NETMOD Working Group M. Wang
Internet-Draft Q. Wu
Intended status: Standards Track Huawei
Expires: April 3, 2019 C. Xie
China Telecom
September 30, 2018

A YANG Data model for Event Management
draft-wwx-netmod-event-yang-00

Abstract

This document defines an YANG data model for event management [RFC7950]. The Event YANG provides the ability to monitor state change on the local system or on a remote system and take simple action when a trigger condition on system state is met.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines an Event YANG data model [RFC7950]. The Event YANG provides the ability to monitor state changes on the local system or on a remote system and take simple action when a trigger condition on system state is met.

The data model in this document is designed to be compliant with the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Conventions used in this document

2.1. Terminology

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]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

This document uses the following terms:

Error A deviation of a system from normal operation [RFC3877].

Fault Lasting error or warning condition [RFC3877].

Event Something that happens which may be of interest or trigger the invocation of the rule. A fault, an alarm, a change in network state, network security threat, hardware malfunction, buffer untilization crossing a threshold, network connection setup, an external input to the system, for example [RFC3877].

2.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Objectives

This section describes some of the design objectives for the Event Data Model:

The Event YANG should provide the ability to monitor yang instance on a local or remote system using the NETCONF/RESTCONF, and initiates simple actions whenever a trigger condition is met. For example, a NETCONF subscribed notification can be generated when an instance value exceeds the threshold.

Clear and precise identification of Event types and instances.

Allow the server to inform the client that certain Events are related to other Events.

Allow one event to be able to trigger another external event or generate derived events.

The event data model defined in this document can be implemented on a system that also implements EVENT-MIB; thus, the mapping between the event data model and ENTITY-MIB should be clear.

4. Relationship to YANG PUSH

YANG-push mechanism provides a subscription service for updates from a datastore. And it support two types of subscriptions which are distinguished by how updates are triggered: periodic and on-change.

The On-change PUSH allow receivers to receive updates whenever changes to targeted objects occur. This document specifies a mechanism that provides three trigger conditions:

- Existence: When a specific yang instance appears, the trigger fires. E.g. Reserved ports are configured.

- Boolean: If the test result is true the trigger fires. The trigger will not fire again until the value has become false and come back to true. And the user can set the type of boolean comparison (e.g. unequal, equal, less, less-or-equal, greater, greater-or-equal, etc), so for example if the boolean comparison type is 'less' the trigger will be fired if the value of YANG Instance is less than the value of Boolean Value.

- Threshold: The event that may be triggered when a YANG instance at data-instance list is found. If the first sample after this instance becomes active is greater than or equal to 'rising-value' and the 'startup' is equal to 'rising' then one threshold rising event is triggered for that instance. And if the first sample after this instance becomes active is less than or equal to 'falling-value' and the 'startup' is equal to 'falling' then one threshold falling event is triggered for that instance.

And the YANG PUSH mechanism more focus on the remote mirroring and monitoring of configuration and operational state. For example, for on change method, the subscriber will receive a notification if the changes appears. The model defines in this document provides a method which allow automatic setting the value of the corresponding instance node when some event is triggered. It esbalishes connection between network service monitoring and network service provision and can use output generated by network service monitoring as input of network service provision and thereby provide automated network management. The details of the usage example is described in Appendix A.

5. Relationship to EVENT MIB

If the device implements the EVENT-MIB [RFC2981], each entry in the "/events/event/trigger" list is mapped to MteTriggerEntry,MteTriggerExistenceEntry,MteTriggerBooleanEntry,MteTriggerThresholdEntry,MteObjectsEntry,MteEventEntry,MteEventSetEntry. respectively.

The following table lists the YANG data nodes with corresponding objects in the EVENT-MIB [RFC2981].

YANG data node in ietf-event.yang	EVENT-MIB Objects (RFC2981)
evt-smp-min	mteResourceSampleMinimum
evt-smp-instance-max	mteResourceSampleInstanceMaximum
traget	mteObjectsName
event-name	mteEventName
event-description	mteEventComment
value	mteEventSetValue
events/event/trigger/name	mteTriggerName
trigger-description	mteTriggerComment
frequency	mteTriggerFrequency
comparison	mteTriggerBooleanComparison
value	mteTriggerBooleanValue
rising-event	mteTriggerThresholdRising
falling-event	mteTriggerThresholdFalling
delta-rising-event	mteTriggerThresholdDeltaRising
threshold/startup	mteTriggerThresholdStartup
existence/enable	mteTriggerExistenceStartup
boolean/enable	mteTriggerBooleanStartup

6. Model Overview

The event yang has four lists: trigger, target, event, and action. Triggers define the targets meeting some conditions that lead to events. Events trigger corresponding actions.

