Internet Engineering Task Force
Diffserv Working Group
Internet-Draft
draft-ietf-diffserv-pib-05.txt
Expires May 2002

M. Fine
K. McCloghrie
Cisco Systems
J. Seligson
K. Chan
Nortel Networks
S. Hahn
C. Bell
Intel
A. Smith
Allegro Networks
F. Reichmeyer
PFN

November 2001

Differentiated Services Quality of Service Policy Information Base

Status of this Memo

This document is a product of the IETF's Differentiated Services Working Group. Comments should be addressed to WG's mailing list at diffserv@ietf.org. The charter for Diffserv may be found at http://www.ietf.org/html.charters/diffserv-charter.html.

1. Glossary

PRC   Provisioning Class. A type of policy data.
PRI   Provisioning 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   Provisioning 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. Relationship to the Diffserv Informal Management Model

This PIB is designed according to the Differentiated Services Informal Management Model documented in [MODEL]. The model describes the way that ingress and egress interfaces of an 'n'-port router are modeled. It describes the configuration and management of a Diffserv interface in terms of a Traffic Conditioning Block (TCB) which contains, by definition, zero or more classifiers, meters, actions, algorithmic droppers, queues and schedulers. These elements are arranged according to the QoS policy being expressed, always in that order. Traffic may be classified; classified traffic may be metered; each stream of traffic identified by a combination of classifiers and meters may have some set of actions performed on it; it may have dropping algorithms applied and it may ultimately be stored into a queue before being scheduled out to its next destination, either onto a link or to another TCB. When the treatment for a given packet must have any of those elements repeated in a way that breaks the permitted sequence {classifier, meter, action, algorithmic dropper, queue, scheduler}, this must be modeled by cascading multiple TCBs.

The PIB represents this cascade by following the "Next" attributes of the various elements. They indicate what the next step in Diffserv processing will be, whether it be a classifier, meter, action, algorithmic dropper, queue, scheduler or a decision to now forward a packet.

The PIB models the individual elements that make up the TCBs. The higher level concept of a TCB is not required in the parameterization or in the linking together of the individual elements, hence it is not used in the PIB itself and only mentioned in the text for relating the PIB with the [MODEL]. The actual distinguishing of which TCB a specific element is a part of is not needed for the instrumentation of a device to support the functionalities of DiffServ, but it is useful for conceptual reasons. By not using the TCB concept, this PIB allows any grouping of elements to construct TCBs, using rules indicated by the [MODEL]. This will minimize changes to this PIB if rules in [MODEL] change.

The notion of a Data Path is used in this PIB to indicate the DiffServ processing a packet may experience. This Data Path is distinguished based on the Role Combination and the Direction of the flow the packet is part of. A Data Path Table Entry indicates the first of possibly multiple elements that will apply DiffServ treatment to the packet.

3.1. PIB Overview

This PIB is structured based on the need to configure the sequential DiffServ treatments being applied to a packet, and the parameterization of these treatments. These two aspects of the configuration are kept separate throughout the design of the PIB, and are fulfilled using separate tables and data definitions.

In addition, the PIB includes tables describing the capabilities and limitations of the device using a general extensible framework. These tables are reported to the PDP and assist the PDP with the configuration of functional elements that can be realized by the device.

This capabilities and limitations exchange allows a single or multiple devices to support many different variations of a functional datapath element. Allowing diverse methods of providing a general functional datapath element.

In this PIB, the ingress and egress portions of a router are configured independently but in the same manner. The difference is distinguished by an attribute in a table describing the start of the data path. Each interface performs some or all of the following high-level functions:

- Classify each packet according to some set of rules

- Determine whether the data stream the packet is part of is within or outside its metering parameters

oPerform a set of resulting actions such as counting and marking of the traffic with aDifferentiated Services Code Point (DSCP) as defined in[DSFIELD].

- Apply appropriate drop policy, either simple or complex algorithmic drop functionality.

Enqueue the traffic for output in the appropriate queue, whosescheduler may shape the traffic or simply forward it with some minimum rate or maximum latency.

The PIB therefore contains the following elements:

Data Path Table
This describes the starting point of DiffServ data paths within a single DiffServ device. This table describes interface role combination and interface direction specific data paths.

Classifier Tables
A general extensible framework for specifying a group of filters.

Meter Tables
A general extensible framework and one example of a parameterization table - TBParam table, applicable for Simple Token Bucket Meter, Average Rate Meter, Single Rate Three Color Meter, Two Rate Three Color Meter, and Sliding Window Three Color Meter.

Action Tables
A general extensible framework and examples of parameterization tables for Mark actions. The "multiplexer" and "null" actions described in [MODEL] are accomplished implicitly by means of the Prid structures of the other elements.

Algorithmic Dropper Tables
A general extensible framework for describing the dropper functional datapath element. This includes the absolute dropper and other queue measurement dependent algorithmic droppers.

Queue and Scheduler Tables
A general extensible framework for parameterizing queuing and scheduler systems. Notice Shaper is considered as a type of scheduler and is included here.

Capabilities Tables
A general extensible framework for defining the capabilities and limitations of the elements listed above. The capability tables

allow intelligent configuration of the elements by a PDP.

4. Structure of the PIB

4.1. General Conventions

The PIB consists of classes that represent functional elements in the data path (e.g. classifiers, meters, actions), and classes that specify parameters that apply to a certain type of functional element (e.g. a Token Bucket meter or a Mark action). Parameters are typically specified in a separate PRC to enable the use of parameter classes by multiple policies.

Functional element PRCs use the Prid TC (defined in [SPPI]) to indicate indirection. A Prid is an object identifier that is used to specify an instance of a PRC in another table. A Prid is used to point to parameter PRC that applies to a functional element, such as which filter should be used for a classifier element. A Prid is also used to specify an instance of a functional element PRC that describes what treatment should be applied next for a packet in the data path.

Note that the use of Prids to specify parameter PRCs allows the same functional element PRC to be extended with a number of different types of parameter PRC's. In addition, using Prids to indicate the next functional datapath element allows the elements to be ordered in any way.

4.2. DiffServ Data Paths

This part of the PIB provides instrumentation for connecting the DiffServ Functional Elements within a single DiffServ device. Please refer to the [MODEL] for discussions on the valid sequencing and grouping of DiffServ Functional Elements. Given some basic information, e.g. the interface capability, role combination and direction, the first DiffServ Functional Element is determined. Subsequent DiffServ Functional Elements are provided by the "Next" pointer attribute of each entry of data path tables. A description of how this "Next" pointer is used in each table is provided in their respective DESCRIPTION clauses.

4.2.1. Data Path PRC

The Data Path PRC provides the DiffServ treatment starting points for all packets of this DiffServ device. Each instance of this PRC specifies the interface capability, role combination and direction for the packet flow. There should be at most two entries for each (interface type, role combination) pair, one for ingress and one for egress. Each instance provides the first DiffServ Functional Element each packet at a specific interface (identified by the roles assigned to the interface) traveling in a specific relative direction should experience. Notice this table is interface specific, with the use of interface type and RoleCombination. To indicate explicitly that there are no Diffserv treatments for a particular interface type, role combination and direction, an instance of the Data Path PRC can be created with zeroDotZero in the qosDataPathStart attribute. This situation can also be indicated implicitly by not supplying an instance of a Data Path PRC for that particular interface type, role combination and direction. The explicit/implicit selection is up to the implementation. This means that the PEP should perform normal IP device processing when zeroDotZero is used in the qosDataPathStart attribute, or when the entry does not exist. Normal IP device processing will depend on the device; for example, this can be forwarding the packet.

Based on implementation experience of network devices where data path functional elements are implemented in separate physical processors or application specific integrated circuits, separated by switch fabric, it seems that more complex notions of data path are required within the network device to correlate the different physically separate data path functional elements. For example, ingress processing may have determined a specific ingress flow that gets aggregated with other ingress flows at an egress data path functional element. Some of the information determined at the ingress data path functional element may need to be used by the egress data path functional element. In numerous implementations, such information has been carried by adding it to the frame/memory block used to carry the flow within the network device; some implementers have called such information a "preamble" or a "frame descriptor". Different implementations use different formats for such information. Initially one may think such information is implementation details within the network device that does not need to be exposed outside of the network device. But from Policy Control point of view, such information will be very useful in determining network resource usage feedback from the network device to the policy server. This is accomplished by using the Internal Label Marker and Filter PRCs defined in [FR-PIB].

4.3. Classifiers

The classifier and classifier element tables determine how traffic is sorted out. They identify separable classes of traffic, by reference to appropriate filters, which may select anything from an individual micro-flow to aggregates identified by DSCP.

The classification is used to send these separate streams to appropriate Meter, Action, Algorithmic Dropper, Queue and Scheduler elements. For example, to indicate a multi-stage meter, sub-classes of traffic may be sent to different meter stages: e.g. in an implementation of the Assured Forwarding (AF) PHB [AF-PHB], AF11 traffic might be sent to the first meter, AF12 traffic might be sent to the second and AF13 traffic sent to the second meter stage's out-of-profile action.

The concept of a classifier is the same as described in [MODEL].

The structure of the classifier and classifier element tables, is the same as the classifier described in [MODEL]. Classifier elements have an associated precedence order solely for the purpose of resolving ambiguity between overlapping filters. Filter with higher values of precedence are compared first; the order of tests for entries of the same precedence is unimportant.

A datapath may consist of more than one classifier. There may be overlap of filter specification between filters of different classifiers. The first classifier functional datapath element encountered, as determined by the sequencing of diffserv functional datapath elements, will be used first.

An important form of classifier is "everything else": the final stage of the classifier i.e. the one with the lowest precedence, must be "complete" since the result of an incomplete classifier is not necessarily deterministic - see [MODEL] section 4.1.2.

When a classifier PRC is instantiated at the PEP, it should always have at least one classifier element table entry, the "everything else" classifier element, with its filter matching all IP packets. This "everything else" classifier element should be created by the PDP as part of the classifier setup. The PDP have full control of all classifier PRIs instantiated at the PEP.

The definition of the actual filter to be used by the classifier is referenced via a Prid: this enables the use of any sort of filter table that one might wish to design, standard or proprietary. No filters are defined in this PIB. However, standard filters for IP packets are defined in the Framework PIB [FR-PIB].

4.3.1. Classifier PRC

Classifiers, used in various ingress and egress interfaces, are organized by the instances of the Classifier PRC. A data path entry points to a classifier entry. A classifier entry identifies a list of classifier elements. A classifier element effectively includes the filter entry, and points to a "next" classifier entry or other data path functional element.

4.3.2. Classifier Element PRC

Classifier elements point to the filters which identify various classes of traffic. The separation between the "classifier element" and the "filter" allows us to use many different kinds of filters with the same essential semantics of "an identified set of traffic". The traffic matching the filter corresponding to a classifier element is given to the "next" data path functional element identified in the classifier element.

An example of a filter that may be pointed to by a Classifier

Element PRI is the frwkIpFilter PRC, defined in [FR-PIB].

4.4. Meters

A meter, according to [MODEL] section 5, measures the rate at which packets making up a stream of traffic pass it, compares this rate to some set of thresholds and produces some number (two or more) of potential results. A given packet is said to "conform" to the meter if, at the time that the packet is being looked at, the stream appears to be within the meter's profile. PIB syntax makes it easiest to define this as a sequence of one or more cascaded pass/fail tests, modeled here as if-then-else constructs. It is important to understand that this way of modeling does not imply anything about the implementation being "sequential": multi-rate/multi-profile meters e.g. those designed to support [SRTCM], [TRTCM], or [TSWTCM] can still be modeled this way even if they, of necessity, share information between the stages: the stages are introduced merely as a notational convenience in order to simplify the PIB structure.

4.4.1. Meter PRC

The generic meter PRC is used as a base for all more specific forms of meter. The definition of parameters specific to the type of meter used is referenced via a pointer to an instance of a PRC containing those specifics. This enables the use of any sort of specific meter table that one might wish to design, standard or proprietary. One specific meter table is defined in this PIB module. Other meter tables may be defined in other PIB modules.

4.4.2. Token-Bucket Parameter PRC

This is included as an example of a common type of meter. Entries in this table are referenced from the qosMeterSpecific attributes of meter PRC instances. The parameters are represented by a rate qosTBParamRate, a burst size qosTBParamBurstSize, and an interval qosTBparamInterval. The type of meter being parameterized is indicated by the qosTBParamType attribute. This is used to determine how the rate, burst and rate interval parameters are used. Additional meter parameterization classes can be defined in other PIBs when necessary.

4.5. Actions

Actions include "no action", "mark the traffic with a DSCP" or "specific action". Other tasks such as "shape the traffic" or "drop based on some algorithm" are handled in other functional datapath elements rather than in actions. The "multiplexer", "replicator" and "null" actions described in [MODEL] are accomplished implicitly through various combinations of the other elements.

