

Network Working Group
Internet Draft
Expires September 2000

M. Fine
K. McCloghrie
Cisco Systems
J. Seligson
K. Chan
Nortel Networks
S. Hahn
Intel
A. Smith
Extreme Networks
Francis Reichmeyer
IPHighway

March 10, 2000

Differentiated Services Quality of Service Policy Information Base

[draft-ietf-diffserv-pib-00.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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.''

To view the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in an Internet-Drafts Shadow Directory, see <http://www.ietf.org/shadow.html>.

1. Glossary

PRC	Policy Rule Class. A type of policy data.
PRI	Policy Rule Instance. An instance of a PRC.
PIB	Policy Information Base. The database of policy information.
PDP	Policy Decision Point. See [RAP-FRAMEWORK].
PEP	Policy Enforcement Point. See [RAP-FRAMEWORK].
PRID	Policy Rule Instance Identifier. Uniquely identifies an instance of a PRC.

2. Introduction

[SPPI] describes a structure for specifying policy information that can then be transmitted to a network device for the purpose of configuring policy at that device. The model underlying this structure is one of well defined policy rule classes and instances of these classes residing in a virtual information store called the Policy Information Base (PIB).

This document specifies a set of policy rule classes specifically for configuring QoS Policy for Differentiated Services [[DSARCH](#)].

One way to provision policy is by means of the COPS protocol [[COPS](#)] with the extensions for provisioning [[COPS-PR](#)]. This protocol supports multiple clients, each of which may provision policy for a specific policy domain such as QoS. The PRCs defined in this DiffServ QoS PIB are intended for use by the COPS-PR QoS client type. Furthermore, these PRCs are in addition to any other PIBs that may be defined for the QoS client type in the future, as well as the PRCs defined in the Framework PIB [[FR-PIB](#)]

3. DiffServ PIB Concepts

3.1. Filters, Filter Groups and Classifiers

The basis of differential QoS treatment of packets is a filter. This is simply a general specification for matching a pattern to appear in packets belonging to flows, e.g. microflows or bandwidth aggregates. Associated with each filter is a permit/deny flag which effectively gives a negation operation.

Sets of these filters are used to create classifiers. Classifiers are applied to interfaces with a direction flag to indicate an ingress or egress classifier. Filters are combined, in order, into filter groups;

filter groups are then combined, in order, to build a classifier. This allows a rudimentary classification grammar to be defined. On input, each packet is checked against the ingress classifier on the interface. Similarly, on output each packet is checked against the egress classifier on the interface. The result of the classifier then feeds into appropriate meters and actions to be applied to packets.

For each classifier, the packet is checked against the set of filter groups in the appropriate order. The detailed operation of the PIB syntax is as follows. If a packet matches a filter in the first filter group of a classifier and the sense is "permit" then the subsequent meters and actions associated with that classifier are applied to that packet and no further filters are compared. If the sense is "deny" then the rest of the filters in the current filter group are skipped and operation proceeds with the first filter of the next filter group. If the packet does not match any of the filters in the filter group then the next filter group is tried. This process is continued until a definitive match is obtained. Each classifier must cover all possible matches i.e., it must be complete.

3.2. Applying QoS Policy Using Targets

The task of applying QoS policy within a network requires the specification of several components. The flows to which QoS policy should be applied must be identified. The interfaces of the device on which the policy should be enforced must be known. A certain set of parameters to support flow metering is also required. The combination of these components provides the target against which QoS policy is to be applied. Within the context of the QoS PIB, the association between these components is defined efficiently using the Target class.

The Target class serves to logically link several other QoS policy classes. Flow classification rules, specifying behavior aggregate (BA) or multi-field (MF) classification parameters, are indirectly identified using the PRC for the appropriate classification class (e.g., IP, 802) coupled with an identifier for a specific classifier. Interface information is specified using the role combination tag, defined in the Interface Type class, to identify the group of interfaces on which classification is to be performed. The direction of packet flow on the identified interfaces is provided as well. A link to the metering component is provided using the PRC for the appropriate metering class instance.

Once a target has been defined, actions based on the classification and metering phases must be specified. Action class instances are linked

with the Target entry through the associated Meter class instance. A precedence component is also provided so that a definitive order of evaluation may be defined for Target class instances being applied to the same interface role and flow direction targets. The Target class thus functions as the integration point for the range of components used for the application of QoS policy.

3.3. Interface Modeling with Queue Sets

The traffic processing capabilities of an interface are determined by the queuing resources that are associated with the interface. These capabilities are represented abstractly using queue sets. A queue set is comprised of one or more individual queues. The PDP creates the queue sets, configures the parameters of the individual queues, configures the scheduling discipline to be used to schedule the queues and then assigns a queue set to each <interface type, role combination> tuple. In this way, the PDP sets the scheduling policy for each interface based on the role combination of the interface and the type of the interface.

In order for the PDP to configure a queue set that can be properly realized by an interface, the PEP reports to the PDP the types of interfaces it has together with various capabilities and configuration limits (such as the maximum number of queues an interface could support) of the interface types.

It should be emphasized that the PDP does not configure individual interfaces directly. Rather, it configures them indirectly by specifying the configuration for each interface type and role combination pair. It is the responsibility of the PEP to apply the queue set characteristics, and hence the interface scheduling configuration, to the individual interfaces on the basis of the type and role combination information.

3.3.1. Queue Scheduling

There are two basic scheduling disciplines supported by queue sets: priority queueing and weighted fair queueing. To support these, each queue is assigned a priority which is then used to determine a strict processing order between queues. However, several queues may be assigned the same priority. In this case, these queues form a group, called a priority group, and are scheduled using WFQ. In other words, service is given to the priority group with the highest priority that has any non-empty queue. Within a priority group queues are serviced

using WFQ.

3.3.2. Assigning Packets To Queues and Thresholds

In keeping with the DiffServ model of classifying packets into behaviour classes and then providing service suitable for that behaviour, packets are assigned to queues on the basis of their final DSCP values. Furthermore, each queue is configured with a set of thresholds to support multiple discard priorities for the PHBs in a PHB group. Packets are assigned to thresholds within a queue on the basis of their DSCPs. The PDP is responsible for this assignment of DSCP values to queues and the associated thresholds.

3.3.3. Hierarchies of Queues

Sometimes policy may require hierarchies of queues. For example, a department might have some set of traffic classes with a defined scheduling policy between these classes. Multiple departments might then share a link with there being a defined scheduling policy between traffic from the various departments.

The PIB does not support hierarchical queueing at this time. However, we expect to add this support in the future by allowing the traffic from one queue set to feed into the queues of another queue set.

3.4. IP Mapping to and from Layer 2

The PIB specifies QoS policy by assigning DSCP values to specific queues, but in order to provide a complete QoS picture, the PIB must consider that not all devices on the network are DiffServ capable, i.e., capable of setting/inspecting a packet's DSCP value. Specifically, the network might include layer 2 devices (switches) that can only support IEEE 802.1p classes of service. In order to support network configuration that consists of DiffServ capable devices and devices that can only support IEEE 802.1p, the PIB has included a mapping table that can allow the DSCP values to be mapped to specific IEEE 802.1p tag values.

DSCP ----- DSCP ----- DSCP ----- DSCP


```

----->|diffserv|----->|L2      |----->|diffserv|----->
      | router | 802.1p  |switch| 802.1p  | router | 802.1p
      ----- priority ----- priority ----- priority

```

A second case exists where packets coming into the network are arriving from a non-DiffServ enabled device and no DSCP exists with in the packet, but an 802.1p tag does exist. In the case where the DiffServ device has the ability to set a DSCP in the packet, the DiffServ router can map the layer 2 tag into a DSCP value. The PIB supports a mapping table that can be used to map from the layer 2 tag to a DSCP value. This mapping would be configured to apply to those ports where the upstream device marks packets using a L2 802.1p tag as shown in the figure below.

```

----- DSCP
-->| L2      |----->| diffserv |----->
-->| switch | 802.1p  | router  | 802.1p
      ----- priority ----- priority

```

Alternatively, the DiffServ router can have policies applied to it that cause it to reclassify the incoming packet using a MF classifier, ignoring the incoming 802.1p tag.

4. Summary of the DiffServ PIB

The DiffServ PIB consists of two modules. The first, DIFFSERV-BASE-PIB contains all the base PRCs for setting DiffServ policy, queues, classifiers, meters, etc., and also contains filters for matching IP packets. The second, the DIFFSERV-802-PIB contains PRCs to map the IEEE [802.3 class of service values to DiffServ code points and filters for matching on the datalink headers of 802.3 frames](#). Devices are not required to support the second module.

These two modules comprise several groups which are summarized in this section.

QoS Interface Group

This group consists of PRCs to indicate to the PDP the types of interface supported on the PEP in terms of their QoS capabilities and PRCs that the PDP can install in order to configure these interfaces (queues, scheduling parameters, buffer sizes, etc.) to affect the desired policy. This group describes capabilities in

terms of the types of interfaces and takes configuration in terms of interface types and role combinations [[FR-PIB](#)]; it does not deal with individual interfaces on the device.

QoS Metering Group

This group contains configuration of meters. These meters can then be used to by target classes to specify metering policy.

QoS Action Group

This group contains the policies that define the action to be taken after the result of the classification and metering. This group also contains the policies that associate the classifiers, meters and actions.

IP Classification and Policing Group

This group contains the policies that define the IP classifier elements.

[802](#) Config Group

This group configures the relationship between 802.3 class of service values and DiffServ codepoints.

[802](#) Classification and Policing Group

This group contains the policies that define the IEEE 802 classifier elements.

[5.](#) PIB Operational Overview

This section provides an operation overview of configuring DiffServ QoS policy.

After initial PEP to PDP communication setup, using [[COPS-PR](#)] for example, the PEP will provide to the PDP the PIB Policy Rule Classes (PRCs), interface types, and interface type capabilities it supports.

The PRCs supported by the PEP are reported to the PDP in the PRC Support Table, policyPrcSupportTable defined in the framework PIB. Each instance of the policyPrcSupportTable indicates a PRC that the PEP understands and for which the PDP can send class instances as part of the policy information.

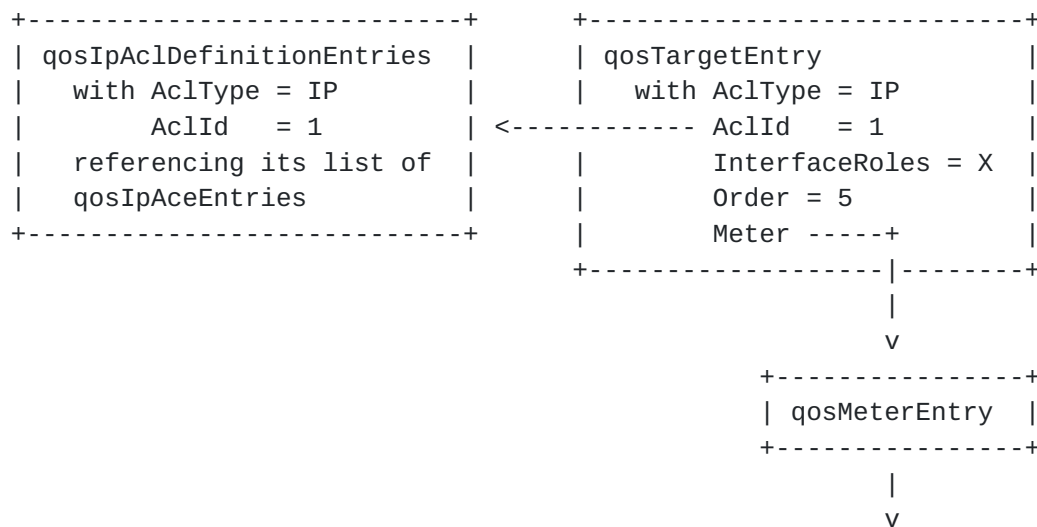
The interface types the PEP supports are described by rows in the interface type table, qosIfTypeTable. Each row, or instance of this class describes the characteristics of an interface type. The PEP

informs the PDP of these interface types and then the PDP configures the interfaces, per role combination, by means of installing queue sets.

The PDP, with knowledge of the PEP's capabilities, then provides the PEP with:

- (1) Administration domain policy information in
 - qosIfDscpAssignTable
 - qos802DscpMappingTable
 - qos802CosToDscpTable
- (2) Interface type and role specific IP policy information in
 - qosIpAceTable
 - qosIpAclDefinitionTable
 - qosActionTable
 - qosTargetTable
 - qosMeterTable
- (3) Interface type and role specific IEEE 802 policy information in
 - qos802AceTable
 - qos802AclDefinitionTable

Instances of the `qosTargetTable` define how the Traffic Conditioning Elements are combined into Traffic Conditioning Blocks, as described in [\[MODEL\]](#). Each instance of the `qosTargetTable` applies to an interface type defined by its roles and direction (ingress or egress). This is pictured in the following diagram where the InterfaceRoles X, and Y would be used by the network device to associate the traffic conditioning block with the interfaces needing each of these policies.



