Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 5, 2015

A. Choudhary
S. Shah
Cisco Systems
M. Jethanandani
Ciena Corporation
Y. Yan
B. Liu
Huawei Technologies
N. Strahle
Juniper Networks

March 04, 2015

YANG Model for Diffserv
draft-asechoud-netmod-diffserv-model-01

Abstract

This document describes a YANG model of Differentiated Services for configuration and operations.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for the configuration, state data of Differentiated Services. Any RPC or notification definition is not part of this document. As many vendors have different object constructs to represent the same data, it has been tried to design this model in a very flexible, extensible and generic way to fit into most of the vendor requirements. The model is based on Differentiated Services (Diffserv) architecture and various references have been made to already available standard architecture documents.

Diffserv is a preferred approach for network service providers to offer services to different customers based on their different kinds of network quality-of-service (QoS) objectives. The traffic streams are differentiated based on Differentiated Services Code Points (DSCP) carried in the IP header of each packet. The DSCP markings are applied by upstream node or by the edge router on entry to the Diffserv network.

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. Diffserv Model Design

Diffserv architecture [RFC3289] [RFC2475] describes network node packet classification function and packet conditioning functions.

The complex classification is done at the edge of network and non-edge network devices conditions appropriately marked aggregate traffic based on per-hop behavior rules. Accordingly, a Multi-Field classifier matches the different fields in a packet and a Behavior Aggregated Classifier matches on DS codepoint field of a packet.

Packets MAY be grouped when a logical set of rules are applied on different packet header fields. Also, packet grouping MAY be done based on different values or range of values of same packet header field. Packet grouping MAY also be done based on presence of some values or range of values of a packet field or absence of such values or ranges. This diffserv model is flexible enough to support such logical grouping of packets.

A classifier entry can be stored as an object and used across different interfaces for either of inbound or outbound traffic. Any modification or deletion of such object will in turn results in such changes to the classifier on the corresponding interfaces. A classifier entry contains one or more packet conditioning functions. A packet conditioning function is typically based on direction of traffic and may drop, mark or delay network packets. A set of such classifier entries with corresponding conditioning functions when arranged in order of priority represents a diffserv policy. Any new classifier entry in a policy MAY be inserted before or after any other existing classifier-entry [RFC6020]. Such policies is stored as an object and used across different network device interfaces.

A meter qualifies if the traffic arrival rate is based on agreed upon rate and variability. A meter is generically modeled as qualifying rate and variability defined as a token bucket. Single rate meter [RFC2697] can be defined as two such token buckets with first defining the rate and committed burst and excess burst for second bucket. Similarly, two rates meter [RFC2698][RFC2859] can be defined as two such token buckets with first and second defining the committed rate and committed burst parameters and peak rate and peak burst respectively. Different Vendors can extend it to have other types of meters as well.

Metered traffic to each token bucket MAY either be marked or remarked appropriately of the diffserv codepoint packet field or even MAY be dropped. Classified packets through a classifier entry MAY directly be marked.

Packets can be always dropped if exceed agreed upon rates or it could be queued and then dropped based on any of various algorithms. Queue dropping is based on the threshold configured and can head-drop, tail-drop or dropped based on Active Queue Management algorithm like Random Early Detection (RED). Packets can be scheduled out based on priority with minimum-rate or WFQ with bandwidth sharing. Priority scheduler allow queue to use the entire capacity of the interface unless higher priority traffic is queued to be scheduled. If combination of EF [RFC3246] and multiple AF [RFC3260] classes of traffic needs to be scheduled, a combination of priority and WFQ scheduler SHOULD be used. Traffic can be shaped by defining a max rate and burst for a leaky bucket profile.

4. Diffserv Model

The model have four YANG modules. ietf-diffserv-classifier consists of classifier entries identified by a classifier entry name. Each such entry contains list of filter entries. Each filter entry represent any of the filter type [RFC6991] of a multi-field classifier which can be logically AND/OR with other filter types in the same classifier-entry. The model is flexible enough to take multiple values of the same filter type.

```
   module: ietf-diffserv-classifier
    +--rw classifiers
       +--rw classifier-entry* [classifier-entry-name]
          +--rw classifier-entry-name                string
          +--rw classifier-entry-descr?              string
          +--rw classifier-entry-filter-operation?   identityref
          +--rw filter-entry* [filter-type filter-logical-not]
             +--rw filter-type                   identityref
             +--rw filter-logical-not            boolean
             +--rw (filter-param)?
                +--:(dscp)
                |  +--rw dscp-cfg* [dscp-min dscp-max]
                |     +--rw dscp-min    inet:dscp
                |     +--rw dscp-max    inet:dscp
                +--:(source-ip-address)
                |  +--rw source-ip-address-cfg* [source-ip-addr]
                |     +--rw source-ip-addr    inet:ip-prefix
                +--:(destination-ip-address)
                |  +--rw destination-ip-address-cfg*
                            [destination-ip-addr]
                |     +--rw destination-ip-addr    inet:ip-prefix
                +--:(source-port)
                |  +--rw source-port-cfg*
                            [source-port-min source-port-max]
                |     +--rw source-port-min    inet:port-number
                |     +--rw source-port-max    inet:port-number
                +--:(destination-port)
                |  +--rw destination-port-cfg*
                            [destination-port-min destination-port-max]
                |     +--rw destination-port-min    inet:port-number
                |     +--rw destination-port-max    inet:port-number
                +--:(protocol)
                |  +--rw protocol-cfg* [protocol-min protocol-max]
                |     +--rw protocol-min    uint8
                |     +--rw protocol-max    uint8
                +--:(flow-label)
                   +--rw flow-label-cfg* [flow-label-min flow-label-max]
                      +--rw flow-label-min    uint32
                      +--rw flow-label-max    uint32
```