This PIB uses the Action PRC qosActionTable to organize one Action's relationship with the element(s) before and after it. It allows Actions to be cascaded to enable multiple Actions be applied to a single traffic stream by using each entry's qosActionNext attribute. The qosActionNext attribute of the last action entry in the chain points to the next element in the TCB, if any, e.g. a Queueing element. It may also point at a next TCB.

The parameters needed for the Action element will depend on the type of Action to be taken. Hence the PIB allows for specific Action Tables for the different Action types. This flexibility allows additional Actions be specified in other PIBs and also allows for the use of proprietary Actions without impact on those defined here.

One may consider packet dropping as an Action element. Packet dropping is handled by the Algorithm Dropper datapath functional element.

4.5.1. DSCP Mark Action PRC

This Action is applied to traffic in order to mark it with a Diffserv Codepoint (DSCP) value, specified in the qosDscpMarkActTable.

4.6. Queueing Elements

These include Algorithmic Droppers, Queues and Schedulers, which are all inter-related in their use of queueing techniques.

4.6.1. Algorithmic Dropper PRC

Algorithmic Droppers are represented in this PIB by instances of the Algorithmic Dropper PRC. An Algorithmic Dropper is assumed to operate indiscriminately on all packets that are presented at its input, all traffic separation should be done by classifiers and meters preceding it.

Algorithmic Dropper includes many types of droppers, from the simple always dropper to the more complex random dropper. This is indicated by the qosAlgDropType attribute.

Algorithmic Droppers have a close relationship with queuing, each Algorithmic Dropper Table entry contains a qosAlgDropQMeasure attribute, indicating which queue's state affects the calculation of the Algorithmic Dropper. Each entry also contains a qosAlgDropNext attribute which indicates to which queue the Algorithmic Dropper sinks its traffic.

Algorithmic Droppers may also contain a pointer to specific detail of the drop algorithm, qosAlgDropSpecific. This PIB defines the detail for three drop algorithms: Tail Drop, Head Drop and Random

Drop; other algorithms are outside the scope of this PIB module but the general framework is intended to allow for their inclusion via other PIB modules.

One generally-applicable parameter of a dropper is the specification of a queue-depth threshold at which some drop action is to start. This is represented in this PIB, as a base attribute, qosAlgDropQThreshold, of the Algorithmic Dropper entry. The attribute, qosAlgDropQMeasure, specifies which queue's depth qosAlgDropQThreshold is to compare against.

- An Always Dropper drops every packet presented to it. This type of dropper does not require any other parameter.

- A Tail Dropper requires the specification of a maximum queue depth threshold: when the queue pointed at by qosAlgDropQMeasure reaches that depth threshold, qosAlgDropQThresh, any new traffic arriving at the dropper is discarded. This algorithm uses only parameters that are part of the qosAlgDropEntry.

- A Head Dropper requires the specification of a maximum queue depth threshold: when the queue pointed at by qosAlgDropQMeasure reaches that depth threshold, qosAlgDropQThresh, traffic currently at the head of the queue is discarded. This algorithm uses only parameters that are part of the qosAlgDropEntry.

- Random Droppers are recommended as a way to control congestion, in [QUEUEMGMT] and called for in the [AF-PHB]. Various implementations exist, which agree on marking or dropping just enough traffic to communicate with TCP-like protocols about congestion avoidance, but differ markedly on their specific parameters. This PIB attempts to offer a minimal set of controls for any random dropper, but expects that vendors will augment the PRC with additional controls and status in accordance with their implementation. This algorithm requires additional parameters on top of those in qosAlgDropEntry; these are discussed below.

4.6.2. Random Dropper PRC

One example of a random dropper is a RED-like dropper. An example of the representation chosen in this PIB for this element is shown in Figure 1.

Random droppers often have their drop probability function described as a plot of drop probability (P) against averaged queue length (Q). (Qmin, Pmin) then defines the start of the characteristic plot. Normally Pmin=0, meaning with average queue length below Qmin, there will be no drops. (Qmax, Pmax) defines a "knee" on the plot, after which point the drop probability become more progressive (greater slope). (Qclip, 1) defines the queue length at which all packets will be dropped. Notice this is different from Tail Drop because this uses an averaged queue length. Although it is possible for

Qclip = Qmax.

In the PIB module, qosRandomDropMinThreshBytes and qosRandomDropMinThreshPkts represent Qmin. QosRandomDropMaxThreshBytes and qosRandomDropMaxThreshPkts represent

Qmax. QosAlgDropQThreshold represents Qclip. qosRandomDropProbMax represents Pmax. This PIB does not represent Pmin (assumed to be zero unless otherwise represented).

In addition, since message memory is finite, queues generally have some upper bound above which they are incapable of storing additional traffic. Normally this number is equal to Qclip, specified by qosAlgDropQThreshold.

Each random dropper specification is associated with a queue. This allows multiple drop processes (of same or different types) to be associated with the same queue, as different PHB implementations may require. This also allows for sequences of multiple droppers if necessary.

```
         AlgDrop                                  Queue
         +-----------------+                   +--------+
     --->| Next   --------- --                          +  +----------------->|
Next -+-->
         | QMeasure -------+--+                 | ...   |
         | QThreshold      |   RandomDrop       +--------+
         | Type=randomDrop |   +----------------+
         | Specific -------+-->| MinThreshBytes |
         +-----------------+   | MaxThreshBytes |
                               | ProbMax        |
                               | InvWeight      |
                               | SamplingRate   |
                               +----------------+
```

Figure 1: Example Use of the RandomDropTable for Random Droppers

The calculation of a smoothed queue length may also have an important bearing on the behavior of the dropper: parameters may include the sampling interval or rate, and the weight of each sample. The performance may be very sensitive to the values of these parameters and a wide range of possible values may be required due to a wide range of link speeds. Most algorithms include a sample weight, represented here by qosRandomDropWeight. The availability of qosRandomDropSamplingRate as readable is important, the information provided by Sampling Rate is essential to the configuration of qosRandomDropWeight. Having Sampling Rate be configurable is also helpful, as line speed increases, the ability to have queue sampling be less frequent than packet arrival is needed. Note however that there is ongoing research on this topic, see e.g. [ACTQMGMT] and [AQMROUTER].

Additional parameters may be added in an enterprise PIB module, e.g. by using AUGMENTS on this table, to handle aspects of random drop algorithms that are not standardized here.

NOTE: Deterministic Droppers can be viewed as a special case of Random Droppers with the drop probability restricted to 0 and 1. Hence Deterministic Droppers might be described by a Random Dropper with Pmin = 0, Pmax = 1, Qmin = Qmax = Qclip, the averaged queue length at which dropping occurs.

4.6.3. Queues and Schedulers

The Queue PRC models simple FIFO queues, as described in [MODEL] section 7.1.1. The Scheduler PRC allows flexibility in constructing both simple and somewhat more complex queueing hierarchies from those queues. Of course, since TCBs can be cascaded multiple times on an interface, even more complex hierarchies can be constructed that way also.

Queue PRC instances are pointed at by the "next" attributes of the upstream elements e.g. qosMeterSucceedNext. Note that multiple upstream elements may direct their traffic to the same Queue PRI. For example, the Assured Forwarding PHB suggests that all traffic marked AF11, AF12 or AF13 be placed in the same queue, after metering, without reordering. This would be represented by having the qosMeterSucceedNext of each upstream meter point at the same Queue PRI.

NOTE: Queue and Scheduler PRIs are for data path description, they both use Scheduler Parameterization Table entries for diffserv treatment parameterization.

Queue Table entries specify the scheduler it wants service from by use of its Next pointer.

Each Scheduler Table entry represents the algorithm in use for servicing the one or more queues that feed it. The [MODEL] section 7.1.2 describes a scheduler with multiple inputs: this is represented in the PIB by having the scheduling parameters be associated with each input. In this way, sets of Queues can be grouped together as inputs to the same Scheduler. This table serves to represent the example scheduler described in the [MODEL]: other more complex representations might be created outside of this PIB.

Both the Queue PRC and the Scheduler PRC use instances of the Scheduler Parameterization PRC to specify diffserv treatment parameterization. Scheduler Parameter PRC instances are used to parameterize each input that feeds into a scheduler. The inputs can be a mixture of Queue PRI's and Scheduler PRI's. Scheduler Parameter PRI's can be used/reused by one or more Queue and/or Scheduler Table entries.

For representing a Strict Priority scheduler, each scheduler input is assigned a priority with respect to all the other inputs feeding the same scheduler, with default values for the other parameters. A higher-priority input which contains traffic that is not being delayed for shaping will be serviced before a lower-priority input.

For Weighted Scheduling methods e.g. WFQ, WRR, the "weight" of a given scheduler input is represented with a Minimum Service Rate leaky-bucket profile which provides guaranteed minimum bandwidth to that input, if required. This is represented by a rate qosMinRateAbsolute; the classical weight is the ratio between that rate and the interface speed, or perhaps the ratio between that rate and the sum of the configured rates for classes. The rate may, alternatively, be represented by a relative value, as a fraction of the interface's current line rate, qosMinRateRelative to assist in cases where line rates are variable or where a higher-level policy might be expressed in terms of fractions of network resources. The two rate parameters are inter-related and changes in one may be reflected in the other.

For weighted scheduling methods, one can say loosely, that WRR focuses on meeting bandwidth sharing, without concern for relative delay amongst the queues; where WFQ control both queue service order and amount of traffic serviced, providing meeting bandwidth sharing and relative delay ordering amongst the queues.

A queue or scheduled set of queues (which is an input to a scheduler) may also be capable of acting as a non-work-conserving [MODEL] traffic shaper: this is done by defining a Maximum Service Rate leaky-bucket profile in order to limit the scheduler bandwidth available to that input. This is represented by a rate qosMaxRateAbsolute; the classical weight is the ratio between that rate and the interface speed, or perhaps the ratio between that rate and the sum of the configured rates for classes. The rate may, alternatively, be represented by a relative value, as a fraction of the interface's current line rate, qosMaxRateRelative. There was discussion in the working group about alternative modeling approaches, such as defining a shaping action or a shaping element. We did not take this approach because shaping is in fact something a scheduler does to its inputs, (which we model as a queue with a maximum rate or a scheduler whose output has a maximum rate) and we felt it was simpler and more elegant to simply describe it in that context.

Other types of priority and weighted scheduling methods can be defined using existing parameters in qosMinRateEntry. NOTE: qosSchedulerMethod uses OBJECT IDENTIFIER syntax, with the different types of scheduling methods defined as OBJECT-IDENTITY. Future scheduling methods may be defined in other PIBs. This requires an OBJECT-IDENTITY definition, a description of how the existing objects are reused, if they are, and any new objects they require.

NOTE: hierarchical schedulers can be parameterized using this PIB by having Scheduler Table entries feeds into Scheduler Table entry.

4.7. Specifying Device Capabilities

The Diffserv PIB uses the Base PRC classes frwkPrcSupportTable and frwkCompLimitsTable defined in [FR-PIB] to specify what PRC's are supported by a PEP and to specify any limitations on that support. The PIB also uses the capability PRC's frwkIfCapSetTable and frwkIfRoleComboTable defined in [FR-PIB] to specify the device's interface types and role combinations. Each instance of the capability PRC frwkIfCapSetTable contains an OID that points to an instance of a PRC that describes some capability of that interface type. The Diffserv PIB defines several of these capability PRCs, which assist the PDP with the configuration of Diffserv functional elements that can be implemented by the device. Each of these capability PRCs contains a direction attribute that specifies the direction for which the capability applies. This attribute is defined in a base capability PRC, which is extended by each specific capability PRC.

Classification capabilities, which specify the information elements the device can use to classify traffic, are reported using the qosIfClassificationCaps PRC. Metering capabilities, which indicate what the device can do with out-of-profile packets, are specified using the qosIfMeteringCaps PRC. Scheduling capabilities, such as the number of inputs supported, are reported using the qosIfSchedulingCaps PRC. Algorithmic drop capabilities, such as the types of algorithms supported, are reported using the qosIfAlgDropCaps PRC. Queue capabilities, such as the maximum number of queues, are reported using the qosIfQueueCaps PRC. Maximum Rate capabilities, such as the maximum number of max rate Levels, are reported using the qosIfMaxRateCaps PRC.

Two PRC's are defined to allow specification of the element linkage capabilities of the PEP. The qosIfElmDepthCaps PRC indicates the maximum number of functional datapath elements that can be linked consecutively in a datapath. The qosIfElmLinkCaps PRC indicates what functional datapath elements may follow a specific type of element in a datapath.

The capability reporting classes in the DiffServ and Framework PIB are meant to allow the PEP to indicate some general guidelines about what the device can do. They are intended to be an aid to the PDP when it constructs policy for the PEP. These classes do not necessarily allow the PEP to indicate every possible configuration that it can or cannot support. If a PEP receives a policy that it cannot implement, it must notify the PDP with a failure report.

