

RTGWG  
Internet-Draft  
Intended status: Standards Track  
Expires: April 22, 2017

I. Chen  
Kuatro Technologies  
October 19, 2016

YANG Data Model for QoS  
draft-chen-rtgwg-qos-yang-03

## Abstract

This document defines a YANG data model to describe quality-of-service (QoS) configurations.

## 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 <http://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 April 22, 2017.

## Copyright Notice

Copyright (c) 2016 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 (<http://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.

Internet-Draft

QoS YANG

October 2016

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Tree Diagram . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Design of the Data Model . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Related Work . . . . .	<a href="#">4</a>
<a href="#">4.</a>	YANG Module . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Security Consideration . . . . .	<a href="#">55</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">55</a>
<a href="#">7.</a>	Acknowledgments . . . . .	<a href="#">55</a>
<a href="#">8.</a>	References . . . . .	<a href="#">55</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">55</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">56</a>
	Author's Address . . . . .	<a href="#">56</a>

[1.](#) Introduction

This document defines a YANG [[RFC7950](#)] data model to specify Quality-of-Service (QoS) configurations. QoS configurations include policies and profiles that allow an administrator of a device to classify, mark, rate-limit, and perform other actions on incoming and outgoing packets.

[1.1.](#) Tree Diagram

A simplified graphical representation of the data model is presented in [Section 2](#).

The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node and "\*" denotes a "list" or "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

o Ellipsis ("...") stands for contents of subtrees that are not shown.

## [2.](#) Design of the Data Model

This data model intends to describe the processing that a device performs on incoming and outgoing packets. These actions include, but are not limited to, classification of packets, mapping of packets, and rate limiting, scheduling and queuing of packets.

This data model has one top level container, qos, which consists of several second level containers. Each of these second level containers describes a particular category of traffic handling, such as mapping traffic from one class to another class. A second-level container further consists of a list of different permutations of options within a traffic processing category.

For example, the "class-maps" processing category below consists of a list of "class-map", which can be used to describe the mapping of a certain type of incoming traffic to a certain differentiated service (DS) codepoint [[RFC2475](#)] such that the device handles that type of incoming traffic as prescribed by the classification [[RFC2474](#)].

```
module: ietf-qos
  +--rw qos
    +--rw class-definitions
      |   +--rw class-definition* [name]
      |   |   +--rw name      string
      |   |   .
      |   |   .
      |   |   .
    +--rw class-maps
      |   +--rw class-map* [name]
      |   |   +--rw name      string
      |   |   .
      |   |   .
      |   |   .
    +--rw congestion-avoidance-maps
      |   +--rw congestion-avoidance-map* [name]
      |   |   +--rw name      string
      |   |   .
      |   |   .
      |   |   .
    +--rw profiles
      |   +--rw profile* [name]
      |   |   +--rw name      string
      |   |   .
      |   |   .
      |   |   .
    +--rw queue-maps
      |   +--rw queue-map* [name]
      |   |   +--rw name      string
```

```

|      .
|      .
|      .
+--rw policies
    +--rw policy* [name]
        +--rw name string
        .
        .
        .

```

### 3. Related Work

An important aspect of QoS configurations involves the pattern of specifying a set of conditions, where matching a particular condition results in one or more actions. This is a common pattern, and it is seen in /qos/policies branch in [Section 2](#), as well as several other proposed YANG models [[aclyang](#)] [[pbfyang](#)] [[supayang](#)]. Another example of a YANG module that models such a condition-action pattern is the YANG module `ietf-qos-policy@2016-03-03.yang` [[dsyang](#)], which

prescribes a high-level but basic QoS policy as a set of conditions and associated actions, but does not include details of the conditions or the actions.

Besides the condition-action pattern, the other similarity between the YANG model proposed in this document and `ietf-qos-policy@2016-03-03.yang` is that both claim to model QoS. Consequently, the authors of the two documents have put in contact to understand the similarities and differences between the two YANG models. While there are no resolutions yet, the goal of all involved is to eliminate unnecessary duplication and design one coherent QoS model (or more if needed).

### 4. YANG Module

```

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

module ietf-qos {
  namespace "urn:ietf:params:xml:ns:yang:ietf-qos";
  prefix qos;

  organization "IETF when standardized";

```

```

contact "IETF when standardized";

description
  "QOS YANG Model";

revision "2015-06-04" {
  description "Initial revision";
  reference "";
}

typedef dscp-code-all-values-type {
  type union {
    type uint8 {
      range "0..63";
    }
    type enumeration {
      enum "af11" {
        description
          "DSCP 10";
      }
      enum "af12" {
        description
          "DSCP 12";
      }
      enum "af13" {
        description

```

```

      "DSCP 14";
    }
    enum "af21" {
      description
        "DSCP 18";
    }
    enum "af22" {
      description
        "DSCP 20";
    }
    enum "af23" {
      description
        "DSCP 22";
    }
    enum "af31" {

```

```

        description
            "DSCP 26";
    }
    enum "af32" {
        description
            "DSCP 28";
    }
    enum "af33" {
        description
            "DSCP 30";
    }
    enum "af41" {
        description
            "DSCP 34";
    }
    enum "af42" {
        description
            "DSCP 36";
    }
    enum "af43" {
        description
            "DSCP 38";
    }
    enum "cs0" {
        description
            "Class selector 000 000 (0)";
    }
    enum "cs1" {
        description
            "Class Selector 001 000 (8)";
    }
    enum "cs2" {
        description

```

```

        "Class Selector 010 000 (16)";
    }
    enum "cs3" {
        description
            "Class Selector 011 000 (24)";
    }
    enum "cs4" {
        description

```

```

        "Class Selector 100 000 (32)";
    }
    enum "cs5" {
        description
            "Class Selector 101 000 (40)";
    }
    enum "cs6" {
        description
            "Class Selector 110 000 (48)";
    }
    enum "cs7" {
        description
            "Class Selector 111 000 (56)";
    }
    enum "df" {
        description
            "Default forwarding 000 000 (0)";
    }
    enum "ef" {
        description
            "Expedited forwarding 101 110 (46)";
    }
    enum "all" {
        description
            "All classes";
    }
    }
}
description "dscp-code-all-values-type";
}
grouping radius-guided-grp {
    description "radius-guided-grp";
    leaf radius-guided {
        type empty;
        description "radius-guided";
    }
}

grouping mdrdrr-congestion-avoidance-map-grp {
    description "mdrrdrr-congestion-avoidance-map-grp";
}

