

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 13, 2019

L. Zheng
G. Zheng
Huawei Technologies
G. Mirsky
ZTE Corp.
R. Rahman
F. Iqbal
Cisco Systems
January 9, 2019

YANG Data Model for LSP-Ping
draft-zheng-mpls-lsp-ping-yang-cfg-10

Abstract

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. RFC 8029 defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in RFC 6020 and RFC 7950, is a data modeling language used to specify the contents of a conceptual data stores that allows networked devices to be managed using NETCONF, as specified in RFC 6241. This document defines a YANG data model that can be used to configure and manage LSP-Ping.

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 July 13, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Support of Long Running Command with NETCONF	3
2. Scope	3
3. Design of the Data Model	4
3.1. The Configuration of Control Information	4
3.2. The Configuration of Schedule Parameters	5
3.3. Display of Result Information	6
4. Data Hierarchy	7
5. Interaction with other MPLS OAM Tools Models	9
6. LSP-Ping YANG Module	10
7. Examples	21
7.1. Configuration of Control Information	21
7.2. The Configuration of Schedule Parameters	22
7.3. Display of Result Information	23
8. Security Considerations	25
9. IANA Considerations	26
Contributors	26
Acknowledgments	27
12. References	27
12.1. Normative References	27
12.2. Informative References	27
Authors' Addresses	28

1. Introduction

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. [RFC8029] defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in [RFC6020] and [RFC7950], is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. This document defines a YANG data model that can be used to configure and manage LSP-Ping [RFC8029].

The rest of this document is organized as follows. Section 2 presents the scope of this document. Section 3 provides the design of the LSP-Ping configuration data model in details by containers. Section 4 presents the complete data hierarchy of LSP-Ping YANG model. Section 5 discusses the interaction between LSP-Ping data model and other MPLS tools data models. Section 6 specifies the YANG module and section 7 lists examples which conform to the YANG module specified in this document. Finally, security considerations are discussed in Section 8.

This version of the LSP Ping data model conforms 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Support of Long Running Command with NETCONF

LSP Ping is one of the examples of what can be described as "long-running operation". Unlike most of the configuration operations that result in single response execution of an LSP Ping triggers multiple responses from a node under control. The question of implementing the long-running operation in NETCONF is still open and possible solutions being discussed:

1. Consecutive Remote Processing Calls (RPC) to poll for results.
2. Model presented in [RFC4560].
3. The one outlined in [I-D.mahesh-netconf-persistent].

The problem of long-running operation as well can be considered as a case of controlling and obtaining results from a Measurement Agent (MA) as defined in [RFC7594].

2. Scope

The fundamental mechanism of LSP-Ping is defined in [RFC8029]. Extensions of LSP-Ping has been developed over the years. There are extensions for performing LSP ping, for example, over P2MP MPLS LSPs [RFC6425] or for Segment Routing IGP Prefix and Adjacency SIDs with an MPLS data plane [RFC8287]. These extensions will be considered in a later update of this document.

3. Design of the Data Model

This YANG data model is defined to be used to configure and manage LSP-Ping and it provides the following features:

1. The configuration of control information of an LSP-Ping test.
2. The configuration of schedule parameters of an LSP-Ping test.
3. Display of result information of an LSP-Ping test.

The top-level container `lsp-pings` holds the configuration of the control information, schedule parameters and result information for multiple instances of LSP-Ping test.

3.1. The Configuration of Control Information

Container `lsp-pings:lsp-ping:control-parameters` defines the configuration parameters which control an LSP-Ping test. Examples are the `target-fec-type/target-fec` of the echo request packet and the `reply mode` of the echo reply packet. Values of some parameters may be auto-assigned by the system, but in several cases, there is a requirement for configuration of these parameters. Examples of such parameters are source address and outgoing interface.