The trigger list defines what managed objects or targets are to be monitored and how and relates each trigger to an event. In this model, the trigger list provides three trigger conditions:

- Existence: When a specific yang instance appears, the trigger fires.

- Boolean: If the test result is true the trigger fires.

- Threshold: The event that may be triggered when a YANG instance at data-instance list is met the threshold.

Each trigger can be seen as a logical test that, if satisfied or evaluated to be true, cause the action to be carried out.

The target list defines managed objects that can be added to notifications based or be set to a new value on the trigger, the trigger test type, or the event that resulted in the actions.

The event list defines what happens when an event is triggered, i.e., trigger corresponding action, e.g.,sending a notification, setting a value to the managed object or both.

The action list consists of updates or innvcations on local managed object attributes and defines a set of actions which will be performed (e.g. notification, set, another event, etc) when corresponding event be triggered. The value to be set can use many variations on rule structure.

This document defines the YANG module "ietf-event", which has the following structure:

```
 module: ietf-event
    +--rw events
       +--rw evt-smp-min?            uint32
       +--rw evt-smp-instance-max?   uint32
       +--rw event* [event-name type]
          +--rw event-name           string
          +--rw type                 identityref
          +--rw event-description?   string
          +--rw target*              target
          +--rw clear?               boolean
          +--rw related-event* [event-name type]
          |  +--rw event-name    string
          |  +--rw type          identityref
          +--rw trigger* [name]
          |  +--rw name                   string
          |  +--rw type?                  enumeration
```

```
          |  +--rw trigger-description?   string
          |  +--rw frequency
          |  |  +--rw type?         identityref
          |  |  +--rw periodic
          |  |  |  +--rw interval    uint32
          |  |  |  +--rw start?      yang:date-and-time
          |  |  |  +--rw end?        yang:date-and-time
          |  |  +--rw scheduling
          |  |  |  +--rw month*          string
          |  |  |  +--rw day-of-month*   uint8
          |  |  |  +--rw day-of-week*    uint8
          |  |  |  +--rw hour*           uint8
          |  |  |  +--rw minute*         uint8
          |  |  |  +--rw second*         uint8
          |  |  |  +--rw start?          yang:date-and-time
          |  |  |  +--rw end?            yang:date-and-time
          |  |  +--rw immediate
          |  |     +--rw immediate    empty
          |  +--rw (test)?
          |     +--:(existences)
          |     |  +--rw existences
          |     |     +--rw target*   -> /events/event/target
          |     |     +--rw enable?   boolean
          |     +--:(boolean)
          |     |  +--rw boolean
          |     |     +--rw comparison?   enumeration
          |     |     +--rw value?        match-value
          |     |     +--rw target*       -> /events/event/target
          |     |     +--rw enable?       boolean
          |     +--:(threshold)
          |        +--rw threshold
          |           +--rw rising-value?          match-value
          |           +--rw rising-target*         -> /events/event/target
          |           +--rw falling-value?         match-value
          |           +--rw falling-target*        -> /events/event/target
          |           +--rw delta-rising-value?    match-value
          |           +--rw delta-rising-target*   -> /events/event/target
          |           +--rw startup?               enumeration
          +--rw action* [action-name]
             +--rw action-name           string
             +---n event-notification
             |  +---- event-name?   -> /events/event/event-name
             |  +---- type?         -> /events/event/type
             |  +---- target*       target
             +---x set
             |  +---w input
             |     +---w target*   target
             |     +---w value?    <anydata>
```

```
             +--rw trigger-event*        -> ../../event-name
```

The relation between Event, Trigger, Target and Action is described as follows:

```
   +-------------------------------+
   | Event                         |
   |  +--------+                   |
   |  |Target |                    |
   |  +---|---+                    |   +-------------------------------+
   |      |                        |   | Event                         |
   | +----V---+        +---------+  |   |  +--------+                   |
   | |Trigger |------->| Action |-------->|Target |                    |
   | +---------+        +---------+  |   |  +---|---+                    |
   +-------------------------------+   |      |                        |
                                       | +----|---+        +---------+  |
                                       | |Trigger |------- | Action |  |
                                       | +---------+        +---------+  |
                                       +-------------------------------+
```

One event may trigger another event, but if it does not trigger another event, the right part of above figure should be ignored.