An ietf-diffserv-policy module contains list of policy objects identified by a policy name which MUST be provided. Each policy object contains list of classifier-entries either configured inline or referred as an object. Each such classifier entry is augmented by set of actions. A policy object MAY contain a child-policy in each classifier-entry. A child policy MAY further classify the traffic and execute actions on classified packets.

```
   module: ietf-diffserv-policy
   +--rw policies {policy-template-support}?
      +--rw policy-entry* [policy-name]
         +--rw policy-name         string
         +--rw policy-descr?       string
         +--rw classifier-entry* [classifier-entry-name]
            +--rw classifier-entry-name           string
            +--rw classifier-entry-inline?        boolean
            +--rw classifier-entry-filter-oper?   identityref
            +--rw filter-entry* [filter-type filter-logical-not]
                                   {policy-inline-classifier-config}?
            |  +--rw filter-type                   identityref
            |  +--rw filter-logical-not            boolean
            |  +--rw (filter-param)?
            |     +--:(dscp)
            |     |  +--rw dscp-cfg* [dscp-min dscp-max]
            |     |     +--rw dscp-min    inet:dscp
            |     |     +--rw dscp-max    inet:dscp
            |     +--:(source-ip-address)
            |     |  +--rw source-ip-address-cfg* [source-ip-addr]
            |     |     +--rw source-ip-addr    inet:ip-prefix
            |     +--:(destination-ip-address)
            |     |  +--rw destination-ip-address-cfg*
                                   [destination-ip-addr]
            |     |     +--rw destination-ip-addr    inet:ip-prefix
            |     +--:(source-port)
            |     |  +--rw source-port-cfg*
                                   [source-port-min source-port-max]
            |     |     +--rw source-port-min    inet:port-number
            |     |     +--rw source-port-max    inet:port-number
            |     +--:(destination-port)
            |     |  +--rw destination-port-cfg*
                           [destination-port-min destination-port-max]
            |     |     +--rw destination-port-min    inet:port-number
            |     |     +--rw destination-port-max    inet:port-number
            |     +--:(protocol)
            |     |  +--rw protocol-cfg* [protocol-min protocol-max]
            |     |     +--rw protocol-min    uint8
            |     |     +--rw protocol-max    uint8
            |     +--:(flow-label)
            |        +--rw flow-label-cfg*
                                   [flow-label-min flow-label-max]
            |           +--rw flow-label-min    uint32
            |           +--rw flow-label-max    uint32
            +--rw classifier-action-entry-cfg* [action-type]
            |  +--rw action-type                          identityref
            |  +--rw (action-cfg-params)?
            |     +--:(marking)
            |     |  +--rw diffserv-action:marking-cfg
            |     |     +--rw diffserv-action:dscp?   inet:dscp
            |     +--:(priority)
            |     |  +--rw diffserv-action:priority-cfg
            |     |     +--rw diffserv-action:priority-level?   uint8
            |     |     +--rw diffserv-action:priority-rate?    uint64
            |     +--:(meter)
            |     |  +--rw diffserv-action:meter-cfg
            |     |     +--rw diffserv-action:meter-list* [meter-id]
            |     |        +--rw diffserv-action:meter-id       uint16
            |     |        +--rw diffserv-action:meter-rate?    uint64
            |     |        +--rw (burst-type)?
            |     |        |  +--:(size)
            |     |        |  |  +--rw diffserv-action:burst-size?
                                                                uint64
            |     |        |  +--:(interval)
            |     |        |     +--rw diffserv-action:burst-interval?
                                                                uint64
            |     |        +--rw diffserv-action:color
            |     |        |  +--rw diffserv-action:
                                        classifier-entry-name?  string
            |     |        |  +--rw diffserv-action:
                                        classifier-entry-descr? string
            |     |        |  +--rw diffserv-action:
                                      classifier-entry-filter-operation?
                                                            identityref
            |     |        +--rw diffserv-action:meter-action-type?
                                                            identityref
            |     |        +--rw (val)?
            |     |           +--:(meter-action-mark)
            |     |           |  +--rw diffserv-action:dscp?  inet:dscp
            |     |           +--:(meter-action-drop)
            |     |              +--rw diffserv-action:drop-action?
                                                                boolean