The data hierarchy for control information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
        +--rw target-fec-type?      target-fec-type
        +--rw (target-fec)?
          +--:(ip-prefix)
            +--rw ip-address?        inet:ip-address
          +--:(bgp)
            +--rw bgp?                inet:ip-address
          +--:(rsvp)
            +--rw tunnel-interface?  string
          +--:(vpn)
            +--rw vrf-name?           uint32
            +--rw vpn-ip-address?    inet:ip-address
          +--:(pw)
            +--rw vcid?               uint32
          +--:(vpls)
            +--rw vsi-name?           string
        +--rw traffic-class?        uint8
        +--rw reply-mode?            reply-mode
        +--rw timeout?               uint32
        +--rw timeout-units?         units
        +--rw interval?              uint32
        +--rw interval-units?        units
        +--rw probe-count?           uint32
        +--rw data-size?              uint32
        +--rw data-fill?              string
        +--rw description?            string
        +--rw source-address?         inet:ip-address
        +--rw ttl?                    uint8
        +--rw (outbound)?
          +--:(interface)
            +--rw interface-name?     string
          +--:(nexthop)
            +--rw nexthop?             inet:ip-address

```

3.2. The Configuration of Schedule Parameters

Container `lsp-pings:lsp-ping:scheduling-parameters` defines the schedule parameters of an LSP-Ping test, which describes when to start and when to end the test. Four start modes and three end modes are defined respectively. To be noted that, the configuration of "interval" and "probe-count" parameter defined in container `lsp-pings:lsp-ping:control-parameters` could also determine when the test ends implicitly. All these three parameters are optional. If the user

does not configure either "interval" or "probe-count" parameter, then the default values will be used by the system. If the user configures "end-test", then the actual end time of the LSP-Ping test is the smaller one between the configuration value of "end-test" and the time implicitly determined by the configuration value of "interval"/"probe-count".

The data hierarchy for schedule information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
        +--rw (start-test)?
          +--:(now)
            | +--rw start-test-now?          empty
          +--:(at)
            | +--rw start-test-at?           yang:date-and-time
          +--:(delay)
            | +--rw start-test-delay?        uint32
            | +--rw start-test-delay-units?  units
          +--:(daily)
            | +--rw start-test-daily?        yang:date-and-time
        +--rw (end-test)?
          +--:(at)
            | +--rw end-test-at?             yang:date-and-time
          +--:(delay)
            | +--rw end-test-delay?          uint32
            | +--rw end-test-delay-units?    units
          +--:(lifetime)
            | +--rw end-test-lifetime?       uint32
            | +--rw lifetime-units?         units

```

3.3. Display of Result Information

Container `lsp-pings:lsp-ping:result-info` shows the result of the current LSP-Ping test. Both the statistical result e.g. `min-rtt`, `max-rtt`, and per test probe result e.g. `return code`, `return subcode`, are shown.

The data hierarchy for display of result information is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
      ...
      +--ro result-info
        +--ro operational-status?    operational-status
        +--ro source-address?        inet:ip-address
        +--ro target-fec-type?       target-fec-type
        +--ro (target-fec)?
          +--:(ip-prefix)
            | +--ro ip-address?       inet:ip-address
          +--:(bgp)
            | +--ro bgp?              inet:ip-address
          +--:(rsvp)
            | +--ro tunnel-interface? string
          +--:(vpn)
            | +--ro vrf-name?         uint32
            | +--ro vpn-ip-address?   inet:ip-address
          +--:(pw)
            | +--ro vcid?              uint32
          +--:(vpls)
            | +--ro vsi-name?         string
        +--ro min-rtt?               uint32
        +--ro max-rtt?               uint32
        +--ro average-rtt?           uint32
        +--ro probe-responses?       uint32
        +--ro sent-probes?           uint32
        +--ro sum-of-squares?        uint32
        +--ro last-good-probe?       yang:date-and-time
        +--ro probe-results
          +--ro probe-result* [probe-index]
            +--ro probe-index         uint32
            +--ro return-code?        uint8
            +--ro return-sub-code?    uint8
            +--ro rtt?                uint32
            +--ro result-type?        result-type

```

4. Data Hierarchy

The complete data hierarchy of LSP-Ping YANG model is presented below.