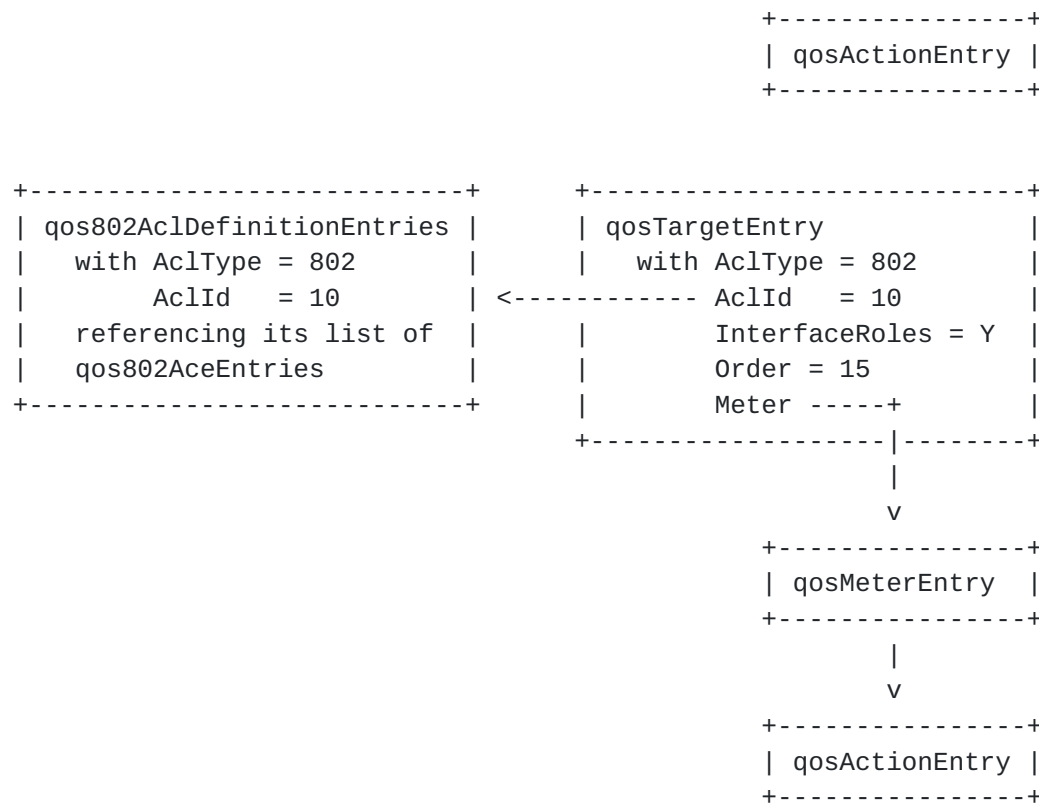


Figure 7.1 DiffServ PIB Table Relationships

Notice in the above diagram, IEEE 802 type classifiers are intermixed with the IP type classifiers, sharing the same pool of Traffic Conditioning Elements. The qosTargetTable allows use of heterogeneous classifiers with same instance of qosMeterTable. Using IP and IEEE 802 classifiers together is just one example. Other types of classifiers may be used heterogeneously.

After receiving the PIB, the PEP will associate the Classifier, Meter and Action with the corresponding interfaces supporting the specific interface type and roles.

6. PIB Definitions

NOTE

In these PIB definitions, we use the term "access control entry" (ACE) synonymous with filter, "access control list" (ACL) synonymous with filter group, and sets of ACLs synonymous with classifier.

6.1. The DiffServ Base PIB

```
DIFFSERV-BASE-PIB PIB-DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    Unsigned32, IpAddress, Integer32,  
    MODULE-IDENTITY, OBJECT-TYPE  
        FROM COPS-PR-SPPI  
    TruthValue, TEXTUAL-CONVENTION  
        FROM SNMPv2-TC  
    PolicyInstanceId, PolicyReferenceId  
        FROM COPS-PR-SPPI;  
    RoleCombination  
        FROM POLICY-FRAMEWORK-PIB;
```

```
qosPolicyIpPib MODULE-IDENTITY  
    CLIENT-TYPE { tbd    -- QoS Client Type  
    }  
    LAST-UPDATED "200003101800Z"  
    ORGANIZATION "IETF DIFFSERV WG"  
    CONTACT-INFO "  
        Michael Fine  
        Cisco Systems, Inc.  
        170 West Tasman Drive  
        San Jose, CA 95134-1706 USA  
        Phone: +1 408 527 8218  
        Email: mfine@cisco.com  
  
        Keith McCloghrie  
        Cisco Systems, Inc.  
        170 West Tasman Drive,  
        San Jose, CA 95134-1706 USA  
        Phone: +1 408 526 5260  
        Email: kzm@cisco.com  
  
        John Seligson
```


Nortel Networks, Inc.
4401 Great America Parkway
Santa Clara, CA 95054 USA
Phone: +1 408 495 2992
Email: jseligso@nortelnetworks.com"

DESCRIPTION

"The PIB module containing a set of policy rule classes that describe quality of service (QoS) policies for DiffServ. It includes general classes that may be extended by other PIB specifications as well as a set of PIB classes related to IP processing."

::= { tbd }

qosPolicyGenPibClasses OBJECT IDENTIFIER ::= { qosPolicyIpPib 1 }
qosPolicyIpPibClasses OBJECT IDENTIFIER ::= { qosPolicyIpPib 2 }

--

-- Textual Conventions

--

--

-- DiffServ Codepoint

--

Dscp ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An integer that is in the range of the DiffServ codepoint values."

SYNTAX INTEGER (0..63)

--

-- Interface types

--

QosInterfaceQueueCount ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An integer that describes the number of queues an interface supports. It is limited to the number of DSCP values."

SYNTAX INTEGER (1..64)


```
--
-- QoS Interface Group
--
--
-- This group specifies the configuration of the various interface
-- types including the configuration of queue sets, setting of
-- queueing parameters and the mapping of DSCPs to thresholds in
-- queues.

qosIfParameters OBJECT IDENTIFIER ::= { qosPolicyGenPibClasses 1 }

--
-- Interface Type Table
--

qosIfTypeTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosIfTypeEntry
    POLICY-ACCESS    notify
    STATUS           current
    DESCRIPTION
        "Interface type definitions. This class describes the types
        of interfaces that exist on the device. An interface type is
        defined by its name. Associated with each interface type is a
        set of capabilities. These capabilities are used by the PDP
        to determine suitable queue sets for interfaces of this type."

    ::= { qosIfParameters 1 }

qosIfTypeEntry OBJECT-TYPE
    SYNTAX          QosIfTypeEntry
    STATUS           current
    DESCRIPTION
        "An instance of this class describes the characteristics
        of a type of an interface."

    INDEX { qosIfTypePrid }
    UNIQUENESS { qosIfTypeName,
                 qosIfTypeCapabilityPrc }

    ::= { qosIfTypeTable 1 }

QosIfTypeEntry ::= SEQUENCE {
    qosIfTypePrid          PolicyInstanceId,
    qosIfTypeName          SnmpAdminString
```



```
        qosIfTypeCapabilityPrc  OBJECT IDENTIFIER,  
        qosIfTypeCapabilityId   PolicyReferenceId  
    }
```

```
qosIfTypePrid OBJECT-TYPE  
    SYNTAX      PolicyInstanceId  
    STATUS      current  
    DESCRIPTION  
        "An arbitrary integer index that uniquely identifies a  
        instance of the class."  
  
    ::= { qosIfTypeEntry 1 }
```

```
qosIfTypeName OBJECT-TYPE  
    SYNTAX      SnmpAdminString  
    STATUS      current  
    DESCRIPTION  
        "The name of the interface type. The interface type name is  
        the unique identifier of an interface type."  
  
    ::= { qosIfTypeEntry 2 }
```

```
qosIfTypeCapabilityPrc OBJECT-TYPE  
    SYNTAX      OBJECT IDENTIFIER  
    STATUS      current  
    DESCRIPTION  
        "A reference to a PRC that specifies a set of interface  
        capabilities. Together with the qosIfTypeCapabilityId,  
        this specifies a PRI that contains a set of capabilities  
        associated with this interface type.  
  
        A given interface type may have several set of capabilities  
        associated with it as long no two sets are of the same PRC.  
        In this way, new capabilities can easily be added to the PIB  
        as new interface types are developed by simply adding new  
        capability PRCs."  
  
    ::= { qosIfTypeEntry 3 }
```

```
qosIfTypeCapabilityId OBJECT-TYPE  
    SYNTAX      PolicyReferenceId  
    STATUS      current  
    DESCRIPTION  
        "The instance ID of the PRI of class  
        qosIfTypeCapabilityPrc containing capabilities of the
```



```
        interface."

 ::= { qosIfTypeEntry 4 }

--
-- Interface Type Role Combination Table
--

qosIfTypeRoleComboTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosIfTypeRoleComboEntry
    POLICY-ACCESS    notify
    STATUS           current
    DESCRIPTION
        "Policy for an interface may depend not only on the type
        of interface but also on its roles.  This table specifies all
        the <interface type, role combination> tuples currently on the
        device."

 ::= { qosIfParameters 2 }

qosIfTypeRoleComboEntry OBJECT-TYPE
    SYNTAX          QosIfTypeRoleComboEntry
    STATUS           current
    DESCRIPTION
        "An instance of this class describes the combinations of
        interface type and role combination."

    INDEX { qosIfTypeRoleComboPrid }
    UNIQUENESS { qosIfTypeRoleComboName,
                 qosIfTypeRoleComboRoles }

 ::= { qosIfTypeTable 1 }

QosIfTypeRoleComboEntry ::= SEQUENCE {
    qosIfTypeRoleComboPrid  PolicyInstanceId,
    qosIfTypeRoleComboIfName SnmpAdminString,
    qosIfTypeRoleComboRoles RoleCombination
}

qosIfTypeRoleComboPrid OBJECT-TYPE
    SYNTAX          PolicyInstanceId
    STATUS           current
    DESCRIPTION
        "An arbitrary integer index that uniquely identifies a
```


instance of the class."

::= { qosIfTypeRoleComboEntry 1 }

qosIfTypeRoleComboIfName OBJECT-TYPE

SYNTAX SnmpAdminString

STATUS current

DESCRIPTION

"The name of the interface type. This name must exist in qosIfTypeTable."

::= { qosIfTypeRoleComboEntry 2 }

qosIfTypeRoleComboRoles OBJECT-TYPE

SYNTAX RoleCombination

STATUS current

DESCRIPTION

"A role combination. The PEP requires policy for interface with this role combination and of type qosIfTypeRoleComboName"

::= { qosIfTypeRoleComboEntry 3 }

--

-- Interface Type Capability Tables

--

-- The Interface type capability tables define capabilities that may
-- be associated with an interface of a specific type. This PIB
-- defines three such tables: a classification capabilities table, a
-- policing capabilities table and a scheduling capabilities table.
-- Other PIBs may define other capability tables to augment the
-- capability definitions of these tables or to introduce completely
-- new capabilities.

--

-- Classification Capabilities

--

qosIfTypeClassificationCapsTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfTypeClassificationCapsEntry

POLICY-ACCESS notify

STATUS current

DESCRIPTION

"This table specifies the classification capabilities of an


```
interface type"
```

```
::= { qosIfParameters 3 }
```

```
qosIfTypeClassificationCapsEntry OBJECT-TYPE
```

```
SYNTAX          QosIfTypeClassificationEntry
```

```
STATUS          current
```

```
DESCRIPTION
```

```
"An instance of this class describes the classification
capabilities of an interface."
```

```
INDEX { qosIfTypeClassificationCapsPrid }
```

```
UNIQUENESS { qosIfTypeClassificationCaps }
```

```
::= { qosIfTypeClassificationCapsTable 1 }
```

```
QosIfTypeClassificationCapsEntry ::= SEQUENCE {
```

```
    qosIfTypeClassificationCapsPrid PolicyInstanceId,
```

```
    qosIfTypeClassificationCaps      BITS
```

```
}
```

```
qosIfTypeClassificationCapsPrid OBJECT-TYPE
```

```
SYNTAX          PolicyInstanceId
```

```
STATUS          current
```

```
DESCRIPTION
```

```
"An arbitrary integer index that uniquely identifies a
instance of the class."
```

```
::= { qosIfTypeClassificationCapsEntry 1 }
```

```
qosIfTypeClassificationCaps OBJECT-TYPE
```

```
SYNTAX          BITS {
```

```
    inputIpClassification(1),
```

```
    outputIpClassification(2),
```

```
    -- Indicates the ability to classify IP
```

```
    -- packets on ingress and on egress,
```

```
    -- respectively.
```

```
    ipAddrClassification(3),
```

```
    -- indicates the ability to classify based on
```

```
    -- IP addresses
```

```
    ipProtoClassification(4),
```

```
    -- indicates the ability to classify based on
```

```
    -- IP protocol numbers
```

```
    ipL4Classification(4)
```



```

        -- indicates the ability to classify based on
        -- IP layer 4 port numbers for UDP and TCP
    }
    STATUS          current
    DESCRIPTION
        "Bit set of supported classification capabilities.  In
        addition to these capabilities, other PIBs may define other
        capabilities that can then be specified in addition to the
        ones specified here (or instead of the ones specified here if
        none of these are specified)."
```

::= { qosIfTypeClassificationCapsEntry 2 }

```

--
-- Policing Capabilities
--

qosIfTypePolicingCapsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosIfTypePolicingCapsEntry
    POLICY-ACCESS    notify
    STATUS           current
    DESCRIPTION
        "This table specifies the policing capabilities of an
        interface type"

    ::= { qosIfParameters 4 }

qosIfTypePolicingEntry OBJECT-TYPE
    SYNTAX          QosIfTypePolicingEntry
    STATUS           current
    DESCRIPTION
        "An instance of this class describes the classification
        capabilities of an interface."

    INDEX { qosIfTypePolicingCapsPrid }
    UNIQUENESS { qosIfTypePolicingCaps }

    ::= { qosIfTypePolicingCapsTable 1 }

QosIfTypePolicingCapsEntry ::= SEQUENCE {
    qosIfTypePolicingCapsPrid      PolicyInstanceId,
    qosIfTypePolicingCaps          BITS
}
```


qosIfTypePolicingCapsPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies a instance of the class."

::= { qosIfTypePolicingCapsEntry 1 }

qosIfTypePolicingCaps OBJECT-TYPE

SYNTAX BITS {
 policeByRemarking (1),
 policeByDropping (2),
 -- These capabilities indicate if the interface
 -- can remark out of profile packets or drop them,
 -- respectively

 inputShaping (3),
 outputShaping (4)
 -- indicate if the interface can shape on ingress
 -- or on egress, respectively.