            |     +--:(max-rate)
            |     |  +--rw diffserv-action:max-rate-cfg
            |     |     +--rw diffserv-action:absolute-rate?    uint64
            |     |     +--rw (burst-type)?
            |     |        +--:(size)
            |     |        |  +--rw diffserv-action:burst-size? uint64
            |     |        +--:(interval)
            |     |           +--rw diffserv-action:burst-interval?
                                                                uint64
            |     +--:(algorithmic-drop)
            |     |  +--rw (drop-algorithm)?
            |     |     +--:(always-drop)
            |     |     |  +--rw diffserv-action:drop-cfg
            |     |     |     +--rw diffserv-action:drop-action? boolean
            |     |     +--:(tail-drop)
            |     |     |  +--rw diffserv-action:tail-drop-cfg
            |     |     |     +--rw diffserv-action:qlimit-dscp-thresh*
                                                     [dscp-min dscp-max]
            |     |     |        +--rw diffserv-action:dscp-min
                                                               inet:dscp
            |     |     |        +--rw diffserv-action:dscp-max
                                                               inet:dscp
            |     |     |        +--rw diffserv-action:threshold
            |     |     |           +--rw (threshold-type)?
            |     |     |              +--:(size)
            |     |     |              |  +--rw diffserv-action:
                                                  threshold-size? uint64
            |     |     |              +--:(interval)
            |     |     |                 +--rw diffserv-action:
                                                  threshold-interval?
                                                                  uint64
            |     |     +--:(random-detect)
            |     |        +--rw diffserv-action:random-detect-cfg
                                                     {aqm-wred-support}?
            |     |           +--rw diffserv-action:exp-weighting-const?
                                                                  uint32
            |     |           +--rw diffserv-action:mode-aggregate?
                                                                 boolean
            |     |           +--rw diffserv-action:ecn-enabled?
                                                                 boolean
            |     |           +--rw diffserv-action:wred-profile*
                                                       [wred-profile-id]
            |     |              +--rw diffserv-action:wred-profile-id
                                                                 uint16
            |     |              +--rw diffserv-action:wred-dscp*
                                                     [dscp-min dscp-max]
            |     |              |  +--rw diffserv-action:dscp-min
                                                               inet:dscp
            |     |              |  +--rw diffserv-action:dscp-max
                                                               inet:dscp
            |     |              +--rw diffserv-action:wred-min-thresh
            |     |              |  +--rw diffserv-action:threshold
            |     |              |     +--rw (threshold-type)?
            |     |              |        +--:(size)
            |     |              |        |  +--rw diffserv-action:
                                                     threshold-size?
                                                                  uint64
            |     |              |        +--:(interval)
            |     |              |           +--rw diffserv-action:
                                                     threshold-interval?
                                                                  uint64
            |     |              +--rw diffserv-action:wred-max-thresh
            |     |              |  +--rw diffserv-action:threshold
            |     |              |     +--rw (threshold-type)?
            |     |              |        +--:(size)
            |     |              |        |  +--rw diffserv-action:
                                                  threshold-size?
                                                              uint64
            |     |              |        +--:(interval)
            |     |              |           +--rw diffserv-action:
                                                  threshold-interval?
                                                              uint64
            |     |              +--rw diffserv-action:mark-probability?
                                                              uint32
            |     +--:(min-rate)
            |        +--rw diffserv-action:min-rate-cfg
            |           +--rw diffserv-action:min-rate?   uint64
            +--rw child-policy?                  leafref
                                   {hierarchial-policy-support}?
```