```

```

leaf congestion-avoidance-map-reference {

```



```

    type leafref {
      path "/qos:qos/qos:congestion-avoidance-maps"
        + "/qos:congestion-avoidance-map/qos:name";
    }
    must
      "/qos:qos/qos:congestion-avoidance-maps" +
      "/qos:congestion-avoidance-map/" +
      "[qos:name = current()]/qos:mdrr" {
        description "Must be an mdrr congestion avoidance map";
      }
    description "congestion-avoidance-map-reference";
  }
}

grouping pwfq-congestion-avoidance-map-grp {
  description "pwfq-congestion-avoidance-map-grp";
  leaf congestion-avoidance-map-reference {
    type leafref {
      path "/qos:qos/qos:congestion-avoidance-maps"
        + "/qos:congestion-avoidance-map/qos:name";
    }
    must
      "/qos:qos/qos:congestion-avoidance-maps/" +
      "/qos:congestion-avoidance-map/" +
      "[qos:name = current()]/qos:pwfq" {
        description "Must be a pwfq congestion avoidance map";
      }
    description "congestion-avoidance-map-reference";
  }
}

grouping queue-map-grp {
  description "queue-map-grp";
  leaf queue-map-reference {
    type leafref {
      path "/qos:qos/qos:queue-maps/qos:queue-map/qos:name";
    }
    description "queue-map-reference";
  }
}

grouping rate-option-grp {
  description "rate-option-grp";
  choice burst-option {
    description "burst-option";
    case burst {
      leaf burst {

```

```
        type uint32 {
            range "1..4250000000";
        }
        units "byte";
        description "burst";
    }
}
case time-burst {
    leaf time-burst {
        type uint32 {
            range "1..10000";
        }
        units "millisecond";
        description "time-burst";
    }
}
}
choice excess-burst-option {
    description "excess-burst-option";
    case excess-burst {
        leaf excess-burst {
            type uint32 {
                range "1..4250000000";
            }
            units "byte";
            description "excess-burst";
        }
    }
    case time-excess-burst {
        leaf time-excess-burst {
            type uint32 {
                range "1..10000";
            }
            description "time-excess-burst";
        }
    }
}
}
}

grouping rate-counters-grp {
    description "rate-counters-grp";
    choice counter-choice {
        description "counter-choice";
        case counters {
            container counters {
                presence "";
            }
        }
    }
}
```

```
description "counters";
container hierarchical-counters {
```

```
    presence "";
    description "hierarchical-counters";
    leaf dual-stack {
        type empty;
        description "dual-stack";
    }
}
}
}
case hierarchical-counters {
    container hierarchical-counters {
        presence "";
        description "hierarchical-counters";
        leaf dual-stack {
            type empty;
            description "dual-stack";
        }
    }
}
}
}
}

grouping rate-percentage-grp {
    description "rate-percentage-grp";
    leaf percentage {
        type uint8 {
            range "1..100";
        }
        units "percent";
        description "percentage";
    }
}

grouping marking-priority-detail-grp {
    description "marking-priority-detail-grp";
    choice marking-priority-detail {
        description "mark-priority-detail";
        case drop-precedence {
            leaf drop-precedence {
```

```

    type union {
        type uint8 {
            range "1..7";
        }
        type enumeration {
            enum "ignore" {
                description "ignore";
            }
        }
    }

```

```

    }
    description "drop-precedence";
}
}
case af-drop {
    leaf af-drop {
        type uint8 {
            range "1..3";
        }
        description "af-drop";
    }
}
}
}
}

grouping marking-grp {
    description "marking-grp";
    choice marking {
        description "marking";
        case dscp {
            leaf mark-dscp {
                type dscp-code-all-values-type;
                description "mark-dscp";
            }
        }
    }
    case precedence {
        leaf mark-precedence {
            type uint8 {
                range "1..3";
            }
            description "mark-precedence";
        }
    }
}

```

```

    }
    case priority {
        leaf mark-priority {
            type union {
                type uint8 {
                    range "1..7";
                }
                type enumeration {
                    enum "ignore" {
                        description "ignore";
                    }
                }
            }
            description "mark-priority";
        }
        uses marking-priority-detail-grp;
    }

```

Chen

Expires April 22, 2017

[Page 11]

---

Internet-Draft

QoS YANG

October 2016

```

    }
  }
}

grouping rate-conform-grp {
  description "rate-conform-grp";
  container conform-handling {
    description "conform-handling";
    choice options {
      description "options";
      case mark-packet {
        uses marking-grp;
      }
      case no-action {
        leaf no-action {
          type empty;
          description "no-action";
        }
      }
    }
  }
}

grouping rate-exceed-grp {
  description "rate-exceed-grp";
}

```

```

container exceed-handling {
  description "exceed-handling";
  choice options {
    description "options";
    case drop-packet {
      leaf drop-packet {
        type empty;
        description "drop-packet";
      }
    }
    case mark-packet {
      uses marking-grp;
    }
    case no-action {
      leaf no-action {
        type empty;
        description "no-action";
      }
    }
  }
}

```

```

grouping rate-violate-grp {
  description "rate-violate-grp";
  container violate-handling {
    description "violate-handling";
    choice options {
      description "options";
      case drop-packet {
        leaf drop-packet {
          type empty;
          description "drop-packet";
        }
      }
      case mark-packet {
        uses marking-grp;
      }
      case no-action {
        leaf no-action {
          type empty;
        }
      }
    }
  }
}

```

```

        description "no-action";
    }
}
}
}
}

grouping policy-packet-handling-grp {
    description "policy-packet-handling-grp";
    choice policy-packet-handling {
        description "policy-packet-handling";
        case mark-dscp {
            leaf mark-dscp {
                type dscp-code-all-values-type;
                description "mark-dscp";
            }
        }
        case mark-precedence {
            leaf mark-precedence {
                type uint8 {
                    range "1..3";
                }
                description "mark-precedence";
            }
        }
        case mark-priority {
            leaf mark-priority {
                type union {
                    type uint8 {
                        range "1..7";
                    }
                }
            }
        }
    }
}

```

```

    }
    type enumeration {
        enum "ignore" {
            description "ignore";
        }
    }
}
description "mark-priority";
}
uses marking-priority-detail-grp;
}

```

```

case rate-limiting {
  choice rate-type {
    description "rate-type";
    case bit-rate-type {
      container bit-rate {
        description "bit-rate";
        uses rate-option-grp;
        uses rate-counters-grp;
        container actions {
          description "actions";
          uses rate-conform-grp;
          uses rate-exceed-grp;
        }
      }
    }
    case percentage-type {
      container percentage {
        description "percentage";
        leaf percentage {
          type uint8 {
            range "1..100";
          }
          units "percent";
          description "percentage";
        }
        uses rate-counters-grp;
      }
    }
  }
}

grouping inline-rate-specifications-packet-handling-grp {
  description "inline-rate-specifications-packet-handling-grp";
  choice inline-rate-specifications {
    description "inline-rate-specifications";

```

```

case mark-dscp {
  leaf mark-dscp {
    type dscp-code-all-values-type;
    description "mark-dscp";

