Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2017

X. Liu
Jabil
I. Bryskin
Huawei Technologies
V. Beeram
Juniper Networks
T. Saad
Cisco Systems Inc
H. Shah
Ciena
O. Gonzalez de Dios
Telefonica

March 9, 2017

A YANG Data Model for Configuration Scheduling
draft-liu-netmod-yang-schedule-03

Abstract

This document describes a data model for configuration scheduling.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document introduces a YANG [RFC6020] data model for configuration scheduling. This model can be used together with other YANG data models to specify a schedule applied on a configuration data node, so that the configuration data can take effect according to the schedule. Such a configuration schedule can be one-time or recurring, with its properties persistently saved in the datastores of the management system server.

The mechanism described in this document is designed to complement the one described in [RFC7758], which defines a capability extension to NETCONF to allow time-triggered RPCs. Such RPCs can be executed at a future time moment, but cannot be repeated and is not saved in the persistent datastores.

1.1. Terminology

The keywords "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, [RFC2119].

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

2. Motivation

Some applications benefit from resource scheduling to allow operators to plan ahead of time. Traffic engineering is one of such examples [RFC7399]. When configuration and state models are designed for such applications, it has been considered that certain data objects need to be configured according to predefined schedules. In other situations, operators need to de-configure certain data objects at predefined schedules for the purposes such as maintenance. These data objects are interpreted and implemented by the applicable applications.

Delay/Disruption Tolerant Networking (DTN) is another example for which the scheduled configuration can be used, where a long-lived, reliable, low-latency sequenced data delivery session is unsustainable. Section 4.3 of [I-D.birrane-dtn-ama] describes the Autonomous Parameterized Control. Time-based event is one of the two types of triggers in such a system.

3. Configuration Scheduling YANG Data Model Overview

This document defines a YANG data model that specifies configuration schedules for other YANG data models. For each targeted configuration data object or a group of configuration data objects, an entry is specified along with requested schedules using this configuration schedule model. The application implementing the targeted schema nodes implements the configuration schedules, configuring or de-configuring the specified objects according to the specified schedules. The model schema of the targeted application does not need changes, so the data model described in this document can be used for any data model. The configuration scheduling YANG data model has the following structure:

```
module: ietf-schedule
   +--rw configuration-schedules
      +--rw target* [object]
         +--rw object        yang:xpath1.0
         +--rw operation?    operation
         +--rw data-value?   anydata
         +--rw schedules
         |  +--rw schedule* [schedule-id]
         |     +--rw schedule-id            uint32
         |     +--rw inclusive-exclusive?   enumeration
         |     +--rw start?                 yang:date-and-time
         |     +--rw schedule-duration?     string
         |     +--rw repeat-interval?       string
         +--ro state
         |  +--ro future-executions
         |     +--ro execution* [start]
         |        +--ro start        yang:date-and-time
         |        +--ro duration?    string
         |        +--ro operation?   operation
         +---n execution
            +---- operation    operation
            +---- datetime?    yang:date-and-time
            +---- results?     anydata
```

4. Usage Example

The following model defines a list of TE (Traffic Engineering) links which can be configured with specified schedules:

```
module: example
   +--rw te-links
      +--rw te-link* [id]
         +--rw id         string
```

```
            +--rw enabled?   boolean
```

The following configuration requests that

- o link-1 is configured weekly for five one-day periods, starting from 2016-09-12T23:20:50.52Z.

- o link-2 is de-configured for two hours, starting from 2016-09-15T01:00:00.00Z.

```
   <configuration-schedules>
     <target xmlns:ex=""urn:example">
       <object>/ex:te-links</object>
       <operation>configure</operation>
       <data-value>
         <te-link>
      <id>link-1</id>
      <enabled>true</enabled>
         </te-link>
       </data-value>
       <schedules>
         <schedule>01
      <schedule-id>11<schedule-id>
      <start>2016-09-12T23:20:50.52Z</start>
      <schedule-duration>P1D</schedule-duration>
      <repeat-interval>R5/P1W</repeat-interval>
         </schedule>
       </schedules>
     </target>
     <target xmlns:ex=""urn:example">
       <object>/ex:te-links</object>
       <operation>configure</operation>
       <data-value>
         <te-link>
      <id>link-2</id>
      <enabled>true</enabled>
         </te-link>
       </data-value>
       <schedules>
         <schedule>
      <schedule-id>12<schedule-id>
      <inclusive-exclusive>exclusive</inclusive-exclusive>
      <start>2016-09-15T01:00:00.00Z</start>
      <schedule-duration>P2H</schedule-duration>
         </schedule>
       </schedules>
     </target>
   </configuration-schedules>
```

The following configuration requests that

- link-1 is enabled weekly for five one-day periods, starting from 2016-09-12T23:20:50.52Z.

