

Network Working Group
Internet Draft
Expires December 2000

M. Fine
K. McCloghrie
Cisco Systems
J. Seligson
K. Chan
Nortel Networks
S. Hahn
Intel
A. Smith
No Affiliation
Francis Reichmeyer
IPHighway

July 14, 2000

Differentiated Services Quality of Service Policy Information Base

[draft-ietf-diffserv-pib-01.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 behavior 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 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.

4. Summary of the DiffServ PIB

The DiffServ PIB consists of one module containing the base PRCs for setting DiffServ policy, queues, classifiers, meters, etc., and also contains filters for matching IP packets. This module comprises 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.

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, `frwkPrcSupportTable` defined in the framework PIB [\[FR-PIB\]](#). Each instance of the `frwkPrcSupportTable` 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, `frwkIfCapsSetTable`. 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 administration domain and interface-specific policy information.

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.

```

+-----+
+-----+ | qosTargetEntry |

```

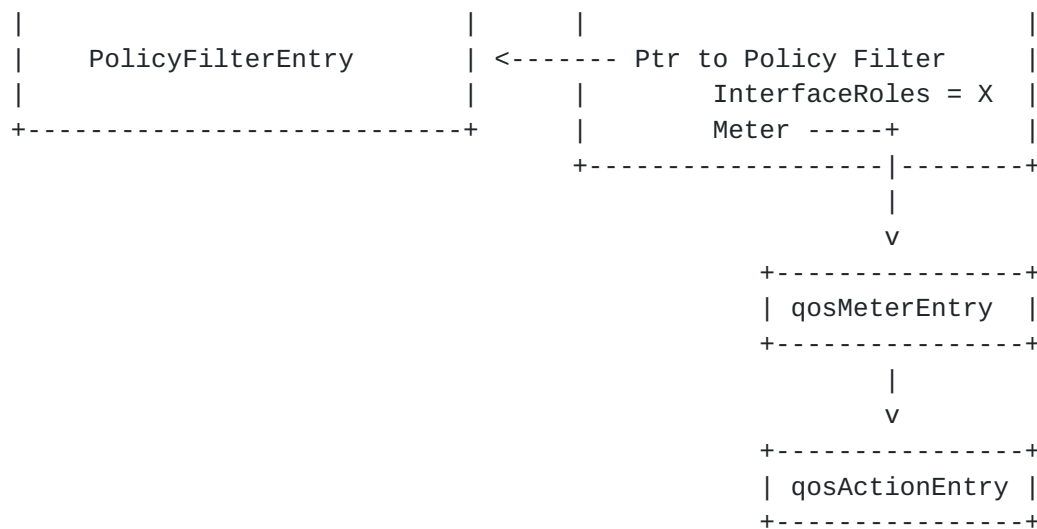



Figure 7.1 DiffServ PIB Table Relationships

Notice that the `qosTargetTable` allows the use of heterogeneous classifiers with same instance of `qosMeterTable`. For example, if classifiers operating on layer 2 addresses were to be defined, those classifiers could be used together with the IP ones.

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

6.1. The DiffServ Base PIB

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

```
IMPORTS
```

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

```
qosPolicyIpPib MODULE-IDENTITY
```

```
    CLIENT-TYPE { tbd    -- QoS Client Type  
    }
```

```
    LAST-UPDATED "200007141800Z"
```

```
    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 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
-- metering 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
--

qosIfClassificationCapsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF QosIfClassificationCapsEntry
    PIB-ACCESS       notify, 3
    STATUS           current
    DESCRIPTION
        "This table specifies the classification capabilities of an
        interface type"

    ::= { qosIfParameters 1 }

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

    INDEX { qosIfClassificationCapsPrid }
    UNIQUENESS { qosIfClassificationCaps }
```



```
::= { qosIfClassificationCapsTable 1 }
```

```
QosIfClassificationCapsEntry ::= SEQUENCE {  
    qosIfClassificationCapsPrid PolicyInstanceId,  
    qosIfClassificationCaps     BITS  
}
```

qosIfClassificationCapsPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

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

```
::= { qosIfClassificationCapsEntry 1 }
```