```



```

    }
  }
  case mark-precedence {
    leaf mark-precedence {
      type uint8 {
        range "1..3";
      }
      description "mark-precedence";
    }
  }
  case mark-priority {
    leaf mark-priority {
      type union {
        type uint8 {
          range "1..7";
        }
        type enumeration {
          enum "ignore" {
            description "ignore";
          }
        }
      }
      description "mark-priority";
    }
    uses marking-priority-detail-grp;
  }
  case rate-limiting {
    container rate {
      description "Specify rate limits inline";
      uses rate-option-grp;
      uses rate-counters-grp;
      container actions {
        description "actions";
        uses rate-conform-grp;
        uses rate-exceed-grp;
        uses rate-violate-grp;
      }
    }
  }
}

grouping policy-card-family-1-3-bit-rate-grp {
  description "policy-card-family-1-3-bit-rate-grp";
}

```

```
    leaf bit-rate {
      type uint32 {
        range "66..1000000000";
      }
      description "bit-rate";
    }
  }

  grouping policy-card-family-2-bit-rate-grp {
    description "policy-card-family-2-bit-rate-grp";
    leaf bit-rate {
      type uint32 {
        range "5..1000000000";
      }
      description "bit-rate";
    }
  }

  grouping informational-grp {
    description "informational-grp";
    leaf informational {
      type empty;
      description "informational";
    }
  }

  grouping metering-or-policing-grp {
    description "metering-or-policing-grp";
    uses radius-guided-grp;
    choice card-family {
      description "card-family";
      case card-family-1 {
        container card-family-1 {
          description "card-family-1";
          uses metering-or-policing-card-family-1-grp;
        }
      }
      case card-family-2 {
        container card-family-2 {
          description "card-family-2";
          uses metering-or-policing-card-family-2-grp;
        }
      }
      case card-family-3 {
        container card-family-3 {
          description "card-family-3";
          uses metering-or-policing-card-family-3-grp;
        }
      }
    }
  }
```

}

```
    }
    case default-card-family {
      /* The default card-family is card-family-1. */
      uses metering-or-policing-card-family-1-grp;
    }
  }
}

grouping metering-or-policing-card-family-1-grp {
  description "metering-or-policing-card-family-1-grp";
  uses policy-group-grp {
    augment "policy/group/class-group/class-group/classes/class/" +
      "packet-handling" {
      description "augment packet-handling";
      uses policy-packet-handling-grp {
        augment "policy-packet-handling/" +
          "rate-limiting/rate-type/bit-rate-type/bit-rate" {
            description "bit-rate";
            uses policy-card-family-1-3-bit-rate-grp;
          }
      }
    }
  }
}

uses inline-rate-specifications-packet-handling-grp {
  augment "inline-rate-specifications/rate-limiting/rate" {
    description "augment rate";
    uses policy-card-family-1-3-bit-rate-grp;
    uses informational-grp;
  }
}

uses rate-calculation-exclude-layer-2-overhead-grp;
}

grouping metering-or-policing-card-family-2-grp {
  description "metering-or-policing-card-family-2-grp";
  uses policy-group-grp {
    augment "policy/group/class-group/class-group/classes/class/" +
      "packet-handling" {
      description "packet-handling";
      uses policy-packet-handling-grp {
```

```

        augment "policy-packet-handling/" +
            "rate-limiting/rate-type/bit-rate-type/bit-rate" {
            description "augment bit-rate";
            uses policy-card-family-2-bit-rate-grp;
        }
    }
}

```

```

    uses inline-rate-specifications-packet-handling-grp {
        augment "inline-rate-specifications/rate-limiting/rate" {
            description "augment rate";
            uses policy-card-family-2-bit-rate-grp;
            uses informational-grp;
        }
    }
    uses rate-calculation-exclude-layer-2-overhead-grp;
}

grouping metering-or-policing-card-family-3-grp {
    description "metering-or-policing-card-family-3-grp";
    uses policy-group-grp {
        augment "policy/group/class-group/class-group/classes/class/" +
            "packet-handling" {
            description "augment packet-handling";
            uses policy-packet-handling-grp {
                augment "policy-packet-handling/" +
                    "rate-limiting/rate-type/bit-rate-type/bit-rate" {
                    uses policy-card-family-2-bit-rate-grp;
                    description "augment bit-rate";
                }
            }
        }
    }
}
uses inline-rate-specifications-packet-handling-grp {
    augment "inline-rate-specifications/rate-limiting/rate" {
        description "augment rate";
        uses policy-card-family-1-3-bit-rate-grp;
        uses informational-grp;
    }
}
uses rate-calculation-exclude-layer-2-overhead-grp;

```

```

}

grouping policy-group-grp {
  description "policy-group-grp";
  container policy {
    description "policy";
    choice group {
      description "group";
      case class-group {
        container class-group {
          description "Reference a previously defined class";
          leaf class-group-reference {
            type leafref {
              path "/qos:qos/qos:class-definitions"
                + "/qos:class-definition/qos:name";
            }
          }
        }
      }
    }
  }
}

```

```

    }
    description "class-group-reference";
  }
  container classes {
    description "classes";
    list class {
      key "name";
      max-elements 8;
      description "class";
      leaf name {
        type leafref {
          path "/qos:qos/qos:class-definitions"
            + "/qos:class-definition"
            + "[qos:name = "
            + "current()/../../../class-group-reference]"
            + "/qos:qos/qos:class";
        }
      }
      description "name";
    }
    container packet-handling {
      description "packet-handling";
      leaf parent-class {
        type string;
        description "parent-class";
      }
    }
  }
}

```

```

    }
  }
}
case access-group {
  container access-group {
    description "access-group";
    container ip-access-group {
      presence "Map an IPv4 access policy";
      description "Reference an IPv4 policy";
      leaf policy-name {
        type string;
        mandatory true;
        description "policy-name";
      }
      leaf context-name {
        type string;
        mandatory true;
        description "context-name";
      }
    }
  }
  container ipv6-access-group {

```

```

    presence "Map an IPv6 access policy";
    description "Reference an IPv6 policy";
    leaf policy-name {
      type string;
      mandatory true;
      description "policy-name";
    }
    leaf context-name {
      type string;
      mandatory true;
      description "context-name";
    }
  }
  container l2-access-group {
    presence "Map a layer-2 access policy";
    description "Reference a layer-2 policy";
    leaf policy-name {
      type string;
      mandatory true;

```



```

leaf qos-mode {
  type enumeration {
    enum "priority" {
      description "priority";
    }
    enum "strict" {
      description "strict";
    }
    enum "wrr" {
      description "wrr";
    }
  }
  default "wrr";
  description "qos-mode";
}
}

grouping mdrdrr-queue-grp {
  description "mdrr-queue-grp";
  leaf num-queues {
    type uint8 {
      range "1 | 2 | 4 | 8";
    }
    description "num-queues";
  }
  list queue {
    when "../num-queues and ../qos-mode != 'priority'" {
      description "only when the number of queues is >= 1";
    }
    must "num < ../num-queues" {
      description "queue's num must be less than num-queues";
    }
    key "num";
    description "queue";
    leaf num {
      type uint8 {