7. EVENT YANG Module

<CODE BEGINS> file "ietf-event@2018-09-18.yang"

```
module ietf-event {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-event";
  prefix evt;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF xxx Working Group";
  contact
    "Zitao Wang: wangzitao@huawei.com
     Qin Wu: bill.wu@huawei.com";
  description
    "This module defines a model for the service topology.";

  revision 2018-09-18 {
    description
      "Initial revision.";
    reference "foo";
```

```
  }

  identity event-type {
    description
      "Base identity for event type";
  }

  identity frequency {
    description
      "Base identity for frequency";
  }

  identity periodic {
    base frequency;
    description
      "Identity for periodic trigger";
  }

  identity scheduling {
    base frequency;
    description
      "Identity for scheduling trigger";
  }

  identity immediate {
    base frequency;
    description
      "Identity for immediate trigger";
  }

  typedef match-value {
    type union {
      type yang:xpath1.0;
      type yang:object-identifier;
      type string;
    }
    description
      "This type is used to match resources of type 'target'.
       Since the type 'target' is a union of different types,
       the 'match-value' type is also a union of corresponding
       types.";
  }

  typedef target {
    type union {
      type instance-identifier;
      type yang:object-identifier;
      type yang:uuid;
```

```
        type string;
      }
      description
        "If the target is modelled in YANG, this type will
         be an instance-identifier.
         If the target is an SNMP object, the type will be an
         object-identifier.
         If the target is anything else, for example a distinguished
         name or a CIM path, this type will be a string.
         If the target is identified by a UUID use the uuid
         type.
         If the server supports several models, the presedence should
         be in the order as given in the union definition.";
    }

    grouping start-end-grouping {
      description
        "A grouping that provides start and end times for
         Event objects.";
      leaf start {
        type yang:date-and-time;
        description
          "The date and time when the Event object
           starts to create triggers.";
      }
      leaf end {
        type yang:date-and-time;
        description
          "The date and time when the Event object
           stops to create triggers.
           It is generally a good idea to always configure
           an end time and to refresh the end time as needed
           to ensure that agents that lose connectivity to
           their Controller do not continue executing Schedules
           forever.";
      }
    }

    container events {
      leaf evt-smp-min {
        type uint32;
        description
          "Sets the minimum value for data instance sampling";
      }
      leaf evt-smp-instance-max {
        type uint32;
        description
          "Sets the maximum value for data instance instance sampling.";
```

```
    }
    list event {
      key "event-name type";
      leaf event-name {
        type string;
        description
          "Event name";
      }
      leaf type {
        type identityref {
          base event-type;
        }
        description
          "Type of event";
      }
      leaf event-description {
        type string;
        description
          "Event description";
      }
      leaf-list target {
        type target;
        description
          "targeted objects";
      }
      leaf clear {
        type boolean;
        default "false";
        description
          "A flag indicate whether the event be closed";
      }
      list related-event {
        key "event-name type";
        leaf event-name {
          type string;
          description
            "Event name";
        }
        leaf type {
          type identityref {
            base event-type;
          }
          description
            "Type of event";
        }
        description
          "List for related events";
      }
```

```
      list trigger {
        key "name";
        leaf name {
          type string;
          description
            "Trigger name";
        }
        leaf type {
          type enumeration {
            enum "existence" {
              description
                "Indicates that the trigger type is 'existence'.
                 For 'existence', the specific test is as selected.
                 When an object appears, vanishes
                 or changes value, the trigger fires.";
            }
            enum "boolean" {
              description
                "Indicates that the trigger type is 'boolean'.
                 If the test result is true the trigger fires.  The
                 trigger will not fire again until the value has become false
and
                 come back to true.";
            }
            enum "threshold" {
              description
                "Indicates that the trigger type is 'threshold'.
                 The event that may be triggered when a YANG instance
                 at data-instance list is found.  If the first sample after
this
                 instance becomes active is greater than or equal to 'rising-
value'
                 and the 'startup' is equal to 'rising' then one threshold
rising
                 event is triggered for that instance.";
            }
          }
          description
            "Trigger type";
        }
        leaf trigger-description {
          type string;
          description
            "Trigger description";
        }
        container frequency {
          leaf type {
            type identityref {
```

```
            base frequency;
          }
          description
            "Type of trigger frequency";
```

```
         }
         container periodic {
           when "derived-from-or-self(../type, 'periodic')";
           description
             "A periodic timing object triggers periodically
              according to a regular interval.";
           leaf interval {
             type uint32 {
               range "1..max";
             }
             units "seconds";
             mandatory true;
             description
               "The number of seconds between two triggers
                generated by this periodic timing object.";
           }
           uses start-end-grouping;
         }
         container scheduling {
           when "derived-from-or-self(../type, 'scheduling')";
           description
             "A scheduling timing object triggers.";
           leaf-list month {
             type string;
             description
               "A set of months at which this scheduling timing
                will trigger.";
           }
           leaf-list day-of-month {
             type uint8 {
               range "0..59";
             }
             description
               "A set of days of the month at which this
                scheduling timing will trigger.";
           }
           leaf-list day-of-week {
             type uint8 {
               range "0..59";
             }
             description
               "A set of weekdays at which this scheduling timing
                will trigger.";
           }
           leaf-list hour {
             type uint8 {
               range "0..59";
             }
```

```
              description
                "A set of hours at which the scheduling timing will
                 trigger.";
            }
            leaf-list minute {
              type uint8 {