}

STATUS current

DESCRIPTION

"Bit set of supported classification capabilities. As with classification capabilities, these policing capabilities may be augmented by capabilities specified in other PRCs (in other PIBs)."

::= { qosIfTypePolicingCapsEntry 2 }

--

-- Scheduling Capabilities

--

qosIfTypeSchedulingCapsTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfTypeSchedulingCapsEntry

POLICY-ACCESS notify

STATUS current

DESCRIPTION

"This table specifies the scheduling capabilities of an interface type"


```
::= { qosIfParameters 5 }
```

qosIfTypeSchedulingEntry OBJECT-TYPE

SYNTAX QosIfTypeSchedulingEntry

STATUS current

DESCRIPTION

"An instance of this class describes the classification capabilities of an interface."

INDEX { qosIfTypeSchedulingCapsPrid }

UNIQUENESS { qosIfTypeSchedulingCapsMaxQueues,
qosIfTypeSchedulingCapsMaxThresholds,
qosIfTypeSchedulingCapsMaxPriorities,
qosIfTypeSchedulingCapsServiceDisc,
qosIfTypeSchedulingCapsMinQueueSize,
qosIfTypeSchedulingCapsMaxQueueSize,
qosIfTypeSchedulingCapsTotalQueueSize,
qosIfTypeSchedulingCapsWredCapable }

```
::= { qosIfTypeSchedulingCapsTable 1 }
```

QosIfTypeSchedulingCapsEntry ::= SEQUENCE {

qosIfTypeSchedulingCapsPrid	PolicyInstanceId,
qosIfTypeSchedulingCapsMaxQueues	INTEGER
qosIfTypeSchedulingCapsMaxThresholds	INTEGER
qosIfTypeSchedulingCapsMaxPriorities	INTEGER
qosIfTypeSchedulingCapsServiceDisc	BITS
qosIfTypeSchedulingCapsMinQueueSize	INTEGER
qosIfTypeSchedulingCapsMaxQueueSize	INTEGER
qosIfTypeSchedulingCapsTotalQueueSize	INTEGER
qosIfTypeSchedulingCapsWredCapable	TruthValue

}

qosIfTypeSchedulingCapsPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies a instance of the class."

```
::= { qosIfTypeSchedulingCapsEntry 1 }
```

qosIfTypeSchedulingCapsMaxQueues OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The maximum number of queues that this interface type can support. The queues set assigned to this interface type may not have more queues than this maximum. A value of zero means that there is no maximum."

::= { qosIfTypeSchedulingCapsEntry 2 }

qosIfTypeSchedulingCapsMaxThresholds OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The maximum number of drop thresholds that each queue supports. If the interface has a different number of thresholds for each of its queues, it must report the maximum number of thresholds any of the queues supports. The value of this attribute must be one or more."

::= { qosIfTypeSchedulingCapsEntry 3 }

qosIfTypeSchedulingCapsMaxPriorities OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The maximum number of priority groups that the the queues of the interface may be grouped into. A value of zero means there is no maximum."

::= { qosIfTypeSchedulingCapsEntry 4 }

qosIfTypeSchedulingCapsServiceDisc OBJECT-TYPE

SYNTAX BITS {
 fq(1), -- fair queueing (a.k.a. round robin)
 wfq(2) -- weighted fq (a.k.a. wrr)

STATUS current

DESCRIPTION

"The scheduling disciplines supported for servicing queues in the same priority group that the interface supports. Several general purpose and well-known queuing disciplines are supported by this attribute. Other queueing disciplines may be specified by setting this attribute to other(1) and providing another capabilities PRC specifying the other scheduling discipline.

A value of fq indicates that the interface supports fair

queuing, i.e., each queue is treated equally and is serviced in a round-robin fashion.

A value of wfq indicates that the queue is serviced using a weighted fair queuing discipline. Queues are serviced in a round robin fashion but each queue is given bandwidth in proportion to its weight.

If none is specified then the service discipline is either unspecified or specified by another capabilities PRC."

::= { qosIfTypeSchedulingCapsEntry 5 }

qosIfTypeSchedulingCapsMinQueueSize OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"Some interfaces may allow the size of a queue to be configured. This attribute specifies the minimum size the queue can be configured to specified in bytes.

Some interfaces set queue size in terms of packets. These devices must report the minimum queue size in bytes by assuming an average packet size suitable for the particular interface."

::= { qosIfTypeSchedulingCapsEntry 6 }

qosIfTypeSchedulingCapsMaxQueueSize OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"Some interfaces may allow the size of a queue to be configured. This attribute specifies the maximum size the queue can be configured to specified in bytes. As with qosIfTypeSchedulingCapsMinQueueSize, devices that set queue size in terms of packets must report the maximum queue size in bytes by assuming an average packet size suitable for the particular interface."

::= { qosIfTypeSchedulingCapsEntry 7 }

qosIfTypeSchedulingCapsTotalQueueSize OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"Some interfaces may have a limited buffer space to be share among all queues of that interface while also allowing the size of each queue to be configurable. To prevent the situation where the PDP configures the sizes of the queues in excess of the total buffer available to the interface, the PEP can report the total buffer space available with this capability. The value is the total number of bytes."

::= { qosIfTypeSchedulingCapsEntry 8 }

qosIfTypeSchedulingCapsWredCapable OBJECT-TYPE

SYNTAX TruthValue

STATUS current

DESCRIPTION

"If true, then this interface supports WRED on (at least one of) its queues. Otherwise it supports only taildrop."

::= { qosIfTypeSchedulingCapsEntry 9 }

--

-- Queue Set Assignment Table

--

qosIfTypeQueueSetAssignTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfTypeQueueSetAssignEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"Contains the assignment of queue sets to interface types per role combination."

::= { qosIfParameters 6 }

qosIfTypeQueueSetAssignEntry OBJECT-TYPE

SYNTAX QosIfTypeQueueSetAssignEntry

STATUS current

DESCRIPTION

"A conceptual row in the qosIfTypeQueueSetAssignTable."

INDEX { qosIfTypeQueueSetAssignPrid }

UNIQUENESS { qosIfTypeQueueSetAssignIfName,

qosIfTypeQueueSetAssignRoleCombo }

::= { qosIfTypeQueueSetAssignTable 1 }

QosIfTypeQueueSetAssignEntry ::= SEQUENCE {
 qosIfTypeQueueSetAssignPrid PolicyInstanceId,
 qosIfTypeQueueSetAssignName SnmpAdminString,
 qosIfTypeQueueSetAssignRoles RoleCombination,
 qosIfTypeQueueSetAssignQueueSetId INTEGER
}

qosIfTypeQueueSetAssignPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"The index that uniquely identifies this row in the table,
i.e., this PRI."

::= { qosIfTypeQueueSetAssignEntry 1 }

qosIfTypeQueueSetAssignName OBJECT-TYPE

SYNTAX SnmpAdminString

STATUS current

DESCRIPTION

"The name of an interface type. This name must exist in
qosIfTypeTable."

::= { qosIfTypeQueueSetAssignEntry 2 }

qosIfTypeQueueSetAssignRoles OBJECT-TYPE

SYNTAX RoleCombination

STATUS current

DESCRIPTION

"The role combination associated with the interface type."

::= { qosIfTypeQueueSetAssignEntry 3 }

qosIfTypeQueueSetAssignQueueSet OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The integer ID of the queue set to be assigned to all interfaces
of type specified by qosIfTypeQueueSetAssignName and with role
combination specified by qosIfTypeQueueSetAssignRoles.
This queue set must exist in qosIfQueueSetTable."


```
::= { qosIfTypeQueueSetAssignEntry 4 }
```

```
--
```

```
-- Interface Queue Table
```

```
--
```

```
-- The Interface Queue Table enumerates the individual queues and
-- groups them into queue sets. Configuration of each queue, and
-- hence an entire queue set is specified by this table.
```

```
--
```

```
qosIfQueueTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF QosIfQueueEntry
```

```
    POLICY-ACCESS   install
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "Contains configuration information for the individual queues
        of the queue sets."
```

```
::= { qosIfParameters 7 }
```

```
qosIfQueueEntry OBJECT-TYPE
```

```
    SYNTAX          QosIfQueueEntry
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "A conceptual row in the qosIfQueueTable.
```

```

        Each row identifies a specific queue within a given queue
        set and contains detailed information about the queue. Queues
        are associated with a given set through this table and
        a queue set is associated with an interface set through
        the qosIfTypeQsetAssignTable."
```

```
    INDEX { qosIfQueuePrid }
```

```
    UNIQUENESS {}
```

```
::= { qosIfQueueTable 1 }
```

```
QosIfQueueEntry ::= SEQUENCE {
```

qosIfQueuePrid	PolicyInstanceId,
qosIfQueueSetId	INTEGER,
qosIfQueueQueueSize	Unsigned32,
qosIfQueueSetThreshSet	INTEGER,
qosIfQueuePriorityGroup	INTEGER,


```
        qosIfQueueServiceDisc      INTEGER,
        qosIfQueueDrainSize         Unsigned32,
        qosIfQueueMinAbsBandwidth   Unsigned64,
        qosIfQueueBandwidthAllocation  INTEGER
```

```
    }
```

qosIfQueuePrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"The index that uniquely identifies this row in the table,
i.e., this PRI."

::= { qosIfQueueEntry 1 }

qosIfQueueSetId OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"An index that uniquely identifies a specific queue set. The
queue set that is identified with this value is associated
with an interface set through the qosIfTypeQueueSet
object in the qosIfTypeTable. The individual queues
that are members of this set all have the same value for
this attribute (i.e., they have the same set ID)."

::= { qosIfQueueEntry 2 }

qosIfQueueQueueSize OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

"The size of the queue in bytes. Some devices set queue size
in terms of packets. These devices must calculate the queue
size in packets by assuming an average packet size suitable
for the particular interface.

Some devices have a fixed size buffer to be shared among all
queues. These devices must allocate a fraction of the
total buffer space to this queue calculated as the the ratio
of the queue size to the sum of the queue sizes for the
interface."

::= { qosIfQueueEntry 3 }

qosIfQueueThreshSet OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The threshold set in the threshold set table that is to be used to configure the thresholds of this queue. The threshold set specifies how to configure the taildrop or RED thresholds for this queue.

"The threshold set may contain less thresholds than the queue actually supports. In this case the queue is free to configure the extra thresholds any way it likes since no packets will ever be assigned to those thresholds."

::= { qosIfQueueEntry 4 }

qosIfQueuePriorityGroup OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"This attribute specifies the priority group that the queue belongs to. Queues with a larger priority group number are given a higher priority than those with a smaller group number. For example, a queue in priority group 2 will be serviced (i.e., drained) before some other queue with a group number of 1.

Queues with the same priority group number have the same priority. For these another scheduling discipline (other than priority scheduling) must be specified. This is done with the qosIfQueueServiceDisc attribute."

::= { qosIfQueueEntry 5 }

qosIfQueueServiceDisc OBJECT-TYPE

SYNTAX INTEGER {
 na(1), -- only one queue in group
 other(2), -- specified by augmented attributes
 fq(3), -- Fair Queuing
 wfq(4) -- Weighted Fair Queuing
}

STATUS current

DESCRIPTION

"This attribute identifies the service discipline used to service the queues in the same priority group. It must have

the same value for all queues in the priority group. Several general purpose and well-known queuing disciplines are supported by this attribute. Queuing disciplines that differ from those that are supported by this attribute are specified by setting this attribute to other(1) and augmenting this PRC with additional attributes to specify the desired service discipline.

As an example, an interface that is associated with a queue set supporting two priority queues and three queues that are serviced using WFQ would be modeled as follows:

Id	Q Discipline	Q Drain Size	Priority Group
22	na(1)	-	3
23	na(1)	-	2
24	wfq(3)	500	1
25	wfq(3)	350	1
26	wfq(3)	150	1

The queue set presented in this example would service all queued traffic in queue 22 first, followed by all of the queued traffic in queue 23. Next the queued traffic in queues 24 through 26 would be serviced in a round robin fashion with queue 24 receiving 50% of the available bandwidth, queue 25 receiving 35% of the available bandwidth and queue 26 receiving 15% of the available bandwidth. This example is presented for expository purposes and has been simplified accordingly.

Note that, in this example, queues 24, 25 and 26 form a priority group. The qosIfQueueDrainSize attribute is used to determine the additional processing characteristics of the individual queues in a this priority group."

```
::= { qosIfQueueEntry 6 }
```

qosIfQueueDrainSize OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

"The maximum number of bytes that may be drained from the queue in one cycle. The percentage of the interface bandwidth allocated to this queue can be calculated from this attribute and the sum of the drain sizes of all the queues in a specific priority group in a queue set.

This attribute when compared with the drain size of other queues, represents the minimum bandwidth available to this queue. The minimum bandwidth specified in absolute terms is specified by the attribute qosIfQueueMinAbsBandwidth. Which of these two applies is specified by the attribute qosIfQueueBandwidthAllocation."

::= { qosIfQueueEntry 7 }

qosIfQueueMinAbsBandwidth OBJECT-TYPE

SYNTAX Unsigned64

STATUS current

DESCRIPTION

"The maximum interface bandwidth that is available for consumption when servicing this queue. This bandwidth is specified in terms of bits per second.

This attribute represents the absolute bandwidth that is available to a given queue. The relative bandwidth that is available to a given queue, with respect to other queues with which it is associated, is specified by the attribute qosIfQueueDrainSize. Which of these two applies is specified by the attribute qosIfQueueBandwidthAllocation.

::= { qosIfQueueEntry 8 }

qosIfQueueBandwidthAllocation OBJECT-TYPE

SYNTAX INTEGER {
absolute(1), --use qosIfQueueMinAbsBandwidth
relative(2) --use qosIfQueueDrainSize
}

STATUS current

DESCRIPTION

"This attribute specifies whether to configure the queue for an absolute bandwidth limit or one that is relative to other queues of the priority group. i.e., whether to configure the queue using qosIfQueueMinAbsBandwidth or qosIfQueueDrainSize."