```

```

      range "0..8";
    }
    description "num";
  }
  leaf weight {

```



```

        when "../num != 0 or ../../qos-mode != 'strict'" {
            description
                "queue 0's weight is not configurable in strict mode";
        }
        type uint8 {
            range "5..100";
        }
        description "weight";
    }
}
}

```

```

grouping queue-0-weight-grp {
    description "queue-0-weight-group";
    leaf queue-0-weight {
        type uint8 {
            range "5..100";
        }
        units "percentage";
        must "../qos-mode != 'strict'" {
            description "qos-mode must not be strict mode";
        }
        description "queue-0-weight";
    }
}

```

```

grouping mdrd-rate-choice-grp {
    description "mdrd-rate-choice-grp";
    choice mdrd-rate {
        description "mdrd-rate";
        case specify-rate {
            leaf rate {
                type uint32 {
                    range "56..100000000";
                }
                units "kbps";
                mandatory true;
                description "rate";
            }
            leaf burst {
                type uint32 {
                    range "1..80000000";
                }
            }
        }
    }
}

```

```

        units "byte";
        mandatory true;
        description "burst";
    }
}
}
}

```

```

grouping mdrd-grp {
    description "mdrd-grp";
    uses mdrd-congestion-avoidance-map-grp;
    uses queue-map-grp;
    uses mdrd-qos-mode-grp;
    uses mdrd-queue-grp;
    uses mdrd-rate-choice-grp;
}

```

```

grouping pwfq-queue-choice-priority-or-rate-grp {
    description "pwfq-queue-choice-priority-or-rate-grp";
    choice priority-or-rate {
        mandatory true;
        description "priority-or-rate";
        case priority {
            leaf priority {
                type uint8 {
                    range "0..7";
                }
                mandatory true;
                description "priority";
            }
            leaf weight {
                type uint16 {
                    range "1..1023";
                }
                mandatory true;
                description "weight";
            }
        }
    }
    case rate {
        choice absolute-or-percentage {
            description "absolute-or-percentage";
            case absolute {
                leaf rate-maximum {
                    type uint32 {
                        range "4..100000000";
                    }
                    units "kbps";
                    description "rate-maximum";
                }
            }
        }
    }
}

```

Internet-Draft

QoS YANG

October 2016

```
    }
  }
  case percentage {
    leaf rate-maximum-percentage {
      type uint8 {
        range "1..100";
      }
      units "percent";
      description "rate-maximum-percentage";
    }
  }
}

grouping pwfq-queue-choice-grp {
  description "pwfq-queue-choice-grp";
  leaf num-queues {
    type uint8 {
      range "1 | 2 | 4 | 8";
    }
    description "num-queues";
  }
  list queue {
    when "../num-queues > 0" {
      description "queue entries exist only when num-queues > 0";
    }
    must "priority < ../num-queues" {
      description "queue priority must be < num-queues";
    }
    key "num";
    description "queue";
    leaf num {
      type uint8 {
        range "0..7";
      }
      description "priority";
    }
    uses pwfq-queue-choice-priority-or-rate-grp;
  }
}
```

```

grouping pwfq-rate-card-family-1-grp {
  description "pwfq-rate-card-family-1-grp";
  choice rate-or-weight {
    description "rate-or-weight";
    case rate-minimum {

```

```

    container rate-minimum {
      presence "Configure minimum rate";
      description "rate-minimum";
      leaf rate-minimum {
        type uint32 {
          range "8..10000000";
        }
        units "kbps";
        mandatory true;
        description "rate-minimum";
      }
      leaf burst {
        type uint32 {
          range "64..1875000000";
        }
        units "byte";
        description "burst";
      }
    }
  }
}
case weight {
  leaf weight {
    type uint16 {
      range "1..1023";
    }
    description "weight";
  }
}
}
}
container rate-maximum {
  presence "Configure maximum rate";
  description "rate-maximum";
  leaf rate-maximum {
    type uint32 {
      range "8..100000000";

```

```

    }
    units "kbps";
    mandatory true;
    description "rate-maximum";
  }
  leaf burst {
    type uint32 {
      range "64..1875000000";
    }
    units "byte";
    description "burst";
  }
}

```

```

}

grouping pwfq-rate-card-family-2-grp {
  description "pwfq-rate-card-family-2-grp";
  choice rate-or-weight {
    description "rate-or-weight";
    case rate-minimum {
      container rate-minimum {
        presence "Configure minimum rate";
        description "rate-minimum";
        leaf rate-minimum {
          type uint32 {
            range "8..1000000";
          }
          units "kbps";
          mandatory true;
          description "rate-minimum";
        }
      }
    }
    case weight {
      leaf weight {
        type uint16 {
          range "1..4096";
        }
        description "weight";
      }
    }
  }
}

```

```

    }
    container rate-maximum {
        presence "Configure maximum rate";
        description "rate-maximum";
        leaf rate-maximum {
            type uint32 {
                range "64..1000000";
            }
            units "kbps";
            mandatory true;
            description "rate-maximum";
        }
    }
}

grouping pwfq-rate-card-family-3-grp {
    description "pwfq-rate-card-family-3-grp";
    container rate-minimum {
        presence "Configure minimum rate";
        description "rate-minimum";
    }
}

```

```

    leaf rate-minimum {
        type uint32 {
            range "8..1000000";
        }
        units "kbps";
        mandatory true;
        description "rate-minimum";
    }
    leaf burst {
        type uint32 {
            range "64..1875000000";
        }
        units "byte";
        description "burst";
    }
}

container rate-maximum {
    presence "Configure maximum rate";
    description "rate-maximum";
    leaf rate-maximum {
        type uint32 {

```

```

        range "8..100000000";
    }
    units "kbps";
    mandatory true;
    description "rate-maximum";
}
container counters {
    presence "Enable counting";
    description "counters";
    leaf dual-stack {
        type empty;
        description "Enable dual-stack costing";
    }
}
}
}

grouping pwfq-queue-priority-group-grp {
    description "pwfq-queue-priority-group-grp";
    list queue-priority-group {
        key "priority-group";
        description "queue-priority-group";
        leaf priority-group {
            type uint8 {
                range "0..7";
            }
            description "priority-group";
        }
    }
}

```

```

    }
    choice absolute-minimum-percentage {
        mandatory true;
        description "absolute-minimum-percentage";
        /* Allow users of this grouping to add different cases */
    }
}
}

grouping pwfq-grp {
    description "pwfq-grp";
    choice card-family {
        description "card-family";
        case card-family-1 {