5. PIB Usage Example

This section provides some examples on how the different table entries of this PIB may be used together for a Diffserv Device. The usage of each individual attribute is defined within the PIB module itself. For the figures, all the PIB table entry and attribute names are assumed to have "qos" as their first common initial part of the name, with the table entry name assumed to be their second common initial part of the name. "0.0" is being used to mean zeroDotZero.

And for Scheduler Method "= X" means "using the OID of qoxSchedulerX".

5.1. Data Path Example

Notice Each entry of the DataPath table is used for a specific interface type handling a flow in a specific direction for a specific functional role-combination. For our example, we just define one of such entry.

```
+---------------------+
|DataPath             |
| IfName ="IfCap1"    |
| Roles = "A+B"       |
| IfDirection=Ingress |    +---------+
| Start --------------+--->|Clfr     |
+---------------------+    | Id=Dept |
                           +---------+
```

Figure 2: DataPath Usage Example

In Figure 2, we are using IfCap1 to indicate interface type with capability set 1 handling ingress flow for functional roles of _A+B_. We are using classifier for departments to lead us into the Classifier Example below.

5.2. Classifier and Classifier Element Example

We want to show how a multilevel classifier can be built using the classifier tables provided by this PIB. Notice we didn't go into details of the filters because they are not defined by this PIB. Continuing from the Data Path example from the previous section, let say we want to perform the following classification functionality to do flow separation based on department and application type:

```
if (Dept1) then take Dept1-action
{
  if (Appl1) then take Dept1-Appl1-action.
  if (Appl2) then take Dept1-Appl2-action.
  if (Appl3) then take Dept1-Appl3-action.

}
if (Dept2) then take Dept2-action
{
  if (Appl1) then take Dept2-Appl1-action.
  if (Appl2) then take Dept2-Appl2-action.
  if (Appl3) then take Dept2-Appl3-action.
```

```
}
if (Dept3) then take Dept3-action
{
  if (Appl1) then take Dept3-Appl1-action.
```

```
    if (Appl2) then take Dept3-Appl2-action.
    if (Appl3) then take Dept3-Appl3-action.
  }
```

The above classification logic is translated into PIB table entries below, with two levels of classifications.

First for department:

```
+---------+
|Clfr     |
| Id=Dept |
+---------+

+-------------+      +-----------+
|ClfrElement  |  +-->|Clfr       |
| Id=Dept1    |  |   | Id=D1Appl |
| ClfrId=Dept |  |   +-----------+
| Preced=NA   |  |
| Next -------+--+   +------------+
| Specific ---+----->|Filter Dept1|
+-------------+      +------------+

+-------------+      +-----------+
|ClfrElement  |  +-->|Clfr       |
| Id=Dept2    |  |   | Id=D2Appl |
| ClfrId=Dept |  |   +-----------+
| Preced=NA   |  |
| Next -------+--+   +------------+
| Specific ---+----->|Filter Dept2|
+-------------+      +------------+

+-------------+      +-----------+
|ClfrElement  |  +-->|Clfr       |
| Id=Dept3    |  |   | Id=D3Appl |
| ClfrId=Dept |  |   +-----------+
| Preced=NA   |  |
| Next -------+--+   +------------+
| Specific ---+----->|Filter Dept3|
+-------------+      +------------+
```

Second for application:

```
+-----------+
|Clfr       |
| Id=D1Appl |
+-----------+

+---------------+                     +--------------+
|ClfrElement    |  +----------------->|Meter         |
| Id=D1Appl1    |  |                  | Id=D1A1Rate1 |
| ClfrId=D1Appl |  |                  | SucceedNext -+--->...
| Preced=NA     |  |                  | FailNext ----+--->...
| Next ---------+--+  +------------+  | Specific ----+--->...
```

```
   | Specific -----+---->|Filter Appl1|  +--------------+
   +---------------+     +------------+

   +---------------+                     +--------------+
   |ClfrElement    |  +----------------->|Meter         |
   | Id=D1Appl2    |  |                  | Id=D1A2Rate1 |
   | ClfrId=D1Appl |  |                  | SucceedNext -+--->...
   | Preced=NA     |  |                  | FailNext ----+--->...
   | Next ---------+--+  +------------+  | Specific ----+--->...
   | Specific -----+---->|Filter Appl2|  +--------------+
   +---------------+     +------------+

   +---------------+                     +--------------+
   |ClfrElement    |  +----------------->|Meter         |
   | Id=D1Appl3    |  |                  | Id=D1A3Rate1 |
   | ClfrId=D1Appl |  |                  | SucceedNext -+--->...
   | Preced=NA     |  |                  | FailNext ----+--->...
   | Next ---------+--+  +------------+  | Specific ----+--->...
   | Specific -----+---->|Filter Appl3|  +--------------+
   +---------------+     +------------+
```

Figure 3: Classifier Usage Example

The application classifiers for department 2 and 3 will be very much like the application classifier for department 1 shown above. Notice in this example, Filters for Appl1, Appl2, and Appl3 are reusable across the application classifiers.

This classifier and classifier element example assumes the next differentiated services functional datapath element is Meter and lead us into the Meter Example section.

5.3. Meter Example

A single rate simple Meter may be easy to envision, hence we will do a Two Rate Three Color [TRTCM] example, using two Meter table entries and two TBParam table entries.

```
   +--------------+    +---------+     +--------------+    +----------+
   |Meter         | +->|Action   |  +->| Meter        | +->|Action    |
   | Id=D1A1Rate1 | |  | Id=Green|  |  | Id=D1A1Rate2 | |  | Id=Yellow|
   | SucceedNext -+ +  +                   -    ---------+  |  | SucceedNext -
+-+  +----------+
   | FailNext ----+-----------------+  | FailNext ----+--+  +-------+
   | Specific -+  |                    | Specific -+  |  +->|Action |
   +-----------+--+                    +-----------+--+     | Id=Red|
               |                                   |        +-------+
|  +------------+
+->|TBMeter     |
   | Type=TRTCM |
   | Rate       |
   | BurstSize  |
```

```
|  +------------+
+->|TBMeter     |
   | Type=TRTCM |
   | Rate       |
   | BurstSize  |
                  | Interval   |                      | Interval   |
                  +------------+                      +------------+
```

Figure 4: Meter Usage Example

For [TRTCM], the first level TBMeter entry is used for Committed Information Rate and Committed Burst Size Token Bucket, and the second level TBMeter entry is used for Peak Information Rate and Peak Burst Size Token Bucket.

The other meters needed for this example will depend on the service class each classified flow uses. But their construction will be similar to the example given here. The TBMeter table entries can be shared by multiple Meter table entries.

In this example the differentiated services functional datapath element following Meter is Action, detailed in the following section.

5.4. Action Example

Typically Mark Action will be used, we will continue using the _Action, Id=Green_ branch off the Meter example.
Recall this is the D1A1Rate1 SucceedNext branch, meaning the flow belongs to Department 1 Application 1, within the committed rate and burst size limits for this flow. We would like to Mark this flow with a specific DSCP and also with a device internal label.

```
   +-----------+                     +-----------+  +--->AlgDropAF11
   |Action      |  +----------------->|Action      |  |
   | Next -----+--+  +------------+  | Next -----+--+ +-------------+
   | Specific -+---->|DscpMarkAct |  | Specific -+--->|ILabelMarker |
   +-----------+     | Dscp=AF11  |  +-----------+    | ILabel=D1A1 |
                     +------------+                   +-------------+
```

Figure 5: Action Usage Example

This example uses the frwkILabelMarker PRC defined in [FR-PIB], showing the device internal label being used to indicate the micro flow that feeds into the aggregated AF flow. This device internal label may be used for flow accounting purposes and/or other data path treatments.

5.5. Dropper Examples

The Dropper examples below will continue from the Action example above for AF11 flow. We will provide three different dropper setups, from simple to complex. The examples below may include some queuing structures, they are here only to show the relationship of the droppers to queuing and are not complete. Queuing Example are provided in later sections.

5.5.1. Tail Dropper Example

The Tail Dropper is one of the simplest. For this example we just want to drop part of the flow that exceeds the queue's buffering capacity, 2 Mbytes.

```
   +--------------------+       +-------+
   |AlgDrop             |    +->|Q AF1 |
   | Id=AF11            |    |  +-------+
   | Type=tailDrop      |    |
   | Next --------------+-+--+
   | QMeasure ----------+-+
   | QThreshold=2Mbytes |
   | Specific=0.0       |
   +--------------------+
```

Figure 6: Tail Dropper Usage Example

5.5.2. Single Queue Random Dropper Example

Use of Random Dropper will introduce the usage of qosRandomDropEntry as in the example below.

```
   +-----------------+       +-------+
   |AlgDrop          |    +->|Q AF1 |
   | Id=AF11         |    |  +-------+
   | Type=randomDrop |    |
   | Next -----------+-+--+
   | QMeasure -------+-+
   | QThreshold      |   +----------------+
   | Specific -------+-->|RandomDrop      |
   +-----------------+   | MinThreshBytes |
                         | MinThreshPkts  |
                         | MaxThreshBytes |
                         | MaxThreshPkts  |
                         | ProbMax        |
                         | Weight         |
                         | SamplingRate   |
                         +----------------+
```

Figure 7: Single Queue Random Dropper Usage Example

Notice for Random Dropper, qosAlgDropQThreshold contains the maximum average queue length, Qclip, for the queue being measured as indicated by qosQMeasure, the rest of the Random Dropper parameters are specified by qosRandomDropEntry as referenced by qosSpecific.

In this example, both qosNext and qosQMeasure references the same queue. This is the simple case but qosQMeasure may reference another queue for PEP implementation supporting this feature.

5.5.3. Multiple Queue Random Dropper Example

When network device implementation requires measuring multiple queues for determining the behavior of a drop algorithm, the existing PRCs defined in this PIB will be sufficient for the simple case, as indicated by this example.

```
   +-------------+                                         +------+
   |AlgDrop      | +----------------+-------------------+->|Q_AF1 |
   | Id=AF11     | |                |                   |  +------+
   | Type=mQDrop | |                |                   |
   | Next -------+-+ +------------+ |    +------------+ |
   | QMeasure ---+-->|MQAlgDrop   | | +->|MQAlgDrop   | |
   | QThreshold  |   | Id=AF11A   | | |  | Id=AF11B   | |
   | Specific    |   | Type       | | |  | Type       | |
   +-------------+   | Next ------+-+ |  | Next ------+-+
                     | ExceedNext +---+  | ExceedNext |   +------+
                     | QMeasure --+-+    | QMeasure --+-->|Q AF2 |
                     | QThreshold | |    | QThreshold |   +------+
                     | Specific + | |    | Specific + |
                     +----------+-+ |    +----------+-+
                                |   |           +---+
                         +------+   |  +------+ |
                         |          +->|Q AF1 | |
                         |             +------+ |
                         |                      |
                         |  +----------------+  |  +----------------+
                         +->|RandomDrop      |  +->|RandomDrop      |
                            | MinThreshBytes |     | MinThreshBytes |
                            | MinThreshPkts  |     | MinThreshPkts  |
                            | MaxThreshBytes |     | MaxThreshBytes |
                            | MaxThreshPkts  |     | MaxThreshPkts  |
                            | ProbMax        |     | ProbMax        |
                            | Weight         |     | Weight         |
                            | SamplingRate   |     | SamplingRate   |
                            +----------------+     +----------------+
```

Figure 8: Multiple Queue Random Dropper Usage Example

For this example, we have two queues, Q_AF1 and Q_AF2, sharing the same buffer resources. We want to make sure the common buffer resource is sufficient to service the AF11 traffic, and we want to measure the two queues for determining the drop algorithm for AF11 traffic feeding into _Q_AF1_. Notice mQDrop is used for qosAlgDropType of qosAlgDropEntry to indicate Multiple Queue Dropping Algorithm.

The common shared buffer resource is indicated by the use of qosAlgDropEntry, with their attributes used as follows:

- qosAlgDropType indicates the algorithm used, mQDrop.
- qosAlgDropNext is used to indicate the next functional data path element to handle the flow when no drop occurs.
- qosAlgDropQMeasure is used as the anchor for the list of qosMQAlgDropEntry, one for each queue being measured.
- qosAlgDropQThreshold is used to indicate the size of the shared buffer pool.
- qosAlgDropSpecific can be used to reference instance of additional PRC (not defined in this PIB) if more parameters are required to describe the common shared buffer resource.