```

module: ietf-lsp-ping

```

```

+--rw lsp-pings
  +--rw lsp-ping* [lsp-ping-name]
    +--rw lsp-ping-name          string
    +--rw control-parameters
      +--rw target-fec-type?      target-fec-type
      +--rw (target-fec)?
        +--:(ip-prefix)
          +--rw ip-address?       inet:ip-address
        +--:(bgp)
          +--rw bgp?              inet:ip-address
        +--:(rsvp)
          +--rw tunnel-interface? string
        +--:(vpn)
          +--rw vrf-name?         uint32
          +--rw vpn-ip-address?   inet:ip-address
        +--:(pw)
          +--rw vcid?             uint32
        +--:(vpls)
          +--rw vsi-name?         string
      +--rw traffic-class?        uint8
      +--rw reply-mode?           reply-mode
      +--rw timeout?              uint32
      +--rw timeout-units?        units
      +--rw interval?             uint32
      +--rw interval-units?       units
      +--rw probe-count?          uint32
      +--rw data-size?            uint32
      +--rw data-fill?            string
      +--rw description?          string
      +--rw source-address?       inet:ip-address
      +--rw ttl?                  uint8
      +--rw (outbound)?
        +--:(interface)
          +--rw interface-name?   string
        +--:(nexthop)
          +--rw nexthop?          inet:ip-address
    +--rw scheduling-parameters
      +--rw (start-test)?
        +--:(now)
          +--rw start-test-now?    empty
        +--:(at)
          +--rw start-test-at?     yang:date-and-time
        +--:(delay)
          +--rw start-test-delay?   uint32
          +--rw start-test-delay-units? units
        +--:(daily)
          +--rw start-test-daily?   yang:date-and-time
      +--rw (end-test)?

```



```

    +---:(at)
    |   +---rw end-test-at?                yang:date-and-time
    +---:(delay)
    |   +---rw end-test-delay?            uint32
    |   +---rw end-test-delay-units?      units
    +---:(lifetime)
    |   +---rw end-test-lifetime?         uint32
    |   +---rw lifetime-units?           units
+---ro result-info
+---ro operational-status?               operational-status
+---ro source-address?                  inet:ip-address
+---ro target-fec-type?                 target-fec-type
+---ro (target-fec)?
|   +---:(ip-prefix)
|   |   +---ro ip-address?               inet:ip-address
+---:(bgp)
|   +---ro bgp?                         inet:ip-address
+---:(rsvp)
|   +---ro tunnel-interface?            string
+---:(vpn)
|   +---ro vrf-name?                    uint32
|   +---ro vpn-ip-address?              inet:ip-address
+---:(pw)
|   +---ro vcid?                        uint32
+---:(vpls)
|   +---ro vsi-name?                    string
+---ro min-rtt?                         uint32
+---ro max-rtt?                         uint32
+---ro average-rtt?                     uint32
+---ro probe-responses?                 uint32
+---ro sent-probes?                     uint32
+---ro sum-of-squares?                  uint32
+---ro last-good-probe?                 yang:date-and-time
+---ro probe-results
    +---ro probe-result* [probe-index]
        +---ro probe-index              uint32
        +---ro return-code?              uint8
        +---ro return-sub-code?          uint8
        +---ro rtt?                      uint32
        +---ro result-type?              result-type

```

5. Interaction with other MPLS OAM Tools Models

TBA

6. LSP-Ping YANG Module

```
<CODE BEGINS> file "ietf-lsp-ping@2018-11-29.yang"
module ietf-lsp-ping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lsp-ping";
  //namespace need to be assigned by IANA
  prefix "lsp-ping";

