CATS Working Group
Internet Draft
Intended status: Standards Track
Expires: March 10, 2026

H. Yao
China Mobile
C. Lin
New H3C Technologies
Z. Li
China Mobile
Q. Xiong
ZTE Corporation
L. M. Contreras
Telefonica

December 1, 2025

Data Model for Computing-Aware Traffic Steering (CATS)
draft-yl-cats-data-model-05

Abstract

This document defines a YANG data model for the configuration and management of Computing-Aware Traffic Steering (CATS) systems.

The YANG module defined in this document conforms to the Network Management Datastore Architecture (NMDA).

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 May 10, 2026.

Table of Contents

1. Introduction

[I-D.ietf-cats-framework] introduces a framework for Computing-Aware Traffic Steering (CATS). This document defines a YANG data model for configuring and managing the CATS system. The model extends the ietf-routing YANG model defined in [RFC8349], is agnostic to specific protocol extensions, and conforms to the framework definition to serve as a critical diagnostic or control mechanism.

1.1. Terminology

The following terms as defined in [I-D.draft-ietf- cats-framework] are used:

CS-ID: CATS Service ID.

CSCI-ID: CATS Service Contact Instance ID.

C-SMA: CATS Service Metric Agent.

C-NMA: CATS Network Metric Agent.

C-PS: CATS Path Selector.

C-TC: CATS Traffic Classifier.

The new defined terms:

C-CIB: CATS Computing Information Base, as shown in Figure 1, responsible for maintaining CATS network computing information. Provides basic data for the C-SMA.

C-NIB: CATS Network Information Base, as shown in Figure 1, responsible for maintaining CATS network information. Provides basic data for the C-NMA.

CATS-SBI: The CATS Southbound Interface, as shown in Figure 1. The interface between the CATS Forwarders and the CATS Control Plane (regardless of the protocols used in the Control Plane). It could be used to report computing metric information from CATS Forwarders to the Control Plane, and also could be used to send path and service policy information or service information from the Control Plane to CATS-Forwarders.

C-SMA API: An extended interface between the Control Plane or Management Plane and the C-SMA, or between CATS-Forwarders and service-instances, as shown in Figure 1, it is used to report service metric information to the Control Plane or CATS-Forwarders.

1.2. Conventions Used in This Document

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.

1.3. Tree Diagrams

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

1.4. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
ip	ietf-ip	[RFC8344]
rt	ietf-routing	[RFC8349]
cats	ietf-cats	Section 3
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

2. CATS System Overview

As defined in [I-D.ietf-cats-framework], the CATS framework structure consists of the C-SMA (responsible for maintaining service metrics), the C-NMA (responsible for maintaining network metrics), the C-PS (responsible for maintaining forwarding table entries), and the C-TC (responsible for traffic classification), as shown in Figure 1.

The C-CIB and C-NIB are defined in the management plane. The CATS-SBI is used to exchange information between CATS Forwarders and the Control Plane. The C-SMA API is an extended interface to report service metric information between the Control Plane and the C-SMA or between CATS-Forwarders and service-instance.