For this example, there are two subsequent qosMQAlgDropEntry, one for each queue being measured, with its attributes used as follows:

- qosMQAlgDropType indicates the algorithm used, for this example, both qosMQAlgDropType uses randomDrop.
- qosMQAlgDropQMeasure indicates the queue being measured.
- qosMQAlgDropNext indicates the next functional data path element to handle the flow when no drop occurs.
- qosMQAlgDropExceedNext is used to indicate the next queue's qosMQAlgDropEntry. With the use of zeroDotZero to indicate the last queue.
- qosMQAlgDropQMeasure is used to indicate the queue being measured. For this example, _Q AF1_ and _Q_AF2_ are the two queues used.
- qosAlgDropQThreshold is used as in single queue Random Dropper.
- qosAlgDropSpecific is used to reference the PRID that describes the dropper parameters as in its normal usage. For this example both qosAlgDropSpecific reference qosRandomDropEntrys.

Notice the anchoring qosAlgDropEntry and the two qosMQAlgDropEntrys all have their Next attribute pointing to Q_AF1. This indicates:

- If the packet does not need to be checked with the individual queue's drop processing because of abundance of common shared buffer resources, then the packet is sent to Q_AF1.
- If the packet is not dropped due to current Q_AF1 conditions, then it is sent to Q_AF1.
- If the packet is not dropped due to current Q_AF2 conditions, then it is sent to Q_AF1.

This example also uses two qosRandomDropEntry for the two queues it measures. Their attribute usage is the same as if for single queue random dropper.

Other more complex result combinations can be achieved by specifying a new PRC and referencing this new PRC with qosAlgDropSpecific of the anchoring qosAlgDropEntry. More simple usage can also be achieved when a single set of drop parameters are used for all queues being measured. This again can be referenced by the anchoring qosAlgDropSpecific. These are not defined in this PIB.

5.6. Queue and Scheduler Example

The queue and scheduler example will continue from the dropper example in previous section. Concentrating in the queue and scheduler Diffserv datapath functional elements. Notice a shaper is constructed using queue and scheduler with MaxRate parameters.

```
        +------------+                           +-----------------+
   ---->|Q           |                        +->|Scheduler        |
        | Id=EF      |                        |  | Id=Diffserv     |
        | Next ------+------------------------+  | Next=0.0        |
        | MinRate ---+--+                     |  | Method=Priority |
        | MaxRate -+ |  |   +----------+      |  | MinRate=0.0     |
        +----------+-+  +-->|MinRate   |      |  | MaxRate=0.0     |
                   |        | Priority |      |  +-----------------+
        +----------+        | Absolute |      |
        |                   | Relative |      |
        |  +-----------+    +----------+      |
        +->|MaxRate    |                      |
           | Level     |                      |
           | Absolute  |                      |
           | Relative  |                      |
           | Threshold |                      |
           +-----------+                      +-------------+
                                                            |
        +----------+                        +------------+  |
   ---->|Q         |                    +-->|Scheduler   |  |
        | Id=AF1   |                    |   | Id=AF      |  |
        | Next ----+--------------------+   | Next ------+--+
        | MinRate -+-+                  |   | Method=WRR |
        | MaxRate  | |  +----------+    |   | MinRate -+ |
        +----------+ +->|MinRate   |    |   | MaxRate  | |
                        | Priority |    |   +----------+-+
                        | Absolute |    |              |
                        | Relative |    |   +----------+
                        +----------+    |   |
        +----------+                    |   |  +------------+
   ---->|Q         |                    |   +->|MinRate     |
        | Id=AF2   |                    |      | Priority   |
        | Next ----+--------------------+      | Absolute   |
        | MinRate -+-+                  |      | Relative   |
        | MaxRate  | |  +----------+    |      +------------+
        +----------+ +->|MinRate   |    |
                        | Priority |    |
                        | Absolute |    |
                        | Relative |    |
                        +----------+    |
        +----------+                    |
   ---->|Q         |                    |
        | Id=AF3   |                    |
        | Next ----+--------------------+
        | MinRate -+-+
        | MaxRate  | |  +----------+
        +----------+ +->|MinRate   |
                        | Priority |
        | Absolute |
        | Relative |
        +----------+
```

Figure 9: Queue and Scheduler Usage Example

This example shows the queuing system for handling EF, AF1, AF2, and AF3 traffic. It is assumed AF11, AF12, and AF13 traffic feeds into Queue AF1. And likewise for AF2x and AF3x traffic.

The AF1, AF2, and AF3 Queues are serviced by the AF Scheduler using a Weighed Round Robin method. The AF Scheduler will service each of the queues feeding into it based on the minimum rate parameters of each queue.

The AF and EF traffic are serviced by the DiffServ Scheduler using a Strict Priority method. The DiffServ Scheduler will service each of its inputs based on their priority parameter.

Notice there is an upper bound to the servicing of EF traffic by the DiffServ Scheduler. This is accomplished with the use of maximum rate parameters. DiffServ Scheduler uses both the maximum rate and priority parameters when servicing the EF Queue.

The DiffServ Scheduler is the last Diffserv datapath functional element in this datapath. It uses zeroDotZero in its Next attribute.

6. 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 capability PRC's that allow a PEP to specify its device characteristics to the PDP. This module contains two groups, which are summarized in this section.

QoS Capabilities 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 Policy Group
This group contains configuration of the functional elements that comprise the QoS policy that applies to an interface and the specific parameters that describe those elements. This group

contains classifiers, meters, actions, droppers, queues and schedulers. This group also contains the PRC that associates the datapath elements with role combinations.

7. 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 Provisioning 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 contains a pointer to a instance of a PRC that describes the capabilities of the interface type. The capability objects may reside in the qosIfClassifierCapsTable, the qosIfMeterCapsTable, the qosIfSchedulerCapsTable, the qosIfElmDepthCapsTable, the qosIfElmOutputCapsTable, or in a table defined in another PIB.

The PDP, with knowledge of the PEP's capabilities, then provides the PEP with administration domain and interface-specific policy information.

Instances of the qosDataPathTable are used to specify the first element in the set of functional elements applied to an interface. Each instance of the qosDataPathTable applies to an interface type defined by its roles and direction (ingress or egress).

8. PIB Definitions

8.1. The DiffServ Base PIB

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

   IMPORTS
       Unsigned32, Integer32,
       MODULE-IDENTITY, OBJECT-TYPE
               FROM COPS-PR-SPPI
       zeroDotZero
            FROM SNMPv2-SMI
       TruthValue, TEXTUAL-CONVENTION
               FROM SNMPv2-TC
       InstanceId, ReferenceId, TagId, TagReferenceId, pib
               FROM COPS-PR-SPPI
       RoleCombination, PrcIdentifier, AttrIdentifier
               FROM FRAMEWORK-ROLE-PIB
       Dscp
               FROM DIFFSERV-DSCP-TC
       IfDirection
               FROM DIFF-SERV-MIB
       BurstSize
               FROM INTEGRATED-SERVICES-MIB;


   qosPolicyPib  MODULE-IDENTITY
       SUBJECT-CATEGORIES { tbd } -- DiffServ QoS COPS Client Type
                                  -- to be assigned by IANA
       LAST-UPDATED "200111071800Z"
       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 provisioning 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."

       ::= { pib xxx } -- xxx to be assigned by IANA


   qosCapabilityClasses    OBJECT IDENTIFIER ::= { qosPolicyPib 1 }
   qosPolicyClasses        OBJECT IDENTIFIER ::= { qosPolicyPib 2 }
   qosPolicyParameters     OBJECT IDENTIFIER ::= { qosPolicyPib 3 }
   qosPolicyPibConformance OBJECT IDENTIFIER ::= { qosPolicyPib 4 }


   --
   -- Interface Capabilities Group
   --

   --
   -- Interface Type Capability Tables
   --
   -- The Interface type capability tables define capabilities that may
   -- be associated with interfaces 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.

   --
   -- The Base Capability Table
   --

   qosBaseIfCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosBaseIfCapsEntry
       PIB-ACCESS     notify
       STATUS         current
       DESCRIPTION
         "The Base Interface Capability class.  This class represents
          a generic capability supported by a device in the ingress,
          egress or both directions."
       ::= { qosCapabilityClasses 1 }

   qosBaseIfCapsEntry OBJECT-TYPE
       SYNTAX         QosBaseIfCapsEntry
       STATUS         current
       DESCRIPTION
         "An instance of this class describes the qosBaseIfCaps class."
```

```
        PIB-INDEX { qosBaseIfCapsPrid }
    ::= { qosBaseIfCapsTable 1 }

QosBaseIfCapsEntry ::= SEQUENCE {
```

```
           qosBaseIfCapsPrid           InstanceId,
           qosBaseIfCapsDirection      Integer32
   }

   qosBaseIfCapsPrid OBJECT-TYPE
       SYNTAX         InstanceId
       STATUS         current
       DESCRIPTION
           "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosBaseIfCapsEntry 1 }


   qosBaseIfCapsDirection OBJECT-TYPE
       SYNTAX         Integer32 {
                           inbound(1),
                           outbound(2),
                           inAndOut(3)
                      }
       STATUS         current
       DESCRIPTION
         "This object specifies the direction(s) for which the
         capability applies. A value of 'inbound(1)' means the
         capability applies only to the ingress direction.  A value of
         'outbound(2)' means the capability applies only to the egress
         direction.  A value of 'inAndOut(3)' means the capability
         applies to both directions."
       ::= { qosBaseIfCapsEntry 2 }


   --
   -- The Classification Capability Table
   --


   qosIfClassificationCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosIfClassificationCapsEntry
       PIB-ACCESS     notify
       STATUS         current
       DESCRIPTION
           "This table specifies the classification capabilities of an
           interface type"
       ::= { qosCapabilityClasses 2 }


   qosIfClassificationCapsEntry OBJECT-TYPE
       SYNTAX         QosIfClassificationCapsEntry
       STATUS         current
       DESCRIPTION
           "An instance of this class describes the classification
```

```
        capabilities of an interface."

    EXTENDS { qosBaseIfCapsEntry }
```

```
       UNIQUENESS { qosBaseIfCapsDirection,
                    qosIfClassificationCapsSpec }
       ::= { qosIfClassificationCapsTable 1 }


   QosIfClassificationCapsEntry ::= SEQUENCE {
           qosIfClassificationCapsSpec BITS
   }


   qosIfClassificationCapsSpec OBJECT-TYPE
       SYNTAX       BITS {
                          ipSrcAddrClassification(1),
                          -- indicates the ability to classify based on
                          -- IP source addresses
                          ipDstAddrClassification(2),
                          -- indicates the ability to classify based on
                          -- IP destination addresses
                          ipProtoClassification(3),
                          -- indicates the ability to classify based on
                          -- IP protocol numbers
                          ipDscpClassification(4),
                          -- indicates the ability to classify based on
                          -- IP DSCP
                          ipL4Classification(5),
                          -- indicates the ability to classify based on
                          -- IP layer 4 port numbers for UDP and TCP
                          ipV6FlowID(6)
                          -- indicates the ability to classify based on
                          -- IPv6 FlowIDs.
                         }
       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 1 }


   --
   -- Metering Capabilities
   --


   qosIfMeteringCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosIfMeteringCapsEntry
       PIB-ACCESS     notify
```