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

  organization "IETF Multiprotocol Label Switching Working Group";

  contact
    "WG Web: http://tools.ietf.org/wg/mppls/
    WG List: mppls@ietf.org

    Editor: Greg Mirsky
      gregimirsky@gmail.com
    Editor: Lianshu Zheng
      vero.zheng@huawei.com
    Editor: Guangying Zheng
      zhengguangying@huawei.com
    Editor: Reshad Rahman
      rrahman@cisco.com
    Editor: Faisal Iqbal
      faiqbal@cisco.com";

  description
    "This YANG module specifies a vendor-independent model
    for the LSP Ping.

    This YANG data model is defined to be used to configure and manage
    LSP-Ping and it provides the following features:
    1. The configuration of control information of an LSP-Ping test.
    2. The configuration of schedule parameters of an LSP-Ping test.
    3. Display of result information of an LSP-Ping test.

    Copyright (c) 2018 IETF Trust and the persons identified as
    the document authors. 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 Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

```
reference "draft-zheng-mpls-lsp-ping-yang-cfg";

revision "2018-11-29" {
  description
    "10 version, refine the target fec type,
    as per RFC8029 and update Security Considerations section.";
  reference "draft-zheng-mpls-lsp-ping-yang-cfg";
}

typedef target-fec-type {
  type enumeration {
    enum ip-prefix {
      value "0";
      description "IPv4/IPv6 prefix";
    }
    enum bgp {
      value "1";
      description "BGP IPv4/IPv6 prefix";
    }
    enum rsvp {
      value "2";
      description "Tunnel interface";
    }
    enum vpn {
      value "3";
      description "VPN IPv4/IPv6 prefix";
    }
    enum pw {
      value "4";
      description "FEC 128 pseudowire IPv4/IPv6";
    }
    enum vpls {
      value "5";
      description "FEC 129 pseudowire IPv4/IPv6";
    }
  }
  description "Target FEC type, as defined in RFC 8029";
}
```

```
typedef reply-mode {
  type enumeration {
    enum do-not-reply {
      value "1";
      description "Do not reply";
    }
    enum reply-via-udp {
      value "2";
      description "Reply via an IPv4/IPv6 UDP packet";
    }
    enum reply-via-udp-router-alert {
      value "3";
      description
        "Reply via an IPv4/IPv6 UDP packet with Router Alert";
    }
    enum reply-via-control-channel {
      value "4";
      description
        "Reply via application level control channel";
    }
  }
  description "Reply mode";
}

typedef units {
  type enumeration {
    enum seconds {
      description "Seconds";
    }
    enum milliseconds {
      description "Milliseconds";
    }
    enum microseconds {
      description "Microseconds";
    }
    enum nanoseconds {
      description "Nanoseconds";
    }
  }
  description "Time units";
}

typedef operational-status {
  type enumeration {
    enum enabled {
      value "1";
      description "The Test is active";
    }
  }
}
```

```
    enum disabled {
        value "2";
        description "The test has stopped";
    }
    enum completed {
        value "3";
        description "The test is completed";
    }
}
description "Operational state of an LSP Ping test";
}

typedef result-type {
    type enumeration {
        enum success {
            value "1";
            description "The test probe is successful";
        }
        enum fail {
            value "2";
            description "The test probe has failed";
        }
        enum timeout {
            value "3";
            description "The time of the test probe has expired";
        }
    }
    description "Result of each LSP Ping test probe";
}

container lsp-pings {
    description "Multi-instance of the LSP Ping test";
    list lsp-ping {
        key "lsp-ping-name";
        description "LSP Ping test";
        leaf lsp-ping-name {
            type string {
                length "1..31";
            }
            mandatory "true";
            description "LSP Ping test name";
        }
        container control-parameters {
            description "Control information of the LSP Ping test";
            leaf target-fec-type {