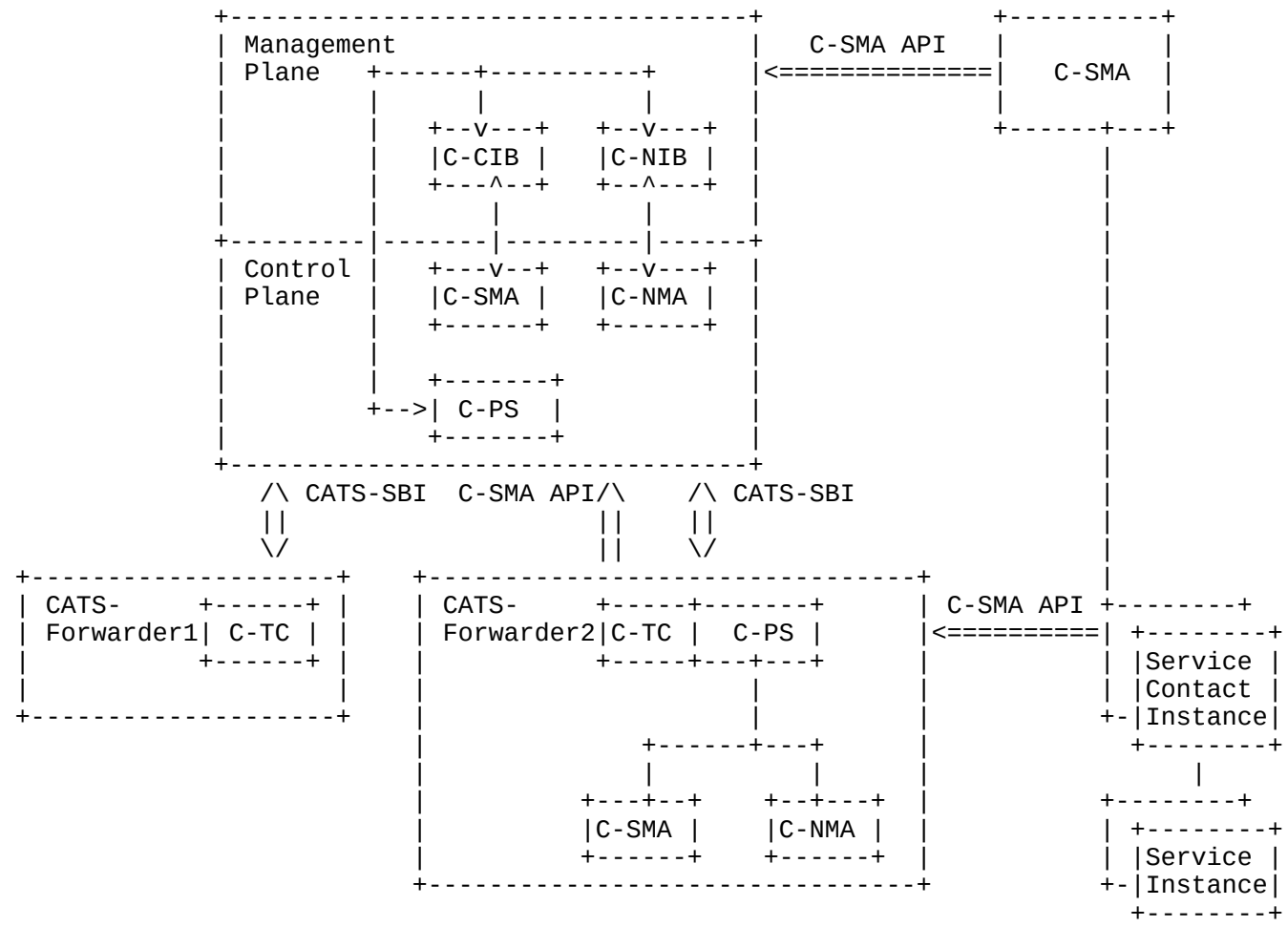

Figure 1: CATS System Architecture

3. Design of CATS Data Model

3.1. Scope of Model

The model covers CATS systems as described in [I-D.ietf-cats-framework]. This model can be used to configure and manage CATS systems.

The operational state data and statistics can be retrieved by this model. The subscription and push mechanism defined in [RFC8639] and [RFC8641] can be implemented by the user to subscribe to notifications on the data nodes in this model.

The model contains all the basic configuration parameters to operate the CATS system. Depending on the implementation choices, some systems might not allow some of the advanced parameters to be configurable.

Occasionally-implemented parameters are modeled as optional features in this model. This model can be extended, and it has been structured in a way that such extensions can be conveniently made.

3.2. Specification

This model imports from and augments the ietf-routing YANG model defined in [RFC8349]. Both configuration branch and state branch of [RFC8349] are augmented. The configuration branch covers node base and policy configuration. The container "cats" is the top level container in this data model.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407].

3.3. Model Structure

This document defines a YANG data model for the configuration and management of CATS corresponding data. The data model is applicable to the CATS-SBI interface and the C-SMA API interface in Figure 1.

The CATS YANG model includes basic CATS objects, traffic-classifier objects, and service-metric objects. The model also includes YANG notifications.

- The CATS base table provides interfaces for the functionality of the C-PS component, which can be used for communication between the Control Plane and the C-SMA, as well as for the interface between the Control Plane and the CATS-Forwarder.

- The CATS traffic-classifier table provides interfaces for the functionality of the C-TC component, which can be used for communication between the Control Plane and the CATS-Forwarder. The Control Plane can distribute directly CATS traffic-classifier table to the CATS-Forwarder, allowing the CATS-Forwarder to proactively select paths according to forwarding policies.

- The CATS service-metric table provides interfaces for the C-SMA component, which can be used for communication between the Control Plane and the CATS-Forwarder, as well as for transmitting service metric information from the C-SMA to the Control Plane. It is also used for forwarding service metric information from the Control Plane to the CATS-Forwarder.