ietf-diffserv-action module contains set of diffserv actions which are augmented to ietf-diffserv-policy module and to ietf-diffserv-target module. Marking sets Diffserv codepoint value in the classified packet. Color-aware and Color-blind meters can be configured. Action counters are defined as grouping and are currently not augmented to any diffserv module.

ietf-diffserv-target module contains reference of diffserv-policy for either direction of network traffic and is augmented to ietf-interfaces [RFC7223] module.

```
   module: ietf-diffserv-target
   augment /if:interfaces/if:interface:
   +--rw diffserv-target-entry* [direction policy-name]
      +--rw policy-name                         string
      +--rw policy-descr?                       string
      +--rw direction                           identityref
      +--rw diffserv-target-classifier-entry*
                [classifier-entry-name parent-path]
                    {target-inline-policy-config}?
         +--rw classifier-entry-name           leafref
         +--rw parent-path                     string
         +--rw classifier-entry-inline?        boolean
         +--rw classifier-entry-filter-oper?   identityref
         +--rw filter-entry* [filter-type filter-logical-not]
                                  {policy-inline-classifier-config}?
         |  +--rw filter-type                   identityref
         |  +--rw filter-logical-not            boolean
         |  +--rw (filter-param)?
         |     +--:(dscp)
         |     |  +--rw dscp-cfg* [dscp-min dscp-max]
         |     |     +--rw dscp-min    inet:dscp
         |     |     +--rw dscp-max    inet:dscp
         |     +--:(source-ip-address)
         |     |  +--rw source-ip-address-cfg* [source-ip-addr]
         |     |     +--rw source-ip-addr    inet:ip-prefix
         |     +--:(destination-ip-address)
         |     |  +--rw destination-ip-address-cfg*
                                   [destination-ip-addr]
         |     |     +--rw destination-ip-addr    inet:ip-prefix
         |     +--:(source-port)
         |     |  +--rw source-port-cfg*
                                   [source-port-min source-port-max]
         |     |     +--rw source-port-min    inet:port-number
         |     |     +--rw source-port-max    inet:port-number
         |     +--:(destination-port)
         |     |  +--rw destination-port-cfg*
                           [destination-port-min destination-port-max]
         |     |     +--rw destination-port-min    inet:port-number
         |     |     +--rw destination-port-max    inet:port-number
         |     +--:(protocol)
         |     |  +--rw protocol-cfg* [protocol-min protocol-max]
         |     |     +--rw protocol-min    uint8
         |     |     +--rw protocol-max    uint8
         |     +--:(flow-label)
         |        +--rw flow-label-cfg* [flow-label-min flow-label-max]
         |           +--rw flow-label-min    uint32
         |           +--rw flow-label-max    uint32
         +--rw classifier-action-entry-cfg* [action-type]
            +--rw action-type                          identityref
            +--rw (action-cfg-params)?
               +--:(marking)
               |  +--rw diffserv-action:marking-cfg
               |     +--rw diffserv-action:dscp?   inet:dscp
               +--:(priority)
               |  +--rw diffserv-action:priority-cfg
               |     +--rw diffserv-action:priority-level?   uint8
               |     +--rw diffserv-action:priority-rate?    uint64
               +--:(meter)
               |  +--rw diffserv-action:meter-cfg
               |     +--rw diffserv-action:meter-list* [meter-id]
               |        +--rw diffserv-action:meter-id          uint16
               |        +--rw diffserv-action:meter-rate?       uint64
               |        +--rw (burst-type)?
               |        |  +--:(size)
               |        |  |  +--rw diffserv-action:burst-size? uint64
               |        |  +--:(interval)
               |        |     +--rw diffserv-action:burst-interval?
                                                                uint64
               |        +--rw diffserv-action:color
               |        |  +--rw diffserv-action:classifier-entry-name?
                                                                string
               |        |  +--rw diffserv-action:classifier-entry-descr?
                                                               string
               |        |  +--rw diffserv-action:
                                   classifier-entry-filter-operation?
                                                            identityref
               |        +--rw diffserv-action:meter-action-type?
                                                            identityref
               |        +--rw (val)?
               |           +--:(meter-action-mark)
               |           |  +--rw diffserv-action:dscp?
                                         inet:dscp
               |           +--:(meter-action-drop)
               |              +--rw diffserv-action:drop-action?
                                         boolean
               +--:(max-rate)
               |  +--rw diffserv-action:max-rate-cfg
               |     +--rw diffserv-action:absolute-rate?    uint64
               |     +--rw (burst-type)?
               |        +--:(size)
               |        |  +--rw diffserv-action:burst-size? uint64
               |        +--:(interval)
               |           +--rw diffserv-action:burst-interval? uint64
               +--:(algorithmic-drop)
               |  +--rw (drop-algorithm)?
               |     +--:(always-drop)
               |     |  +--rw diffserv-action:drop-cfg
               |     |     +--rw diffserv-action:drop-action?   boolean
               |     +--:(tail-drop)
               |     |  +--rw diffserv-action:tail-drop-cfg
               |     |     +--rw diffserv-action:qlimit-dscp-thresh*
                                                     [dscp-min dscp-max]
               |     |        +--rw diffserv-action:dscp-min   inet:dscp
               |     |        +--rw diffserv-action:dscp-max   inet:dscp
               |     |        +--rw diffserv-action:threshold
               |     |           +--rw (threshold-type)?
               |     |              +--:(size)
               |     |              |  +--rw diffserv-action:
                                             threshold-size?   uint64
               |     |              +--:(interval)
               |     |                 +--rw diffserv-action:
                                             threshold-interval? uint64
               |     +--:(random-detect)
               |        +--rw diffserv-action:random-detect-cfg
```

```
                                                     {aqm-wred-support}?
               |           +--rw diffserv-action:exp-weighting-const?
                                                                  uint32
               |           +--rw diffserv-action:mode-aggregate?
                                                                 boolean
               |           +--rw diffserv-action:ecn-enabled?
                                                                 boolean
               |           +--rw diffserv-action:wred-profile*
                                                       [wred-profile-id]
               |              +--rw diffserv-action:wred-profile-id
                                                                  uint16
               |              +--rw diffserv-action:wred-dscp*
                                                     [dscp-min dscp-max]
               |              |  +--rw diffserv-action:dscp-min
                                                               inet:dscp
               |              |  +--rw diffserv-action:dscp-max
                                                               inet:dscp
               |              +--rw diffserv-action:wred-min-thresh
               |              |  +--rw diffserv-action:threshold
               |              |     +--rw (threshold-type)?
               |              |        +--:(size)
               |              |        |  +--rw diffserv-action:
                                                  threshold-size? uint64
               |              |        +--:(interval)
               |              |           +--rw diffserv-action:
                                                  threshold-interval?
                                                                  uint64
               |              +--rw diffserv-action:wred-max-thresh
               |              |  +--rw diffserv-action:threshold
               |              |     +--rw (threshold-type)?
               |              |        +--:(size)
               |              |        |  +--rw diffserv-action:
                                                  threshold-size? uint64
               |              |        +--:(interval)
               |              |           +--rw diffserv-action:
                                                  threshold-interval?
                                                                  uint64
               |              +--rw diffserv-action:mark-probability?
                                                                  uint32
               +--:(min-rate)
                  +--rw diffserv-action:min-rate-cfg
                     +--rw diffserv-action:min-rate?   uint64
```

