Workgroup: Internet
Published: 7 July 2023
Intended Status: Standards Track
Expires: 8 January 2024
Authors: Y. Qu (Futurewei)
A. Lindem (LabN Consulting LLC)
M. Blanchet (Viagenie)

YANG Model for Scheduled Attributes

Abstract

The YANG model in this document specifies a recurring schedule for changing the attributes of resources that are defined in other YANG data models. An example use case augmentation is also included, i.e., using the schedule to change the OSPF interface metric.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2024.

Table of Contents

1. Overview
   1.1. Requirements Language
2. Tree Diagrams
3. Design of the model
   3.1. Schedule Definition
   3.2. Model Augmentations
4. TVR Schedule YANG Trees
5. TVR Schedule YANG Module
6. Security Considerations
7. IANA Considerations
8. Acknowledgements
9. Normative References
10. Informative References

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

Appendix B. Example: LNE scheduled power configuration

Authors' Addresses

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.

The YANG model in this document specifies a recurring schedule for changing the attributes of resources that are defined in other YANG YANG data models.

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 Definition

The grouping "schedule" is defined in this model, which can be used to augment other models with time variant attributes.

The grouping "schedule-lifetime" is to define the life time of the "schedule". It has a "schedule-start-time" using "date-and-time", and the "schedule-end-time" can be an option of infinite, duration or date time. The "schedule-lifetime" can also be set to "always", which means the schedule is always valid and will be effective right away after the configuration. If the "schedule-start-time" is set to a time in the past when a configuration is done, the schedule is effective right away.

The leaf "recurrence" specifies the repetition pattern of the "base-schedule", such as daily or weekly. The "base-schedule", which is the basic unit of this schedule, consists a list of "intervals". These intervals specify the attribute values during the "base-schedule". The "recurrence" defines the duration of each "base-schedule", so the configured "intervals" should be within this duration. For example, if the "recurrence" is configured as daily, the last "end-time" in the list should not be larger than 24 hours, which is 8640000 in timeticks. Also each "end-time" must be greater than the "start-time" in the same interval. The "timeticks" defined in [RFC6991] is used to specify the relative "start-time" and "end-time" in each recurrence.

The "value" leaf is defined as a union, allowing it to be a simple boolean value, an integer represented as a string, or a value function expressed as a string. For instance, the string can represent a value that dynamically changes over time using a function such as "2*t+10". The "value-default" leaf defines the attribute's value when it's not covered by the intervals.

When a schedule starts in the middle of an recurrence, for example, the recurrence is set to "daily", and the "schedule-start-time" is 8:00AM, when the schedule starts, the "value" should be the 8:00AM value defined by the "base-schedule".

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

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 attribute A2 will take on the value vv1 from time t0 until t1 and vv2 from v1 until v3.

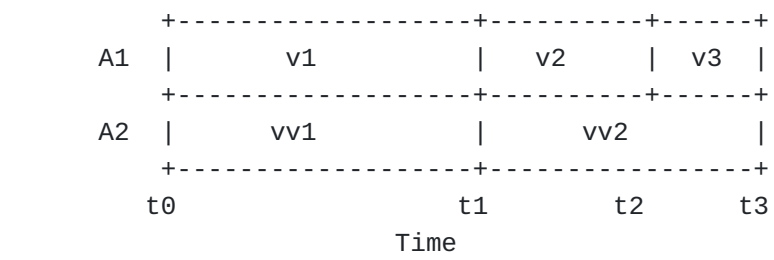


Figure 1: Time Varying Properties

The following is the tree diagram of the grouping "schedule".

```
 +--rw (schedule-lifetime)?
 |  +--:(always)
 |  |  +--rw always?                empty
 |  +--:(start-end-time)
 |     +--rw start-date-time?       yang:date-and-time
 |     +--rw (end-time)?
 |        +--:(infinite)
 |        |  +--rw no-end-time?     empty
 |        +--:(duration)
 |        |  +--rw duration?        uint32
 |        +--:(end-date-time)
 |           +--rw end-date-time?   yang:date-and-time
 +--rw recurrence?                  recurrence-type
 +--rw value-default?               union
 +--rw base-schedule
    +--rw intervals* [start-time]
       +--rw start-time    yang:timeticks
       +--rw end-time?     yang:timeticks
       +--rw value?        union
```

Figure 2: Tree diagram of the schedule

3.2. Model Augmentations

Grouping "schedule" can be used to augment other YANG data models with an attribute that changes its value based on a predefined schedule.