qosIfClassificationCaps 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  
    ipDscpClassification(5),  
    -- indicates the ability to classify based on  
    -- IP DSCP  
    ipL4Classification(6),  
    -- 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)."


```
::= { qosIfClassificationCapsEntry 2 }
```

```
--
```

```
-- Metering Capabilities
```

```
--
```

```
qosIfMeteringCapsTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF QosIfMeteringCapsEntry
```

```
    PIB-ACCESS      notify, 3
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "This table specifies the metering capabilities of an  
        interface type"
```

```
::= { qosIfParameters 2 }
```

```
qosIfMeteringCapsEntry OBJECT-TYPE
```

```
    SYNTAX          QosIfMeteringCapsEntry
```

```
    STATUS          current
```

```
    DESCRIPTION
```

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

```
    INDEX { qosIfMeteringCapsPrid }
```

```
    UNIQUENESS { qosIfMeteringCaps }
```

```
::= { qosIfMeteringCapsTable 1 }
```

```
QosIfMeteringCapsEntry ::= SEQUENCE {
```

```
    qosIfMeteringCapsPrid      PolicyInstanceId,
```

```
    qosIfMeteringCaps          BITS
```

```
}
```

```
qosIfMeteringCapsPrid OBJECT-TYPE
```

```
    SYNTAX          PolicyInstanceId
```

```
    STATUS          current
```

```
    DESCRIPTION
```

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

```
::= { qosIfMeteringCapsEntry 1 }
```

```
qosIfMeteringCaps OBJECT-TYPE
```

```
    SYNTAX          BITS {
```



```
        meterByRemarking (1),
        meterByDropping (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 metering capabilities may
    be augmented by capabilities specified in other PRCs (in other
    PIBs)."
```

::= { qosIfMeteringCapsEntry 2 }

--

-- Scheduling Capabilities

--

qosIfSchedulingCapsTable OBJECT-TYPE

```
    SYNTAX          SEQUENCE OF QosIfSchedulingCapsEntry
    PIB-ACCESS       notify, 10
    STATUS           current
    DESCRIPTION
        "This table specifies the scheduling capabilities of an
        interface type"
```

::= { qosIfParameters 3 }

qosIfSchedulingCapsEntry OBJECT-TYPE

```
    SYNTAX          QosIfSchedulingCapsEntry
    STATUS           current
    DESCRIPTION
        "An instance of this class describes the classification
        capabilities of an interface."
```

INDEX { qosIfSchedulingCapsPrid }

UNIQUENESS { qosIfSchedulingCapsMaxQueues,


```

    qosIfSchedulingCapsMaxThresholds,
    qosIfSchedulingCapsMaxPriorities,
    qosIfSchedulingCapsServiceDisc,
    qosIfSchedulingCapsMinQueueSize,
    qosIfSchedulingCapsMaxQueueSize,
    qosIfSchedulingCapsTotalQueueSize,
    qosIfSchedulingCapsWredCapable }

```

```
 ::= { qosIfSchedulingCapsTable 1 }
```

```

QosIfSchedulingCapsEntry ::= SEQUENCE {
    qosIfSchedulingCapsPrid          PolicyInstanceId,
    qosIfSchedulingCapsMaxQueues     INTEGER
    qosIfSchedulingCapsMaxThresholds INTEGER
    qosIfSchedulingCapsMaxPriorities INTEGER
    qosIfSchedulingCapsServiceDisc   BITS
    qosIfSchedulingCapsMinQueueSize  INTEGER
    qosIfSchedulingCapsMaxQueueSize  INTEGER
    qosIfSchedulingCapsTotalQueueSize INTEGER
    qosIfSchedulingCapsWredCapable   TruthValue
}

```

```

qosIfSchedulingCapsPrid OBJECT-TYPE
    SYNTAX      PolicyInstanceId
    STATUS      current
    DESCRIPTION
        "An arbitrary integer index that uniquely identifies a
        instance of the class."

```

```
 ::= { qosIfSchedulingCapsEntry 1 }
```

```

qosIfSchedulingCapsMaxQueues 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."

```

```
 ::= { qosIfSchedulingCapsEntry 2 }
```

```

qosIfSchedulingCapsMaxThresholds 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."

::= { qosIfSchedulingCapsEntry 3 }

qosIfSchedulingCapsMaxPriorities 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."

::= { qosIfSchedulingCapsEntry 4 }

qosIfSchedulingCapsServiceDisc 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 instead of, or in addition to, these disciplines by setting 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."