                type target-fec-type;
                description "Specifies the address type of the Target FEC";
            }
        }
    }
}
```

```
choice target-fec {
  case ip-prefix {
    leaf ip-address {
      type inet:ip-address;
      description "IPv4/IPv6 Prefix";
    }
  }
  case bgp {
    leaf bgp {
      type inet:ip-address;
      description "BGP IPv4/IPv6 Prefix";
    }
  }
  case rsvp {
    leaf tunnel-interface {
      type string;
      description "Tunnel interface";
    }
  }
  case vpn {
    leaf vrf-name {
      type uint32;
      description "Layer3 VPN Name";
    }
    leaf vpn-ip-address {
      type inet:ip-address;
      description "Layer3 VPN IPv4 Prefix";
    }
  }
  case pw {
    leaf vcid {
      type uint32;
      description "VC ID";
    }
  }
  case vpls {
    leaf vsi-name {
      type string;
      description "VPLS VSI";
    }
  }
  description "Specifies the type of the Target FEC";
}
leaf traffic-class {
  type uint8;
  description "Specifies the Traffic Class";
}
leaf reply-mode {
```

```
    type reply-mode;
    description "Specifies the Reply Mode";
  }
  leaf timeout {
    type uint32;
    description
      "Specifies the time-out value for a LSP Ping operation.";
  }
  leaf timeout-units {
    type units;
    description "Time-out units";
  }
  leaf interval {
    type uint32;
    default 1;
    description
      "Specifies the interval between transmissions
       of LSP Ping echo request packets (probes)
       as part of the LSP Ping test.";
  }
  leaf interval-units {
    type units;
    default seconds;
    description "Interval units";
  }
  leaf probe-count {
    type uint32;
    default 5;
    description
      "Specifies the number of probes sent in the LSP Ping test.";
  }
  leaf data-size {
    type uint32;
    description
      "Specifies the size of the data portion to
       be transmitted in an LSP Ping operation, in octets.";
  }
  leaf data-fill {
    type string{
      length "0..1564";
    }
    description
      "Used together with the corresponding
       data-size value to determine how to fill the data
       portion of a probe packet.";
  }
  leaf description {
    type string{
```

```
        length "1..31";
    }
    description "A descriptive name of the LSP Ping test";
}
leaf source-address {
    type inet:ip-address;
    description "Specifies the source address";
}
leaf ttl {
    type uint8;
    default 255;
    description "Time to live";
}
choice outbound {
    case interface {
        leaf interface-name {
            type string {
                length "1..255";
            }
            description "Specifies the outgoing interface";
        }
    }
    case nexthop {
        leaf nexthop {
            type inet:ip-address;
            description "Specifies the nexthop";
        }
    }
    description "Specifies the out interface or nexthop";
}
}

container scheduling-parameters {
    description "LSP Ping test schedule parameter";
    choice start-test {
        case now {
            leaf start-test-now {
                type empty;
                description "Start test now";
            }
        }
        case at {
            leaf start-test-at {
                type yang:date-and-time;
                description "Start test at a specific time";
            }
        }
        case delay {
```



```
    leaf start-test-delay {
        type uint32;
        description "Start after a specific delay";
    }
    leaf start-test-delay-units {
        type units;
        default seconds;
        description "Delay units";
    }
}
case daily {
    leaf start-test-daily {
        type yang:date-and-time;
        description "Start test daily";
    }
}
description
    "Specifies when the test begins to start,
    include 4 schedule method: start now(1), start at(2),
    start delay(3), start daily(4).";
}

choice end-test{
    case at {
        leaf end-test-at{
            type yang:date-and-time;
            description "End test at a specific time";
        }
    }
    case delay {
        leaf end-test-delay {
            type uint32;