In this document, the IETF logical network element model [RFC8530] is augmented with a scheduled power shutdown using the grouping "schedule".

The interfaces defined in the ietf-interface model [RFC8343] is augmented with a scheduled interface up and down.

More use cases can be found in [I-D.ietf-tvr-use-cases].

4. TVR Schedule YANG Trees

The following figure shows the tree diagram of the interface augmentation and LNE augmentation.

```
module: ietf-tvr-schedule

  augment /if:interfaces/if:interface:
    +--rw scheduled-up-down
       +--rw (schedule-lifetime)?
       |  +--:(always)
       |  |  +--rw always?                empty
       |  +--:(start-end-time)
       |     +--rw start-date-time?       yang:date-and-time
       |     +--rw (end-time)?
       |        +--:(infinite)
       |        |  +--rw no-end-time?     empty
       |        +--:(duration)
       |        |  +--rw duration?        uint32
       |        +--:(end-date-time)
       |           +--rw end-date-time?   yang:date-and-time
       +--rw recurrence?                  recurrence-type
       +--rw value-default?               union
       +--rw base-schedule
          +--rw intervals* [start-time]
             +--rw start-time    yang:timeticks
             +--rw end-time?     yang:timeticks
             +--rw value?        union
  augment /lne:logical-network-elements/lne:logical-network-element:
    +--rw scheduled-power
       +--rw (schedule-lifetime)?
       |  +--:(always)
       |  |  +--rw always?                empty
       |  +--:(start-end-time)
       |     +--rw start-date-time?       yang:date-and-time
       |     +--rw (end-time)?
       |        +--:(infinite)
       |        |  +--rw no-end-time?     empty
       |        +--:(duration)
       |        |  +--rw duration?        uint32
       |        +--:(end-date-time)
       |           +--rw end-date-time?   yang:date-and-time
       +--rw recurrence?                  recurrence-type
       +--rw value-default?               union
       +--rw base-schedule
          +--rw intervals* [start-time]
             +--rw start-time    yang:timeticks
             +--rw end-time?     yang:timeticks
             +--rw value?        union
```