```
::= { qosIfSchedulingCapsEntry 5 }
```

qosIfSchedulingCapsMinQueueSize 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."

```
::= { qosIfSchedulingCapsEntry 6 }
```

qosIfSchedulingCapsMaxQueueSize 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 qosIfSchedulingCapsMinQueueSize, 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."

```
::= { qosIfSchedulingCapsEntry 7 }
```

qosIfSchedulingCapsTotalQueueSize 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."

```
::= { qosIfSchedulingCapsEntry 8 }
```


qosIfSchedulingCapsWredCapable 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."

::= { qosIfSchedulingCapsEntry 9 }

--

-- Queue Set Assignment Table

--

qosIfQueueSetAssignTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosIfQueueSetAssignEntry

PIB-ACCESS install, 6

STATUS current

DESCRIPTION

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

Contains the assignment of DSCPs to queues and thresholds for each interface type. 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."

::= { qosIfParameters 4 }

qosIfQueueSetAssignEntry OBJECT-TYPE

SYNTAX QosIfQueueSetAssignEntry

STATUS current

DESCRIPTION

"A conceptual row in the qosIfQueueSetAssignTable.

INDEX { qosIfQueueSetAssignPrid }

UNIQUENESS { qosIfQueueSetAssignIfName,
qosIfQueueSetAssignRoles }

::= { qosIfQueueSetAssignTable 1 }

QosIfQueueSetAssignEntry ::= SEQUENCE {


```

        qosIfQueueSetAssignPrid          PolicyInstanceId,
        qosIfQueueSetAssignName          SnmpAdminString,
        qosIfQueueSetAssignRoles         RoleCombination,
        qosIfQueueSetAssignQueueSetId    PolicyTagReference,
        qosIfQueueSetAssignDscpMap       PolicyTagReference
    }

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

    ::= { qosIfQueueSetAssignEntry 1 }

qosIfQueueSetAssignName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    STATUS      current
    DESCRIPTION
        "The name of an interface type. This name must exist in
        frwkIfCapSetTable."

    ::= { qosIfQueueSetAssignEntry 2 }

qosIfQueueSetAssignRoles OBJECT-TYPE
    SYNTAX      RoleCombination
    STATUS      current
    DESCRIPTION
        "The role combination associated with the interface type."

    ::= { qosIfQueueSetAssignEntry 3 }

qosIfQueueSetAssignQueueSet OBJECT-TYPE
    SYNTAX      PolicyTagReference
    PIB-TAG     qosIfQueueSetId
    STATUS      current
    DESCRIPTION
        "The integer ID of the queue set to be assigned to all interfaces
        of type specified by qosIfQueueSetAssignName and with role
        combination specified by qosIfQueueSetAssignRoles.
        This queue set must exist in qosIfQueueTable."

    ::= { qosIfQueueSetAssignEntry 4 }
```


qosIfQueueSetAssignDscpMap OBJECT-TYPE

SYNTAX PolicyTagReference

PIB-TAG qosIfDscpMapMapId

STATUS current

DESCRIPTION

"The DSCP map to apply to interfaces of type
qosIfQueueSetAssignName and role combo
qosIfQueueSetAssignRoles."

::= { qosIfQueueSetAssignEntry 5 }

--

-- 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

PIB-ACCESS install, 10

STATUS current

DESCRIPTION

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

::= { qosIfParameters 5 }

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 qosIfQsetAssignTable."

INDEX { qosIfQueuePrid }

UNIQUENESS { }


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

```
QosIfQueueEntry ::= SEQUENCE {
    qosIfQueuePrid          PolicyInstanceId,
    qosIfQueueSetId         PolicyTagId,
    qosIfQueueQueueSize     Unsigned32,
    qosIfQueueSetThreshSet  PolicyTagReference,
    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 PolicyTagId

STATUS current

DESCRIPTION

"An index that uniquely identifies a specific queue set. The queue set identified by this value is associated with an interface set through the qosIfQueueSetAssignQueueSetId object in the qosIfQueueSetAssignTable. 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 ratio of the queue size to the sum of the queue sizes for the interface."