                range "0..59";
              }
              description
                "A set of minutes at which this scheduling timing
                 will trigger.";
            }
            leaf-list second {
              type uint8 {
                range "0..59";
              }
              description
                "A set of seconds at which this calendar timing
                 will trigger.";
            }
            uses start-end-grouping;
          }
          container immediate {
            when "derived-from-or-self(../type, 'immediate')";
            leaf immediate {
             type empty;
             mandatory true;
             description
              "This immediate Event object triggers immediately
               when it is configured.";
          }
          description
           "This immediate Event object triggers immediately
            when it is configured.";
         }
          description
            "Container for frequency";
        }
        choice test {
          description
            "Choice test";
          container existences {
            leaf-list target {
              type leafref {
                path "/events/event/target";
              }
              description
                "List for target objects";
```

```
            }
            leaf enable {
              type boolean;
              description
                "Startup";
            }
            description
              "Container for existence";
          }
          container boolean {
            leaf comparison {
              type enumeration {
                enum "unequal" {
                  description
                    "Indicates that the comparision type is 'unequal'.";
                }
                enum "equal" {
                  description
                    "Indicates that the comparision type is 'equal'.";
                }
                enum "leass" {
                  description
                    "Indicates that the comparision type is 'less'.";
                }
                enum "less-or-equal" {
                  description
                    "Indicates that the comparision type is 'less or equal'.";
                }
                enum "greater" {
                  description
                    "Indicates that the comparision type is 'greater'.";
                }
                enum "greater-or-equal" {
                  description
                    "Indicates that the comparision type is 'greater or
equal'.";
                }
              }
              description
                "Comparison type";
            }
            leaf value {
              type match-value;
              description
                "Compartion value";
            }
            leaf-list target {
              type leafref {
                path "/events/event/target";
```

```
              }
              description
                "List for target objects";
            }
            leaf enable {
              type boolean;
              description
                "Startup";
            }
            description
              "Container for boolean test";
          }
          container threshold {
            leaf rising-value {
              type match-value;
              description
                "Sets the rising threshold to the specified value, ]
                 When the current sampled value is greater than or equal to
                 this threshold, and the value at the last sampling interval
                 was less than this threshold, the event is triggered. ";
            }
            leaf-list rising-target {
              type leafref {
                path "/events/event/target";
              }
              description
                "List for target objects";
            }
            leaf falling-value {
              type match-value;
              description
                "Sets the falling threshold to the specified value";
            }
            leaf-list falling-target {
              type leafref {
                path "/events/event/target";
              }
              description
                "List for target objects";
            }
            leaf delta-rising-value {
              type match-value;
              description
                "Sets the delta rising threshold to the specified value";
            }
            leaf-list delta-rising-target {
              type leafref {
                path "/events/event/target";
```

```
              }
              description
                "List for target objects";
            }
            leaf startup {
              type enumeration {
                enum "rising" {
                  description
                    "If the first sample after this
                     instance becomes active is greater than or equal to
'rising-value'
                     and the 'startup' is equal to 'rising' then one threshold
rising
                     event is triggered for that instance.";
                }
                enum "falling" {
                  description
                    "If the first sample
                     after this instance becomes active is less than or equal
to
                     'falling-value' and the 'startup' is equal to 'falling'
then one
                     threshold falling event is triggered for that instance.";
                }
                enum "rising-or-falling" {
                  description
                    "That event is
                       also triggered if the first sample after this entry
becomes
                       active is less than or equal or rising than to this
threshold and
                       'startup' is equal to 'rising-or-falling'.";
                }
              }
              description
                "Startup";
            }
            description
              "Container for threshold";
          }
        }
        description
          "List for trigger";
      }
      list action {
        key "action-name";
        leaf action-name {
          type string;
          description
            "Action name";
        }
        notification event-notification {
          leaf event-name {
            type leafref {
              path "/events/event/event-name";
            }
            description
              "Report the event name";
          }
          leaf type {
            type leafref {
              path "/events/event/type";
            }
            description
              "Report the event type";
          }
          leaf-list target {
            type target;
            description
              "Report the target objects";
          }
          description
            "This notification is used to report that an operator
             acted upon an Event.";
        }
        action set {
          input {
            leaf-list target {
              type target;
              description
                "Report the target objects";
            }
            anydata value {
              description
                "Inline set content.";
            }
          }
        }
        leaf-list trigger-event {
          type leafref {
            path "../../event-name";
          }
          description
            "This action trigger another event";
        }
        description
          "List for Actions";
      }
      description
        "List for Events";
    }
    description
```

```
      "YANG data module for defining event triggers
       and actions for network management purposes";
  }
}

   <CODE ENDS>
```