            description "End after a specific delay";
        }
        leaf end-test-delay-units {
            type units;
            default seconds;
            description "Delay units";
        }
    }
}
case lifetime {
    leaf end-test-lifetime {
        type uint32;
        description "Set the test lifetime";
    }
    leaf lifetime-units {
        type units;
        default seconds;
    }
}
```

```
        description "Lifetime units";
    }
}
description
    "Specifies when the test ends, include 3
    schedule method: end at(1), end delay(2),
    end lifetime(3).";
}
}

container result-info {
    config "false";
    description "LSP Ping test result information";
    leaf operational-status {
        type operational-status;
        description "Operational state of a LSP Ping test";
    }
    leaf source-address {
        type inet:ip-address;
        description "The source address of the test";
    }
    leaf target-fec-type {
        type target-fec-type;
        description "The Target FEC address type";
    }
    choice target-fec {
        case ip-prefix {
            leaf ip-address {
                type inet:ip-address;
                description "IPv4/IPv6 Prefix";
            }
        }
        case bgp {
            leaf bgp {
                type inet:ip-address;
                description "BGP IPv4/IPv6 Prefix";
            }
        }
        case rsvp {
            leaf tunnel-interface {
                type string;
                description "Tunnel interface";
            }
        }
        case vpn {
            leaf vrf-name {
                type uint32;
                description "Layer3 VPN Name";
            }
        }
    }
}
```

```
    }
    leaf vpn-ip-address {
      type inet:ip-address;
      description "Layer3 VPN IPv4 Prefix";
    }
  }
  case pw {
    leaf vcid {
      type uint32;
      description "VC ID";
    }
  }
  case vpls {
    leaf vsi-name {
      type string;
      description "VPLS VSI";
    }
  }
  description "The Target FEC address";
}
leaf min-rtt {
  type uint32;
  description
    "The minimum LSP Ping round-trip-time (RTT)
    received measured in usec.";
}
leaf max-rtt {
  type uint32;
  description
    "The maximum LSP Ping round-trip-time (RTT)
    received measured in usec.";
}
leaf average-rtt {
  type uint32;
  description
    "The current average LSP Ping round-trip-time
    (RTT) measured in usec.";
}
leaf probe-responses {
  type uint32;
  description
    "Number of responses received for the
    corresponding LSP Ping test.";
}
leaf sent-probes {
  type uint32;
  description
    "Number of probes sent for the
```

```
        corresponding LSP Ping test.";
    }
    leaf sum-of-squares {
        type uint32;
        description
            "The sum of the squares of RTT,
             calculated as the sum of the squared
             differences between each RTT and the overall
             mean RTT, for all replies received.";
    }
    leaf last-good-probe {
        type yang:date-and-time;
        description
            "Date and time when the last response
             was received for a probe.";
    }
}

container probe-results {
    description "Result info of test probes";
    list probe-result {
        key "probe-index";
        description "Result info of each test probe";
        leaf probe-index {
            type uint32;
            config false;
            description "Probe index";
        }
        leaf return-code {
            type uint8;
            config false;
            description "The Return Code set in the echo reply";
        }
        leaf return-sub-code {
            type uint8;
            config false;
            description
                "The Return Sub-code set in the echo reply.";
        }
        leaf rtt {
            type uint32;
            config false;
            description "The round-trip-time (RTT) received";
        }
        leaf result-type {
            type result-type;
            config false;
            description "The probe result type";
        }
    }
}
```

```
    }  
  }  
}  
}  
}  
<CODE ENDS>
```