::= { qosIfQueueEntry 3 }

qosIfQueueThreshSet OBJECT-TYPE

SYNTAX PolicyTagReference
PIB-TAG qosIfThresholdSetId
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.

A value of zero indicates no threshold set is associated with the queue. In this case the queue is configured with a single threshold at 100% qosIfThresholdDropMethod of tailDrop."

::= { 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

PIB-ACCESS install, 6

STATUS current

DESCRIPTION

"Contains configuration information for the individual thresholds of the threshold sets."

::= { qosIfParameters 6 }

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     PolicyTagId,
    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      PolicyTagId
    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 than 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, in bytes, 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, in bytes, above which all packets assigned
to this threshold are dropped."

::= { qosIfThresholdEntry 5 }

--

-- 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

PIB-ACCESS install, 6

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 7 }

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 PolicyTagId,

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 PolicyTagId

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

PIB-REFERENCE qosIfQueueTable

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

PIB-REFERENCE qosIfThresholdTable

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

PIB-ACCESS install, 10

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

PIB-REFERENCE qosActionTable

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

PIB-REFERENCE qosActionTable

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

PIB-REFERENCE qosActionTable

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 Action Group

--

qosAction OBJECT IDENTIFIER ::= { qosPolicyGenPibClasses 3 }

--

-- The QoS Action Table

--

-- The QoS Action Table describes actions that are associated with
-- specific meters through the QoS Target Table. An action specifies
-- whether to mark, drop, or leave the packet unchanged.

qosActionTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosActionEntry

PIB-ACCESS install, 4

STATUS current

DESCRIPTION

"Contains the current set of configured actions. The actions are associated with meters 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.

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 using augmentation."

```
INDEX { qosActionPrid }
UNIQUENESS { qosActionDrop,
             qosActionUpdatedDSCP }
```

```
::= { qosActionTable 1 }
```

```
QosActionEntry ::= SEQUENCE {
    qosActionPrid      PolicyInstanceId,
    qosActionAction     INTEGER,
    qosActionUpdatedDSCP Dscp,
}
```

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 Meter class) using this attribute."

```
::= { qosActionEntry 1 }
```

qosActionAction OBJECT-TYPE

SYNTAX INTEGER {

drop(1),

mark(2),

unchange(3) -- don't alter the DSCP

}

STATUS current

DESCRIPTION

"This action attribute specifies the action to be taken on the packet.

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 }

qosActionUpdatedDSCP OBJECT-TYPE

SYNTAX Dscp

STATUS current

DESCRIPTION

"This attribute specifies the value to write into the DSCP field of the packet if the action to be taken is to mark the packet.

::= { qosActionEntry 3 }

--

-- The QoS Target Table

--

-- The QoS Target Table supports the association of filters,
-- interfaces, meters and actions. It allows filter instances, as
-- defined in various filter 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 and metering. Furthermore, it allows heterogeneous
-- filter definition class instances to be applied to the same
-- interface group in a prescribed order of precedence.

--

qosTargetTable OBJECT-TYPE

SYNTAX SEQUENCE OF QosTargetEntry

PIB-ACCESS install, 7

STATUS current

DESCRIPTION

"A class that applies a set of filters to interfaces specifying, for each interface, the precedence order of the filters with respect to other filters applied to the same interface and, for each filter, the meter to apply to packets accepted by the filter. Interfaces are specified abstractly in terms of interface roles.

This class may contain filters that specify different types of traffic classification."


```
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 { qosTargetFilterId,
 qosTargetInterfaceRoles,
 qosTargetInterfaceDirection,
 qosTargetOrder }

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

QosTargetEntry ::= SEQUENCE {

qosTargetPrid	PolicyInstanceId,
qosTargetFilterId	PolicyTagReference,
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 }
```

qosTargetFilterId OBJECT-TYPE

SYNTAX PolicyTagReference

PIB-TAG frwkFilterGroupDefinitionId

STATUS current

DESCRIPTION

"This attribute identifies the filter group that is associated with this target. This filter group must be specified in frwkFilterGroupDefinitionTable and the specific group is identified by the value of this attribute."

```
::= { qosTargetEntry 2 }
```

qosTargetInterfaceRoles OBJECT-TYPE

SYNTAX RoleCombination

STATUS current

DESCRIPTION

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

```
::= { qosTargetEntry 3 }
```

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 filter applies."