```
STATUS          current
DESCRIPTION
    "This table specifies the metering capabilities of an
    interface type"
::= { qosCapabilityClasses 3 }
```

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

       EXTENDS { qosBaseIfCapsEntry }
       UNIQUENESS { qosBaseIfCapsDirection,
                    qosIfMeteringCapsSpec }
       ::= { qosIfMeteringCapsTable 1 }


   QosIfMeteringCapsEntry ::= SEQUENCE {
           qosIfMeteringCapsSpec       BITS
   }


   qosIfMeteringCapsSpec OBJECT-TYPE
       SYNTAX  BITS {
                     SimpleTokenBucket(1),
                     AvgRate(2),
                     SrTCMBlind(3),
                     SrTCMAware(4),
                     TrTCMBlind(5),
                     TrTCMAware(6),
                     TswTCM(7)
                    }
       STATUS       current
       DESCRIPTION
         "Bit set of supported metering capabilities.  As with
         classification capabilities, these metering capabilities may
         be augmented by capabilities specified in other PRCs (in other
         PIBs)."
       ::= { qosIfMeteringCapsEntry 1 }



   --
   -- Algorithmic Dropper Capabilities
   --
   -- This capability table indicates the types of algorithmic
   -- drop supported by an interface type for a specific flow
   -- direction.
   -- Additional capabilities affecting the drop functionalities
   -- are determined based on queue capabilities associated with
   -- specific instance of a dropper, hence not specified by
   -- this table.
```

```
--


qosIfAlgDropCapsTable OBJECT-TYPE
```

```
        SYNTAX         SEQUENCE OF QosIfAlgDropCapsEntry
        PIB-ACCESS     notify
        STATUS         current
        DESCRIPTION
            "This table specifies the algorithmic dropper
            capabilities of an interface type"
        ::= { qosCapabilityClasses 4 }

    qosIfAlgDropCapsEntry OBJECT-TYPE
        SYNTAX         QosIfAlgDropCapsEntry
        STATUS         current
        DESCRIPTION
            "An instance of this class describes the algorithm dropper
            capabilities of an interface."
        EXTENDS { qosBaseIfCapsEntry }
        UNIQUENESS { qosBaseIfCapsDirection,
                     qosIfAlgDropCapsType,
                     qosIfAlgDropCapsMQCount }
        ::= { qosIfAlgDropCapsTable 1 }

    QosIfAlgDropCapsEntry ::= SEQUENCE {
            qosIfAlgDropCapsType                BITS,
            qosIfAlgDropCapsMQCount             Unsigned32
    }

    qosIfAlgDropCapsType OBJECT-TYPE
        SYNTAX      BITS {
                         tailDrop(2),
                         headDrop(3),
                         randomDrop(4),
                         alwaysDrop(5),
                         mQDrop(6) }
        STATUS      current
        DESCRIPTION
          "The type of algorithm that droppers associated with queues
          may use.

          The tailDrop(2) algorithm means that packets are dropped from
          the tail of the queue when the associated queue's MaxQueueSize
          is exceeded.  The headDrop(3) algorithm means that packets are
          dropped from the head of the queue when the associated queue's
          MaxQueueSize is exceeded. The randomDrop(4) algorithm means
          that an algorithm is executed which may randomly
          drop the packet, or  drop  other  packet(s) from  the  queue
          in  its place.  The specifics of the algorithm may be
          proprietary.  However, parameters would be specified in the
          qosRandomDropTable.  The alwaysDrop(5) will drop every packet
          presented to it.  The mQDrop(6) algorithm will drop packets
          based on measurement from multiple queues."
    ::= { qosIfAlgDropCapsEntry 1 }

qosIfAlgDropCapsMQCount OBJECT-TYPE
    SYNTAX      Unsigned32
    STATUS      current
```

```
       DESCRIPTION
         "Indicates the number of queues measured for the drop
         algorithm.
         This attribute is ignored when alwaysDrop(5) algorithm is
         used.  This attribute contains the value of 1 for all drop
         algorithm types except for mQDrop(6), where this attribute
         is used to indicate the maximum number of qosMQAlgDropEntry
         that can be chained together."
       DEFVAL   { 1 }
       ::= { qosIfAlgDropCapsEntry 2 }


   --
   -- Queue Capabilities
   --

   qosIfQueueCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosIfQueueCapsEntry
       PIB-ACCESS     notify
       STATUS         current
       DESCRIPTION
           "This table specifies the scheduling capabilities of an
           interface type"
       ::= { qosCapabilityClasses 5 }

   qosIfQueueCapsEntry OBJECT-TYPE
       SYNTAX         QosIfQueueCapsEntry
       STATUS         current
       DESCRIPTION
           "An instance of this class describes the queue
           capabilities of an interface type."
       EXTENDS { qosBaseIfCapsEntry }
       UNIQUENESS { qosBaseIfCapsDirection,
                    qosIfQueueCapsMinQueueSize,
                    qosIfQueueCapsMaxQueueSize,
                    qosIfQueueCapsTotalQueueSize }
       ::= { qosIfQueueCapsTable 1 }

   QosIfQueueCapsEntry ::= SEQUENCE {
           qosIfQueueCapsMinQueueSize          Unsigned32,
           qosIfQueueCapsMaxQueueSize          Unsigned32,
           qosIfQueueCapsTotalQueueSize        Unsigned32
   }

   qosIfQueueCapsMinQueueSize OBJECT-TYPE
       SYNTAX      Unsigned32
       STATUS      current
       DESCRIPTION
           "Some interfaces may allow the size of a queue to be
```

```
        configured.  This attribute specifies the minimum size that
        can be configured for a queue, specified in bytes."
    ::= { qosIfQueueCapsEntry 1 }

qosIfQueueCapsMaxQueueSize OBJECT-TYPE
```

```
           SYNTAX      Unsigned32
           STATUS      current
           DESCRIPTION
               "Some interfaces may allow the size of a queue to be
               configured.  This attribute specifies the maximum size that
               can be configured for a queue, specified in bytes."
           ::= { qosIfQueueCapsEntry 2 }

       qosIfQueueCapsTotalQueueSize OBJECT-TYPE
           SYNTAX      Unsigned32
           STATUS      current
           DESCRIPTION
             "Some interfaces may have a limited buffer space to be shared
             amongst 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 in bytes available with this
             capability."
           ::= { qosIfQueueCapsEntry 3 }


       --
       -- Scheduler Capabilities
       --

       qosIfSchedulerCapsTable OBJECT-TYPE
           SYNTAX         SEQUENCE OF QosIfSchedulerCapsEntry
           PIB-ACCESS     notify
           STATUS         current
           DESCRIPTION
             "This table specifies the scheduler capabilities of an
             interface type"
           ::= { qosCapabilityClasses 6 }

       qosIfSchedulerCapsEntry OBJECT-TYPE
           SYNTAX         QosIfSchedulerCapsEntry
           STATUS         current
           DESCRIPTION
             "An instance of this class describes the scheduler
             capabilities of an interface type."
           EXTENDS { qosBaseIfCapsEntry }
           UNIQUENESS { qosBaseIfCapsDirection,
                        qosIfSchedulerCapsServiceDisc,
                        qosIfSchedulerCapsMaxInputs }
           ::= { qosIfSchedulerCapsTable 1 }

       QosIfSchedulerCapsEntry ::= SEQUENCE {
               qosIfSchedulerCapsServiceDisc      OBJECT IDENTIFIER,
        qosIfSchedulerCapsMaxInputs        Unsigned32,
        qosIfSchedulerCapsMinMaxRate       BITS
}

qosIfSchedulerCapsServiceDisc OBJECT-TYPE
```

```
       SYNTAX      OBJECT IDENTIFIER
       STATUS      current
       DESCRIPTION
         "The scheduling discipline for which the set of capabilities
         specified in this object apply. Object identifiers for several
         general purpose and well-known scheduling disciplines are
   defined
         in the Scheduler Method Parameters section of this PIB.
         These include Priority, WRR, WFQ."
       ::= { qosIfSchedulerCapsEntry 1 }

   qosIfSchedulerCapsMaxInputs OBJECT-TYPE
       SYNTAX      Unsigned32
       STATUS      current
       DESCRIPTION
         "The maximum number of queues and/or schedulers that can
         feed into a scheduler indicated by this capability entry
         for this interface type.  A value of zero means there
         is no maximum."
       ::= { qosIfSchedulerCapsEntry 2 }

   qosIfSchedulerCapsMinMaxRate OBJECT-TYPE
       SYNTAX BITS {
                    MinRate(1),
                    MaxRate(2),
                    MinAndMaxRates(3)
                   }
       STATUS      current
       DESCRIPTION
         "Scheduler capability indicating ability to handle inputs
         with minimum rate, maximum rate, or both."
       ::= { qosIfSchedulerCapsEntry 3 }


   --
   -- Maximum Rate Capabilities
   --

   qosIfMaxRateCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosIfMaxRateCapsEntry
       PIB-ACCESS     notify
       STATUS         current
       DESCRIPTION
           "This table specifies the maximum rate capabilities of an
           interface type"
       ::= { qosCapabilityClasses 7 }

   qosIfMaxRateCapsEntry OBJECT-TYPE
       SYNTAX         QosIfMaxRateCapsEntry
```

```
STATUS         current
DESCRIPTION
    "An instance of this class describes the maximum rate
    capabilities of an interface type."
EXTENDS { qosBaseIfCapsEntry }
```

```
        UNIQUENESS { qosBaseIfCapsDirection,
                     QosIfMaxRateCapsMaxLevels }
        ::= { qosIfMaxRateCapsTable 1 }

    QosIfMaxRateCapsEntry ::= SEQUENCE {
            qosIfMaxRateCapsMaxLevels           Unsigned32
    }

    qosIfMaxRateCapsMaxLevels OBJECT-TYPE
        SYNTAX      Unsigned32
        STATUS      current
        DESCRIPTION
            "The maximum number of levels a maximum rate specification
            may have for this interface type and flow direction."
        ::= { qosIfMaxRateCapsEntry 1 }


    --
    -- Datapath Element Linkage Capabilities
    --

    --
    -- Datapath Element Cascade Depth
    --


    qosIfElmDepthCapsTable OBJECT-TYPE
        SYNTAX         SEQUENCE OF QosIfElmDepthCapsEntry
        PIB-ACCESS     notify
        STATUS         current
        DESCRIPTION
            "This table specifies the number of elements of the same
            type that can be cascaded together in a data path."
        ::= { qosCapabilityClasses 8 }


    qosIfElmDepthCapsEntry OBJECT-TYPE
        SYNTAX         QosIfElmDepthCapsEntry
        STATUS         current
        DESCRIPTION
            "An instance of this class describes the cascade depth
            for a particular functional datapath element PRC.  A
            functional datapath element not represented in this
            table can be assumed to have no specific maximum
            depth."
        EXTENDS { qosBaseIfCapsEntry }
        UNIQUENESS { qosBaseIfCapsDirection,
                     qosIfElmDepthCapsPrc }
        ::= { qosIfElmDepthCapsTable 1 }
```

```
QosIfElmDepthCapsEntry ::= SEQUENCE {
        qosIfElmDepthCapsPrc                PrcIdentifier,
        qosIfElmDepthCapsCascadeMax         Unsigned32
```

```
   }


   qosIfElmDepthCapsPrc OBJECT-TYPE
       SYNTAX         PrcIdentifier
       STATUS         current
       DESCRIPTION
         "The object identifier of a PRC that represents a functional
         datapath element.  This may be one of:  qosClfrElementEntry,
         qosMeterEntry, qosActionEntry, qosAlgDropEntry, qosQEntry, or
         qosSchedulerEntry.  The value is the OID of the table entry.
         There may not be more than one instance of this class with
         the same value of qosIfElmDepthCapsPrc."
       ::= { qosIfElmDepthCapsEntry 1 }


   qosIfElmDepthCapsCascadeMax OBJECT-TYPE
       SYNTAX         Unsigned32
       STATUS         current
       DESCRIPTION
         "The maximum number of elements of type qosIfElmDepthCapsPrc
         that can be linked consecutively in a data path.  A value of
         zero indicates there is no specific maximum."
       ::= { qosIfElmDepthCapsEntry 2 }


   --
   -- Datapath Element Linkage Types
   --


   qosIfElmLinkCapsTable OBJECT-TYPE
       SYNTAX         SEQUENCE OF QosIfElmLinkCapsEntry
       PIB-ACCESS     notify
       STATUS         current
       DESCRIPTION
           "This table specifies what types of datapath functional
           elements may be used as the next downstream element for
           a specific type of functional element."
       ::= { qosCapabilityClasses 9 }


   qosIfElmLinkCapsEntry OBJECT-TYPE
       SYNTAX         QosIfElmLinkCapsEntry
       STATUS         current
       DESCRIPTION
           "An instance of this class specifies a PRC that may
            be used as the next functional element after a specific
            type of element in a data path."
       EXTENDS { qosBaseIfCapsEntry }
```

```
UNIQUENESS { qosBaseIfCapsDirection,
             qosIfElmLinkCapsPrc,
             qosIfElmLinkCapsAttr,
             qosIfElmLinkCapsNextPrc }
```

```
        ::= { qosIfElmLinkCapsTable 1 }


   QosIfElmLinkCapsEntry ::= SEQUENCE {
           qosIfElmLinkCapsPrc               PrcIdentifier,
           qosIfElmLinkCapsAttr              AttrIdentifier,
           qosIfElmLinkCapsNextPrc           PrcIdentifier
   }







   qosIfElmLinkCapsPrc OBJECT-TYPE
       SYNTAX         PrcIdentifier
       STATUS         current
       DESCRIPTION
         "The value is the OID of a PRC that represents a
         functional datapath element. This OID must not have
         the value zeroDotZero."
       ::= { qosIfElmLinkCapsEntry 1 }


   qosIfElmLinkCapsAttr OBJECT-TYPE
       SYNTAX         AttrIdentifier
       STATUS         current
       DESCRIPTION
         "The value represents the attribute in the PRC
         indicated by qosIfElmLinkCapsPrc that is used to
         specify the next functional element in the datapath.
         The attribute value corresponds to the order in which
         the attribute appears in the definition of the PRC.
         A value of 1 indicates the first attribute of the PRC,
         a value of 2 indicates the second attribute of the
         PRC, and so forth."
       ::= { qosIfElmLinkCapsEntry 2 }


   qosIfElmLinkCapsNextPrc OBJECT-TYPE
       SYNTAX         PrcIdentifier
       STATUS         current
       DESCRIPTION
         "The value is the OID of a PRC table entry from which
         instances can be referenced by the attribute indicated
         by qosIfElmLinkCapsPrc and qosIfElmLinkAttr.

         For example, suppose a meter's success output can be an
         action or another meter, and the fail output can only be
         an action.  This can be expressed as follows:
```