If some queues have their bandwidth requirement specified in absolute terms and others in relative terms then the requirements of the absolute specification is met first. That is, the drain sizes of the absolute queues must be calculated based on the interface speed so as to ensure the absolute bandwidth requirement.


```
::= { qosIfQueueEntry 9 }
```

```
--
-- Interface Threshold Table
--
-- The Interface Threshold Table enumerates the individual thresholds
-- and groups them into sets that can be applied to queues.
-- Configuration of individual thresholds and hence the threshold sets
-- of individual queues, is done through this table.
--
```

```
qosIfThresholdTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosIfThresholdEntry
    POLICY-ACCESS    install
    STATUS           current
    DESCRIPTION
        "Contains configuration information for the individual thresholds
        of the threshold sets."

    ::= { qosIfParameters 8 }
```

```
qosIfThresholdEntry OBJECT-TYPE
    SYNTAX          QosIfThresholdEntry
    STATUS           current
    DESCRIPTION
        "A conceptual row in the qosIfThresholdTable.

        Each row identifies a specific threshold within a given
        set and contains detailed information about the
        threshold. Threshold sets are associated with a queue set through
        the qosIfQueueThreshSet attribute of the qosIfQueueTable."

    INDEX { qosIfThresholdPrid }
    UNIQUENESS { qosIfThresholdSetId,
                  qosIfThresholdDropMethod,
                  qosIfThresholdMinThresh,
                  qosIfThresholdMaxThresh }

    ::= { qosIfThresholdSetTable 1 }
```

```
QosIfThresholdSetEntry ::= SEQUENCE {
    qosIfThresholdPrid      PolicyInstanceId,
    qosIfThresholdSetId     INTEGER,
    qosIfThresholdDropMethod INTEGER,
```



```
        qosIfThresholdMinThresh    INTEGER,
        qosIfThresholdMaxThresh    INTEGER
    }

qosIfThresholdPrid OBJECT-TYPE
    SYNTAX      PolicyInstanceId
    STATUS      current
    DESCRIPTION
        "The index that uniquely identifies this row in the table,
        i.e., this PRI."

    ::= { qosIfThresholdEntry 1 }

qosIfThresholdSetId OBJECT-TYPE
    SYNTAX      INTEGER
    STATUS      current
    DESCRIPTION
        "An index that uniquely identifies a specific threshold set.
        The individual thresholds that are members of this set all
        have the same value for this attribute (i.e., they have the
        same set ID)."

    ::= { qosIfThresholdEntry 2 }

qosIfThresholdDropMethod OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    tailDrop(2),
                    randomDrop(3)
                }
    STATUS      current
    DESCRIPTION
        "The drop method to apply to packets exceeding the threshold.
        If the mechanism is other then another policy may be specified
        by an additional attribute augmenting this table."

    ::= { qosIfThresholdEntry 3 }

qosIfThresholdMinThresh OBJECT-TYPE
    SYNTAX      INTEGER
    STATUS      current
    DESCRIPTION
        "The queue depth below which no packets are dropped.  If the
        queue depth is above this value and below the value of
        qosIfThresholdMaxThresh then packets assigned to this
```


threshold are dropped randomly by the random drop process if random drop is in effect. If tail drop is in effect, this attribute has no relevance."

::= { qosIfThresholdEntry 4 }

qosIfThresholdMaxThresh OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The queue depth above which all packets assigned to this threshold are dropped."

::= { qosIfThresholdEntry 5 }

--

-- DSCP Assignment Table

--

-- Supports the assignment of DSCPs to queues and thresholds for each
-- interface type indirectly through a DSCP map table. So, after
-- classification and metering, when the packet has a final DSCP mark,
-- the packet is enqueued on the appropriate queue at the appropriated
-- threshold based on the mapping of the DSCP to thresholds in
-- queues.

--

qosIfDscpAssignTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfDscpAssignEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"Assigns a DSCP map specified in qosIfDscpMapTable an interface type with a specific role combination. This map then determines to which queue and threshold a packet with a given DSCP gets assigned.

We use a map rather than assigning individual DSCPs in this table because then the map can be used for multiple interface type and role combo combinations."

::= { qosIfParameters 9 }

qosIfDscpAssignEntry OBJECT-TYPE

SYNTAX QosIfDscpAssignEntry
STATUS current
DESCRIPTION
 "An instance of the qosIfDscpAssign class."

INDEX { qosIfDscpAssignPrid }
UNIQUENESS { qosIfDscpAssignName,
 qosIfDscpAssignRoles }

::= { qosIfDscpAssignTable 1 }

QosIfDscpAssignEntry ::= SEQUENCE {
 qosIfDscpAssignPrid PolicyInstanceId,
 qosIfDscpAssignName SnmpAdminString,
 qosIfDscpAssignRoles RoleCombination,
 qosIfDscpAssignDscpMap INTEGER
}

qosIfDscpAssignPrid OBJECT-TYPE

SYNTAX PolicyInstanceId
STATUS current
DESCRIPTION
 "An index that is used to uniquely identify the
 instance of the qosIfDscpAssign class."

::= { qosIfDscpAssignEntry 1 }

qosIfDscpAssignName OBJECT-TYPE

SYNTAX SnmpAdminString
STATUS current
DESCRIPTION
 "The type of interfaces to which this PRI applies."

::= { qosIfDscpAssignEntry 2 }

qosIfDscpAssignRoles OBJECT-TYPE

SYNTAX RoleCombination
STATUS current
DESCRIPTION
 "The role combination with which an interface must be
 configured to which this PRI applies."

::= { qosIfDscpAssignEntry 2 }

qosIfDscpAssignDscpMap OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"The DSCP map to apply to interfaces of type
qosIfDscpAssignName and role combo qosIfDscpAssignRoles."

::= { qosIfDscpAssignEntry 3 }

--

-- DSCP to Queue and Threshold Mapping Table

--

-- Supports the assignment of DSCPs to queues and thresholds for each
-- interface type

--

qosIfDscpMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfDscpMapEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"Assigns DSCP values to queues and thresholds for an arbitrary
DSCP map. This map can then be assigned to various interface
and role combination pairs."

::= { qosIfParameters 10 }

qosIfDscpMapEntry OBJECT-TYPE

SYNTAX QosIfDscpMapEntry

STATUS current

DESCRIPTION

"An instance of the qosIfDscpMap class."

INDEX { qosIfDscpMapPrid }

UNIQUENESS { qosIfDscpMapMapId,
 qosIfDscpMapDscp }

::= { qosIfDscpMapTable 1 }

QosIfDscpMapEntry ::= SEQUENCE {

 qosIfDscpMapPrid PolicyInstanceId,

 qosIfDscpMapMapId INTEGER,

 qosIfDscpMapDscp Dscp,

 qosIfDscpMapQueue PolicyReferenceId,


```
        qosIfDscpMapThresh      PolicyReferenceId
    }
```

qosIfDscpMapPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An index that is used to uniquely identify the instance of the qosIfDscpMap class."

::= { qosIfDscpMapEntry 1 }

qosIfDscpMapMapId OBJECT-TYPE

SYNTAX INTEGER

STATUS current

DESCRIPTION

"An integer that identifies the DSCP map to which this PRI belongs."

::= { qosIfDscpMapEntry 2 }

qosIfDscpMapDscp OBJECT-TYPE

SYNTAX Dscp

STATUS current

DESCRIPTION

"The DSCP that is being assigned to a queue and threshold by this PRI."

::= { qosIfDscpMapEntry 3 }

qosIfDscpMapQueue OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute maps the DSCP specified by qosIfDscpMapDscp to the queue identified by qosIfQueuePrid in qosIfQueueTable. For a given DSCP map, all the queues must belong to a single queue set."

::= { qosIfDscpMapEntry 4 }

qosIfDscpMapThresh OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute maps the DSCP specified by qosIfDscpMapDscp to the threshold identified by qosIfThresholdId in qosIfThresholdTable." The threshold set to which this threshold belongs must be assigned to the queue specified by qosIfDscpMapQueue."

::= { qosIfDscpMapEntry 5 }

--

-- QoS Meter Table

--

-- The QoS Meter Table contains metering specifications that
-- can be used to provide an acceptable flow bandwidth
-- dimension to the Target table.

--

qosMeter OBJECT IDENTIFIER ::= { qosPolicyGenPibClasses 2 }

qosMeterTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosMeterEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"Contains the current set of configured meters. The meters are associated with a classifier during operation through the QoS Target Table."

INSTALL-ERRORS {

invalidCommittedData(1),

invalidPeakData(2)

}

::= { qosMeter 1 }

qosMeterEntry OBJECT-TYPE

SYNTAX QosMeterEntry

STATUS current

DESCRIPTION

"General metering definitions. Each entry specifies an instance of the qosMeter class which specifies metering information in terms of traffic stream bandwidth parameters. An entry can thus be used to support traffic metering based on the specified

service level specification."

INDEX { qosMeterPrid }

UNIQUENESS { qosMeterDataSpecification,
 qosMeterCommittedRate,
 qosMeterCommittedBurst,
 qosMeterPeakRate,
 qosMeterPeakBurst,
 qosMeterHighConfAction,
 qosMeterMedConfAction,
 qosMeterLowConfAction }

::= { qosMeterTable 1 }

QosMeterEntry ::= SEQUENCE {
 qosMeterPrid PolicyInstanceId,
 qosMeterDataSpecification INTEGER,
 qosMeterCommittedRate Unsigned32,
 qosMeterCommittedBurst Unsigned32,
 qosMeterPeakRate Unsigned32,
 qosMeterPeakBurst Unsigned32,
 qosMeterHighConfAction PolicyReferenceId,
 qosMeterMedConfAction PolicyReferenceId,
 qosMeterLowConfAction PolicyReferenceId
 }

qosMeterPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies the instance of the qosMeter class. Meters are associated with specific flows using this attribute through the qosTargetMeter attribute in the QoS Target class."

::= { qosMeterEntry 1 }

qosMeterDataSpecification OBJECT-TYPE

SYNTAX INTEGER {
 noMeterData(1), -- no metering reqd
 committedData(2), -- committed rate only
 peakData(3) -- committed and peak
 }

STATUS current

DESCRIPTION

"Specifies the metering data, and thus the actions, that are defined in a given entry.

A value of noMeterData(1) indicates that no flow metering is necessary. All flows associated with this meter entry are considered to be at a high level of conformance.

A value of committedData(2) indicates that committed rate and committed burst information has been specified and will be applied to associated flows. No peak rate and burst information has been specified meaning that two levels of conformance (high, medium) are supported.

A value of peakData(3) indicates that peak rate and peak burst information has been provided in addition to the committed rate and committed burst information. All provided information will be applied to associated flows meaning that three levels of conformance (high, medium, low) are supported."

::= { qosMeterEntry 2 }

qosMeterCommittedRate OBJECT-TYPE

SYNTAX Unsigned32 (0..'ffffffff'h)

STATUS current

DESCRIPTION

"This object represents the committed information rate (CIR) against which associated traffic streams will be metered. The CIR specifies the rate at which incoming traffic can arrive to be considered to be at a high level of conformance. Typically, this value specifies the rate at which tokens are added to a token bucket used to meter received flows.

This object specifies a rate in bytes per second units such that, for example, a value of 100 equates to a committed information rate of 100 bytes per second.

Committed rate (and burst) information must be present if the qosMeterDataSpecification object has the value committedData(2) or peakRate(3). This, in turn, requires that at least both high and medium conformance actions be specified."


```
::= { qosMeterEntry 3 }
```

qosMeterCommittedBurst OBJECT-TYPE

SYNTAX Unsigned32 (0..'ffffffff'h)

STATUS current

DESCRIPTION

"This object represents the committed burst size (CBS) against which associated traffic streams will be metered. The CBS specifies the maximum burst size that is supported for flows to be considered to be at a high level of conformance. Typically, this value represents the maximum number of tokens in a token bucket.

This object specifies flow data in bytes per second units such that, for example, a value of 100 equates to a committed information rate of 100 bytes per second.

Committed burst (and rate) information must be present if the qosMeterDataSpecification object has the value committedData(2) or peakRate(3). This, in turn, requires that at least both high and medium conformance actions be specified."

```
::= { qosMeterEntry 4 }
```

qosMeterPeakRate OBJECT-TYPE

SYNTAX Unsigned32 (0..'ffffffff'h)

STATUS current

DESCRIPTION

"This object represents the peak information rate (PIR) against which associated traffic streams will be metered. The PIR specifies the rate at which incoming traffic can arrive to be considered to be at a medium level of conformance. Typically, this value specifies the rate at which tokens are added to a token bucket used to meter received flows.

This object specifies a rate in bytes per second units such that, for example, a value of 100 equates to a committed information rate of 100 bytes per second.

Peak rate (and burst) information must be present if the qosMeterDataSpecification object has the value

peakData(3). This, in turn, requires that high, medium and low conformance actions be specified."

::= { qosMeterEntry 5 }

qosMeterPeakBurst OBJECT-TYPE

SYNTAX Unsigned32 (0..'ffffffff'h)

STATUS current

DESCRIPTION

"This object represents the peak burst size (PBS) against which associated traffic streams will be metered. The CBS specifies the maximum burst size that is supported for flows to be considered to be at a medium level of conformance. Typically, this value represents the maximum number of tokens in a token bucket.

This object specifies flow data in bytes per second units such that, for example, a value of 100 equates to a committed information rate of 100 bytes per second.

Peak burst (and rate) information must be present if the qosMeterDataSpecification object has the value peakData(3). This, in turn, requires that high, medium and low conformance actions be specified."

::= { qosMeterEntry 6 }

qosMeterHighConfAction OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the action that is to be initiated for flows that are determined to have a high level of conformance with regard to metering criteria being applied to the flow.