7. Examples

The following examples show the netconf RPC communication between client and server for one LSP-Ping test case.

7.1. Configuration of Control Information

Configure the control-parameters for sample-test-case.

Request from netconf client:

```
<rpc
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <control-parameters>
            <target-fec-type>ip-prefix</target-fec-type>
            <ip-prefix>2001:db8::1:100/64</ip-prefix>
            <reply-mode>reply-via-udp</reply-mode>
            <timeout>1</timeout>
            <timeout-units>seconds</timeout-units>
            <interval>1</interval>
            <interval-units>seconds</interval-units>
            <probe-count>6</probe-count>
            <admin-status>enabled</admin-status>
            <data-size>64</data-size>
            <data-fill>this is a lsp ping test</data-fill>
            <source-address>2001:db8::4</source-address>
            <ttl>56</ttl>
          </control-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.2. The Configuration of Schedule Parameters

Set the scheduling-parameters for sample-test-case to start the test.

Request from netconf client:

```
<rpc
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <scheduling-parameters>
            <start-test-now/>
          </scheduling-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.3. Display of Result Information

Get the result-info of sample-test-case.

Request from netconf client:

```
<rpc
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <result-info/>
        </lsp-ping>
      </lsp-pings>
    </filter>
  </get>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
<data>
  <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
    <lsp-ping>
      <lsp-ping-name>sample-test-case</lsp-ping-name>
      <result-info>
        <operational-status>completed</operational-status>
        <source-address>2001:db8::4</source-address>
        <target-fec-type>ip-prefix</target-fec-type>
        <ip-prefix>2001:db8::1:100/64</ip-prefix>
        <min-rtt>10</min-rtt>
        <max-rtt>56</max-rtt>
        <average-rtt>36</average-rtt>
        <probe-responses>6</probe-responses>
        <sent-probes>6</sent-probes>
        <sum-of-squares>8882</sum-of-squares>
        <last-good-probe>2015-07-01T10:36:56</last-good-probe>
        <probe-results>
          <probe-result>
            <probe-index>0</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>10</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>1</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>56</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>2</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>35</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>3</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>38</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>4</probe-index>
            <return-code>0</return-code>
          </probe-result>
        </probe-results>
      </result-info>
    </lsp-ping>
  </lsp-pings>
```



```
        <return-sub-code>3</return-sub-code>
        <rtt>36</rtt>
        <result-type>success</result-type>
    </probe-result>
    <probe-result>
        <probe-index>5</probe-index>
        <return-code>0</return-code>
        <return-sub-code>3</return-sub-code>
        <rtt>41</rtt>
        <result-type>success</result-type>
    </probe-result>
</probe-results>
</result-info>
</lsp-ping>
</lsp-pings>
</data>
</rpc-reply>
```

8. Security Considerations

The YANG module specified in this document defines 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 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 an adverse effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can adversely affect the routing subsystem of both the local device and the network. This may lead to corruption of the measurement that may result in false corrective action, e.g., false negative or false positive. That could be, for example, prolonged and undetected

deterioration of the quality of service or actions to improve the quality unwarranted by the real network conditions.

Some of the readable data nodes in this 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can disclose the operational state information of VRRP on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

TBD

The LSP ping YANG module inherits all security consideration of [RFC8029].

9. IANA Considerations

The IANA is requested to assign a new namespace URI from the IETF XML registry.

URI:TBA

Contributors

Yanfeng Zhang

Huawei Technologies

zhangyanfeng@huawei.com

Sam Aldrin

Google

aldrin.ietf@gmail.com

Acknowledgments

TBD

12. References

12.1. 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>>.
- [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>>.
- [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>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [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>>.

12.2. Informative References

- [I-D.mahesh-netconf-persistent] Jethanandani, M., "NETCONF and persistent responses", draft-mahesh-netconf-persistent-00 (work in progress), October 2014.
- [RFC4560] Quittek, J., Ed. and K. White, Ed., "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations", RFC 4560, DOI 10.17487/RFC4560, June 2006, <<https://www.rfc-editor.org/info/rfc4560>>.
- [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>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [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>>.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [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>>.

Authors' Addresses

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Guangying Zheng
Huawei Technologies
China

Email: zhengguangying@huawei.com

Greg Mirsky
ZTE Corp.
USA

Email: gregimirsky@gmail.com

Reshad Rahman
Cisco Systems
Canada

Email: rrahman@cisco.com

Faisal Iqbal
Cisco Systems

Email: faiqbal@cisco.com