```

```

container card-family-1 {
  must "rate-minimum/rate <= rate-maximum/rate" {
    description "minimum rate <= maximum rate";
  }
  description "card-family-1";
  uses pwfq-congestion-avoidance-map-grp;
  uses queue-map-grp;
  uses pwfq-queue-choice-grp;
  uses pwfq-queue-priority-group-grp {
    augment "queue-priority-group/"
      + "absolute-minimum-percentage" {
      description "augment absolute-minimum-percentage";
      case rate-absolute {
        container rate-absolute {
          description "rate-absolute";
          leaf rate {
            type uint32 {
              range "8..1000000000";
            }
            units "kbps";
            description "rate";
          }
          leaf exceed {
            type empty;
            description "empty";
          }
        }
      }
    }
  case rate-percentage {
    container rate-percentage {
      description "rate-percentage";
      leaf rate-percentage {
        type uint8 {
          range "1..100";

```

```

    }
    units "percent";
    description "rate-percentage";
  }
}
}
}

```



```

    }
    uses pwfq-rate-card-family-1-grp;
  }
}
case card-family-2 {
  container card-family-2 {
    must "rate-minimum/rate <= rate-maximum/rate" {
      description "minimum rate must be <= maximum rate";
    }
    description "card-family-2";
    uses pwfq-congestion-avoidance-map-grp;
    uses queue-map-grp;
    uses pwfq-queue-choice-grp;
    uses pwfq-queue-priority-group-grp {
      augment "queue-priority-group/"
        + "absolute-minimum-percentage" {
        description "absolute-minimum-percentage";
        case rate-absolute {
          container rate-absolute {
            description "rate-absolute";
            leaf rate {
              type uint32 {
                range "8..100000000";
              }
              units "kpbs";
              description "rate";
            }
            leaf exceed {
              type empty;
              description "exceed";
            }
          }
        }
      }
    case rate-percentage {
      container rate-percentage {
        description "rate-percentage";
        leaf rate-percentage {
          type uint8 {
            range "1..100";
          }
          units "percent";
        }
      }
    }
  }
}

```

```

        description "rate-percentage";
    }
    leaf exceed {
        type empty;
        description "exceed";
    }
}
}
}
}
}
uses pwfq-rate-card-family-2-grp;
}
}
case card-family-3 {
    container card-family-3 {
        must "rate-minimum/rate <= rate-maximum/rate" {
            description "minimum rate must be <= maximum rate";
        }
        description "card-family-3";
        uses pwfq-congestion-avoidance-map-grp;
        uses queue-map-grp;
        uses pwfq-queue-choice-grp;
        uses pwfq-queue-priority-group-grp {
            augment "queue-priority-group/"
                + "absolute-minimum-percentage" {
                description "augment absolute-minimum-percentage";
                case rate-absolute {
                    container rate-absolute {
                        description "rate-absolute";
                        leaf rate {
                            type uint32 {
                                range "8..1000000000";
                            }
                            units "kbps";
                            description "rate";
                        }
                    }
                }
            }
            case rate-minimum {
                container rate-minimum {
                    description "rate-minimum";
                    leaf rate-minimum {
                        type uint32 {
                            range "8..1000000000";
                        }
                        description "rate-minimum";
                    }
                }
            }
        }
    }
}

```

Internet-Draft

QoS YANG

October 2016

```
    }
    case rate-percentage {
      container rate-percentage {
        description "rate-percentage";
        leaf rate-percentage {
          type uint8 {
            range "1..100";
          }
          units "percent";
          description "rate-percentage";
        }
      }
    }
    case rate-minimum-percentage {
      container rate-minimum-percentage {
        description "rate-minimum-percentage";
        leaf rate-minimum-percentage {
          type uint8 {
            range "1..100";
          }
          units "percent";
          description "rate-minimum-percentage";
        }
      }
    }
  }
}
uses pwfq-rate-card-family-3-grp;
}
}
case default-card-family {
  /* Same as card-family-1 */
  uses pwfq-congestion-avoidance-map-grp;
  uses queue-map-grp;
  uses pwfq-queue-choice-grp;
  uses pwfq-queue-priority-group-grp {
    augment "queue-priority-group/"
      + "absolute-minimum-percentage" {
      description "augment absolute-minimum-percentage";
      case rate-absolute {
        container rate-absolute {
          description "rate-absolute";
          leaf rate {
```

```

    type uint32 {
        range "8..1000000000";
    }
    units "kbps";
    description "rate";

```

```

    }
    leaf exceed {
        type empty;
        description "exceed";
    }
}
}
case rate-percentage {
    container rate-percentage {
        description "rate-percentage";
        leaf rate-percentage {
            type uint8 {
                range "1..100";
            }
            units "percent";
            description "rate-percentage";
        }
        leaf exceed {
            type empty;
            description "exceed";
        }
    }
}
}
}
}
}
uses pwfq-rate-card-family-1-grp;
must "rate-minimum/rate <= rate-maximum/rate" {
    description "minimum rate must be <= maximum rate";
}
}
}
}
}

grouping arp-rate-limit-grp {
    description "arp-rate-limit-grp";
    container arp {

```

```

presence "arp";
description "arp";
leaf rate {
    type uint32 {
        range "1..2500000";
    }
    units "packet per second";
    mandatory true;
    description "rate";
}
leaf burst {
    type uint32 {

```

```

        range "1..25000000";
    }
    units "packet";
    mandatory true;
    description "burst";
}
}
}

grouping qos-class-definition-grp {
    description "qos-class-definition";
    list qos {
        key "class";
        unique "dscp-code";
        description "qos class";
        leaf dscp-code {
            type uint8 {
                range "1..63";
            }
            description "dscp-code";
        }
        leaf class {
            type string {
                length "1..39";
            }
            mandatory true;
            description "class name";
        }
    }
}

```

```

}

grouping class-map-grp {
  description "class-map grouping";
  choice marking-type {
    description "marking-type choice";
    case atm-class-map {
      uses atm-class-map-grp;
    }
    case ethernet-class-map {
      uses ethernet-class-map-grp;
    }
    case ip-class-map {
      uses ip-class-map-grp;
    }
    case mpls-class-map {
      uses mpls-class-map-grp;
    }
  }
}

```

```

}

grouping atm-class-map-grp {
  description "atm-class-map-grp";
  container atm {
    presence "atm-class-map";
    description "atm";
    choice direction {
      description "direction choice";
      case in {
        container in {
          presence "atm-class-map-in";
          description "in";
          list atm {
            key "class";
            leaf class {
              type uint8 {
                range "0..1";
              }
              description "class";
            }
          }
          uses atm-in-mapping-grp;
        }
      }
    }
  }
}