Actions must be defined in the qosActionTable prior to being referenced by this attribute. A valid value for this attribute must always be provided."

::= { qosMeterEntry 7 }

qosMeterMedConfAction OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the action that is to be initiated for flows that are determined to have a medium level of conformance with regard to metering criteria being applied to the flow.

Actions must be defined in the qosActionTable prior to being referenced by this attribute. A valid value for this attribute must be provided if the value of the associated qosMeterDataSpecification object is committedRate(2) or peakRate(3)."

::= { qosMeterEntry 8 }

qosMeterLowConfAction OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the action that is to be initiated for flows that are determined to have a low level of conformance with regard to metering criteria being applied to the flow.

Actions must be defined in the qosActionTable prior to being referenced by this attribute. A valid value for this attribute must be provided if the value of the associated qosMeterDataSpecification object is peakRate(3)."

::= { qosMeterEntry 9 }

--

-- The Generic QoS ACL Action Group

--

qosAction OBJECT IDENTIFIER ::= { qosPolicyGenPibClasses 3 }

--

-- The QoS Action Table

--

-- The QoS Action Table describes actions that are associated with


```
-- specific IP, IEEE 802 and other ACLs through the QoS Target
-- Table. An action specification may be simple (i.e., a single
-- action) or complex (i.e., multiple actions that are performed
-- in "parallel").
--
```

qosActionTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosActionEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"Contains the current set of configured actions. The actions are associated with IP, IEEE 802 and other ACLs and interfaces during operation."

::= { qosAction 1 }

qosActionEntry OBJECT-TYPE

SYNTAX QosActionEntry

STATUS current

DESCRIPTION

"General action definitions. Each entry specifies an instance of the qosAction class which describes (potentially) several distinct action attributes. Each action is taken individually regarding the data in question. Several actions can be taken for a single frame.

An instance of this class can not be deleted while it is being referenced in a target instance in another class. This class may be extended with actions that apply to specific QoS policies (e.g., IP, IEEE 802, security) using augmentation."

INDEX { qosActionPrid }

UNIQUENESS { qosActionDrop,
 qosActionUpdatedDSCP,
 qosActionMeter }

::= { qosActionTable 1 }

QosActionEntry ::= SEQUENCE {

 qosActionPrid PolicyInstanceId,

 qosActionDrop TruthValue,

 qosActionUpdatedDSCP Integer32,

 qosActionMeter PolicyReferenceId

}

qosActionPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies the instance of the QoS Action class. Class instances may not be contiguous. Actions are associated with Target instances in other classes (e.g., the QoS Target class) using this attribute."

::= { qosActionEntry 1 }

qosActionDrop OBJECT-TYPE

SYNTAX TruthValue

STATUS current

DESCRIPTION

"This action attribute, when specified, will cause the frame being evaluated to be dropped if the value is 'true(1)'. A value of 'false(2)' indicates that this action will not be initiated (i.e., the frame will not be dropped) based on this attribute."

Prior to discarding a packet, other actions that have been specified should be performed if they make protocol sense. For example, requests for traffic mirroring (if such an action is supported by a device) should be honored. However, updating protocol header values will typically not be necessary."

::= { qosActionEntry 2 }

qosActionUpdatedSCP OBJECT-TYPE

SYNTAX Integer32 (-1 | 0..63)

STATUS current

DESCRIPTION

"This action component, when specified, will cause the value contained in the Differentiated Services (DS) field of an associated IP datagram to be updated with the value of this object."

A value of -1 indicates that this action component has not been set to an appropriate value and should not be used for action initiation. The DSCP should remain unchanged."

::= { qosActionEntry 3 }

qosActionMeter OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This action component, when specified, will identify another level of metering that should be applied to the given flow. This action is only taken if it is not in conflict with other specified actions, i.e., qosActionDrop.

A value of 0 indicates that an additional metering component has not been specified. No additional metering is thus required."

::= { qosActionEntry 4 }

--

-- The QoS Target Table

--

-- The QoS Target Table supports the association of ACLs, interfaces and actions. It allows ACL class instances, as defined in various ACL Definition classes, to be associated with specific interfaces/flow direction (based on interface role combination and traffic direction) and actions to be performed based on traffic classification. Furthermore, it allows heterogeneous ACL Definition class instances (e.g., IP, IEEE 802, security) to be applied to the same interface group in a prescribed order of precedence.

--

qosTargetTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosTargetEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"A class that applies a set of ACLs to interfaces specifying, for each interface, the precedence order of the ACL with respect to other ACLs applied to the same interface and, for each ACL, the action to take for a packet that matches a permit ACE in that ACL. Interfaces are specified abstractly in terms of interface roles.

This class may contain ACLs that specify different types of traffic classification (e.g., IP ACLs and IEEE 802 ACLs

defined in their respective definition tables). An ACL is identified by its class and instance within that class. An ACL association is formed when ACLs apply to the same interfaces, as determined by the specified interface role and direction. ACL evaluation precedence within an association is determined by the precedence attribute."

```
INSTALL-ERRORS {
    priPrecedenceConflict(1) -- precedence conflict detected
}
```

```
::= { qosAction 2 }
```

qosTargetEntry OBJECT-TYPE

SYNTAX QosTargetEntry

STATUS current

DESCRIPTION

"An instance of the qosTarget class. Instance creation may be prohibited based on the status of certain class attributes which must exist prior to class instantiation."

INDEX { qosTargetPrid }

UNIQUENESS { qosTargetAclId,
qosTargetAclType,
qosTargetInterfaceRoles,
qosTargetInterfaceDirection }

```
::= { qosTargetTable 1 }
```

QosTargetEntry ::= SEQUENCE {

qosTargetPrid	PolicyInstanceId,
qosTargetAclId	PolicyReferenceId,
qosTargetAclType	OBJECT IDENTIFIER,
qosTargetInterfaceRoles	RoleCombination,
qosTargetInterfaceDirection	INTEGER,
qosTargetOrder	Unsigned32,
qosTargetMeter	PolicyReferenceId

}

qosTargetPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies the instance of the QoS Target class."

::= { qosTargetEntry 1 }

qosTargetAclId OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the ACL that is associated with this target. It identifies (potentially many) ACL class instances in a specific ACL Definition table where ACLs, and their associated ACEs, are defined. For example, instances in the qosIpAclDefinitionTable are identified by setting the value of this object equal to the qosIpAclDefinitionAclId of the instances being targeted. This value, together with the value of the corresponding qosTargetAclType attribute, uniquely identifies one or more instances of a specific ACL Definition class.

Attempting to specify an unknown ACL class instance will result in an appropriate error indication being returned to the entity that is attempting to install the conflicting entry. For example, a 'priUnknown(2)' error indication is returned to the policy server in this situation."

::= { qosTargetEntry 2 }

qosTargetAclType OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

STATUS current

DESCRIPTION

"The ACL Definition class that is being referenced by this instance of the ACL Target class. This policy class identifier, together with the corresponding qosTargetAclId attribute, uniquely identifies instances of a specific ACL Definition class.

The object identifier value of this attribute must exist in the policyPrcSupportTable."

::= { qosTargetEntry 3 }

qosTargetInterfaceRoles OBJECT-TYPE

SYNTAX RoleCombination

STATUS current

DESCRIPTION

"The interfaces to which this ACL applies specified in terms of a set of roles. The role combination specified by this attribute must exist in the qosInterfaceTypeTable prior to being association with an instance of this class."

```
::= { qosTargetEntry 4 }
```

qosTargetInterfaceDirection OBJECT-TYPE

```
SYNTAX          INTEGER {
                    in(1),
                    out(2)
                  }
```

```
STATUS          current
```

DESCRIPTION

"The direction of packet flow at the interface in question to which this ACL applies."

```
::= { qosTargetEntry 5 }
```

qosTargetOrder OBJECT-TYPE

```
SYNTAX          Unsigned32
```

```
STATUS          current
```

DESCRIPTION

"An integer that determines the precedence order of this ACL in the list of ACLs applied to interfaces of the specified role combination. An ACL with a given precedence order is positioned in the list before one with a higher-valued precedence order."

As an example, consider the following ACL Target association:

Index	IfRoleCombo	IfDirection	AclId	AclType	Order
14	'eth1000+L2+L3'	'in'	8	'802'	1
15	'eth1000+L2+L3'	'in'	3	'802'	2
16	'eth1000+L2+L3'	'in'	12	'IP'	3
17	'eth1000+L2+L3'	'in'	6	'IP'	4
18	'eth1000+L2+L3'	'in'	21	'IP'	5

Five distinct ACL specifications, 3 from an IP ACL Definition class and 2 from an IEEE 802 ACL Definition class, form an Acl Target association (e.g., based on the specified interface role combination and direction attributes) with a prescribed order of evaluation. The AclType and AclId attributes identify the ACL Definition instances in their

respective classes.

Precedence values within an association must be unique otherwise instance installation will be prohibited and an error value will be returned."

::= { qosTargetEntry 6 }

qosTargetMeter OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the meter that is associated with this QoS Target instance. Meters are defined in the qosMeterTable. The corresponding instance in the qosMeter class (i.e., the class instance where the qosMeterPrid is equal to the value of this object) must exist prior to being associated with a Target entry."

::= { qosTargetEntry 7 }

--

-- The IP Classification and Policing Group

--

qosIpQos OBJECT IDENTIFIER ::= { qosPolicyIpPibClasses 1 }

-- The IP ACE Table

qosIpAceTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIpAceEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"ACE definitions. A packet has to match all fields in an ACE. Wildcards may be specified for those fields that are not relevant."

INSTALL-ERRORS {
 invalidDstL4PortData(1),
 invalidSrcL4PortData(2)
}

::= { qosIpQos 1 }

qosIpAceEntry OBJECT-TYPE

SYNTAX QosIpAceEntry

STATUS current

DESCRIPTION

"An instance of the qosIpAce class."

INDEX { qosIpAcePrid }

UNIQUENESS { qosIpAceDstAddr,
 qosIpAceDstAddrMask,
 qosIpAceSrcAddr,
 qosIpAceSrcAddrMask,
 qosIpAceDscp,
 qosIpAceProtocol,
 qosIpAceDstL4PortMin,
 qosIpAceDstL4PortMax,
 qosIpAceSrcL4PortMin,
 qosIpAceSrcL4PortMax }

::= { qosIpAceTable 1 }

QosIpAceEntry ::= SEQUENCE {

qosIpAcePrid PolicyInstanceId,
 qosIpAceDstAddr IpAddress,
 qosIpAceDstAddrMask IpAddress,
 qosIpAceSrcAddr IpAddress,
 qosIpAceSrcAddrMask IpAddress,
 qosIpAceDscp Integer32,
 qosIpAceProtocol INTEGER,
 qosIpAceDstL4PortMin INTEGER,
 qosIpAceDstL4PortMax INTEGER,
 qosIpAceSrcL4PortMin INTEGER,
 qosIpAceSrcL4PortMax INTEGER,
 qosIpAcePermit TruthValue

}

qosIpAcePrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An integer index to uniquely identify this ACE among all the
 ACEs."

::= { qosIpAceEntry 1 }

qosIpAceDstAddr OBJECT-TYPE

SYNTAX IpAddress
STATUS current
DESCRIPTION
 "The IP address to match against the packet's destination IP
 address."

::= { qosIpAceEntry 2 }

qosIpAceDstAddrMask OBJECT-TYPE

SYNTAX IpAddress
STATUS current
DESCRIPTION
 "A mask for the matching of the destination IP address.
 A zero bit in the mask means that the corresponding bit in
 the address always matches."

::= { qosIpAceEntry 3 }

qosIpAceSrcAddr OBJECT-TYPE

SYNTAX IpAddress
STATUS current
DESCRIPTION
 "The IP address to match against the packet's source IP
 address."

::= { qosIpAceEntry 4 }

qosIpAceSrcAddrMask OBJECT-TYPE

SYNTAX IpAddress
STATUS current
DESCRIPTION
 "A mask for the matching of the source IP address."

::= { qosIpAceEntry 5 }

qosIpAceDscp OBJECT-TYPE

SYNTAX Integer32 (-1 | 0..63)
STATUS current
DESCRIPTION
 "The value that the DSCP in the packet can have and
 match this ACE. A value of -1 indicates that a specific
 DSCP value has not been defined and thus all DSCP values
 are considered a match."

::= { qosIpAceEntry 6 }

qosIpAceProtocol OBJECT-TYPE

SYNTAX INTEGER (0..255)

STATUS current

DESCRIPTION

"The IP protocol to match against the packet's protocol.
A value of zero means match all."

::= { qosIpAceEntry 7 }

qosIpAceDstL4PortMin OBJECT-TYPE

SYNTAX INTEGER (0..65535)

STATUS current

DESCRIPTION

"The minimum value that the packet's layer 4 destination
port number can have and match this ACE."

::= { qosIpAceEntry 8 }

qosIpAceDstL4PortMax OBJECT-TYPE

SYNTAX INTEGER (0..65535)

STATUS current

DESCRIPTION

"The maximum value that the packet's layer 4 destination
port number can have and match this ACE. This value must be
equal to or greater that the value specified for this ACE in
qosIpAceDstL4PortMin."

::= { qosIpAceEntry 9 }

qosIpAceSrcL4PortMin OBJECT-TYPE

SYNTAX INTEGER (0..65535)

STATUS current

DESCRIPTION

"The minimum value that the packet's layer 4 source port
number can have and match this ACE."

::= { qosIpAceEntry 10 }

qosIpAceSrcL4PortMax OBJECT-TYPE

SYNTAX INTEGER (0..65535)

STATUS current

DESCRIPTION

"The maximum value that the packet's layer 4 source port
number can have and match this ACE. This value must be equal
to or greater that the value specified for this ACE in

qosIpAceSrcL4PortMin."