5. Diffserv Modules

5.1. IETF-DIFFSERV-CLASSIFIER

```
   module ietf-diffserv-classifier {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv-classifier";
     prefix diffserv-classifier;

     import ietf-inet-types {
       prefix inet;
     }

     revision 2014-09-07 {
       description
         "First revision of diffserv based classifier";
     }

     feature policy-inline-classifier-config {
       description
         " This feature allows classifier configuration directly
           under policy.";
     }

     identity filter-type {
       description
         " This is identity of base filter-type";
     }

     identity dscp {
       base filter-type;
     }

     identity source-ip-address {
       base filter-type;
     }

     identity destination-ip-address {
       base filter-type;
     }

     identity source-port {
       base filter-type;
     }

     identity destination-port {
       base filter-type;
     }
     identity protocol {
       base filter-type;
     }

     identity flow-label {
       base filter-type;
     }

     identity classifier-entry-filter-operation-type {
       description
         "Classifier entry filter logical operation";
     }

     identity match-any-filter {
       base classifier-entry-filter-operation-type;
       description
         "Classifier entry filter logical OR operation";
     }

     identity match-all-filter {
       base classifier-entry-filter-operation-type;
       description
         "Classifier entry filter logical AND operation";
     }

     grouping filters {
       leaf filter-type {
         type identityref {
           base filter-type;
         }
         description
           "This leaf defines type of the filter";
       }
       leaf filter-logical-not {
         type boolean;
         description
           "
            This is logical-not operator for a filter. When true, it
            indicates filter looks for absence of a pattern defined
            by the filter
           ";
       }
       choice filter-param {
         case dscp {
           list dscp-cfg {
             key "dscp-min dscp-max";
             leaf dscp-min {
               type inet:dscp;
             }
             leaf dscp-max {
               type inet:dscp;
             }
           }
           description
             "Filter containing list of dscp ranges";
         }
         case source-ip-address {
           list source-ip-address-cfg {
             key "source-ip-addr";
             leaf source-ip-addr {
               type inet:ip-prefix;
             }
           }
           description
             "Filter containing list of source ip addresses";
         }
         case destination-ip-address {
           list destination-ip-address-cfg {
             key "destination-ip-addr";
             leaf destination-ip-addr {
               type inet:ip-prefix;
             }
           }
           description
             "Filter containing list of destination ip address";
         }
         case source-port {
           list source-port-cfg {
             key "source-port-min source-port-max";
             leaf source-port-min {
               type inet:port-number;
             }
             leaf source-port-max {
               type inet:port-number;
             }
           }
           description
             "Filter containing list of source-port ranges";
         }
         case destination-port {
           list destination-port-cfg {
             key "destination-port-min destination-port-max";
             leaf destination-port-min {
               type inet:port-number;
             }
             leaf destination-port-max {
               type inet:port-number;
             }
           }
           description
             "Filter containing list of destination-port ranges";
         }
         case protocol {
           list protocol-cfg {
             key "protocol-min protocol-max";
             leaf protocol-min {
               type uint8 {
                 range "0..255";
               }
             }
             leaf protocol-max {
               type uint8 {
                 range "0..255";
               }
             }
           }
           description
             "Filter Type Protocol";
         }
         case flow-label {
           list flow-label-cfg {
             key "flow-label-min flow-label-max";
             leaf flow-label-min {
               type uint32 {
                 range "0..1048575";
               }
             }
             leaf flow-label-max {
               type uint32 {
                 range "0..1048575";
               }
             }
           }
           description
             "Filter containing list of flow-label ranges";
         }
       }
     }

     grouping classifier-entry-generic-attr {
       leaf classifier-entry-name {
         type string;
         description
           "Diffserv classifier name";
       }
       leaf classifier-entry-descr {
         type string;
         description
           "Description of the class template";
       }
       leaf classifier-entry-filter-operation {
         type identityref {
           base classifier-entry-filter-operation-type;
         }
         default "match-any-filter";
       }
     }

     grouping classifier-entry-inline-attr {
       leaf classifier-entry-inline {
         type boolean;
         description
           "Indication of inline classifier entry";
         default "false";
       }
       leaf classifier-entry-filter-oper {
         type identityref {
           base classifier-entry-filter-operation-type;
         }
         default "match-any-filter";
       }
       list filter-entry {
         if-feature policy-inline-classifier-config;
         when "classifier-entry-inline == true";
         key "filter-type filter-logical-not";
         uses filters;
       }
     }

     container classifiers {
       description
         "list of classifier entry";
       list classifier-entry {
         key "classifier-entry-name";
         description
           "classifier entry template";
         uses classifier-entry-generic-attr;
         list filter-entry {
           key "filter-type filter-logical-not";
           uses filters;
         }
       }
     }
   }
```