```



```

    }
    must
      "/qos/class-maps/class-map[name = current()]/ethernet "
    + "and "
    + "/qos/class-maps/class-map[name = current()]/ethernet/"
    + "in" {
      description "must use ethernet in class-map";
    }
    description "use-ethernet";
  }
}
case use-ip {
  leaf use-ip {
    type leafref {
      path "/qos/class-maps/class-map/name";
    }
    must "/qos/class-maps/class-map[name = current()]/ip and"
      + "/qos/class-maps/class-map[name = current()]/ip/in" {
      description "must use ip in class-map";
    }
    description "use-ip";
  }
}
}
}

grouping atm-out-mapping {
  description "atm-out-mapping grouping";
  leaf to-atm {
    type uint8 {
      range "0..1";
    }
    description "to-atm";
  }
}

```

```

  }
}

grouping ethernet-class-map-grp {
  description "ethernet-class-map grouping";
  container ethernet {
    presence "ethernet-class-map";
    description "ethernet";
  }
}

```



```

choice direction {
  description "direction choice";
  case in {
    container in {
      presence "ethernet-class-map-in";
      description "in";
      list ethernet {
        key "class";
        leaf class {
          type uint8 {
            range "0..7";
          }
          description "class";
        }
        uses ethernet-in-mapping-grp;
        description "ethernet list";
      }
      uses mapping-schema-grp;
    }
  }
  case out {
    container out {
      presence "ethernet-class-map-out";
      description "out";
      list qos {
        key "dscp-code";
        leaf dscp-code {
          type uint8 {
            range "0..63";
          }
          description "dscp-code";
        }
        uses ethernet-out-mapping-grp;
        description "qos list";
      }
      uses mapping-schema-grp;
    }
  }
}

```

```

grouping ethernet-in-mapping-grp {
  description "ethernet-in-mapping grouping";
  choice ethernet-in-mapping {
    description "ethernet-in-mapping choice";
    case to-qos {
      leaf to-qos {
        type uint8 {
          range "0..63";
        }
        description "to-qos";
      }
    }
  }
  case use-ip {
    leaf use-ip {
      type leafref {
        path "/qos/class-maps/class-map/name";
      }
      must "/qos/class-maps/class-map[name = current()]/ip and"
        + "/qos/class-maps/class-map[name = current()]/ip/in" {
        description "must use ip in class-map";
      }
      description "use-ip";
    }
  }
}

grouping ethernet-out-mapping-grp {
  description "ethernet-out-mapping grouping";
  choice ethernet-out-mapping {
    description "ethernet-out-mapping choice";
    case to-ethernet {
      leaf to-ethernet {
        type uint8 {
          range "0..7";
        }
        description "to-ethernet";
      }
    }
  }
  case use-ethernet {
    leaf use-ethernet {
      type leafref {
        path "/qos/class-maps/class-map/name";
      }
      must
        "/qos/class-maps/class-map[name = current()]/ethernet "

```

```
    + " and"
    + "/qos/class-maps/class-map[name = current()]/ethernet/"
    + "out" {
        description "must use ethernet out class-map";
    }
    description "use-ethernet";
}
}
}
}
```

```
grouping mapping-schema-grp {
    description "mapping-schema grouping";
    leaf mapping-schema {
        type enumeration {
            enum "5P3D" {
                description "5P3D";
            }
            enum "6P2D" {
                description "6P2D";
            }
            enum "7P1D" {
                description "7P1D";
            }
            enum "8P0D" {
                description "7P1D";
            }
        }
        description "mapping-schema";
    }
}
```

```
grouping ip-class-map-grp {
    description "ip-class-map grouping";
    container ip {
        presence "ip-class-map";
        description "ip";
        choice direction {
            description "direction choice";
            case in {
                container in {
                    presence "ip-class-map-in";
                    description "in direction";
                    list ip {
                        must "dscp-code != to-qos" {
                            description
```

```
        "dscp-code must be different from to-qos";
    }
}
```

```
    key "dscp-code";
    leaf dscp-code {
        type uint8 {
            range "0..63";
        }
        description "dscp-code";
    }
    leaf to-qos {
        type uint8 {
            range "0..63";
        }
        description "to-qos";
    }
    description "ip list";
}
}
}
case out {
    container out {
        presence "ip-class-map-out";
        description "out direction";
        list qos {
            must "dscp-code != to-qos" {
                description
                    "dscp-code must be different from to-qos";
            }
            key "dscp-code";
            leaf dscp-code {
                type uint8 {
                    range "0..63";
                }
                description "dscp-code";
            }
            leaf to-ip {
                type uint8 {
                    range "0..63";
                }
                description "to-ip";
            }
        }
    }
}
```

```

        description "qos list";
    }
}
}
}
}
}
}

```

```

grouping mpls-class-map-grp {

```

```

description "mpls-class-map grouping";
container mpls {
  presence "mpls-class-map";
  description "mpls";
  choice direction {
    description "direction choice";
    case in {
      container in {
        presence "mpls-class-map-in";
        description "in direction";
        list mpls {
          key "exp-value";
          leaf exp-value {
            type uint8 {
              range "0..7";
            }
            description "exp-value";
          }
          uses mpls-in-mapping-grp;
          description "mpls list";
        }
        uses mapping-schema-grp;
      }
    }
  }
  case out {
    container out {
      presence "mpls-class-map-out";
      description "out";
      list qos {
        key "dscp-code";
        leaf dscp-code {
          type uint8 {

```

```

        range "0..63";
    }
    description "dscp-code";
}
uses mpls-out-mapping-grp;
description "qos list";
}
uses mapping-schema-grp;
}
}
}
}
}

grouping mpls-in-mapping-grp {
    description "mpls-in-mapping grouping";
}

```

```

choice mapping {
    description "mapping choice";
    case to-qos {
        leaf to-qos {
            type uint8 {
                range "0..63";
            }
            description "to-qos";
        }
    }
    case use-ethernet {
        leaf use-ethernet {
            type leafref {
                path "/qos/class-maps/class-map/name";
            }
            must
                "/qos/class-maps/class-map[name = current()]/ethernet "
                + "and"
                + "/qos/class-maps/class-map[name = current()]/ethernet/"
                + "in" {
                    description "must use ethernet incoming map";
                }
            description "use-ethernet";
        }
    }
}
}

```

```

    case use-ip {
      leaf use-ip {
        type leafref {
          path "/qos/class-maps/class-map/name";
        }
        must "/qos/class-maps/class-map[name = current()]/ip and"
          + "/qos/class-maps/class-map[name = current()]/ip/in" {
          description "must use ip in class-map";
        }
        description "use-ip";
      }
    }
  }
}

```

```

grouping mpls-out-mapping-grp {
  description "mpls-out-mapping grouping";
  choice mapping {
    description "mapping choice";
    case to-mpls {
      leaf to-mpls {
        type uint8 {
          range "0..7";