- link-2 is not enabled for two hours, starting from 2016-09-15T01:00:00.00Z.

```
   <configuration-schedules>
     <target xmlns:ex=""urn:example">
       <object>/ex:te-links/ex:te-link[ex:link-id=’link-
   1’]/ex:enabled</object>
       <operation>set</operation>
       <data-value>true</data-value>
       <schedules>
         <schedule>
      <schedule-id>11<schedule-id>
      <start>2016-09-12T23:20:50.52Z</start>
      <schedule-duration>P1D</schedule-duration>
      <repeat-interval>R5/P1W</repeat-interval>
         </schedule>
       </schedules>
     </target>
     <target xmlns:ex=""urn:example">
       <object>/ex:te-links/ex:te-link[ex:link-id=’link-
   2’]/ex:enabled</object>
       <operation>set</operation>
       <data-value>true</data-value>
       <schedules>
         <schedule>
      <schedule-id>12<schedule-id>
      <inclusive-exclusive>exclusive</inclusive-exclusive>
      <start>2016-09-15T01:00:00.00Z</start>
      <schedule-duration>P2H</schedule-duration>
         </schedule>
       </schedules>
     </target>
   </configuration-schedules>
```

5. Relations to Datastores

NETCONF defines configuration datastores and operations that can be used to access these datastores. The configuration data encoded according to this data model is persistently saved in the proper datastores in the same way as other data model, such as ietf-interfaces.

5.1. Validation

When configuration data based on this model is received, the server MUST perform syntax validations on the received data nodes, and examine the requested schedules. The server does not validate whether requested target configuration data can be applied to the target configuration objects, until the actual scheduled time arrives.

At each scheduled time moment, the server applies the requested target configuration data to the target configuration objects. The server MUST perform the validations on the target configuration data along with the current target configuration objects in the proper datastore.

5.2. Schedules Expansion and Operational States

The server SHOULD expand these schedules and expose them to the client as operational states.

5.3. Server Executions at Scheduled Moments

At each scheduled time moment, the server applies the requested target configuration data to the target configuration objects, as if an RPC request is newly received. Whether such a time-triggered configuration is successfully applied depends on the configuration data of the target object and requested configuration data. The results of such executions are sent to the client through notifications. The notification management mechanism described in [I-D.ietf-netconf-yang-push] and [I-D.ietf-netconf-rfc5277bis] can be used to enable, disable, subscribe, filter, and replay the notifications.

5.4. Interactions with Locks

The rules of datastore lock specified by NETCONF [RFC6241] are checked when the schedule configuration data is received and when the target configuration data is applied.

5.5. Interactions with Authorization Mechanism

If the server implements any authorization mechanism, the authorization rules MUST be checked against this data model schema when the schedule configuration data is received. At each scheduled time moment, the authorization rules MUST be checked against the target objects by using the target configuration data. To check the authorization rules, the server uses the same client credential learned when the initial configuration data was received.

6. Synchronization Aspects

The scheduling mechanisms described in this document assume that servers have access to the wall-clock time. Thus, servers are required to acquire the time-of-day from an external time source, for example using the Network Time Protocol [RFC5905], or the Precision Time Protocol [IEEE1588].

It is assumed that the client and servers rely on a common time source, so as to guarantee that schedules are defined with respect to a common reference. In order to avoid the potential ambiguity of different time zones and daylight saving time, it is recommended to define all schedules in the UTC time zone, using the suffix ’Z’. For example, the time 2016-09-12T23:20:50.52Z, is specified with respect to the UTC time zone.

7. Configuration Scheduling YANG Module

```
<CODE BEGINS> file "ietf-schedule@2017-03-06.yang"
module ietf-schedule {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-schedule";

  prefix "sch";
```

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

     organization "TBD";
     contact "TBD";
     description
       "The model allows time scheduling parameters to be specified.";

     revision "2017-03-06" {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Typedefs
      */
     typedef operation {
       type enumeration {
         enum configure {
           description
           "Create the configuration data.";
         }
         enum deconfigure {
           description
           "Remove the configuration data.";
         }
         enum set {
           description
           "Set the specified configuration data.";
         }
         enum reset {
           description
           "Revert the specified configuration data back to the
            original value.";
         }
       }
       description "Operation type.";
     }
     /*
      * Groupings
      */

     grouping schedule-config-attributes {
       description
         "A group of attributes for a schedule.";

       leaf inclusive-exclusive {
         type enumeration {
           enum inclusive {
             description
               "The schedule element is inclusive, i.e., the schedule
                specifies the time at which the element is enabled.";
           }
           enum exclusive {
             description
               "The schedule element is exclusive. i.e., the schedule
                specifies the time at which the element is disabled.";
           }
         }
         default "inclusive";
         description
           "Whether the list item is inclusive or exclusive.";
       }
       leaf start {
         type yang:date-and-time;
         description "Start time.";
       }
       leaf schedule-duration {
         type string {
           pattern
             ’P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)?’;
         }
         description "Schedule duration in ISO 8601 format.";
       }
       leaf repeat-interval {
         type string {
           pattern
             ’R\d*/P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?’
```