::= { qosIpAceEntry 11 }

qosIpAcePermit OBJECT-TYPE

SYNTAX TruthValue

STATUS current

DESCRIPTION

"If the packet matches this ACE and the value of this attribute is true, then the matching process terminates and the QoS associated with this ACE (indirectly through the ACL) is applied to the packet. If the value of this attribute is false, then no more ACEs in this ACL are compared to this packet and matching continues with the first ACE of the next ACL."

::= { qosIpAceEntry 12 }

--

-- The IP ACL Definition Table

--

qosIpAclDefinitionTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIpAclDefinitionEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"A class that defines a set of ACLs each being an ordered list of ACEs. Each instance of this class identifies one ACE of an ACL and the precedence order of that ACE with respect to other ACEs in the same ACL."

INSTALL-ERRORS {

priPrecedenceConflict(1) -- precedence conflict detected
}

::= { qosIpQos 2 }

qosIpAclDefinitionEntry OBJECT-TYPE

SYNTAX QosIpAclDefinitionEntry

STATUS current

DESCRIPTION

"An instance of the qosIpAclDefinition class."

INDEX { qosIpAclDefinitionPrid }


```
UNIQUENESS { qosIpAclDefinitionAclId,  
             qosIpAclDefinitionAceId }
```

```
::= { qosIpAclDefinitionTable 1 }
```

```
QosIpAclDefinitionEntry ::= SEQUENCE {  
    qosIpAclDefinitionPrid      PolicyInstanceId,  
    qosIpAclDefinitionAclId     INTEGER,  
    qosIpAclDefinitionAceId     PolicyReferenceId,  
    qosIpAclDefinitionAceOrder Unsigned32  
}
```

```
qosIpAclDefinitionPrid OBJECT-TYPE  
    SYNTAX      PolicyInstanceId  
    STATUS      current  
    DESCRIPTION  
        "Unique index of this policy rule instance."
```

```
::= { qosIpAclDefinitionEntry 1 }
```

```
qosIpAclDefinitionAclId OBJECT-TYPE  
    SYNTAX      INTEGER  
    STATUS      current  
    DESCRIPTION  
        "An ID for this ACL. There will be one instance of  
        the class qosIpAclDefinition with this ID for each ACE in  
        the ACL per role combination."
```

```
::= { qosIpAclDefinitionEntry 2 }
```

```
qosIpAclDefinitionAceId OBJECT-TYPE  
    SYNTAX      PolicyReferenceId  
    STATUS      current  
    DESCRIPTION  
        "This attribute specifies the ACE in the qosIpAceTable that  
        is in the ACL specified by qosIpAclDefinitionAclId at the  
        position specified by qosIpAceOrder."
```

Attempting to specify an unknown class instance will result in an appropriate error indication being returned to the entity that is attempting to install the conflicting entry. For example, a 'priUnknown(2)' error indication is returned to the policy server in this situation."

```
::= { qosIpAclDefinitionEntry 3 }
```


qosIpAclDefinitionAceOrder OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

"The precedence order of this ACE. The precedence order determines the position of this ACE in the ACL. An ACE with a given precedence order is positioned in the access control list before one with a higher-valued precedence order.

Precedence values within a group must be unique otherwise instance installation will be prohibited and an error value will be returned."

::= { qosIpAclDefinitionEntry 4 }

--

-- Conformance Section

--

qosPolicyIpPibConformance

OBJECT IDENTIFIER ::= { qosPolicyIpPib 3 }

qosPolicyIpPibCompliances

OBJECT IDENTIFIER ::= { qosPolicyIpPibConformance 1 }

qosPolicyIpPibGroups

OBJECT IDENTIFIER ::= { qosPolicyIpPibConformance 2 }

qosPolicyIpPibCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Describes the requirements for conformance to the QoS Policy IP PIB."

MODULE -- this module

MANDATORY-GROUPS { qosIfTypeGroup,
qosIfTypeRoleComboGroup,
qosIfTypeSchedulingCapsGroup,
qosIfTypeQueueSetAssignGroup,
qosIfQueueGroup,
qosMeterGroup,
qosActionGroup,
qosTargetGroup }

OBJECT qosIfTypeQueueSetAssignName

MIN-ACCESS notify

DESCRIPTION	"Install support is not required."
OBJECT	qosIfTypeQueueSetAssignRoles
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfTypeQueueSetAssignQueueSetId
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueSetId
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueQueueSize
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueSetThreshSet
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueuePriorityGroup
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueServiceDisc
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueDrainSize
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueMinAbsBandwidth
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosIfQueueBandwidthAllocation
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."
OBJECT	qosMeterDataSpecification
MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."

OBJECT qosMeterCommittedRate
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterCommittedBurst
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterPeakRate
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterPeakBurst
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterHighConfAction
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterMedConfAction
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosMeterLowConfAction
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosActionDrop
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosActionUpdatedDSCP
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosActionMeter
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

GROUP qosIfTypeClassificationCapsGroup
DESCRIPTION
 "The qosIfTypeClassificationCapsGroup is mandatory
 if IP datagram classification is supported."

GROUP qosIfTypePolicingCapsGroup

DESCRIPTION

"The qosIfTypePolicingCapsGroup is mandatory if policing and shaping capabilities are supported."

GROUP qosIfThresholdGroup

DESCRIPTION

"The qosIfThresholdGroup is mandatory if queue-based thresholds are supported and if the qosIfDscpMapGroup is supported."

OBJECT qosIfThresholdSetId

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

OBJECT qosIfThresholdDropMethod

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

OBJECT qosIfThresholdMinThresh

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

OBJECT qosIfThresholdMaxThresh

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

GROUP qosIfDscpAssignGroup

DESCRIPTION

"The qosIfDscpAssignGroup is mandatory if traffic queue assignment based on DSCP is supported."

OBJECT qosIfDscpAssignName

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

OBJECT qosIfDscpAssignRoles

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

OBJECT qosIfDscpAssignDscpMap

MIN-ACCESS notify

DESCRIPTION "Install support is not required."

GROUP qosIfDscpMapGroup

DESCRIPTION

"The qosIfDscpMapGroup is mandatory if the qosIfDscpAssignGroup is supported."

OBJECT qosIfDscpMapMapId
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosIfDscpMapDscp
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosIfDscpMapQueue
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qosIfDscpMapThresh
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

GROUP qosIpAceGroup
DESCRIPTION
"The qosIpAceGroup is mandatory if filtering based on IP traffic components is supported."

GROUP qosIpAclDefinitionGroup
DESCRIPTION
"The qosIpAclDefinitionGroup is mandatory if filtering based on IP traffic components is supported."

::= { qosPolicyIpPibCompliances 1 }

qosIfTypeGroup OBJECT-GROUP

OBJECTS {
 qosIfTypeName,
 qosIfTypeCapabilityPrc,
 qosIfTypeCapabilityId
}

STATUS current

DESCRIPTION
"Objects from the qosIfTypeTable."

::= { qosPolicyIpPibGroups 1 }

qosIfTypeRoleComboGroup OBJECT-GROUP


```
OBJECTS {
    qosIfTypeRoleComboIfName,
    qosIfTypeRoleComboRoles
}
STATUS current
DESCRIPTION
    "Objects from the qosIfTypeRoleComboTable."

::= { qosPolicyIpPibGroups 2 }
```

```
qosIfTypeClassificationCapsGroup OBJECT-GROUP
    OBJECTS {
        qosIfTypeClassificationCaps
    }
    STATUS current
    DESCRIPTION
        "Objects from the qosIfTypeClassificationCapsTable."

    ::= { qosPolicyIpPibGroups 3 }
```

```
qosIfTypePolicingCapsGroup OBJECT-GROUP
    OBJECTS {
        qosIfTypePolicingCaps
    }
    STATUS current
    DESCRIPTION
        "Objects from the qosIfTypePolicingCapsTable."

    ::= { qosPolicyIpPibGroups 4 }
```

```
qosIfTypeSchedulingCapsGroup OBJECT-GROUP
    OBJECTS {
        qosIfTypeSchedulingCapsMaxQueues,
        qosIfTypeSchedulingCapsMaxThresholds,
        qosIfTypeSchedulingCapsMaxPriorities,
        qosIfTypeSchedulingCapsServiceDisc,
        qosIfTypeSchedulingCapsMinQueueSize,
        qosIfTypeSchedulingCapsMaxQueueSize,
        qosIfTypeSchedulingCapsTotalQueueSize,
        qosIfTypeSchedulingCapsWredCapable
    }
    STATUS current
    DESCRIPTION
        "Objects from the qosIfTypeSchedulingCapsTable."
```



```
::= { qosPolicyIpPibGroups 5 }
```

```
qosIfTypeQueueSetAssignGroup OBJECT-GROUP
```

```
  OBJECTS {
    qosIfTypeQueueSetAssignName,
    qosIfTypeQueueSetAssignRoles,
    qosIfTypeQueueSetAssignQueueSetId,
  }
  STATUS current
  DESCRIPTION
    "Objects from the qosIfTypeQueueSetAssignTable."
```

```
::= { qosPolicyIpPibGroups 6 }
```

```
qosIfQueueGroup OBJECT-GROUP
```

```
  OBJECTS {
    qosIfQueueSetId,
    qosIfQueueQueueSize,
    qosIfQueueSetThreshSet,
    qosIfQueuePriorityGroup,
    qosIfQueueServiceDisc,
    qosIfQueueDrainSize,
    qosIfQueueMinAbsBandwidth,
    qosIfQueueBandwidthAllocation
  }
  STATUS current
  DESCRIPTION
    "Objects from the qosIfQueueTable."
```

```
::= { qosPolicyIpPibGroups 7 }
```

```
qosIfThresholdGroup OBJECT-GROUP
```

```
  OBJECTS {
    qosIfThresholdSetId,
    qosIfThresholdDropMethod,
    qosIfThresholdMinThresh,
    qosIfThresholdMaxThresh
  }
  STATUS current
  DESCRIPTION
    "Objects from the qosIfThresholdTable."
```

```
::= { qosPolicyIpPibGroups 8 }
```

```
qosIfDscpAssignGroup OBJECT-GROUP
```



```
OBJECTS {
    qosIfDscpAssignName,
    qosIfDscpAssignRoles,
    qosIfDscpAssignDscpMap
}
STATUS current
DESCRIPTION
    "Objects from the qosIfDscpAssignTable."
```

```
::= { qosPolicyIpPibGroups 9 }
```

```
qosIfDscpMapGroup OBJECT-GROUP
    OBJECTS {
        qosIfDscpMapMapId,
        qosIfDscpMapDscp,
        qosIfDscpMapQueue,
        qosIfDscpMapThresh
    }
    STATUS current
    DESCRIPTION
        "Objects from the qosIfDscpMapTable."
```

```
::= { qosPolicyIpPibGroups 10 }
```

```
qosMeterGroup OBJECT-GROUP
    OBJECTS {
        qosMeterDataSpecification,
        qosMeterCommittedRate,
        qosMeterCommittedBurst,
        qosMeterPeakRate,
        qosMeterPeakBurst,
        qosMeterHighConfAction,
        qosMeterMedConfAction,
        qosMeterLowConfAction
    }
    STATUS current
    DESCRIPTION
        "Objects from the qosMeterTable."
```

```
::= { qosPolicyIpPibGroups 11 }
```

```
qosActionGroup OBJECT-GROUP
    OBJECTS {
        qosActionDrop,
        qosActionUpdateDSCP,
```



```
        qosActionMeter
    }
    STATUS    current
    DESCRIPTION
        "Objects from the qosActionTable."

    ::= { qosPolicyIpPibGroups 12 }

qosTargetGroup OBJECT-GROUP
    OBJECTS {
        qosTargetAclId,
        qosTargetAclType,
        qosTargetInterfaceRoles,
        qosTargetInterfaceDirection,
        qosTargetOrder,
        qosTargetMeter
    }
    STATUS    current
    DESCRIPTION
        "Objects from the qosTargetTable."

    ::= { qosPolicyIpPibGroups 13 }

qosIpAceGroup OBJECT-GROUP
    OBJECTS {
        qosIpAceDstAddr,
        qosIpAceDstAddrMask,
        qosIpAceSrcAddr,
        qosIpAceSrcAddrMask,
        qosIpAceDscp,
        qosIpAceProtocol,
        qosIpAceDstL4PortMin,
        qosIpAceDstL4PortMax,
        qosIpAceSrcL4PortMin,
        qosIpAceSrcL4PortMax,
        qosIpAcePermit
    }
    STATUS    current
    DESCRIPTION
        "Objects from the qosIpAceTable."

    ::= { qosPolicyIpPibGroups 14 }

qosIpAclDefinitionGroup OBJECT-GROUP
    OBJECTS {
```



```
        qosIpAclDefinitionAclId,  
        qosIpAclDefinitionAceId,  
        qosIpAclDefinitionAceOrder  
    }  
    STATUS    current  
    DESCRIPTION  
        "Objects from the qosIpAclDefinitionTable."  
  
    ::= { qosPolicyIpPibGroups 15 }
```

END

6.2. The DiffServ IEEE 802 PIB

DIFFSERV-802-PIB PIB-DEFINITIONS ::= BEGIN

IMPORTS

```
    Unsigned32, Integer32,
    MODULE-IDENTITY, OBJECT-TYPE
        FROM COPS-PR-SPPI
    TruthValue, PhysAddress,
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC
    RoleCombination, PolicyInstanceId, PolicyReferenceId
        FROM POLICY-FRAMEWORK-PIB
    Dscp
        FROM QOS-POLICY-IP-PIB;
```

qosPolicy802Pib MODULE-IDENTITY

```
    LAST-UPDATED "9906241800Z"
    ORGANIZATION "IETF RAP WG"
    CONTACT-INFO "
        Michael Fine
        Cisco Systems, Inc.
        170 West Tasman Drive
        San Jose, CA 95134-1706 USA
        Phone: +1 408 527 8218
        Email: mfine@cisco.com
```

```
        Keith McCloghrie
        Cisco Systems, Inc.
        170 West Tasman Drive,
        San Jose, CA 95134-1706 USA
        Phone: +1 408 526 5260
        Email: kzm@cisco.com
```

```
        John Seligson
        Nortel Networks, Inc.
        4401 Great America Parkway
        Santa Clara, CA 95054 USA
        Phone: +1 408 495 2992
        Email: jseligso@nortelnetworks.com"
```

DESCRIPTION

```
    "The PIB module containing an initial set of policy
    rule classes that describe the quality of service
    (QoS) policies supported by devices for IEEE 802-
    based traffic."
```



```
::= { tbd }
```

```
qosPolicy802PibClasses OBJECT IDENTIFIER ::= { qosPolicy802Pib 1 }
```

```
--
```

```
-- Textual Conventions
```

```
--
```

```
--
```

```
-- IEEE 802 CoS
```

```
--
```

```
QosIeee802Cos ::= TEXTUAL-CONVENTION
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "An integer that is in the range of the IEEE 802 CoS  
        values. This corresponds to the 802.1p priority values."
```

```
    SYNTAX INTEGER (0..7)
```

```
--
```

```
-- General configuration information for the entire domain
```

```
--
```

```
qos802DomainConfig OBJECT IDENTIFIER ::= { qosPolicy802PibClasses 1 }
```

```
--
```

```
-- 802 Classification Capabilities
```

```
--
```

```
qos802IfTypeClassificationCapsTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF QosIfType802ClassificationCapsEntry
```

```
    POLICY-ACCESS   notify
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "This table specifies the 803 classification capabilities of  
        interface types. A PRI in the qosIfTypeTable would reference  
        a PRI in this table if the corresponding interface type  
        supports some of these capabilities."
```

```
    ::= { qos802DomainConfig 1 }
```

```
qos802IfTypeClassificationCapsEntry OBJECT-TYPE
```

```
    SYNTAX          QosIfType802ClassificationEntry
```


STATUS current

DESCRIPTION

"An instance of this class describes the 802 classification capabilities of an interface."