```

```

    }
    description "to-mpls";
  }
}
case use-mpls {
  leaf use-mpls {
    type leafref {
      path "/qos/class-maps/class-map/name";
    }
    must
      "/qos/class-maps/class-map[name = current()]/mpls and"
      + "/qos/class-maps/class-map[name = current()]/mpls/out" {
      description "must use mpls out class-map";
    }
    description "use-mpls";
  }
}
}

```

```

}

grouping congestion-avoidance-map-grp {
  description "congestion-avoidance-map grouping";
  choice map-type {
    description "map-type choice";
    case mdr {
      container mdr {
        presence "congestion-avoidance-map-mdr";
        description "mdr";
        uses mdr-map-grp;
      }
    }
    case pwfq {
      container pwfq {
        presence "congestion-avoidance-map-pwfq";
        description "pwfq";
        uses pwfq-map-grp;
      }
    }
  }
}

grouping mdr-map-grp {
  description "mdr-map-grp";
  list queue {
    key "num";
    description "queue";
    leaf num {
      type uint8 {

```

```

      range "0..7";
    }
    description "queue number";
  }
  uses queue-depth-grp;
  uses exponent-weighting-grp;
  uses drop-profile-grp;
}

grouping queue-depth-grp {

```



```

description "queue-depth-grp";
leaf depth {
    type uint32 {
        range "1..65536";
    }
    description "depth";
}
}

grouping exponent-weighting-grp {
    description "exponent-weighting-grp";
    leaf exponential-weight {
        type uint8 {
            range "1..15";
        }
        description "exponential-weight";
    }
}

grouping drop-profile-grp {
    description "drop-profile-grp";
    container default {
        must "min-threshold <= max-threshold" {
            description "min-threshold must be <= max-threshold";
        }
        presence "default drop profile";
        uses threshold-grp;
        description "default drop profile";
    }
    container profile-1 {
        must "min-threshold <= max-threshold" {
            description "min-threshold must be <= max-threshold";
        }
        presence "drop profile-1";
        uses threshold-grp;
        uses dscp-grp;
        description "drop profile-1";
    }
}

```

```

}
container profile-2 {
    must "min-threshold <= max-threshold" {
        description "min-threshold must be <= max-threshold";
    }
}

```

```

    }
    presence "drop profile-2";
    uses threshold-grp;
    uses dscp-grp;
    description "drop profile-2";
  }
}

grouping average-packet-size-grp {
  description "average-packet-size-grp";
  leaf average-packet-size {
    type uint16 {
      range "128..9600";
    }
    description "Packet size must be multiples of 128";
  }
}

grouping threshold-grp {
  description "threshold-grp";
  leaf min-threshold {
    type uint16 {
      range "1..10240";
    }
    mandatory true;
    description "min-threshold";
  }
  leaf max-threshold {
    type uint16 {
      range "2..10240";
    }
    mandatory true;
    description "max-threshold";
  }
  leaf probability {
    type uint16 {
      range "1..1023";
    }
    mandatory true;
    description "probability";
  }
}

grouping dscp-grp {

```

```
description "dscp-grp";
leaf-list dscp {
  type union {
    type enumeration {
      enum "af11" {
        description "DSCP 10";
      }
      enum "af12" {
        description "DSCP 12";
      }
      enum "af13" {
        description "DSCP 14";
      }
      enum "af21" {
        description "DSCP 18";
      }
      enum "af22" {
        description "DSCP 20";
      }
      enum "af23" {
        description "DSCP 22";
      }
      enum "af31" {
        description "DSCP 26";
      }
      enum "af32" {
        description "DSCP 28";
      }
      enum "af33" {
        description "DSCP 30";
      }
      enum "af41" {
        description "DSCP 34";
      }
      enum "af42" {
        description "DSCP 36";
      }
      enum "af43" {
        description "DSCP 38";
      }
      enum "cs1" {
        description "Class Selector 001 000 (8)";
      }
      enum "cs2" {
        description "Class Selector 010 000 (16)";
      }
      enum "cs3" {
```

description "Class Selector 011 000 (24)";

```
    }
    enum "cs4" {
        description "Class Selector 100 000 (32)";
    }
    enum "cs5" {
        description "Class Selector 101 000 (40)";
    }
    enum "cs6" {
        description "Class Selector 110 000 (48)";
    }
    enum "cs7" {
        description "Class Selector 111 000 (56)";
    }
    enum "df" {
        description
            "Default forwarding 000 000 (0). "
            + "This is synonymous to cs0";
    }
    enum "ef" {
        description "Expedited forwarding 101 110 (46)";
    }
}
type uint8 {
    range "1..7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | "
        + "27 | 29 | 31 | 33 | 35 | 37 | 39 | 41..45 | 47 | "
        + "49..55 | 57..63";
}
}
min-elements 1;
description "dscp";
}
}

grouping pwfq-map-grp {
    description "pwfq-map-grp";
    choice card-family {
        description "card-family";
        case card-family-1 {
            container card-family-1 {
                description "card-family-1";
            }
        }
    }
}
```

```

list queue {
  key "num";
  description "queue";
  leaf num {
    type uint8 {
      range "0..7";
    }
    description "queue number";
  }
}

```

```

    }
    uses queue-depth-grp;
    uses drop-profile-grp;
  }
}
case card-family-2 {
  description "card-family-2";
  container card-family-2 {
    description "card-family-2";
    list queue {
      key "num";
      description "queue";
      leaf num {
        type uint8 {
          range "0..7";
        }
        description "queue number";
      }
      uses queue-depth-grp;
      uses drop-profile-grp;
      uses exponent-weighting-grp;
    }
  }
}
case card-family-3 {
  container card-family-3 {
    description "card-family-3";
    list queue {
      key "num";
      description "queue";
      leaf num {
        type uint8 {

```



```

choice encaps-access-line {
    mandatory true;
    description "encaps-access-line";
    case ethernet {
        leaf ethernet {
            type empty;
            description "ethernet";
        }
    }
    case value {
        leaf value {
            type uint8 {
                range "0..255";
            }
            description "value";
        }
    }
}
leaf reserved {
    type uint8 {
        range "1..255";
    }
}

```

```

    }
    description "reserved";
}
}
case card-family-2 {
    container card-family-2 {
        description "2";
        uses qos-grp;
    }
}
case card-family-3 {
    container card-family-3 {
        description "3";
        uses qos-grp;
    }
}
}
description "card-family";

```

```

    }
  }
}
case resource {
  leaf resource {
    type empty;
    description resource ;
  }
  leaf tm-resource {
    type uint8 {
      range "1..4";
    }
    description "tm-resource";
  }
  container slot-port {
    description "slot-port";
    leaf slot-port {
      type string;
      mandatory true;
      description "slot-port";
    }
    leaf tm-resource {
      type uint8 {
        range "1..4";
      }
      description "tm-resource";
    }
  }
}
}