5. TVR Schedule YANG Module

The following RFCs are referenced in the "ietf-tvr-schedule" YANG module: [RFC6991], [RFC8343] and [RFC8530].

```
<CODE BEGINS> file "ietf-tvr-schedule@2023-07-05.yang"
module ietf-tvr-schedule {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tvr-schedule";
  prefix tvr-schedule;

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

  import ietf-interfaces {
    prefix "if";
    reference
      "RFC 8343: A YANG Data Model for Interface
               Management (NMDA Version)";
  }

  import ietf-logical-network-element {
    prefix "lne";
    reference
      "RFC 8530: YANG Model for Logical Network Elements";
  }

  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>";

  description
    "The YANG module defines a schedule for changing
     attributes. It can be used to augment existing models.

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

     Copyright (c) 2023 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.";
reference
  "RFC XXXX: YANG Data Model for Scheduled Attributes";

revision 2023-07-05 {
  description
    "Initial Version";
  reference
    "RFC XXXX: YANG Data Model for Scheduled Attributes.";
}

typedef recurrence-type {
  type enumeration {
    enum once {
      description
         "Repeat the scheduled action or attribute modificaton
         only once";
    }
    enum hourly {
      description
        "Repeat the scheduled action or attribute modificaiton
        once per hour.";
    }
    enum daily {
      description
        "Repeat the scheduled action or attribute modificaiton
         once per day.";
    }
    enum weekly {
      description
        "Repeat the scheduled action or attribute modificaiton
         once per week.";
    }
    enum monthly {
      description
        "Repeat the scheduled action or attribute modificaiton
```

```
          once per week.";
      }
      enum yearly {
        description
          "Repeat the scheduled action or attribute modificaiton
           once per year.";
      }
    }
    description
      "Types of recurrence";
  }

  grouping schedule-lifetime {
    description
      "Schedule lifetime specification.";
    choice schedule-lifetime {
      default "always";
      description
        "Options for specifying schedule lifetimes";
      case always {
        leaf always {
          type empty;
          description
            "Indicates schedule is always valid, and the schedule
             will be effective right away after the configuration.";
        }
      }
      case start-end-time {
        leaf start-date-time {
          type yang:date-and-time;
          description
            "Start time of the schedule.";
        }
        choice end-time {
          default "infinite";
          description
            "End-time setting.";
          case infinite {
            leaf no-end-time {
              type empty;
              description
                "Indicates schedule lifetime end-time is infinite.";
            }
          }
          case duration {
            leaf duration {
              type uint32 {
                range "1..2147483646";
              }
```

```
            units "seconds";
            description
              "Schedule lifetime duration, in seconds";
          }
        }
        case end-date-time {
          leaf end-date-time {
            type yang:date-and-time;
            description
              "End time.";
          }
        }
      }
    }
  }
}

grouping schedule {
  uses schedule-lifetime;
  leaf recurrence {
    type recurrence-type;
    must "(current() != 'once') or ../start-date-time" {
      error-message "A single occurrence requires a start-time";
      description
        "With single recurrence of a schedule, a start-time
         is required.";
     }
    description
      "Type of recurrence";
  }

  leaf value-default {
    type union {
      type boolean;
      type string;
    }
    description
      "Attribute default value. This value is used when the value
       is not covered by the intervals, as well as when the
       schedule ends.";
  }

  container base-schedule {
    list intervals {
      key "start-time";
      leaf start-time {
        type yang:timeticks;
        must "((../../../recurrence != 'hourly') or "
          +      "(current() < 360000)) and "
```

```
    +  "((../../../recurrence != 'daily') or "
    +      "(current() < 8640000)) and "
    +  "((../../../recurrence != 'monthly') or "
    +      "(current() < 267840000)) and "
    +   "((../../../recurrence != 'yearly') or "
    +      "(current() < 3162240000))"  {
    error-message
      "The start-time must not exceed the recurrence
       interval";
    description
      "The start-time must not exceed the recurrrence
       interval. For example, for a 'monthy' recurrence, the
       start-time must not exceed 31 days.";
  }
  description
    "Start time of the scheduled value within one recurrence.
     The calculation of the real start time is to use the
     basestarting time defined by schedule-lifetime, recurrence,
     and the timeticks. For example, if a schedule starts at
     2001/1/1 8:00AM with an hourly recurrence, and the
     timeticks is 180,000 (30 mins), the start-time will be
     every 30 mins pass an hour after 2001/1/1.";
}
leaf end-time {
  type yang:timeticks;
  must "(current() > ../start-time) and "
    +  "((../../../recurrence != 'hourly') or "
    +      "(current() < 360000)) and "
    +  "((../../../recurrence != 'daily') or "
    +      "(current() < 8640000)) and "
    +  "((../../../recurrence != 'monthly') or "
    +      "(current() < 267840000)) and "
    +   "((../../../recurrence != 'yearly') or "
    +      "(current() < 3162240000))"  {
    error-message
      "The end-time must be greater than the end time and
       must not exceed the recurrrence interval.";
    description
      "The end-time must be greater than the end time and
       must not exceed the recurrrence interval. For example,
       for a 'daily' recurrence, the time-tick offset must
       not exceed 24 hours.";
  }
  description
    "End time of the scheduled change. The calculation of the
     end-time is the same as the start-time.";
}
leaf value {
  type union {
```

```
          type boolean;
          type string;
        }
        description
          "Value for the scheduled attribute during the
           specified time period.
           When type string is used, it can be a simple value such
           as an int, or a string representing a value that changes
           its value based on a function.";
      }
      description
        "List of intervals for action or attribute modification.
         The intervals should be contained within each recurrence.
         For example, when the recurrence is set to daily, the list
         of intervals should be within this daily range, e.g., the
         last end-time can't be larger than 8640000 (24hours).";
    }
    description
      "The recurring base-schedule of the action or attribute
       modification";
  }
  description
    "Complete schedule for a action or attribute modification.";
}

augment "/if:interfaces/if:interface" {
  description
    "Augments interface with scheduled attribute.";
  container scheduled-up-down {
    description
      "Scheduled interface up and down. This is to work with the
       leaf 'enabled' for the configured state of the interface.";

    uses schedule;
  }
}

augment "/lne:logical-network-elements"
      + "/lne:logical-network-element" {
  description
    "Augments logical network elements with scheduled power
     shutdown.";
  container scheduled-power {
    description
      "Scheduled power up and down.";
    uses schedule;
  }
}
```

```
}
<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 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-schedule.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. There are the subtrees and data nodes and their sensitivity/vulnerability:

> /if:interfaces/if:interface/scheduled-attributes
> /lne:logical-network-elements/lne:logical-network-element/
> scheduled-attribute Modifications to these scheduled attributes
> may result in a denial of service.

Some of the readable data nodes in the ietf-tvr-schedule.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.

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.

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-schedule
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>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[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>.

[RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <https://www.rfc-editor.org/info/rfc8530>.

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., and Y. Qu, "TVR (Time-Variant Routing) Use Cases", Work in Progress, Internet-Draft, draft-ietf-tvr-use-cases-01, 3 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tvr-use-cases-01>.

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.

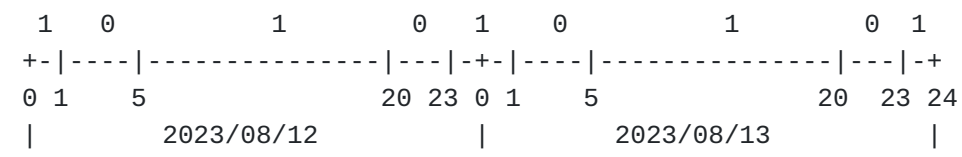
This example demonstrates how an OSPF interface can be extended with a cost that changes with a schedule.