INDEX { qos802IfTypeClassificationCapsPrid }

UNIQUENESS { qos802IfTypeClassificationCaps }

::= { qos802IfTypeClassificationCapsTable 1 }

QosIfType802ClassificationCapsEntry ::= SEQUENCE {
 qos802IfTypeClassificationCapsPrid PolicyInstanceId,
 qos802IfTypeClassificationCaps BITS
}

qos802IfTypeClassificationCapsPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies a instance of the class."

::= { qos802IfTypeClassificationCapsEntry 1 }

qos802IfTypeClassificationCaps OBJECT-TYPE

SYNTAX BITS {
 input802Classification(1),
 output802Classification(2),
 -- Indicates the ability to classify 802.3
 -- packets on ingress and on egress,
 -- respectively.

 etherAddrClassification(3),
 -- indicates the ability to classify based on
 -- addresses
 etherTypeClassification(4),
 etherSapClassification(4),
 etherSnapClassification(4),
 -- indicates the ability to classify based on
 -- ether types, SAPs or SNAP types
}

STATUS current

DESCRIPTION

"Bit set of supported 802.3 classification capabilities."


```
::= { qos802IfTypeClassificationCapsEntry 2 }
```

```
--
```

```
-- Differentiated Services Code Point Mapping Table
```

```
--
```

```
-- Supports the mapping of DSCP values to IEEE CoS values.
```

```
--
```

```
qos802DscpMappingTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF Qos802DscpMappingEntry
```

```
    POLICY-ACCESS    install
```

```
    STATUS           current
```

```
    DESCRIPTION
```

```
        "Maps each DSCP to an QosIeee802Cos. When configured
        for the first time, all 64 entries of the table must
        be specified. Thereafter, instances may be modified but
        not deleted unless all instances are deleted."
```

```
    INSTALL-ERRORS {
```

```
        priInstNotComplete(1)    -- required instances not created
```

```
    }
```

```
::= { qos802DomainConfig 2 }
```

```
qos802DscpMappingEntry OBJECT-TYPE
```

```
    SYNTAX          Qos802DscpMappingEntry
```

```
    STATUS           current
```

```
    DESCRIPTION
```

```
        "An instance of the qos802DscpMapping class. A total of 64
        class instances are constantly maintained after initial device
        configuration."
```

```
    INDEX { qos802DscpMappingPrid }
```

```
    UNIQUENESS { qos802DscpMappingDscp }
```

```
::= { qos802DscpMappingTable 1 }
```

```
Qos802DscpMappingEntry ::= SEQUENCE {
```

```
    qos802DscpMappingPrid      PolicyInstanceId,
```

```
    qos802DscpMappingDscp      Dscp,
```

```
    qos802DscpMapping802Cos    QosIeee802Cos
```

```
}
```

```
qos802DscpMappingPrid OBJECT-TYPE
```


SYNTAX PolicyInstanceId
STATUS current
DESCRIPTION
 "A unique ID for this policy rule instance."

::= { qos802DscpMappingEntry 1 }

qos802DscpMappingDscp OBJECT-TYPE

SYNTAX Dscp
STATUS current
DESCRIPTION
 "The DSCP class instance attribute that is used to determine the appropriate layer 2 CoS mappings. DSCP values 0 through 63 (inclusive) are maintained in the table."

::= { qos802DscpMappingEntry 2 }

qos802DscpMapping802Cos OBJECT-TYPE

SYNTAX QosIeee802Cos
STATUS current
DESCRIPTION
 "The IEEE 802 CoS value to use when mapping the DSCP value specified by the qos802DscpMappingDscp attribute to a IEEE 802 CoS."

::= { qos802DscpMappingEntry 3 }

--
-- Layer 2 CoS-to-DSCP Mapping Table
--
-- Supports the mapping of IEEE CoS values to DSCP values
-- for generic QoS traffic classification
--

qos802CosToDscpTable OBJECT-TYPE

SYNTAX SEQUENCE OF Qos802CosToDscpEntry
POLICY-ACCESS install
STATUS current
DESCRIPTION
 "Maps each of eight layer 2 CoS values to a DSCP. When configured for the first time, all 8 entries of the table must be specified. Thereafter, instances may be modified but not deleted unless all instances are deleted."


```
INSTALL-ERRORS {
    priInstNotComplete(1)    -- required instances not created
}

::= { qos802DomainConfig 3 }

qos802CosToDscpEntry OBJECT-TYPE
    SYNTAX          Qos802CosToDscpEntry
    STATUS          current
    DESCRIPTION
        "An instance of the qosCosToDscp class. A total of 8
        class instances are constantly maintained after initial
        device configuration."

    INDEX { qos802CosToDscpPrid }
    UNIQUENESS { qos802CosToDscpCos }

    ::= { qos802CosToDscpTable 1 }

Qos802CosToDscpEntry ::= SEQUENCE {
    qos802CosToDscpPrid PolicyInstanceId,
    qos802CosToDscpCos  QosIeee802Cos,
    qos802CosToDscpDscp Dscp
}

qos802CosToDscpPrid OBJECT-TYPE
    SYNTAX          PolicyInstanceId
    STATUS          current
    DESCRIPTION
        "A unique index for this policy rule instance."

    ::= { qos802CosToDscpEntry 1 }

qos802CosToDscpCos OBJECT-TYPE
    SYNTAX          QosIeee802Cos
    STATUS          current
    DESCRIPTION
        "The layer 2 CoS class instance attribute that is used to
        determine the appropriate DSCP mappings. CoS values 0
        through 7 (inclusive) are maintained in the table."

    ::= { qos802CosToDscpEntry 2 }

qos802CosToDscpDscp OBJECT-TYPE
    SYNTAX          Dscp
```


STATUS current

DESCRIPTION

"The DSCP value to use when mapping the layer 2 CoS value specified by the qosCosToDscp attribute to a DSCP."

::= { qos802CosToDscpEntry 3 }

--

-- The IEEE 802 Classification and Policing Group

--

qos802Qos OBJECT IDENTIFIER ::= { qosPolicy802PibClasses 2 }

--

-- The IEEE 802 ACE Table

--

-- The IEEE 802 ACE Table supports the specification of IEEE
-- 802-based (e.g., 802.3) information that is used to perform
-- traffic classification.

--

qos802AceTable OBJECT-TYPE

SYNTAX SEQUENCE OF Qos802AceEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"IEEE 802-based ACE definitions. A class that contains attributes of IEEE 802 (e.g., 802.3) traffic that form an association that is used to perform traffic classification."

::= { qos802Qos 1 }

qos802AceEntry OBJECT-TYPE

SYNTAX Qos802AceEntry

STATUS current

DESCRIPTION

"IEEE 802-based ACE definitions. An entry specifies (potentially) several distinct matching components. Each component is tested against the data in a frame individually. An overall match occurs when all of the individual components match the data they are compared against in the frame being processed. A failure of any one test causes the overall match to fail."

Wildcards may be specified for those fields that are not relevant."

```
INDEX { qos802AcePrid }
UNIQUENESS { qos802AceDstAddr,
              qos802AceDstAddrMask,
              qos802AceSrcAddr,
              qos802AceSrcAddrMask,
              qos802AceVlanId,
              qos802AceVlanTagRequired,
              qos802AceEtherType,
              qos802AceUserPriority}
```

```
::= { qos802AceTable 1 }
```

```
Qos802AceEntry ::= SEQUENCE {
    qos802AcePrid          PolicyInstanceId,
    qos802AceDstAddr       PhysAddress,
    qos802AceDstAddrMask   PhysAddress,
    qos802AceSrcAddr       PhysAddress,
    qos802AceSrcAddrMask   PhysAddress,
    qos802AceVlanId        Integer32,
    qos802AceVlanTagRequired INTEGER,
    qos802AceEtherType     Integer32,
    qos802AceUserPriority   BITS,
    qos802AcePermit        TruthValue
}
```

qos802AcePrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies this 802 ACE among all of the 802 ACEs. Note that this identifier is used in instances of the qos802Acl class to associate a 802 ACE with a 802 ACL. An active ACE/ACL association prohibits the deletion of the 802 ACE until the ACE/ACL association is terminated. Class instances may not be contiguous."

```
::= { qos802AceEntry 1 }
```

qos802AceDstAddr OBJECT-TYPE

SYNTAX PhysAddress

STATUS current

DESCRIPTION

"The 802 address against which the 802 DA of incoming traffic streams will be compared. Frames whose 802 DA matches the physical address specified by this object, taking into account address wildcarding as specified by the qos802AceDstAddrMask object, are potentially subject to the processing guidelines that are associated with this entry through the related action class."

::= { qos802AceEntry 2 }

qos802AceDstAddrMask OBJECT-TYPE

SYNTAX PhysAddress

STATUS current

DESCRIPTION

"This object specifies the bits in a 802 destination address that should be considered when performing a 802 DA comparison against the address specified in the qos802AceDstAddr object.

The value of this object represents a mask that is logically and'ed with the 802 DA in received frames to derive the value to be compared against the qos802AceDstAddr address. A zero bit in the mask thus means that the corresponding bit in the address always matches. The qos802AceDstAddr value must also be masked using this value prior to any comparisons.

The length of this object in octets must equal the length in octets of the qos802AceDstAddr. Note that a mask with no bits set (i.e., all zeroes) effectively wildcards the qos802AceDstAddr object."

::= { qos802AceEntry 3 }

qos802AceSrcAddr OBJECT-TYPE

SYNTAX PhysAddress

STATUS current

DESCRIPTION

"The 802 MAC address against which the 802 MAC SA of incoming traffic streams will be compared. Frames whose 802 MAC SA matches the physical address specified by this object, taking into account address wildcarding as specified by the qos802AceSrcAddrMask object, are potentially subject to the processing guidelines that are associated with this entry through the related action class."


```
::= { qos802AceEntry 4 }
```

qos802AceSrcAddrMask OBJECT-TYPE

SYNTAX PhysAddress

STATUS current

DESCRIPTION

"This object specifies the bits in a 802 MAC source address that should be considered when performing a 802 MAC SA comparison against the address specified in the qos802AceSrcAddr object.

The value of this object represents a mask that is logically and'ed with the 802 MAC SA in received frames to derive the value to be compared against the qos802AceSrcAddr address. A zero bit in the mask thus means that the corresponding bit in the address always matches. The qos802AceSrcAddr value must also be masked using this value prior to any comparisons.

The length of this object in octets must equal the length in octets of the qos802AceSrcAddr. Note that a mask with no bits set (i.e., all zeroes) effectively wildcards the qos802AceSrcAddr object."

```
::= { qos802AceEntry 5 }
```

qos802AceVlanId OBJECT-TYPE

SYNTAX Integer32 (-1 | 1..4094)

STATUS current

DESCRIPTION

"The VLAN ID (VID) that uniquely identifies a VLAN within the device. This VLAN may be known or unknown (i.e., traffic associated with this VID has not yet been seen by the device) at the time this entry is instantiated.