Prid	Prc	Attr	NextPrc
1	qosMeterEntry	qosMeterSucceedNext	qosActionEntry
2	qosMeterEntry	qosMeterSucceedNext	qosMeterEntry

```
         3    qosMeterEntry   qosMeterFailNext      qosActionEntry.

         zeroDotZero is a valid value for this attribute to
         specify that the PRC specified in qosIfElmLinkCapsPrc
         is the last functional data path element."
       ::= { qosIfElmLinkCapsEntry 3 }
```

```
   --
   -- Policy Classes
   --


   --
   -- Data Path Table
   --
   -- The Data Path Table enumerates the Differentiated Services
   -- Functional Data Paths within this device.  Each entry specifies
   -- the first functional datapath element to process data flow
   -- for each specific datapath.  Each datapath is defined by the
   -- interface role combination and direction. This table can
   -- therefore have up to two entries for each role combination,
   -- ingress and egress.


   qosDataPathTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosDataPathEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The data path table indicates the start of
          functional data paths in this device."

       ::= { qosPolicyClasses 1 }


   qosDataPathEntry OBJECT-TYPE
       SYNTAX       QosDataPathEntry
       STATUS       current
       DESCRIPTION
          "Each entry in this table indicates the start of a single
          functional data path, defined by its interface name,
          role combination and traffic direction.  The first
          functional datapath element to handle traffic for each
          data path is defined by the qosDataPathStart attribute
          of each table entry.
          Notice for each entry:
          1. qosDataPathIfName must reference an existing interface
             capability name in frwkIfCapSetTable [FR-PIB].
          2. qosDataPathRoles must reference existing Role Combination
             in frwkIfRoleComboTable [FR-PIB].
          3. qosDataPathStart must reference an existing entry in a
             functional data path element table.
          If any one or more of these three requirements is not
          satisfied, the qosDataPathEntry will not be installed."
       PIB-INDEX { qosDataPathPrid }
```

```
UNIQUENESS { qosDataPathIfName,
             qosDataPathRoles,
             qosDataPathIfDirection }
::= { qosDataPathTable 1 }
```

```
   QosDataPathEntry ::= SEQUENCE  {
       qosDataPathPrid           InstanceId,
       qosDataPathIfName         SnmpAdminString,
       qosDataPathRoles          RoleCombination,
       qosDataPathIfDirection    IfDirection,
       qosDataPathStart          Prid
   }


   qosDataPathPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosDataPathEntry 1 }


   qosDataPathIfName OBJECT-TYPE
       SYNTAX       SnmpAdminString
       STATUS       current
       DESCRIPTION
          "The interface capability set to which this data path entry
           applies.  The interface capability name specified by this
           attribute must exist in the frwkIfCapSetTable [FR-PIB] prior
           to association with an instance of this class."
       ::= { qosDataPathEntry 2 }


   qosDataPathRoles OBJECT-TYPE
       SYNTAX       RoleCombination
       STATUS       current
       DESCRIPTION
          "The interfaces to which this data path entry applies,
           specified in terms of roles.  There must exist an entry
           in the frwkIfRoleComboTable [FR-PIB] specifying
           this role combination, together with the interface
           capability set specified by qosDataPathIfName, prior to
           association with an instance of this class."
       ::= { qosDataPathEntry 3 }


   qosDataPathIfDirection OBJECT-TYPE
       SYNTAX       IfDirection
       STATUS       current
       DESCRIPTION
          "Specifies the direction for  which  this  data  path
          entry applies on this interface."
       ::= { qosDataPathEntry 4 }
```

```
qosDataPathStart OBJECT-TYPE
    SYNTAX       Prid
```

```
       STATUS       current
       DESCRIPTION
          "This selects the first functional  datapath  element
          to   handle   traffic   for  this  data  path.   This
          Prid should point to an instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry
            qosQEntry

          The PRI pointed to must exist prior to the installation of
          this datapath start element."
       ::= { qosDataPathEntry 5 }
```

```
   --
   -- Classifiers
   --
   -- Classifier allows multiple classifier elements, of same or
   -- different types, to be used together.
   -- A classifier must completely classify all packets presented to
   -- it. This means all traffic handled by a classifier must match
   -- at least one classifier element within the classifier,
   -- with the classifier element parameters specified by a filter.
   -- It is the PDP's responsibility to create a _catch all_ classifier
   -- element and filter that matches all packet.  This _catch all_
   -- classifier element should have the lowest Precedence value.

   -- If there is ambiguity between classifier elements of different
   -- classifier, classifier linkage order indicates their precedence;
   -- the first classifier in the link is applied to the traffic first.

   -- Each entry in the classifier table represents a classifier, with
   -- classifier element table handling the fan-out functionality of a
   -- classifier, and filter table defining the classification
   -- patterns.
   --


   --
   -- Classifier Table
   --
   -- The Classifier Table enumerates the Diffserv classifiers in this
   -- device.  Each classifier is referenced by its classifier elements
   -- using its classifier ID.
   --


   qosClfrTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosClfrEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "This table enumerates all the Diffserv classifier functional
          data path elements of this device.  The actual classification
          definitions are detailed in qosClfrElementTable entries
          belonging to each classifer.

          An entry in this table, referenced by an upstream functional
          data path element or a datapath table entry, is the entry
          point to the classifier functional data path element.

          The qosClfrId of each entry is used to organize all
          classifier elements belonging to the same classifier."
```

```
REFERENCE
    "[MODEL] section 4.1"
::= { qosPolicyClasses 2 }
```

```
   qosClfrEntry OBJECT-TYPE
       SYNTAX       QosClfrEntry
       STATUS       current
       DESCRIPTION
          "An entry in the classifier table describes a single
          classifier. Each classifier element belonging to this
          classifier must have its qosClfrElementClfrId attribute equal
          to qosClfrId."
       PIB-INDEX { qosClfrPrid }
       UNIQUENESS { qosClfrId }
       ::= { qosClfrTable 1 }


   QosClfrEntry ::= SEQUENCE  {
       qosClfrPrid            InstanceId,
       qosClfrId              TagReferenceId
   }


   qosClfrPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosClfrEntry 1 }


   qosClfrId OBJECT-TYPE
       SYNTAX       TagReferenceId
       PIB-TAG      { qosClfrElementClfrId }
       STATUS       current
       DESCRIPTION
          "Identifies a Classifier.  A  Classifier must be
          complete, this means all traffic handled by a
          Classifier must match at least  one  Classifier
          Element within  the  Classifier."
       ::= { qosClfrEntry 2 }


   --
   -- Classifier Element Table
   --
   -- Entries in the classifier element table serves as
   -- the anchor for each classification pattern, defined
   -- in filter table entries.  Each classifier element
   -- table entry also specifies the subsequent downstream
   -- diffserv functional datapath element when the
   -- classification pattern is satisfied.
```

```
-- Each entry in the classifier element table describes
-- one branch of the fan-out characteristic of a classifier
-- indicated in [MODEL] section 4.1.  A classifier is made up
-- of one or more classifier elements.
```

```
   --
   -- If there is ambiguity between classifier elements of the same
   -- classifier, then qosClfrElementPrecedence needs to be used.
   --


   qosClfrElementTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosClfrElementEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The classifier element table enumerates the relationship
          between classification patterns and subsequent downstream
          diffserv  functional data path elements. Classification
          parameters are defined  by entries of filter tables pointed
          to by qosClfrElementSpecific.   There can be filter tables of
          different types, and they can be inter-mixed and used within
          a classifier. An example of a filter table is the
          frwkIpFilterTable, defined in [FR-PIB], for IP Multi-Field
          Classifiers (MFCs)."
       REFERENCE
           "[MODEL] section 4.1"
       ::= { qosPolicyClasses 3 }


   qosClfrElementEntry OBJECT-TYPE
       SYNTAX       QosClfrElementEntry
       STATUS       current
       DESCRIPTION
          "An entry in the classifier element table describes a
          single element of the classifier."
       PIB-INDEX { qosClfrElementPrid }
       UNIQUENESS { qosClfrElementClfrId,
                    qosClfrElementPrecedence,
                    qosClfrElementSpecific }
       ::= { qosClfrElementTable 1 }


   QosClfrElementEntry ::= SEQUENCE  {
       qosClfrElementPrid        InstanceId,
       qosClfrElementClfrId      TagId,
       qosClfrElementPrecedence  Unsigned32,
       qosClfrElementNext        Prid,
       qosClfrElementSpecific    Prid
   }


   qosClfrElementPrid OBJECT-TYPE
       SYNTAX       InstanceId
```

```
STATUS       current
DESCRIPTION
   "An arbitrary integer index that uniquely identifies an
    instance of the class."
::= { qosClfrElementEntry 1 }
```

```
   qosClfrElementClfrId OBJECT-TYPE
       SYNTAX       TagId
       STATUS       current
       DESCRIPTION
          "A classifier is composed of one or more classifier
           elements. Each classifier element belonging to
           the same classifier uses the same classifier ID.

           Hence, A classifier Id identifies which classifier
           this classifier element is a part of. This needs to be
           the value of qosClfrId attribute for an existing
           instance of qosClfrEntry."
       ::= { qosClfrElementEntry 2 }


   qosClfrElementPrecedence OBJECT-TYPE
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
          "The relative order in which classifier elements are
          applied:  higher  numbers  represent classifier elements
          with higher precedence.  Classifier elements with the same

          precedence  must  be  unambiguous  i.e. they must define
          non-overlapping patterns, and are  considered  to  be
          applied  simultaneously  to the traffic stream. Clas-
          sifier elements with different precedence may overlap
          in their filters: the classifier element with the highest
          precedence that matches is taken.

          On a given interface, there must be a complete  clas-
          sifier  in  place  at  all  times in   the
          ingress direction.  This means that there will always
          be one or more filters that match every possible pat-
          tern  that  could be presented in an incoming packet.
          There is no such requirement in the egress direction."
       DEFVAL { 0 }
       ::= { qosClfrElementEntry 3 }


   qosClfrElementNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
```

```
DESCRIPTION
   "This attribute provides one branch  of  the  fan-out
   functionality  of  a  classifier described in [MODEL]
   section 4.1.
```

```
          This selects the next  diffserv  functional  datapath
          element  to  handle traffic for this data path.

          A value of zeroDotZero marks the end of DiffServ processing
          for this data path.  Any other value must point to a
          valid (pre-existing) instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry
            qosQEntry."
       DEFVAL      { zeroDotZero }
       ::= { qosClfrElementEntry 4 }


   qosClfrElementSpecific OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "A pointer to a valid entry  in  another  table  that
          describes  the applicable classification filter, e.g.
          an entry in frwkIpFilterTable [FR-PIB].

          The PRI pointed to must exist prior to the installation of
          this classifier element.

          The value zeroDotZero is interpreted  to  match  any-
          thing  not  matched  by another classifier element - only one
          such entry may exist for each classifier."
       DEFVAL { zeroDotZero }
       ::= { qosClfrElementEntry 5 }
```

```
   --
   -- Meters
   --
   -- This PIB supports a variety of Meters.  It includes a
   -- specific definition for Meters whose parameter set can
   -- be modelled using Token Bucket parameters.
   -- Other metering parameter sets can be defined by other PIBs.
   --
   -- Multiple meter elements may be logically cascaded
   -- using their qosMeterSucceedNext and qosMeterFailNext pointers if
   -- required.
   -- One example of this might be for an AF PHB implementation
   -- that uses multiple level conformance meters.
   --
   -- Cascading of individual meter elements in the PIB is intended
   -- to be functionally equivalent to multiple level conformance--
   determination of a packet.  The sequential
   -- nature of the representation is merely a notational
   -- convenience for this PIB.
   --
   -- srTCM meters (RFC 2697) can be specified using two sets of
   -- qosMeterEntry and qosTBParamEntry. First set specifies the
   -- Committed Information Rate and Committed Burst Size
   -- token-bucket.  Second set specifies the Excess Burst
   -- Size token-bucket.
   --
   -- trTCM meters (RFC 2698) can be specified using two sets of
   -- qosMeterEntry and qosTBParamEntry. First set specifies the
   -- Committed Information Rate and Committed Burst Size
   -- token-bucket.  Second set specifies the Peak Information
   -- Rate and Peak Burst Size token-bucket.
   --
   -- tswTCM meters (RFC 2859) can be specified using two sets of
   -- qosMeterEntry and qosTBParamEntry. First set specifies the
   -- Committed Target Rate token-bucket. Second set specifies the
   -- Peak Target Rate token-bucket. qosTBParamInterval in each
   -- token bucket reflects the Average Interval.
   --


   qosMeterTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosMeterEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "This table enumerates specific meters that a  system
          may  use  to  police a stream of traffic. The traffic
          stream to be metered is determined by the  element(s)
```

upstream of the meter i.e. by the object(s) that point to each entry in this table. This may include all traffic on an interface.

