

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 1, 2018

A. Choudhary
Cisco Systems
I. Chen
Kuatro Technologies
October 28, 2017

YANG Model for QoS Operational Parameters
draft-asechoud-rtgwg-qos-oper-model-00

Abstract

This document describes a YANG model for Quality of Service (QoS) operational parameters.

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 1, 2018.

Copyright Notice

Copyright (c) 2017 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
2. Terminology	2
3. QoS Operational Model Design	2
4. Modules Tree Structure	3
5. Modules	5
5.1. IETF-QOS-OPER	5
6. Security Considerations	11
7. Revision Tracking	11
8. Acknowledgement	11
9. Normative References	11
Authors' Addresses	12

1. Introduction

This document defines a base YANG [RFC6020] data module for Quality of Service (QoS) operational parameters. Remote Procedure Calls (RPC) or notification definition is currently not part of this document and will be added later if necessary. QoS configuration modules are defined by draft-asechoud-rtgwg-qos-model.

This document doesn't include operational parameters for random-detect (RED, which is left to individual vendor to augment it.

2. 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].

3. QoS Operational Model Design

QoS operational model include QoS policy applied to an interface in each direction of traffic. For each QoS policy applied to an interface the model further includes counters for associated Classifiers, Meters and Queues in a particular direction. To modularize and for reusability, grouping have been defined for various counters of Classifier, Meters and Queues. The target is assumed to be interface but the groupings can be used for any other target type where QoS policy is applied.

draft-asechoud-rtgwg-qos-model defines various building blocks for applying a QoS Policy on a target. It includes QoS Policy configuration, which is a container of various classifiers and corresponding actions which are configured for traffic conditioning. This drafts defines the various counters for these building blocks.

Classifier statistics contains counters for packets and bytes matched to the traffic in a direction and also average rate at which traffic is hitting a classifier.

Statistics of meter is modeled based on commonly used algorithms in industry, Single Rate Tri Color Marking (srTCM) [RFC2697] meter, Two Rate Tri Color Marking (trTCM) [RFC2698] meter. Metering statistics includes counters corresponding to various rates configured. A metering container is referred by a metering identifier. This identifier could be a classifier name if the metering configuration is inline with classifier or it could be metering template name if the metering is configured as separate entity and associated with the classifier.

Queuing statistics includes counters corresponding to various queues associated with the policy. A queuing container is referred by queuing identifier. This identifier could be a classifier name if the queuing configuration is inline with classifier and hence there is one-to-one mapping between a classifier and a queue or it could be a separate queue identifier if one or more than one classifiers are associated with a queue.

4. Modules Tree Structure

This document defines counters for classifiers, meters and queues.

Classifier statistics consists of list of classifier entries identified by a classifier entry name. Classifier counters include matched packets, bytes and average rate of traffic matching a particular classifier.

Metering statistics consists of meters identified by an identifier. Metering counters include conform, exceed, violate and drop packets and bytes.

Queuing counters include instantaneous, peak, average queue length, as well as output conform, exceed, tail drop packets and bytes.

```

module: ietf-qos-oper
augment /if:interfaces/if:interface:
  +--ro qos-classifier-statistics*
    |   +--ro policy-name?          string
    |   +--ro direction?           identityref
    |   +--ro classifier-entry-name? string
    |   +--ro classifier-entry-statistics
    |     +--ro classified-pkts?    uint64
    |     +--ro classified-bytes?   uint64
    |     +--ro classified-rate?    uint64
  +--ro qos-named-statistics*
    |   +--ro stats-name?    string
    |   +--ro pkts?          uint64
    |   +--ro bytes?         uint64
    |   +--ro rate?          uint64
  +--ro metering-statistics*
    |   +--ro policy-name?    string
    |   +--ro direction?     identityref
    |   +--ro meter-id?      string
    |   +--ro conform-pkts?   uint64
    |   +--ro conform-bytes?  uint64
    |   +--ro conform-rate?   uint64
    |   +--ro exceed-pkts?    uint64
    |   +--ro exceed-bytes?   uint64
    |   +--ro exceed-rate?    uint64
    |   +--ro violate-pkts?   uint64
    |   +--ro violate-bytes?  uint64
    |   +--ro violate-rate?   uint64
    |   +--ro meter-drop-pkts? uint64
    |   +--ro meter-drop-bytes? uint64
  +--ro queueing-statistics*
    |   +--ro policy-name?    string
    |   +--ro direction?     identityref
    |   +--ro queue-id?      string
    |   +--ro queueing-statistics
    |     +--ro output-conform-pkts?    uint64
    |     +--ro output-conform-bytes?    uint64
    |     +--ro output-exceed-pkts?     uint64
    |     +--ro output-exceed-bytes?    uint64
    |     +--ro queue-current-size-bytes? uint64
    |     +--ro queue-average-size-bytes? uint64
    |     +--ro queue-peak-size-bytes?   uint64
    |     +--ro tailed-drop-pkts?       uint64
    |     +--ro tailed-drop-bytes?       uint64