8. Security Considerations

The YANG modules defined in this document MAY be accessed via the RESTCONF protocol [RFC8040] or NETCONF protocol ([RFC6241]). The lowest RESTCONF or NETCONF layer requires that the transport-layer protocol provides both data integrity and confidentiality, see Section 2 in [RFC8040] and [RFC6241]. 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 [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- /events/event/event-name

- /events/event/target

- /events/action/set/target

- /events/event/trigger/name

9. 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-event
      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-event
      Namespace:    urn:ietf:params:xml:ns:yang:ietf-event
      Prefix:       evt
      Reference:    RFC xxxx
   --------------------------------------------------------------------
```

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

[RFC2981] Kavasseri, R., Ed., "Event MIB", RFC 2981, DOI 10.17487/RFC2981, October 2000, <https://www.rfc-editor.org/info/rfc2981>.

[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>.

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <https://www.rfc-editor.org/info/rfc6370>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[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>.

[RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <https://www.rfc-editor.org/info/rfc7952>.

Appendix A. Example of Event

For example, some service requires to monitoring the "in-errors" state of the interface, and if the value of "in-errors" exceeds the threshold, the event should reset the interface's enabled value to false:

```
   <events>
    <event>
     <event-name>interface-state-exception</event-name>
     <type>interface-exception</type>
     <target>/if:interfaces/if:interface[if:name='eth1']</target>
     <target>/if:interfaces/if:interface[if:name='eth2']</target>
     <target>/if:interfaces/if:interface[if:name='eth3']</target>
     <trigger>
      <name>evaluate-in-errors</name>
      <trigger-description>evaluate the number of
       the packets that contained errors
       </trigger-description>
      <frequency>10m</frequency>
      <type>threshold</type>
      <test>
       <threshold>
        <startup>rising</startup>
        <rising-value>100</rising-value>
        <rising-target>/if:interfaces/if:interface[if:name='eth1']
          /if:statistic/if:in-errors</rising-target>
        <rising-target>/if:interfaces/if:interface[if:name='eth2']
          /if:statistic/if:in-errors</trising-arget>
       </threshold>
      </test>
     </trigger>
     <action>
      <name>interface-exception</name>
      <event-notification>
       <event-name>interface-state-exception</event-name>
       <type>interface-exception</type>
       <target>/if:interfaces/if:interface[if:name='eth1']</target>
      </event-notification>
      <set>
       <target>/if:interfaces/if:interface[if:name='eth1']</target>
       <interger-value>
        <interfaces>
         <interface>
          <name>eth1</name>
          <enable>false</enable>
         </interface>
        </interfaces>
       </interger-value>
      </set>
     <action>
    </event>
   </events>
```

Authors' Addresses

Michael Wang
Huawei Technologies,Co.,Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Chongfeng Xie
China Telecom

Email: xiechf@ctbri.com.cn