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

qosTargetOrder OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

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

As an example, consider the following Target association:

Index	IfRoleCombo	IfDirection	FilterId	Order
14	'eth1000+L2+L3'	'in'	8	1

15	'eth1000+L2+L3'	'in'	3	2
16	'eth1000+L2+L3'	'in'	12	3
17	'eth1000+L2+L3'	'in'	6	4
18	'eth1000+L2+L3'	'in'	21	5

Five distinct filter specifications form a Target association (e.g., based on the specified interface role combination and direction attributes) with a prescribed order of evaluation. The FilterId attributes identify the filter definition instances.

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

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

qosTargetMeter OBJECT-TYPE

SYNTAX PolicyReferenceId

PIB-REFERENCE qosMeterTable

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 6 }
```

```
--
```

```
-- 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 { qosIfSchedulingCapsGroup,  
                    qosIfQueueSetAssignGroup,  
                    qosIfQueueGroup,  
                    qosMeterGroup,  
                    qosActionGroup,  
                    qosTargetGroup }
```

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

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

OBJECT qosIfQueueSetAssignQueueSetId
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."

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

GROUP qosIfMeteringCapsGroup
DESCRIPTION
 "The qosIfMeteringCapsGroup is mandatory if
 metering 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."

::= { qosPolicyIpPibCompliances 1 }

qosIfClassificationCapsGroup OBJECT-GROUP
 OBJECTS {
 qosIfClassificationCaps
 }
 STATUS current
 DESCRIPTION
 "Objects from the qosIfClassificationCapsTable."

::= { qosPolicyIpPibGroups 3 }

qosIfMeteringCapsGroup OBJECT-GROUP


```
OBJECTS {
    qosIfMeteringCaps
}
STATUS current
DESCRIPTION
    "Objects from the qosIfMeteringCapsTable."
```

```
::= { qosPolicyIpPibGroups 4 }
```

qosIfSchedulingCapsGroup OBJECT-GROUP

```
OBJECTS {
    qosIfSchedulingCapsMaxQueues,
    qosIfSchedulingCapsMaxThresholds,
    qosIfSchedulingCapsMaxPriorities,
    qosIfSchedulingCapsServiceDisc,
    qosIfSchedulingCapsMinQueueSize,
    qosIfSchedulingCapsMaxQueueSize,
    qosIfSchedulingCapsTotalQueueSize,
    qosIfSchedulingCapsWredCapable
}
STATUS current
DESCRIPTION
    "Objects from the qosIfSchedulingCapsTable."
```

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

qosIfQueueSetAssignGroup OBJECT-GROUP

```
OBJECTS {
    qosIfQueueSetAssignName,
    qosIfQueueSetAssignRoles,
    qosIfQueueSetAssignQueueSetId,
}
STATUS current
DESCRIPTION
    "Objects from the qosIfQueueSetAssignTable."
```

```
::= { 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
  }
  STATUS current
  DESCRIPTION
    "Objects from the qosActionTable."
```

```
::= { qosPolicyIpPibGroups 12 }
```

```
qosTargetGroup OBJECT-GROUP
```

```
  OBJECTS {
    qosTargetFilterId,
    qosTargetFilterType,
    qosTargetInterfaceRoles,
    qosTargetInterfaceDirection,
    qosTargetOrder,
    qosTargetMeter
  }
  STATUS current
  DESCRIPTION
    "Objects from the qosTargetTable."
```

```
::= { qosPolicyIpPibGroups 13 }
```


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
Fax: +1 415 345 1827
Email: ah_smith@pacbell.net

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-03.txt](#), July 2000.
- [SPPI] K. McCloghrie, et.al., "Structure of Policy Provisioning Information," [draft-ietf-rap-sppi-01.txt](#), July 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-01.txt](#)>, July 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, D. Grossman "A Conceptual Model for DiffServ Routers", [draft-ietf-diffserv-model-03.txt](#), May 2000.

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
4	Summary of the DiffServ PIB	5
5	PIB Operational Overview	6
6	PIB Definitions	8
6.1	The DiffServ Base PIB	8
7	Security Considerations	48
8	Intellectual Property Considerations	48
9	Authors' Addresses	48
10	References	49