- The CATS notify table is used by the management layer component and can be utilized for the CATS-Forwarder to report events to the Control Plane.

The following is a tree representation of the CATS YANG model:

```
 module: ietf-cats
   augment /rt:routing:
     +--rw cats
     |  +--rw base
     |  |     ...
     |  |
     |  +--rw traffic-classifiers
     |  |     ...
     |  |
     |  +--rw service-metrics
     |  |     ...
     |  |
     |  +--n notify
     |  |    ...
```

Figure 2: Yang Organization and Hierarchy

3.3.1. CATS base

```
        |  +--rw base
        |  |  +--rw enable          bool
        |  |  +--rw update-interval uint32
        |  |  +--rw metric-limits   uint32
        |  |  +--rw flow-limits     uint32
        |  |  +--rw flow-timeout    uint32
        |  |  +--rw service-policy
        |  |     +--rw cs-id        index, type cs-id
        |  |     +--rw policy-type  enumeration
```

Figure 3: Base configuration tree view

In the base, you can set the "enable" attribute to enable/disable CATS function on a per-device basis. When set to false, the CATS function component is shut down.

You can set the "update-interval" parameter to determine the interval at which the C-SMA notifies the C-PS of metric changes. The default value for this interval is 30 seconds.

You can set the maximum number of metric table entries by configuring "metric-limits".

You can set the maximum number of flow tables by configuring "flow-limits".

When flow tables have been inactive for a long period of time, it is necessary to age out the stale state entries. This can be achieved by configuring "flow-timeout" to control the aging time of flow tables.

You can set the service-policy table for traffic steering, which includes affinity-based, service-metric-based, network-metric-based, and combined service-metric and network-metric-based steering.

3.3.2. CATS traffic-classifier

```
      |  +--rw traffic-classifiers
      |  |   +--rw traffic-classifier
      |  |      +--rw cs-id               index,type cs-id
      |  |      +--rw description         string
      |  |      +--rw server-port         uint16
      |  |      +--rw protocol            uint8
```

Figure 4: traffic-classifier tree view

The CATS traffic-classifier table is used to define the features of the service and to classify the traffic.

The "cs-id" is used to uniquely identify a CATS service, and the "description" serves to document its content.

The "server-port" defines the application port for a CATS service. This port number matches the destination port in the IP packets of data flows destined for the service.

The "protocol" defines the application transport protocol for a CATS service. This protocol number matches the protocol in the IP packets of data flows destined for the service.

3.3.3. CATS service-metric

```
|  +--rw service-metrics
|  |   +--rw service-metric
|  |      +--rw cs-id               index, type cs-id
|  |      +--rw csci-id             index, type csci-id
|  |      +--rw source-type         index, source-type
|  |      +--rw priority            uint8
|  |      +--rw affinity            uint8
|  |      +--rw location            inet:ip-address
|  |      +--rw metrics
|  |         +--rw metric
|  |            +--rw metric-type   index, type uint
|  |            +--rw metric-value  uint32
```

Figure 5: Service-metric tree view

The CATS service-metric table is used to control the delivery service metrics on the control plane, thereby generating the forwarding table on the forwarding plane in conjunction with network metrics.

The "cs-id" is used to uniquely identify a CATS service, and the "csci-id" is used to represent an instance of that service. The "source-type" indicates the source protocol of this service metric information, such as a static protocol.

The "priority" indicates the priority of different service instances of the same service, and the "affinity" indicates the affinity between this service and its service site.

The "location" indicates the IP address of the service site where this service instance resides. And the "metrics" indicates different types of service metric values.

3.3.4. CATS notify

```
|  +--n notify
|  |     +--ro entry-limit-reached  boolean
|  |     +--ro flow-limit-reached   boolean
```

Figure 6: Notify tree view

When the number of CATS table entries reaches the maximum and when the number of entries goes from maximum to not being maximum, an event notification will be sent indicating the change.

4. CATS YANG Module

<CODE BEGINS> file "ietf-cats@2025-11-05.yang"

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

  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing
       Management (NMDA Version)";
  }

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

  organization
    "IETF Computing-Aware Traffic Steering
     (CATS) Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/group/cats/>
     WG List: <mailto:cats@ietf.org>
     Author:  Huijuan Yao
              <mailto:yaohuijuan@chinamobile.com>
     Author:  Changwang Lin
              <mailto:linchangwang.04414@h3c.com>
     Author:  Zhenqiang Li
              <mailto:lizhenqiang@chinamobile.com>
     Author:  Quan Xiong
              <mailto:xiong.quan@zte.com.cn>
     Author:  Luis M. Contreras
              <mailto:luismiguel.contrerasmurillo@telefonica.com>
     ";

  description
    "This module describes a YANG model for CATS.
     This YANG model conforms to the Network Management
     Datastore Architecture (NMDA) as described in RFC 8342.