Setting the qos802AceVlanId object to -1 indicates that VLAN data should not be considered during traffic classification."

```
::= { qos802AceEntry 6 }
```

qos802AceVlanTagRequired OBJECT-TYPE

SYNTAX INTEGER {
taggedOnly(1),


```

        priorityTaggedPlus(2),
        untaggedOnly(3),
        ignoreTag(4)
    }
STATUS      current
DESCRIPTION
    "This object indicates whether the presence of an
    IEEE 802.1Q VLAN tag in data link layer frames must
    be considered when determining if a given frame
    matches this 802 ACE entry.

    A value of 'taggedOnly(1)' means that only frames
    containing a VLAN tag with a non-Null VID (i.e., a
    VID in the range 1..4094) will be considered a match.

    A value of 'priorityTaggedPlus(2)' means that only
    frames containing a VLAN tag, regardless of the value
    of the VID, will be considered a match.

    A value of 'untaggedOnly(3)' indicates that only
    untagged frames will match this filter component.

    The presence of a VLAN tag is not taken into
    consideration in terms of a match if the value is
    'ignoreTag(4)'."

 ::= { qos802AceEntry 7 }

```

qos802AceEtherType OBJECT-TYPE

```

SYNTAX      Integer32 (-1 | 0..'ffff'h)
STATUS      current
DESCRIPTION
    "This object specifies the value that will be compared
    against the value contained in the EtherType field of an
    IEEE 802 frame. Example settings would include 'IP'
    (0x0800), 'ARP' (0x0806) and 'IPX' (0x8137).

    Setting the qos802AceEtherTypeMin object to -1 indicates
    that EtherType data should not be considered during traffic
    classification.

    Note that the position of the EtherType field depends on
    the underlying frame format. For Ethernet-II encapsulation,
    the EtherType field follows the 802 MAC source address. For
    802.2 LLC/SNAP encapsulation, the EtherType value follows the

```


Organization Code field in the 802.2 SNAP header. The value that is tested with regard to this filter component therefore depends on the data link layer frame format being used. If this 802 ACE component is active when there is no EtherType field in a frame (e.g., 802.2 LLC), a match is implied."

::= { qos802AceEntry 8 }

qos802AceUserPriority OBJECT-TYPE

SYNTAX BITS {
 matchPriority0(0),
 matchPriority1(1),
 matchPriority2(2),
 matchPriority3(3),
 matchPriority4(4),
 matchPriority5(5),
 matchPriority6(6),
 matchPriority7(7)
 }
STATUS current

DESCRIPTION

"The set of values, representing the potential range of user priority values, against which the value contained in the user priority field of a tagged 802.1 frame is compared. A test for equality is performed when determining if a match exists between the data in a data link layer frame and the value of this 802 ACE component. Multiple values may be set at one time such that potentially several different user priority values may match this 802 ACE component.

Setting all of the bits that are associated with this object causes all user priority values to match this attribute. This essentially makes any comparisons with regard to user priority values unnecessary. Untagged frames are treated as an implicit match."

::= { qos802AceEntry 9 }

qos802AcePermit OBJECT-TYPE

SYNTAX TruthValue
STATUS current
DESCRIPTION

"If the frame matches this ACE and the value of this attribute is true, then the matching process terminates

and the QoS associated with this 802-based ACE (indirectly through the 802 ACL) is applied to the packet. If the value of this attribute is false, then no more 802 ACEs in this 802 ACL are compared to this packet and matching continues with the first 802-based ACE of the next 802 ACL."

```
::= { qos802AceEntry 10 }
```

```
--
```

```
-- The IEEE 802 ACL Definition Table
```

```
--
```

```
-- The IEEE 802 ACL Definition Table supports the association of  
-- distinct IEEE 802-based (e.g., 802.3) traffic classification  
-- specifications into an ordered list.  
--
```

qos802AclDefinitionTable OBJECT-TYPE

SYNTAX SEQUENCE OF Qos802AclDefinitionEntry

POLICY-ACCESS install

STATUS current

DESCRIPTION

"IEEE 802-based ACL definitions. A class that defines a set of 802 ACLs, each of which is comprised of an ordered list of 802 ACEs."

INSTALL-ERRORS {

priPrecedenceConflict(1) -- precedence conflict detected
}

```
::= { qos802Qos 2 }
```

qos802AclDefinitionEntry OBJECT-TYPE

SYNTAX Qos802AclDefinitionEntry

STATUS current

DESCRIPTION

"IEEE 802-based ACL definitions. An entry specifies an instance of this class that associates an 802 ACE with a given 802 ACL. The evaluation order of distinct 802 ACEs that are associated with a specific 802 ACL is specified as well."

INDEX { qos802AclDefinitionPrid }

UNIQUENESS { qos802AclDefinitionAclId,
 qos802AclDefinitionAceId }


```
::= { qos802AclDefinitionTable 1 }
```

```
Qos802AclDefinitionEntry ::= SEQUENCE {
    qos802AclDefinitionPrid      PolicyInstanceId,
    qos802AclDefinitionAclId    INTEGER,
    qos802AclDefinitionAceId    PolicyReferenceId,
    qos802AclDefinitionAceOrder Unsigned32
}
```

```
qos802AclDefinitionPrid OBJECT-TYPE
    SYNTAX      PolicyInstanceId
    STATUS      current
    DESCRIPTION
        "An arbitrary integer index that uniquely identifies this
        802 ACE / 802 ACL association."
```

```
::= { qos802AclDefinitionEntry 1 }
```

```
qos802AclDefinitionAclId OBJECT-TYPE
    SYNTAX      INTEGER
    STATUS      current
    DESCRIPTION
        "An index for this 802 ACL. Each 802 ACL in the device is
        assigned a unique integer index. There will (potentially) be
        multiple instances of the qos802AclDefinition class with this
        identifier, one for each 802 ACE that is associated with the
        specified 802 ACL."
```

For example, assume that 2 802 ACLs, each comprised of 4 802 ACEs, have been installed. The instances of this class may appear as follows:

Index	AclId	AceId	AceOrder
10	6	4	1
11	6	5	2
12	6	9	23
13	6	11	24
65	18	5	8
66	18	9	12
67	18	13	15
70	18	14	16

Note that this identifier is used in instances of the qosAclTarget class to associate an 802 ACL with an interface set and action. An active ACL Target association prohibits

the deletion of all of the qos802AclDefinition instances with a given qos802AclDefinitionAclId (i.e., at least one entry for the specific qos802AclDefinitionAclId must be present in this table) until the ACL Target association is terminated."

::= { qos802AclDefinitionEntry 2 }

qos802AclDefinitionAceId OBJECT-TYPE

SYNTAX PolicyReferenceId

STATUS current

DESCRIPTION

"This attribute identifies the 802 ACE in the qos802AceTable that is associated with the 802 ACL specified by qos802AclDefinitionAclId object. The corresponding instance in the qos802Ace class must exist prior to being associated with a 802 ACL.

Attempting to specify an unknown class instance will result in an appropriate error indication being returned to the entity that is attempting to install the conflicting entry. For example, a 'priUnknown(2)' error indication is returned to the policy server in this situation."

::= { qos802AclDefinitionEntry 3 }

qos802AclDefinitionAceOrder OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

"The precedence of the 802 ACE, identified via the qos802AclDefinitionAceId object, with regard to evaluation order. The precedence determines the order of evaluation of this ACE in relation to related 802 ACEs that are associated with an ACL. An ACE with a given precedence order in the access control list is evaluated before one with a higher-valued precedence order.

Precedence values within a group must be unique otherwise instance installation will be prohibited and an error value will be returned.

Note that qos802AclDefinitionAceOrder values within a given ACL need not be contiguous."


```
 ::= { qos802Ac1DefinitionEntry 4 }

--
-- Conformance Section
--

qosPolicy802PibConformance
    OBJECT IDENTIFIER ::= { qosPolicy802Pib 2 }

qosPolicy802PibCompliances
    OBJECT IDENTIFIER ::= { qosPolicy802PibConformance 1 }
qosPolicy802PibGroups
    OBJECT IDENTIFIER ::= { qosPolicy802PibConformance 2 }

qosPolicy802PibCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Describes the requirements for conformance to the
        QoS Policy 802 PIB."

    MODULE -- this module

        GROUP qos802IfTypeClassificationCapsGroup
        DESCRIPTION
            "The qos802DscpMappingGroup is mandatory if 802.3
            frame classification is supported."

        GROUP qos802DscpMappingGroup
        DESCRIPTION
            "The qos802DscpMappingGroup is mandatory if mapping
            from DSCP to 802.1 user priority is supported."

        OBJECT qos802DscpMappingDscp
        MIN-ACCESS notify
        DESCRIPTION "Install support is not required."

        OBJECT qos802DscpMapping802Cos
        MIN-ACCESS notify
        DESCRIPTION "Install support is not required."

        GROUP qos802CosToDscpGroup
        DESCRIPTION
            "The qos802CosToDscpGroup is mandatory if mapping
            from 802.1 user priority to DSCP is supported."
```


OBJECT qos802CosToDscpCos
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

OBJECT qos802CosToDscpDscp
MIN-ACCESS notify
DESCRIPTION "Install support is not required."

GROUP qos802AceGroup
DESCRIPTION
"The qos802AceGroup is mandatory if filtering
based on 802 traffic criteria is supported."

GROUP qos802AclDefinitionGroup
DESCRIPTION
"The qos802AclDefinitionGroup is mandatory if
filtering based on 802 traffic criteria is supported."

::= { qosPolicy802PibCompliances 1 }

qos802IfTypeClassificationCapsGroup OBJECT-GROUP
OBJECTS {
 qos802IfTypeClassificationCaps
}
STATUS current
DESCRIPTION
"Objects from the qos802IfTypeClassificationCapsTable."

::= { qosPolicy802PibGroups 1 }

qos802DscpMappingGroup OBJECT-GROUP
OBJECTS {
 qos802DscpMappingDscp,
 qos802DscpMapping802Cos
}
STATUS current
DESCRIPTION
"Objects from the qos802DscpMappingTable."

::= { qosPolicy802PibGroups 2 }

qos802CosToDscpGroup OBJECT-GROUP
OBJECTS {
 qos802CosToDscpCos,
 qos802CosToDscpDscp


```
}
STATUS current
DESCRIPTION
    "Objects from the qos802CosToDscpTable."

::= { qosPolicy802PibGroups 3 }

qos802AceGroup OBJECT-GROUP
    OBJECTS {
        qos802AceDstAddr,
        qos802AceDstAddrMask,
        qos802AceSrcAddr,
        qos802AceSrcAddrMask,
        qos802AceVlanId,
        qos802AceVlanTagRequired,
        qos802AceEtherType,
        qos802AceUserPriority,
        qos802AcePermit
    }
    STATUS current
    DESCRIPTION
        "Objects from the qos802AceTable."

    ::= { qosPolicy802PibGroups 4 }

qos802AclDefinitionGroup OBJECT-GROUP
    OBJECTS {
        qos802AclDefinitionAclId,
        qos802AclDefinitionAceId,
        qos802AclDefinitionAceOrder
    }
    STATUS current
    DESCRIPTION
        "Objects from the qos802AclDefinitionTable."

    ::= { qosPolicy802PibGroups 5 }

END
```


7. Security Considerations

The information contained in a PIB when transported by the COPS protocol [[COPS-PR](#)] may be sensitive, and its function of provisioning a PEP requires that only authorized communication take place. The use of IPSEC between PDP and PEP, as described in [[COPS](#)], provides the necessary protection against these threats.

8. Intellectual Property Considerations

The IETF is being notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

9. Authors' Addresses

Michael Fine
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706 USA
Phone: +1 408 527 8218
Email: mfine@cisco.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706 USA
Phone: +1 408 526 5260
Email: kzm@cisco.com

John Seligson
Nortel Networks, Inc.
4401 Great America Parkway
Santa Clara, CA 95054 USA
Phone: +1 408 495 2992
Email: jseligso@nortelnetworks.com

Kwok Ho Chan
Nortel Networks, Inc.
600 Technology Park Drive
Billerica, MA 01821 USA
Phone: +1 978 288 8175
Email: khchan@nortelnetworks.com

Scott Hahn
Intel
2111 NE 25th Avenue
Hillsboro, OR 97124 USA
Phone: +1 503 264 8231
Email: scott.hahn@intel.com

Andrew Smith
Extreme Networks
10460 Bandley Drive
Cupertino CA 95014 USA
Phone: +1 408 342 0999
Email: andrew@extremenetworks.com

Francis Reichmeyer
IPHighway Inc.
Parker Plaza, 16th Floor
400 Kelby St.
Fort-Lee, NJ 07024
Phone: (201) 585-0800
Email: FranR@iphighway.com

10. References

- [COPS] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol" [RFC 2748](#), January 2000.
- [COPS-PR] K. Chan, D. Durham, S. Gai, S. Herzog, K. McCloghrie, F. Reichmeyer, J. Seligson, A. Smith, R. Yavatkar, "COPS Usage for Policy Provisioning," [draft-ietf-rap-cops-pr-02.txt](#), March 2000.
- [SPPI] K. McCloghrie, et.al., "Structure of Policy Provisioning Information," [draft-ietf-rap-sppi-00.txt](#), march 2000.
- [DSARCH] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, and E. Davies, "An Architecture for Differentiated Services", [RFC 2475](#), December 1998
- [FR-PIB] M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn, A. Smith, F. Reichmeyer "Framework Policy Information Base", Internet Draft <[draft-ietf-rap-frameworkpib-00.txt](#)>, March 2000

- [POLICY] M. Stevens, W. Weiss H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen, J. Wheeler, "Policy Framework", [draft-ietf-policy-framework-00.txt](#), September 1999.
- [RAP-FRAMEWORK] R. Yavatkar, D. Pendarakis, "A Framework for Policy-based Admission Control", [draft-ietf-rap-framework-03.txt](#), April 1999.
- [SNMP-SMI] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [MODEL] Y. Bernet, A. Smith, S. Blake, "A Conceptual Model for DiffServ Routers", [draft-ietf-diffserv-model-01.txt](#), October 1999.

Table of Contents

1	Glossary	2
2	Introduction	2
3	DiffServ PIB Concepts	2
3.1	Filters, Filter Groups and Classifiers	2
3.2	Applying QoS Policy Using Targets	3
3.3	Interface Modeling with Queue Sets	4
3.3.1	Queue Scheduling	4
3.3.2	Assigning Packets To Queues and Thresholds	5
3.3.3	Hierarchies of Queues	5
3.4	IP Mapping to and from Layer 2	5
4	Summary of the DiffServ PIB	6
5	PIB Operational Overview	7
6	PIB Definitions	10
6.1	The DiffServ Base PIB	10
6.2	The DiffServ IEEE 802 PIB	63
7	Security Considerations	81
8	Intellectual Property Considerations	81
9	Authors' Addresses	81
10	References	82