```

5. Modules

5.1. IETF-QOS-OPER

```

<CODE BEGINS>file "ietf-qos-oper.yang"

module iETF-qos-oper {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-qos-oper";
  prefix oper;

  import iETF-interfaces {
    prefix if;
  }

  organization
    "IETF RTG (Routing Area) Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/rtgwg/>
    WG List: <mailto:rtgwg@ietf.org>
    WG Chair: Chris Bowers
              <mailto:cbowers@juniper.net>
    WG Chair: Jeff Tantsura
              <mailto:jefftant.ietf@gmail.com>
    Editor: Aseem Choudhary
            <mailto:asechoud@cisco.com>";
  description
    "This module contains a collection of YANG definitions for
    configuring qos specification implementations.
    Copyright (c) 2014 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 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.";

  revision 2017-10-28 {
    description
      "Latest revision qos based policy applied to a target";
    reference "RFC XXXX";
  }

  identity direction {
    description

```

```

        "This is identity of traffic direction";
    }

    identity inbound {
        base direction;
        description
            "Direction of traffic coming into the network entry";
    }

    identity outbound {
        base direction;
        description
            "Direction of traffic going out of the network entry";
    }

    grouping classifier-entry-stats {
        description
            "Classifier Counters";
        container classifier-entry-statistics {
            config false;
            description
                "
                This group defines the classifier filter statistics of
                each classifier entry
                ";
            leaf classified-pkts {
                type uint64;
                description
                    " Number of total packets which filtered
                    to a classifier-entry";
            }
            leaf classified-bytes {
                type uint64;
                description
                    " Number of total bytes which filtered
                    to a classifier-entry";
            }
            leaf classified-rate {
                type uint64;
                units "bits-per-second";
                description
                    " Rate of average data flow through a
                    classifier-entry";
            }
        }
    }

    grouping qos-named-stats {

```

```

description
  "QoS matching statistics associated with a stats-name";
leaf pkts {
  type uint64;
  description
    " Number of total matched packets associated
      to a statistics name";
}
leaf bytes {
  type uint64;
  description
    " Number of total matched bytes associated
      to a statistics name";
}
leaf rate {
  type uint64;
  units "bits-per-second";
  description
    " Rate of average matched data which is associated
      to a statistics name";
}
}

```

```

grouping queuing-stats {
  description
    "Queuing Counters";
  container queuing-statistics {
    description
      "queue related statistics ";
    leaf output-conform-pkts {
      type uint64;
      description
        "Number of packets transmitted from queue ";
    }
    leaf output-conform-bytes {
      type uint64;
      description
        "Number of bytes transmitted from queue ";
    }
    leaf output-exceed-pkts {
      type uint64;
      description
        "Number of packets transmitted from queue ";
    }
    leaf output-exceed-bytes {
      type uint64;
      description
        "Number of bytes transmitted from queue ";
    }
  }
}