     Copyright (c) 2025 IETF Trust and the persons identified as
     authors of the code. All rights reserved.
```

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

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

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
        'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT
        RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to
        be interpreted as described in BCP 14 (RFC 2119) (RFC 8174)
        when, and only when, they appear in all capitals, as shown
        here.";

    revision 2025-11-05 {
      description
        "Initial Version";
      reference
        "RFC XXXX: YANG Data Model for CATS";
     }

    typedef cs-id {
      type union {
        type uint32;
        type inet:ipv4-address;
        type inet:ipv6-address;
      }
      description
        "This type is for CATS CS-ID.";
      reference
        "RFC XXXX: YANG Data Model for CATS";
    }

    typedef csci-id {
      type union {
        type uint32;
        type inet:ipv4-address;
        type inet:ipv6-address;
      }
      description
        "This type is for CATS CSCI-ID.";
      reference
        "RFC XXXX: YANG Data Model for CATS";
    }
```

```
    grouping service-policy-type {
      description
        "service policy type";
      leaf policy-type {
        type enumeration {
          enum base-on-affinity {
            value 0;
            description "base on affinity";
          }
          enum base-on-sm-only {
            value 1;
            description "base on service metric only";
          }
          enum base-on-nm-only {
            value 2;
            description "base on network metric only";
          }
          enum base-on-sm-nm {
            value 3;
            description
              "base on the combination of service metric and network
               metric";
          }
        }
        description "service policy type";
      }
    }

    augment "/rt:routing" {
      description
        "This augments the CATS configuration.";
      container cats {
        description
          "Support for the CATS configuration.";
        container base {
          description
            "CATS base configuration.";
          leaf enable {
            type boolean;
            description
              "enable CATS";
          }
          leaf update-interval {
            type uint32;
            units "seconds";
            default "30";
            description
```

```
            "Update interval of CATS metric. A value of 0
             indicates that metrics are sent only when a
             significant change occurs and not on a periodic
             basis.";
          }
          leaf metric-limits {
            type uint32;
            description
              "CATS metric entry limit";
          }
          leaf flow-limits {
            type uint32;
            description
              "CATS flow entry limit";
          }
          leaf flow-timeout {
            type uint32;
            units "seconds";
            default "120";
            description
              "CATS flow timeout when no flow. A value of 0 means
               that the flow does not time out due to inactivity
               (i.e., infinite timeout), and must be terminated
               explicitly (e.g., via signaling or administrative
               action).";
          }
          container service-policy {
            description
              "CATS service policy";
            leaf cs-id {
              type cs-id;
              description
                "cs-id";
            }
            uses service-policy-type;
          }
        }

        container traffic-classifiers {
          description
            "CATS traffic-classifier feature";
          list traffic-classifier {
            key "cs-id";
            description
              "CATS traffic-classifier feature";
            leaf cs-id {
              type cs-id;
              description
```

```
                "CATS CS-ID";
            }
            leaf description {
              type string;
              description
                "description of this service, example: http";
            }
            leaf server-port {
              type uint16;
              description
                "server-port of the service, matches the
                 destination port in the IP packets of data
                 flows destined for the service.";
            }
            leaf protocol {
              type uint8;
              description
                "protocol of the service.";
            }
          }
        }

        container service-metrics {
          description
            "CATS service metric entry";
          list service-metric {
            key "cs-id csci-id source-type";
            description
              "CATS service metric entry";
            leaf cs-id {
              type cs-id;
              description
                "CATS cs-id";
            }
            leaf csci-id {
              type csci-id;
              description
                "CATS csci-id";
            }
            leaf source-type {
              type enumeration {
                enum static {
                  value 0;
                  description "static configuration";
                }
              }
              description
                "source-type of the service compute-metric";
```

```
            }
            leaf priority {
              type uint8;
              description
                "server priority, the smaller the value,
                 the higher the priority.";
            }
            leaf affinity {
              type uint8;
              description
                "server affinity, the higher the value,
                 the higher the affinity.";
            }
            leaf location {
              type inet:ip-address;
              description
                "server location";
            }
            container metrics {
              description
                "service metric";
              list metric {
                key "metric-type";
                description "Different types of service.";
                leaf metric-type {
                  type enumeration {
                    enum delay {
                      value 0;
                      description
                        "Calculate the metric based on transmission
                         delay, where the metric value is the delay
                         time in milliseconds.";
                    }
                    enum service-ratio {
                      value 1;
                      description
                        "Calculate the metric based on business
                         capacity, where the metric value is the
                         current capacity percentage.";
                    }
                    enum memory-ratio {
                      value 2;
                      description
                        "Calculate the metric based on memory
                         utilization percentage, where the metric
                         value is the current memory utilization
                         percentage";
                    }
```

```
                    }
                    description "metric type";
                    reference
                      "RFC YYYY: CATS Metrics Definition";
                  }
                  leaf metric-value {
                    type uint32;
                    description "metric value";
                  }
                }
              }
            }
          }
          container notify {
            description "event notify";
            leaf entry-limit-reached {
              type boolean;
              description "entry limit reached";
            }
            leaf flow-limit-reached {
              type boolean;
              description "flow entry limit reached";
            }
          }
        }
      }
    }
    <CODE ENDS>
```