```
               + '(\d+S)?';
           }
           description "Repeat interval in ISO 8601 format.";
         }
       } // schedule-config-attributes

       grouping schedule-config-notification {
         description
           "A group of attributes for a schedule notification.";

         notification execution {
           description
             "Notification event for an execution performed on a target
              object.";
           leaf operation {
             type operation;
             mandatory true;
             description "Operation type.";
           }
           leaf datetime {
             type yang:date-and-time;
             description
               "The date and time when the execution was performed.";
           }
           anydata results {
             description
               "This chunk of data contains the results of the execution
                performed on the target object. The results are the same
                or equivalent to the contents of a <rpc-reply> message,
                Because of the nature of such a target execution, a
                <rpc-reply> message is not used to return the execution
                results. Instead, this notification is used to serve
                the same purpose.";
           }
         }
       } // schedule-config-notification

       grouping schedule-state-attributes {
         description
           "State attributes for a schedule.";
         container future-executions {
           description
             "The state information of the nexte scheduled event.";
           list execution {
             key "start";
             description
               "List of scheduled future executions.";
             leaf start {
               type yang:date-and-time;
               description "Start time.";
             }
             leaf duration {
               type string {
                 pattern
                   'P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)?';
               }
               description "Schedule duration in ISO 8601 format.";
             }
             leaf operation {
               type operation;
               description "Operation type.";
             }
           } // event
         } // future-events
       } // schedule-state-attributes

       grouping schedules {
         description
           "A list of schedules defining when a particular
            configuration takes effect.";
         container schedules {
           description
             "Container of a schedule list defining when a particular
              configuration takes effect.";
           list schedule {
             key "schedule-id";
             description "A list of schedule elements.";
             leaf schedule-id {
               type uint32;
               description "Identifies the schedule element.";
             }
             uses schedule-config-attributes;
           }
         }
       } // schedules

       /*
        * Configuration data and operational state nodes
        */
       container configuration-schedules {
         description
           "Serves as top-level container for a list of configuration
            schedules.";
         list target {
           key "object";
           description
             "A list of targets that configuration schedules are
              applied.";
           leaf object {
             type yang:xpath1.0;
             description
               "Xpath defining the data items of interest.";
           }
           leaf operation {
             type operation;
             default "configure";
             description
               "Operation type.";
           }
           anydata data-value {
             description
               "The data value applied to the leaf data node
                specified by data-objects.
                The format of the data value depends on the value of the
                leaf operation defined above:
                configure:   data-value is the sub-tree added to the
                             target object;
                deconfigure: data-value is the child to be deleted from
                             the target object;
                set:         the target object MULST be a leaf, and
```

```
                        data-value is the new value to be set to
                        the target object;
               reset:       data-value is ignored.";
          }
          uses schedules;
          container state {
            config false;
            description
              "Operational state data.";
            uses schedule-state-attributes;
          } // state

          uses schedule-config-notification;
        } // target
      } // configuration-schedules
    }
    <CODE ENDS>
```

8. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and contents.

The functionality defined in this memo can potentially allow network reconnaissance; by gathering information about schedules an attacker can learn about the network policy, its temporal behavior, and future events.

The schedule YANG model defines schedules that are writable, creatable, and deletable. Therefore, this model may be considered sensitive or vulnerable in some network environments. An attacker may maliciously configure a schedule in a way that disrupts the normal behavior of the network. Furthermore, an attacker may attempt to maliciously set a schedule or a set of schedules in a way that amplifies an attack, or schedules an attack to a particularly sensitive time instant.

The use of configuration scheduling implicitly assumes that there is an underlying synchronization or time distribution mechanism. Therefore, an attack on the synchronization mechanism may compromise the configuration scheduling. The security considerations of time protocols are discussed further in [RFC 7384].

9. Contributors

Tal Mizrahi

Email: talmi@marvell.com

10. References

10.1. Normative References

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard 1588.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, October 2014.

[RFC7399] Farrel, A. and King, D., "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, October 2014.

[RFC7758] Mizrahi, T. and Moses, Y., "Time Capability in NETCONF", RFC7758, February 2016.

[I-D.birrane-dtn-ama] Birrane, E., "Asynchronous Management Architecture", draft-birrane-dtn-ama-04 (work in progress), October 2016. [I-D.ietf-netconf-yang-push] Clemm, A., "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push (Work in Progress).

[I-D.ietf-netconf-rfc5277bis] Clemm, A., "Subscribing to Event Notifications", draft-ietf-netconf-rfc5277bis (Work in Progress).

10.2. Informative References

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, January 2011.

Authors' Addresses

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean, VA 22102
USA

Email: Xufeng_Liu@jabil.com

Igor Bryskin
Huawei Technologies
Email: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc
Email: tsaad@cisco.com

Himanshu Shah
Ciena
Email: hshah@ciena.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com