```

```

    }
    leaf queue-current-size-bytes {
        type uint64;
        description
            "Number of bytes currently buffered ";
    }
    leaf queue-average-size-bytes {
        type uint64;
        description
            "Average queue size in number of bytes";
    }
    leaf queue-peak-size-bytes {
        type uint64;
        description
            "Peak buffer queue size in bytes ";
    }
    leaf tailed-drop-pkts {
        type uint64;
        description
            "Total number of packets tail-dropped ";
    }
    leaf tailed-drop-bytes {
        type uint64;
        description
            "Total number of bytes tail-dropped ";
    }
}

grouping meter-stats {
    description
        "Metering Statistics";
    leaf conform-pkts {
        type uint64;
        description
            "Number of conform packets";
    }
    leaf conform-bytes {
        type uint64;
        description
            "Bytes of conform packets";
    }
    leaf conform-rate {
        type uint64;
        units "bits-per-second";
        description
            "Traffic Rate measured as conforming";
    }
}

```



```

    leaf exceed-pkts {
        type uint64;
        description
            "Number of packets counted as exceeding";
    }
    leaf exceed-bytes {
        type uint64;
        description
            "Bytes of packets counted as exceeding";
    }
    leaf exceed-rate {
        type uint64;
        units "bits-per-second";
        description
            "Traffic Rate measured as exceeding";
    }
    leaf violate-pkts {
        type uint64;
        description
            "Number of packets counted as violating";
    }
    leaf violate-bytes {
        type uint64;
        description
            "Bytes of packets counted as violating";
    }
    leaf violate-rate {
        type uint64;
        units "bits-per-second";
        description
            "Traffic Rate measured as violating";
    }
    leaf meter-drop-pkts {
        type uint64;
        description
            "Number of packets dropped by meter";
    }
    leaf meter-drop-bytes {
        type uint64;
        description
            "Bytes of packets dropped by meter";
    }
}

augment "/if:interfaces/if:interface" {
    description
        "Augments Qos Target Entry to Interface module";
    list qos-classifier-statistics {

```

```

    config false;
    description
        "Statistics for each Classifier Entry in a Policy applied
        in a particular direction";
    leaf policy-name {
        type string;
        description
            "Policy entry name";
    }
    leaf direction {
        type identityref {
            base direction;
        }
        description
            "Direction fo the traffic flow either inbound or outbound";
    }
    leaf classifier-entry-name {
        type string;
        description
            "Classifier Entry Name";
    }
    uses classifier-entry-stats;
}
list qos-named-statistics {
    config false;
    description
        "Matched Statistics for a statistics-name";
    leaf stats-name {
        type string;
        description
            "Statistics name";
    }
    uses qos-named-stats;
}
list metering-statistics {
    config false;
    description
        "Statistics for each Meter associated with the Policy";
    leaf policy-name {
        type string;
        description
            "Policy entry name";
    }
    leaf direction {
        type identityref {
            base direction;
        }
        description

```

```
        "Direction fo the traffic flow either inbound or outbound";
    }
    leaf meter-id {
        type string;
        description
            "Meter Identifier";
    }
    uses meter-stats;
}
list queueing-statistics {
    config false;
    description
        "Statistics for each Queue associated with the Policy";
    leaf policy-name {
        type string;
        description
            "Policy entry name";
    }
    leaf direction {
        type identityref {
            base direction;
        }
        description
            "Direction fo the traffic flow either inbound or outbound";
    }
    leaf queue-id {
        type string;
        description
            "Queue Identifier";
    }
    uses queueing-stats;
}
}
```

<CODE ENDS>

6. Security Considerations
7. Revision Tracking
8. Acknowledgement
9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, DOI 10.17487/RFC2697, September 1999, <<https://www.rfc-editor.org/info/rfc2697>>.
- [RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, DOI 10.17487/RFC2698, September 1999, <<https://www.rfc-editor.org/info/rfc2698>>.
- [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>>.

Authors' Addresses

Aseem Choudhary
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
US

Email: asechoud@cisco.com

Ing-Wher Chen
Kuatro Technologies

Email: ichen@kuatrotech.com