```
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-schedule";
  }

  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-schedule:schedule;
    }
  }
}
```

Appendix B. Example: LNE scheduled power configuration

In this configuration example, LNE power is scheduled between 2023/8/12 0:00 UTC to the end of 2023/08/13 with a daily schedule. In each day, the router is scheduled to be off between 1:00 to 5:00 and 20:00 to 23:00. The two intervals and the corresponding value are defined, the "value-default" is to define the value that's not covered by the intervals.

THe following figure shows the intended power schedule.



The following XML demonstrate the configuration.

```
<?xml version='1.0' encoding='UTF-8'?>
  <logical-network-elements xmlns="urn:ietf:params:xml:ns:yang:
                                  ietf-logical-network-element">
    <logical-network-element>
      <name>"Router ABC"</name>
      <scheduled-power xmlns="urn:ietf:params:xml:ns:yang:
                              ietf-tvr-schedule">
        <start-date-time>2023-08-12T00:00:00.00Z</start-date-time>
        <end-date-time>2023-08-13T24:00:00.00Z</end-date-time>
        <recurrence>daily</recurrence>
        <value-default>1</value-default>
        <base-schedule>
          <intervals>
            <start-time>360000</start-time>
            <end-time>1800000</end-time>
            <value>0</value>
          </intervals>
          <intervals>
            <start-time>7200000</start-time>
            <end-time>8280000</end-time>
            <value>0</value>
          </intervals>
        </base-schedule>
      </scheduled-power>
    </logical-network-element>
  </logical-network-elements>
```

Authors' Addresses

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America

Email: yingzhen.qu@futurewei.com

Acee Lindem
LabN Consulting LLC
301 Midenhall Way
Cary, NC 27513

Email: acee.ietf@gmail.com

Marc Blanchet
Viagenie

Email: marc.blanchet@viagenie.ca