5.2. IETF-DIFFSERV-POLICY

```
   module ietf-diffserv-policy {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv-policy";
     prefix diffserv-policy;

     import ietf-diffserv-classifier {
       prefix classifier;
     }

     revision 2014-09-07 {
       description
         "First revision of diffserv policy";
     }

     feature policy-template-support {
       description
         " This feature allows policy template to be configured";
     }

     feature hierarchial-policy-support {
       description
         " This feature allows hierarchial policy to be configured";
     }

     grouping policy-generic-attr {
       leaf policy-name {
         type string;
         description
           "Diffserv policy name";
       }
       leaf policy-descr {
         type string;
         description
           "Diffserv policy description";
       }
     }

     identity action-type {
       description
         "This base identity type defines action-types";
     }
     grouping classifier-action-entry-cfg {
       list classifier-action-entry-cfg {
         key "action-type";
         ordered-by user;
         leaf action-type {
           type identityref {
             base action-type;
           }
           description
             "This defines action type ";
         }
         choice action-cfg-params;
       }
     }

     container policies {
       description
         "list of policy templates";
       if-feature policy-template-support;
       list policy-entry {
         key "policy-name";
         description
           "policy template";
         uses policy-generic-attr;
         list classifier-entry {
           key "classifier-entry-name";
           ordered-by user;
           leaf classifier-entry-name {
             type string;
             description
               "Diffserv classifier entry name";
           }
           uses classifier:classifier-entry-inline-attr;
           uses classifier-action-entry-cfg;
           leaf child-policy {
             if-feature hierarchial-policy-support;
             type leafref {
               path "/policies/policy-entry/policy-name";
             }
           }
         }
       }
     }
   }
```

