not sure this makes sense for default value

Commenté [BMI32]: Redundant with the unit stateme

5.3. TVR Network Topology Module

This module uses types defined in [RFC6991].

```
<CODE BEGINS> file "ietf-tvr-topology@2024-07-05.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";
    reference
      "RFC XXXX: YANG Data Model for Scheduled Attributes";

  }

  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 defines a contains YANG definitions for describing
    network topology with an time-variant availability schedules.

    Copyright (c) 2024 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

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

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

    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 2024-07-05 {
    description
      "Initial revision";
    reference
      "RFC XXXX: YANG Data Model for Scheduled Attributes";
  }

  container topology-schedule {
    description
      "Network topology schedules.";
    list nodes {
      key "node-id";
      description
        "List of nodes with schedules.";
      leaf node-id {
        type inet:uri;
        description
          "Identifier for a node, uniquely identifies a node within a
topology. 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 offSpecifies the default
value of node availability.";
        }

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

    list links link {
      key "source-node source-link-id";
      description
        "List of links.";
      leaf source-node {
        type inet:uri;
        description
```

Commenté [BMI33]: I assume that there is always one single topo. Right?

Commenté [BMI34]: You may elaborate, e.g., whether that is for correlation between a network topology and schedule topology

Commenté [BMI35]: You may recall the directionality assumption of modelling links.

```
          "Specifies a name that identifies A name refers to the source
node of the link.";
      }
      leaf destination-node {
        type inet:uri;
        description
          " Specifies a name that identifies A name refers to the
destination node of the link.";
      }
      leaf source-link-id {
        type string;
        description
          " Specifies a name that identifies A name refers to the link of
the source node.";
      }
      container available {
        description
          "The time at which this a link becomes (un)available.";
        uses tvr-schd:tvr-schedule {
          augment "schedules/attr-value" {
            description
              "Augment scheduled values.";
            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 default-link-available {
          type boolean;
          default "false";
          description
            "The default link availibility. During times when the
schedule
             does not specifiy an availability, this value is used.";
        }
        leaf default-bandwidth {
```

Commenté [BMI36]: This is distinct from the design in 8543

Commenté [BMI37]: Where the predicted value is defined?

```
          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. 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 Network Configuration Access Control Model (NACM) [RFC8341] ~~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.

Commenté [BMI38]: You should call these out.

7. 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
maintained by IANA? N
reference: RFC XXXX

name: ietf-tvr-node
namespace: urn:ietf:params:xml:ns:yang:ietf-tvr-node
prefix: tvr-node
maintained by IANA? N
reference: RFC XXXX

name: ietf-tvr-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-tvr-topology
prefix: tvr-topo
maintained by IANA? N
reference: RFC XXXX

8. Acknowledgements

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

9. 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-02, 25 June 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-schedule-yang-02>.

10. 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>.

[I-D.ietf-tvr-use-cases] Birrane, E. J., Kuhn, N., Qu, Y., Taylor, R., and L.

Zhang, "TVR (Time-Variant Routing) Use Cases", Work in
Progress, Internet-Draft, draft-ietf-tvr-use-cases-09, 29
February 2024, <https://datatracker.ietf.org/doc/html/
draft-ietf-tvr-use-cases-09>.

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~~ exemplifies how an OSPF interface can be extended with
a cost that changes with a schedule.

```
module ietfexample-tvr-ospf-schedule {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-ospf-schedule";
  prefix ex-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 "schedules/attr-value" {
          leaf cost {
            type uint32;
            description
             "interface cost";
          }
        }
      }
    }
  }
}
```

Authors' Addresses

Commenté [BMI39]: Please refer to 8407

Commenté [BMI40]: Please refer to 8407bis

Yingzhen Qu
Futurewei Technologies
United States of America
Email: yingzhen.ietf@gmail.com

Acee Lindem
LabN Consulting, L.L.C.
301 Midenhall Way
Cary, NC 27513
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