Specific meter details are to be found in table entry

```
          referenced by qosMeterSpecific."
      REFERENCE  "[MODEL] section 5.1"
       ::= { qosPolicyClasses 4 }


   qosMeterEntry OBJECT-TYPE
       SYNTAX       QosMeterEntry
       STATUS       current
       DESCRIPTION
          "An entry in the  meter  table  describes  a  single
          conformance level of a meter."
       PIB-INDEX { qosMeterPrid }
       UNIQUENESS { qosMeterSucceedNext,
                    qosMeterFailNext,
                    qosMeterSpecific }
       ::= { qosMeterTable 1 }


   QosMeterEntry ::= SEQUENCE  {
       qosMeterPrid              InstanceId,
       qosMeterSucceedNext       Prid,
       qosMeterFailNext          Prid,
       qosMeterSpecific          Prid
   }


   qosMeterPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosMeterEntry 1 }


   qosMeterSucceedNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "If the traffic does conform, this selects  the  next
          diffserv   functional   datapath  element  to  handle
          traffic for this data path.

          The value zeroDotZero in this variable  indicates  no
          further Diffserv treatment is performed on traffic of
          this datapath.  Any other value must point to a valid
          (pre-existing) instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
```

```
     qosAlgDropEntry
     qosQEntry."
DEFVAL      { zeroDotZero }
::= { qosMeterEntry 2 }
```

```
   qosMeterFailNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "If the traffic does not conform,  this  selects  the
          next  diffserv  functional datapath element to handle
          traffic for this data path.

          The value zeroDotZero in this variable  indicates  no
          further Diffserv treatment is performed on traffic of
          this datapath.  Any other value must point to a valid
          (pre-existing) instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry
            qosQEntry."
       DEFVAL      { zeroDotZero }
       ::= { qosMeterEntry 3 }


   qosMeterSpecific OBJECT-TYPE
       SYNTAX       Prid
        STATUS       current
       DESCRIPTION
          "This indicates the behaviour of the meter by  point-
          ing  to an entry containing detailed parameters. Note
          that entries in that specific table must  be  managed
          explicitly.

          For example, qosMeterSpecific may  point  to  an
          entry  in  qosTBMeterTable,  which  contains  an
          instance of a single set of Token Bucket parameters.

          The PRI pointed to must exist prior to installing this
          Meter datapath element."
       ::= { qosMeterEntry 4 }


   --
   -- Token-Bucket Parameter Table
   --
   -- Each entry in the Token Bucket Parameter Table parameterizes
   -- a single token bucket.  Multiple token buckets can be
   -- used together to parameterize multiple levels of
   -- conformance.
   --
   -- Note that an entry in the Token Bucket Parameter Table can
```

```
-- be shared, pointed to, by multiple qosMeterTable entries.
--
```

```
   qosTBParamTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosTBParamEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "This table enumerates token-bucket meter parameter sets
          that  a system may use to police a stream of traffic.
          Such parameter sets are modelled here as each having a single
   rate
          and a single burst size.  Multiple entries are used
          when multiple rates/burst sizes are needed."
       REFERENCE
           "[MODEL] section 5.1"
       ::= { qosPolicyClasses 5 }


   qosTBParamEntry OBJECT-TYPE
       SYNTAX       QosTBParamEntry
       STATUS       current
       DESCRIPTION
          "An entry that describes a single token-bucket
          parameter set."
       PIB-INDEX { qosTBParamPrid }
       UNIQUENESS { qosTBParamType,
                    qosTBParamRate,
                    qosTBParamBurstSize,
                    qosTBParamInterval }
       ::= { qosTBParamTable 1 }


   QosTBParamEntry ::= SEQUENCE  {
       qosTBParamPrid            InstanceId,
       qosTBParamType            OBJECT IDENTIFIER,
       qosTBParamRate            Unsigned32,
       qosTBParamBurstSize       BurstSize,
       qosTBParamInterval        Unsigned32
   }


   qosTBParamPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosTBParamEntry 1 }


   qosTBParamType OBJECT-TYPE
```

```
SYNTAX       OBJECT IDENTIFIER
STATUS       current
DESCRIPTION
  "The Metering algorithm associated with the
  Token-Bucket parameters.  zeroDotZero indicates this
```

```
         is unknown.
         Standard values for generic algorithms are as follows:

         qosTBParamSimpleTokenBucket, qosTBParamAvgRate,
         qosTBParamSrTCMBlind, qosTBParamSrTCMAware,
         qosTBParamTrTCMBlind, qosTBParamTrTCMAware,
         qosTBParamTswTCM

         These are specified in this PIB as OBJECT-IDENTITYs
         under qosPolicyParameters; additional values may be
         further specified in other PIBs."
       REFERENCE
           "[MODEL] section 5"
       ::= { qosTBParamEntry 2 }


   qosTBParamRate OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "kilobits per second"
       STATUS       current
       DESCRIPTION
          "The  token-bucket  rate,  in  kilobits  per   second
          (kbps).   This  attribute  is used for:
       1. CIR in RFC 2697 for srTCM
          2. CIR and PIR in RFC 2698 for trTCM
          3. CTR and PTR in RFC 2859 for TSWTCM
       4. AverageRate  in [MODEL] section 5."
       ::= { qosTBParamEntry 3 }


   qosTBParamBurstSize OBJECT-TYPE
       SYNTAX       BurstSize
       UNITS        "Bytes"
       STATUS       current
       DESCRIPTION
          "The maximum number of bytes in a single transmission
          burst.  This attribute is used for:
       1. CBS and EBS in RFC 2697 for srTCM
          2. CBS and PBS in FRC 2698 for trTCM
          3. Burst Size in [MODEL] section 5."
       ::= { qosTBParamEntry 4 }


   qosTBParamInterval OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "microseconds"
       STATUS       current
       DESCRIPTION
          "The time interval used with the token bucket.   For:
```

1. Average Rate Meter, [MODEL] section 5.2.1,
   -Delta.
2. Simple Token Bucket Meter, [MODEL] section
   5.1, - time interval t.
3. RFC 2859 TSWTCM, - AVG_INTERVAL.

```
              4. RFC 2697 srTCM, RFC 2698 trTCM, -
                 token bucket update time interval."
           ::= { qosTBParamEntry 5 }
```

```
   --
   -- Actions
   --


   --
   -- The Action Table allows enumeration of the different
   -- types of actions to be applied to a traffic flow.
   --


   qosActionTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosActionEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The Action Table enumerates actions that can be per-
          formed  to  a stream of traffic. Multiple actions can
          be concatenated.  For example, after marking a stream
          of  traffic  exiting  from a meter, a device can then
          perform a mark  action  of  the  conforming  or  non-
          conforming traffic.

          Specific actions  are  indicated  by  qosAction-
          Specific  which  points  to  an  entry  of a specific
          action type parameterizing the action in detail."
       REFERENCE
           "[MODEL] section 6."
       ::= { qosPolicyClasses 6 }


   qosActionEntry OBJECT-TYPE
       SYNTAX       QosActionEntry
       STATUS       current
       DESCRIPTION
          "Each entry in the action table allows description of
          one specific action to be applied to traffic."
       PIB-INDEX { qosActionPrid }
       UNIQUENESS { qosActionNext,
                    QosActionSpecific }
       ::= { qosActionTable 1 }


   QosActionEntry ::= SEQUENCE  {
       qosActionPrid             InstanceId,
       qosActionNext             Prid,
       qosActionSpecific         Prid
   }
```

```
qosActionPrid OBJECT-TYPE
    SYNTAX       InstanceId
    STATUS       current
```

```
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosActionEntry 1 }


   qosActionNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This selects the next diffserv  functional  datapath
          element  to  handle traffic for this data path.

          The value zeroDotZero in this variable  indicates  no
          further Diffserv treatment is performed on traffic of
          this datapath.  Any other value must point to a valid
          (pre-existing) instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry
            qosQEntry."
       DEFVAL      { zeroDotZero }
       ::= { qosActionEntry 2 }


   qosActionSpecific OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "A pointer to an object instance providing additional
          information  for the type of action indicated by this
          action table entry.

          For the standard actions defined by this PIB  module,
          this should  point to an instance of qosDscpMarkActEntry.
          For other actions, it may point to an instance of a
          PRC defined in some other PIB.

          The PRI pointed to must exist prior to installing this
          action datapath entry."
       ::= { qosActionEntry 3 }


   -- DSCP Mark Action Table
   --
   -- Rows of this table are pointed to by qosActionSpecific
   -- to provide detailed parameters specific to the DSCP
   -- Mark action.
   -- This table should at most contain one entry for each supported
```

```
-- DSCP value.  These entries should be reused by different
-- qosActionEntry in same or different data paths.
--
```

```
   qosDscpMarkActTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosDscpMarkActEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "This table enumerates specific DSCPs used for marking or
          remarking  the DSCP field of IP packets. The entries of this
          table may be referenced by a qosActionSpecific attribute."
       REFERENCE
           "[MODEL] section 6.1"
       ::= { qosPolicyClasses 7 }


   qosDscpMarkActEntry OBJECT-TYPE
       SYNTAX       QosDscpMarkActEntry
       STATUS       current
       DESCRIPTION
         "An entry in the DSCP mark action table that describes a
         single DSCP used for marking."
       PIB-INDEX { qosDscpMarkActPrid }
       UNIQUENESS { qosDscpMarkActDscp }
       ::= { qosDscpMarkActTable 1 }


   QosDscpMarkActEntry ::= SEQUENCE  {
       qosDscpMarkActPrid          InstanceId,
       qosDscpMarkActDscp          Dscp
   }


   qosDscpMarkActPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosDscpMarkActEntry 1 }


   qosDscpMarkActDscp OBJECT-TYPE
       SYNTAX       Dscp
       STATUS       current
       DESCRIPTION
          "The DSCP that this Action uses for marking/remarking
          traffic.  Note that a DSCP value of -1 is not permit-
          ted in this table.  It is  quite  possible  that  the
          only  packets  subject  to  this  Action  are already
          marked with this DSCP. Note also  that  Diffserv  may
          result  in packet remarking both on ingress to a net-
   work and on egress from it and it is  quite  possible
   that  ingress  and  egress  would  occur  in the same
   router."
::= { qosDscpMarkActEntry 2 }
```

```
   --
   -- Algorithmic Drop Table
   --

   -- Algorithmic Drop Table is the entry point for the Algorithmic
   -- Dropper functional data path element.

   -- For a simple algorithmic dropper, a single algorithmic drop entry
   -- will be sufficient to parameterize the dropper.

   -- For more complex algorithmic dropper, the qosAlgDropSpecific
   -- attribute can be used to reference an entry in a parameter table,
   -- e.g. qosRandomDropTable for random dropper.

   -- For yet more complex dropper, for example, dropper that measures
   -- multiple queues, each queue with its own algorithm, can use a
   -- qosAlgDropTable entry as the entry point for Algorithm Dropper
   -- functional data path element, leaving the dropper parameters
   -- for each queue be specified by entries of qosMQAlgDropTable.
   -- In such usage, the anchoring qosAlgDropEntry's qosAlgDropType
   -- should be mQDrop, and its qosAlgDropQMeasure should reference
   -- the subsequent qosMQAlgDropEntry's, its qosAlgDropSpecific
   -- should be used to reference parameters applicable to all the
   -- queues being measured.
   -- The subsequent qosMQAlgDropEntry's will provide the parameters,
   -- one for each queue being measured.  The qosMQAlgDropEntry's are
   -- chained using their qosMQAlgDropNext attributes.
   --


   qosAlgDropTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosAlgDropEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The algorithmic drop table contains entries describ-
          ing  a functional data path element that drops
          packets according to some algorithm."
       REFERENCE
           "[MODEL] section 7.1.3"
       ::= { qosPolicyClasses 9 }


   qosAlgDropEntry OBJECT-TYPE
       SYNTAX       QosAlgDropEntry
       STATUS       current
       DESCRIPTION
          "An entry describes  a  process  that  drops  packets
```

```
according  to some algorithm.  Further details of the
algorithm type are to be found in qosAlgDropType
and  with  more  detail parameter entry pointed to by
qosAlgDropSpecific when necessary."
```

```
       PIB-INDEX { qosAlgDropPrid }
       UNIQUENESS { qosAlgDropType,
                    qosAlgDropNext,
                    qosAlgDropQMeasure,
                    qosAlgDropQThreshold,
                    qosAlgDropSpecific }
       ::= { qosAlgDropTable 1 }


   QosAlgDropEntry ::= SEQUENCE  {
       qosAlgDropPrid             InstanceId,
       qosAlgDropType             INTEGER,
       qosAlgDropNext             Prid,
       qosAlgDropQMeasure         Prid,
       qosAlgDropQThreshold       Unsigned32,
       qosAlgDropSpecific         Prid
   }


   qosAlgDropPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosAlgDropEntry 1 }


   qosAlgDropType OBJECT-TYPE
       SYNTAX       INTEGER {
                        other(1),
                        tailDrop(2),
                        headDrop(3),
                        randomDrop(4),
                        alwaysDrop(5),
                        mQDrop(6)
                    }
       STATUS       current
       DESCRIPTION
          "The type of algorithm used by this dropper. A value
          of tailDrop(2), headDrop(3), or alwaysDrop(5) represents
   an algorithm that is completely specified by this PIB.

          A value of other(1) indicates that the specifics of
          the drop algorithm are specified in some other PIB
          module, and that the qosAlgDropSpecific attribute
          points to an instance of a PRC in that PIB that
          specifies the information necessary to implement the
          algorithm.
```