5. Security Considerations

The YANG module "ietf-cats" defines a data model designed to be accessed via YANG-based management protocols such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] 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 the ietf-cats.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:

* /cats:base/cats:enable

* /cats:base/cats:update-interval

* /cats:base/cats:metric-limits

* /cats:base/cats:flow-limits

* /cats:base/cats:flow-timeout

* /cats:base/cats:service-policy/cats:cs-id

* /cats:base/cats:service-policy/cats:policy-type

* /cats:traffic-classifiers/cats:traffic-classifier/cats:cs-id

* /cats:traffic-classifiers/cats:traffic-classifier /cats:description

* /cats:traffic-classifiers/cats:traffic-classifier /cats:server-port

* /cats:traffic-classifiers/cats:traffic-classifier/cats:protocol

* /cats:service-metrics/cats:service-metric/cats:cs-id

* /cats:service-metrics/cats:service-metric/cats:csci-id

* /cats:service-metrics/cats:service-metric/cats:source-type

* /cats:service-metrics/cats:service-metric/cats:priority

* /cats:service-metrics/cats:service-metric/cats:affinity

* /cats:service-metrics/cats:service-metric/cats:location

* /cats:service-metrics/cats:service-metric/cats:metrics /cats:metric/cats:metric-type

* /cats:service-metrics/cats:service-metric/cats:metrics /cats:metric/cats:metric-value

For these augmentations to ietf-routing.yang, the ability to delete, add, and modify CATS base configuration, traffic classifier

configuration and service metric configuration would allow traffic to be misrouted.

Some of the readable data nodes in the ietf-cats.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:

* /cats:notify/cats:entry-limit-reached

* /cats:notify/cats:flow-limit-reached

Exposing the notification will expose the CATS table information of the network device. This may be undesirable due to the fact that such exposure may facilitate other attacks. Additionally, network operators may consider their CATS table information to be sensitive confidential data.

All the security considerations for writable and readable data nodes defined in [RFC8349] apply to the augmentations described herein.

6. IANA Considerations

The IANA is requested to assign the following URI in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-cats
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

This document registers the following YANG module in the "YANG Module Names" registry [RFC6020]:

```
Name:       ietf-cats
Maintained by IANA?  N
Namespace:  urn:ietf:params:xml:ns:yang:ietf-cats
Prefix:     cats
Reference:  RFC XXXX
```

7. References

7.1. 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>.

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[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>.

[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>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <https://www.rfc-editor.org/info/rfc8344>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

7.2. Informative References

[I-D.ietf-cats-framework] C. Li.,Z. Du.,M. Boucadair.,L. M. Contreras., J. Drake., "A Framework for Computing-Aware Traffic Steering (CATS)", draft-ietf-cats-framework-13(work in progress), September 2025.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[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>.

[RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <https://www.rfc-editor.org/info/rfc7895>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[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>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[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>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Contributors

TBD

Authors' Addresses

Huijuan Yao
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: yaohuijuan@chinamobile.com

Changwang Lin
New H3C Technologies
Beijing
China
Email: linchangwang.04414@h3c.com

Zhenqiang Li
China Mobile
China
Email: lizhenqiang@chinamobile.com

Quan Xiong
ZTE Corporation
Email: xiong.quan@zte.com.cn

Luis M. Contreras
Telefonica
Spain
Email: luismiguel.contrerasmurillo@telefonica.com