```

```

    }
  }

  grouping qos-grp {
    description "qos";
    container encaps-access-line {
      description "encaps-access-line";
      choice encaps-access-line {
        description "encaps-access-line";
        case ether-aal5-llc {
          leaf ether-aal5-llc {
            type empty;
          }
        }
      }
    }
  }
}

```



```

        description "ether-aal5-llc";
    }
}
case ether-aal5-llc-fcs {
    leaf ether-aal5-llc-fcs {
        type empty;
        description "ether-aal5-llc-fcs";
    }
}
case ether-aal5-null {
    leaf ether-aal5-null {
        type empty;
        description "ether-aal5-null";
    }
}
case ether-aal5-null-fcs {
    leaf ether-aal5-null-fcs {
        type empty;
        description "ether-aal5-null-fcs";
    }
}
case ethernet {
    leaf ethernet {
        type empty;
        description "ethernet";
    }
}
case ipoa-llc {
    leaf ipoa-llc {
        type empty;
        description "ipoa-llc";
    }
}
case ipoa-null {
    leaf ipoa-null {
        type empty;

```

```

        description "ipoa-null";
    }
}
case pppoa-llc {
    leaf pppoa-llc {

```

```

        type empty;
        description "pppoa-llc";
    }
}
case pppoa-null {
    leaf pppoa-null {
        type empty;
        description "pppoa-null";
    }
}
case value {
    container value {
        description "value";
        leaf value {
            type uint8 {
                range "0..255";
            }
            mandatory true;
            description "value";
        }
        container data-link {
            description "data-link";
            choice data-link {
                description "data-link";
                case atm {
                    leaf atm {
                        type empty;
                        description "atm";
                    }
                }
                case ethernet {
                    leaf ethernet {
                        type empty;
                        description "ethernet";
                    }
                }
            }
        }
    }
}
}
leaf rate-factor {

```

```

    type uint8 {
        range "1..100";
    }
    description "rate-factor";
}
leaf reserved {
    type uint8 {
        range "1..255";
    }
    description "reserved";
}
}

grouping qos-queue-map-grp {
    description "qos";
    /* List which enter the sub-grouping num-queues-grp */
    list num-queues {
        key "num";
        description "num-queues";
        leaf num {
            type uint8 {
                range "2 | 4 | 8";
            }
            description "num-queues";
        }
        uses num-queues-grp {
            refine "queue" {
                must "id < num" {
                    description
                        "A queue's id must be less than number of queues";
                }
            }
        }
    }
}

grouping num-queues-grp {
    description "num-queues";
    list queue {
        key "id";
        description "queue";
        leaf id {
            type uint8 {
                range "0..7";
            }
            description "queue";
        }
        leaf-list priority {

```

Internet-Draft

QoS YANG

October 2016

```
        type uint8 {
            range "0..7";
        }
        description "priority";
    }
}
}
container qos {
    description "qos";

    container class-definitions {
        description "class-definition";
        list class-definition {
            key "name";
            max-elements 14;
            leaf name {
                type string;
                description "qos-class-definition name";
            }
            uses qos-class-definition-grp;
            description "class-definition";
        }
    }
}
container class-maps {
    description "class-maps";
    list class-map {
        key "name";
        leaf name {
            type string;
            description "class-map-name";
        }
        description "class-map";
        uses class-map-grp;
    }
}
container congestion-avoidance-maps {
    description "congestion-avoidance-maps";
    list congestion-avoidance-map {
        key "name";
        leaf name {
            type string;
            description "congestion-avoidance-map name";
        }
    }
}
```

```

    }
    uses congestion-avoidance-map-grp;
    description "congestion-avoidance-map";
  }
}
container profiles {

```

```

    description "profile";
    list profile {
      key "name";
      uses qos-profile-grp;
      description "profile";
    }
  }
}
container queue-maps {
  description "queue-maps";
  list queue-map {
    key "name";
    leaf name {
      type string;
      description "queue-map-name";
    }
    description "queue-map";
    uses qos-queue-map-grp;
  }
}
container policies {
  description "policies";
  list policy {
    key "name";
    description "policy";
    leaf name {
      type string;
      description "name";
    }
  }
  choice policy-type {
    description "policy-type";
    case metering {
      container metering {
        presence "metering policy";
        description "metering policy";
        uses metering-or-policing-grp;
      }
    }
  }
}

```

```

    }
  }
  case policing {
    container policing {
      presence "policing policy";
      description "policing policy";
      uses metering-or-policing-grp;
    }
  }
  case mdrd {
    container mdrd {
      presence "mdrd policy";
      description "mdrd policy";

```

```

    uses mdrd-grp;
  }
}
case pwfq {
  container pwfq {
    presence "pwfq policy";
    description "pwfq policy";
    uses pwfq-grp;
  }
}
case protocol-rate-limit {
  container protocol-rate-limit {
    presence "protocol-rate-limit policy";
    description "protocol-rate-limit policy";
    uses arp-rate-limit-grp;
  }
}
}
}
}
}
}
<CODE ENDS>

```

## [5.](#) Security Consideration

TBD.

## 6. IANA Considerations

TBD.

## 7. Acknowledgments

The author thanks Sudhakar Chunduru from Ericsson for answering my questions about QoS.

## 8. References

### 8.1. Normative References

- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), DOI 10.17487/RFC2474, December 1998, <<http://www.rfc-editor.org/info/rfc2474>>.

Chen

Expires April 22, 2017

[Page 55]

---

Internet-Draft

QoS YANG

October 2016

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

### 8.2. Informative References

- [aclyang] Bogdanovic, D., Sreenivasa, K., Huang, L., and D. Blair, "Work-in-progress, '[draft-ietf-netmod-acl-model-09](#)'", October 2016, <<https://tools.ietf.org/html/draft-ietf-netmod-acl-model-09>>.
- [dsyang] Choudhary, A., Jethanandani, J., Strahle, N., and E. Aries, "Work-in-progress, '[draft-asechoud-netmod-qos-model-01](#)'", July 2016, <<https://tools.ietf.org/html/draft-asechoud-netmod-qos-model-01>>.
- [pbfyang] Hares, S., Wu, Q., and R. White, "Work-in-progress, '[draft-ietf-i2rs-pkt-eca-data-model-02](#)'", October 2016, <<https://tools.ietf.org/html/draft-ietf-i2rs-pkt-eca-data-model-02>>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", [RFC 2475](#), DOI 10.17487/RFC2475, December 1998, <<http://www.rfc-editor.org/info/rfc2475>>.

[supayang]

Halpern, J., Strassner, J., and S. Van der Meer, "Work-in-progress, '[draft-ietf-supra-generic-policy-data-model-02](#)'", October 2016, <<https://tools.ietf.org/html/draft-ietf-supra-generic-policy-data-model-02>>.

#### Author's Address

I. Chen  
Kuatro Technologies

Email: [ichen@kuatrotech.com](mailto:ichen@kuatrotech.com)