5.3. IETF-DIFFSERV-ACTION

```
   module ietf-diffserv-action {
     namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv-action";
     prefix diffserv-action;

     import ietf-interfaces {
       prefix if;
     }
     import ietf-inet-types {
       prefix inet;
     }
     import ietf-diffserv-classifier {
       prefix diffserv-classifier;
     }
     import ietf-diffserv-policy {
       prefix diffserv-policy;
     }
     import ietf-diffserv-target {
       prefix diffserv-target;
     }

     revision 2014-09-07 {
       description
         "Initial revision for diffserv actions on network packets";
     }

     feature aqm-wred-support {
       description
         " This feature allows AQM WRED to be configured";
     }

     grouping dscp-range {
       leaf dscp-min {
         type inet:dscp;
       }
       leaf dscp-max {
         type inet:dscp;
       }
     }

     grouping burst {
       choice burst-type {
         case size {
           leaf burst-size {
             units "bytes";
             type uint64;
```

```
          }
        }
        case interval {
          leaf burst-interval {
            units "microsecond";
            type uint64;
          }
        }
      }
    }

    grouping threshold {
      container threshold {
        description
          "threshold";
        choice threshold-type {
          case size {
            leaf threshold-size {
              units "bytes";
              type uint64;
            }
          }
          case interval {
            leaf threshold-interval {
              units "microsecond";
              type uint64;
            }
          }
        }
      }
    }

    identity min-rate {
      base diffserv-policy:action-type;
    }

    identity marking {
      base diffserv-policy:action-type;
    }

    identity priority {
      base diffserv-policy:action-type;
    }

    identity meter {
      base diffserv-policy:action-type;
    }
```

```
     identity max-rate {
       base diffserv-policy:action-type;
     }

     identity algorithmic-drop {
       base diffserv-policy:action-type;
     }

     identity meter-action-type {
       description
         "conform/violate/exceed action type in a meter";
     }

     identity meter-action-drop {
       base meter-action-type;
     }

     identity meter-action-set {
       base meter-action-type;
     }

     grouping drop {
       leaf drop-action {
         type boolean;
       }
       description
         "the drop action";
     }

     grouping queuelimit {
       list qlimit-dscp-thresh {
         key "dscp-min dscp-max";
         uses dscp-range;
         uses threshold;
       }
     }

     grouping meter-action-params {
       leaf meter-action-type {
         type identityref {
           base meter-action-type;
         }
       }
       choice val {
         case meter-action-mark {
           uses marking;
           description
             "meter action: mark";
         }
         case meter-action-drop {
           description
             "meter action: drop";
           uses drop;
         }
       }
     }

     grouping meter {
       leaf meter-id {
         type uint16;
       }
       leaf meter-rate {
         units "bits-per-second";
         type uint64;
       }
       uses burst;
       container color {
         uses diffserv-classifier:classifier-entry-generic-attr;
       }
       uses meter-action-params;
     }

     grouping priority {
       leaf priority-level {
         type uint8;
         description
           "priority level";
       }
       leaf priority-rate {
         units "bits-per-second";
         type uint64;
       }
     }

     grouping min-rate {
       leaf min-rate {
         units "bits-per-second";
         type uint64;
       }
       description
         "min guanteed bandwidth";
     }

     grouping marking {
       leaf dscp {
         type inet:dscp;
```

```
      }
    }

    grouping max-rate {
      leaf absolute-rate {
        units "bits-per-second";
        type uint64;
      }
      uses burst;
    }

    grouping wred-threshold {
      container wred-min-thresh {
        uses threshold;
        description
          "Minimum threshold";
      }
      container wred-max-thresh {
        uses threshold;
        description
          "Maximum threshold";
      }
      leaf mark-probability {
        type uint32 {
          range "1..1000";
        }
        description
          "Mark probability";
      }
    }

    grouping randomdetect {
      leaf exp-weighting-const {
        type uint32;
        description
          "Exponential weighting constant factor for wred profile ";
      }
      leaf mode-aggregate {
        type boolean;
        default "false";
        description
          "
           Indicates aggregate mode or non-aggregate mode.
           Non-aggregate mode by default creates sub-class for each
           code-point value with different min and max threshold.
           Aggregate mode defaults to only one subclass unless
           explicitly configured by the user
          ";
```

```
       }
       leaf ecn-enabled {
         type boolean;
         default "false";
       }
       list wred-profile {
         key "wred-profile-id";
         leaf wred-profile-id {
           type uint16;
         }
         list wred-dscp {
           key "dscp-min dscp-max";
           uses dscp-range;
         }
         uses wred-threshold;
       }
     }

     augment "/diffserv-policy:policies/diffserv-policy:policy-entry
              /diffserv-policy:classifier-entry
              /diffserv-policy:classifier-action-entry-cfg
              /diffserv-policy:action-cfg-params" {
       case marking {
         container marking-cfg {
           uses marking;
         }
       }
       case priority {
         container priority-cfg {
           uses priority;
         }
       }
       case meter {
         container meter-cfg {
           list meter-list {
             key "meter-id";
             uses meter;
           }
         }
       }
       case max-rate {
         container max-rate-cfg {
           uses max-rate;
         }
       }
       case algorithmic-drop {
         choice drop-algorithm {
           case always-drop {
             container drop-cfg {
               uses drop;
             }
           }
           case tail-drop {
             container tail-drop-cfg {
               uses queuelimit;
             }
           }
           case random-detect {
             container random-detect-cfg {
               if-feature aqm-wred-support;
               uses randomdetect;
             }
           }
         }
       }
       case min-rate {
         container min-rate-cfg {
           uses min-rate;
         }
       }
     }
     augment "/if:interfaces/if:interface
              /diffserv-target:diffserv-target-entry
              /diffserv-target:diffserv-target-classifier-entry
              /diffserv-target:classifier-action-entry-cfg
              /diffserv-target:action-cfg-params" {
       case marking {
         container marking-cfg {
           uses marking;
         }
       }
       case priority {
         container priority-cfg {
           uses priority;
         }
       }
       case meter {
         container meter-cfg {
           list meter-list {
             key "meter-id";
             uses meter;
           }
         }
       }
       case max-rate {
         container max-rate-cfg {
           uses max-rate;
         }
       }
       case algorithmic-drop {
         choice drop-algorithm {
           case always-drop {
             container drop-cfg {
               uses drop;
             }
           }
           case tail-drop {
             container tail-drop-cfg {
               uses queuelimit;
             }
           }
           case random-detect {
             container random-detect-cfg {
               if-feature aqm-wred-support;
               uses randomdetect;
             }
           }
         }
       }
       case min-rate {
         container min-rate-cfg {
           uses min-rate;
         }
       }
     }
   }
```

5.4. IETF-DIFFSERV-TARGET

```
   module ietf-diffserv-target {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv-target";
     prefix diffserv-target;

     import ietf-interfaces {
       prefix if;
     }
     import ietf-diffserv-classifier {
       prefix classifier;
     }
     import ietf-diffserv-policy {
       prefix policy;
     }
     revision 2014-09-07 {
       description
         "First revision diffserv based policy applied to a target";
     }

     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";
     }

     feature target-inline-policy-config {
       description
         "This feature allows the policy configuration directly
          under a target.";
     }

     grouping policy-target-generic-attr {
       uses policy:policy-generic-attr;
       leaf direction {
         type identityref {
           base direction;
         }
       }
     }

     grouping wred-class-stats {
       leaf early-drop-pkts {
         type uint64;
         description
           "Early drop packets ";
       }
       leaf early-drop-bytes {
         type uint64;
         description
           "Early drop bytes ";
       }
     }

     grouping classifier-entry-stats {
       container classifier-entry-statistics {
         config false;
         description
           "
            This container defines the classifier filter statistics
            pertaining to each classifier entry.
           ";
         leaf classified-pkts {
           type uint64;
           description
             " Number of total packets which filtered
               to the classifier-entry";
         }
         leaf classified-bytes {
           type uint64;
           description
             " Number of total bytes which filtered
               to the classifier-entry";
         }
         leaf classified-rate {
           units "bits-per-second";
           type uint64;
           description
             " Rate of average data flow through the
               classifier-entry";
         }
         container queuing-statistics {
           description
             "queue related statistics ";
           config false;
           leaf output-pkts {
             type uint64;
             description
               "Number of packets transmitted from queue ";
           }
           leaf output-bytes {
             type uint64;
             description
               "Number of bytes transmitted from queue ";
           }
           leaf queue-size-pkts {
             type uint64;
             description
               "Number of packets currently buffered ";
           }
           leaf queue-size-bytes {
             type uint64;
             description
               "Number of bytes currently buffered ";
           }
           leaf drop-pkts {
             type uint64;
             description
               "Total number of packets dropped ";
           }
           leaf drop-bytes {
             type uint64;
             description
               "Total number of bytes dropped ";
           }
           list wred-statistics {
             key "wred-profile-id";
             description
               "WRED statistics for a dscp range ";
             leaf wred-profile-id {
               type uint16;
             }
             uses wred-class-stats;
           }
         }
       }
     }

     grouping meter-action-stats {
       list meter-action-statistics {
         description
           "Meter statistics";
         config false;
         key "meter-id";
         leaf meter-id {
           type uint16;
         }
         leaf metered-pkts {
           type uint64;
           description
             "Number of packets counted by the meter";
         }
         leaf metered-bytes {
           type uint64;
           description
             "Bytes of packets counted by the meter";
         }
         leaf metered-rate {
           units "bits-per-second";
           type uint64;
           description
             "Traffic Rate measured by the meter";
         }
       }
     }

     augment "/if:interfaces/if:interface" {
       list diffserv-target-entry {
         key "direction policy-name";
         description
           "policy target for inbound or outbound direction";
         uses policy-target-generic-attr;
         list diffserv-target-classifier-entry {
           if-feature target-inline-policy-config;
           key "classifier-entry-name parent-path";
           ordered-by user;
           leaf classifier-entry-name {
             type leafref {
               path "/classifier:classifiers/classifier:classifier-entry
                     /classifier:classifier-entry-name";
             }
           }
           leaf parent-path {
             type string;
           }
           uses classifier:classifier-entry-inline-attr;
           uses policy:classifier-action-entry-cfg;
         }
       }
     }
   }
```

6. Security Considerations

7. Acknowledgement

The editor of this document wishes to thank Fred Baker for overviewing the document and provide useful comments, Andrew Mao for the guidance and support, Fred Yip and Aleksandr Zhdankin for helpful suggestions and contributions.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, September 1999.

[RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999.

[RFC2859] Fang, W., Seddigh, N., and B. Nandy, "A Time Sliding Window Three Colour Marker (TSWTCM)", RFC 2859, June 2000.

[RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, April 2002.

[RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014.

8.2. Informative References

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

Appendix A. Open Items

The current model represents hierarchical QoS alike with the non-leaf and leaf nodes, in a scheduling hierarchy, without any restrictions of actions, such as AQM, that should not be allowed at non-leaf nodes. This is to be addressed in subsequent revisions.

Authors’ Addresses

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

Email: asechoud@cisco.com

Shitanshu Shah
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
US

Email: svshah@cisco.com

Mahesh Jethanandani
Ciena Corporation
3939 North 1st Street
San Jose, CA 95134
US

Email: mjethanandani@gmail.com

Gang Yan
Huawei Technologies
Huawei Bld., No. 156 Beiqing Rd
Beijing 100095
China

Email: yangang@huawei.com

Bing Liu
Huawei Technologies
Q14, Huawei Campus, No.156 Beiqing Rd
Beijing 100095
China

Email: Leo.liubing@huawei.com

Norm Strahle
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
US

Email: nstrahle@juniper.net