The tailDrop(2) algorithm is described as follows: qosAlgDropQThreshold represents the depth of the queue, pointed to by qosAlgDropQMeasure, at which all newly arriving packets will be dropped.

```
          The headDrop(3) algorithm is described as follows: if
          a packet arrives when the current depth of the queue,
          pointed to by qosAlgDropQMeasure, is at
          qosAlgDropQThreshold,  packets currently at the head of
          the queue are dropped to make room for the new packet
          to be enqueued at the tail of the queue.

          The randomDrop(4) algorithm is described as  follows:
          on packet arrival, an algorithm is executed which may
          randomly drop the packet,  or  drop  other  packet(s)
          from  the  queue  in  its place. The specifics of the
          algorithm may be  proprietary.  For  this  algorithm,
          qosAlgDropSpecific  points  to a qosRandomDropEntry
          that describes  the  algorithm.   For  this
          algorithm,  qosAlgQThreshold is understood to be
          the absolute maximum size of the queue and additional
          parameters are described in qosRandomDropTable.

          The alwaysDrop(5) algorithm always drops packets. In
          this case, the other configuration values in this Entry
          are not meaningful; The queue is not used, therefore,
          qosAlgDropNext, qosAlgDropQMeasure, and
          qosAlgDropSpecific should be all set to zeroDotZero.

          The mQDrop(6) algorithm measures multiple queues for
          the drop algorithm.  The queues measured are represented
          by having qosAlgDropQMeasure referencing a qosMQAlgDropEntry.
          Each of the chained qosMQAlgDropEntry is used to describe
          the drop algorithm for one of the measured queues."

       ::= { qosAlgDropEntry 2 }



   qosAlgDropNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This selects the next diffserv functional datapath
          element to  handle traffic for this data path.

          The value zeroDotZero in this attribute indicates  no
          further Diffserv treatment is performed on traffic of
          this datapath.  Any other value must point to a valid
          (pre-existing) instance of one of:
            qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry
```

```
  qosQEntry.

When qosAlgDropType is alwaysDrop(5), this attribute is
Ignored."
```

```
       DEFVAL      { zeroDotZero }
       ::= { qosAlgDropEntry 3 }


   qosAlgDropQMeasure OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "Points to a PRI to indicate the queues that a drop algorithm
          is to monitor when deciding whether to drop a packet.

          For alwaysDrop(5), this attribute should be zeroDotZero.
          For tailDrop(2), headDrop(3), randomDrop(4), this should
          point to an entry in the qosQTable.
          For mQDrop(6), this should point to a qosMQAlgDropEntry that
          Describe one of the queues being measured for multiple
          queue dropper.

          The PRI pointed to must exist prior to installing
          this dropper element."
       ::= { qosAlgDropEntry 4 }


   qosAlgDropQThreshold OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "Bytes"
       STATUS       current
       DESCRIPTION
          "A threshold on the depth in bytes of the queue being
          measured at which a trigger is generated to the drop-
          ping algorithm, unless qosAlgDropType is alwaysDrop(5)
          where this attribute is ignored.

          For the tailDrop(2) or headDrop(3)  algorithms,  this
          represents  the  depth  of  the  queue, pointed to by
          qosAlgDropQMeasure, at  which  the  drop  action
          will take place. Other algorithms will need to define
          their own semantics for this threshold."
       ::= { qosAlgDropEntry 5 }


   qosAlgDropSpecific OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "Points to a table entry that provides further detail
          regarding a drop algorithm.  The PRI pointed to
          must exist prior to installing this dropper element.

          Entries with qosAlgDropType equal to other(1)
```

must have this point to an instance of a PRC
defined in another PIB module.

Entries with qosAlgDropType equal to random-

```
          Drop(4)   must   have  this  point  to  an  entry  in
          qosRandomDropTable.

          Entries with qosAlgDropType equal to mQDrop(6) can use this
          attribute to reference parameters that is used by all the
          queues of the multiple queues being measured.

          For all other algorithms, this should take the  value
          zeroDotZero."
       ::= { qosAlgDropEntry 6 }




   --
   -- Multiple Queue Algorithmic Drop Table
   --
   -- Entries of this table should be referenced by qosAlgDropQMeasure
   -- when qosAlgDropType is mQDrop(6) for droppers measuring multiple
   -- queues for its drop algorithm.
   -- Each entry of the table is used to describe the drop algorithm
   -- for a single queue within the multiple queues being measured.
   --
   -- Entries of this table, qosMQAlgDropEntry, is extended from
   -- qosAlgDropEntry, with usage of corresponding parameters the same
   -- except:
   --   qosMQAlgDropNext is used to point to the next diffserv
   --     functional data path element when the packet is not dropped.
   --   qosMQAlgDropExceedNext is used to point to the next
   --     qosMQAlgDropEntry for chaining together the multiple
   --     qosMQAlgDropEntry's for the multiple queues being measured.
   --


   qosMQAlgDropTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosMQAlgDropEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The multiple queue algorithmic drop table contains entries
          describing each queue being measured for the multiple queue
          algorithmic dropper."
       ::= { qosPolicyClasses 10 }


   qosMQAlgDropEntry OBJECT-TYPE
       SYNTAX       QosMQAlgDropEntry
       STATUS       current
       DESCRIPTION
          "An entry describes a process that drops packets
          according to some algorithm.  Each entry is used for
```

each of the multiple queues being measured. Each entry
extends the basic qosAlgDropEntry with adding of a
qosMQAlgDropExceedNext attribute.
Further details of the algorithm type are to be found in

```
          qosAlgDropType and with more detail parameter entry pointed
          to by qosMQAlgDropSpecific when necessary."
       EXTENDS { qosAlgDropEntry }
       UNIQUENESS { qosMQAlgDropExceedNext }
       ::= { qosMQAlgDropTable 1 }


   QosMQAlgDropEntry ::= SEQUENCE  {
       qosMQAlgDropExceedNext     Prid
   }


   qosMQAlgDropExceedNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "Used for linking of multiple qosMQAlgDropEntry for mQDrop.
          A value of zeroDotZero indicates this is the last of a
          chain of qosMQAlgDropEntry."
       DEFVAL      { zeroDotZero }
       ::= { qosMQAlgDropEntry 1 }



   --
   -- Random Drop Table
   --

   qosRandomDropTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosRandomDropEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The random drop table contains entries describing  a
          process  that drops packets randomly. Entries in this
          table is  intended  to  be  pointed  to  by
          qosAlgDropSpecific."
       REFERENCE
           "[MODEL] section 7.1.3"
       ::= { qosPolicyClasses 11 }


   qosRandomDropEntry OBJECT-TYPE
       SYNTAX       QosRandomDropEntry
       STATUS       current
       DESCRIPTION
          "An entry describes  a  process  that  drops  packets
          according to a random algorithm."
       PIB-INDEX { qosRandomDropPrid }
       UNIQUENESS { qosRandomDropMinThreshBytes,
                    qosRandomDropMinThreshPkts,
                    qosRandomDropMaxThreshBytes,
                    qosRandomDropMaxThreshPkts,
                    qosRandomDropProbMax,
                    qosRandomDropWeight,
                    qosRandomDropSamplingRate
                  }
           ::= { qosRandomDropTable 1 }


   QosRandomDropEntry ::= SEQUENCE  {
       qosRandomDropPrid             InstanceId,
       qosRandomDropMinThreshBytes   Unsigned32,
       qosRandomDropMinThreshPkts    Unsigned32,
       qosRandomDropMaxThreshBytes   Unsigned32,
       qosRandomDropMaxThreshPkts    Unsigned32,
       qosRandomDropProbMax          Unsigned32,
       qosRandomDropWeight           Unsigned32,
       qosRandomDropSamplingRate     Unsigned32
   }


   qosRandomDropPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
          "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosRandomDropEntry 1 }


   qosRandomDropMinThreshBytes OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "bytes"
       STATUS       current
       DESCRIPTION
          "The average queue depth in bytes, beyond which traffic has a
          non-zero probability of being dropped."
        ::= { qosRandomDropEntry 2 }


   qosRandomDropMinThreshPkts OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "packets"
       STATUS       current
       DESCRIPTION
         "The average queue depth in packets, beyond which traffic has
         a non-zero probability of being dropped."
       ::= { qosRandomDropEntry 3 }


   qosRandomDropMaxThreshBytes OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "bytes"
```

```
STATUS       current
DESCRIPTION
  "The average queue depth beyond which traffic has a
  probability indicated by qosRandomDropProbMax of being dropped
```

```
         or marked. Note that this differs from the physical queue
         limit, which is stored in qosAlgDropQThreshold."
       ::= { qosRandomDropEntry 4 }




   qosRandomDropMaxThreshPkts OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "packets"
       STATUS       current
       DESCRIPTION
         "The average queue depth beyond which traffic has a
         probability indicated by qosRandomDropProbMax of being dropped
         or marked. Note that this differs from the physical queue
         limit, which is stored in qosAlgDropQThreshold."
       ::= { qosRandomDropEntry 5 }


   qosRandomDropProbMax OBJECT-TYPE
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
         "The worst case random drop probability, expressed in drops
         per thousand packets.

         For example, if every packet may be dropped in the worst case
         (100%), this has the value 1000. Alternatively, if in the
         worst case one percent (1%) of traffic may be dropped, it has
         the value 10."
       ::= { qosRandomDropEntry 6 }


   qosRandomDropWeight OBJECT-TYPE
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
         "The weighting of past history in affecting the Exponentially
         Weighted Moving Average function which calculates the current
         average queue depth.  The equation uses
         qosRandomDropWeight/MaxValue as the coefficient for the new
         sample in the equation, and
         (MaxValue - qosRandomDropWeight)/MaxValue as the coefficient
         of the old value, where, MaxValue is determined via capability
         reported by the PEP.

         Implementations may further limit the values of
         qosRandomDropWeight via the capability tables."
       ::= { qosRandomDropEntry 7 }
```

```
qosRandomDropSamplingRate OBJECT-TYPE
    SYNTAX       Unsigned32
    STATUS       current
```

```
       DESCRIPTION
         "The number of times per second the queue is sampled for queue
         average calculation. A value of zero means the queue is
         sampled approximately each time a packet is enqueued (or
         dequeued)."
       ::= { qosRandomDropEntry 8 }
```

```
   --
   -- Queue Table
   --

   --
   -- An entry of qosQTable represents a FIFO queue diffserv
   -- functional data path element as described in [MODEL] section
   -- 7.1.1.
   -- Notice the specification of scheduling parameters for a queue
   -- as part of the input to a scheduler functional data path
   -- element as described in [MODEL] section 7.1.2.  This allows
   -- building of hierarchical queuing/scheduling.
   -- A queue therefore is parameterized by:
   -- 1. Which scheduler will service this queue, qosQNext.
   -- 2. How the scheduler will service this queue, with respect
   --    to all the other queues the same scheduler needs to service,
   --    qosQMinRate and qosQMaxRate.
   --
   -- Notice one or more upstream diffserv functional data path element
   -- may share, point to, a qosQTable entry as described in [MODEL] --
   section 7.1.1.
   --

   qosQTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosQEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
       "The Queue Table enumerates the queues."
       ::= { qosPolicyClasses 12 }


   qosQEntry OBJECT-TYPE
       SYNTAX       QosQEntry
       STATUS       current
       DESCRIPTION
          "An entry in the Queue Table describes a single queue
          as a functional data path element."
       PIB-INDEX { qosQPrid }
       UNIQUENESS { qosQNext,
                    qosQMinRate,
                    qosQMaxRate }
       ::= { qosQTable 1 }


   QosQEntry ::= SEQUENCE  {
       qosQPrid                    InstanceId,
       qosQNext                    Prid,
        qosQMinRate                     Prid,
        qosQMaxRate                     Prid
    }
```

```
   qosQPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
           "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosQEntry 1 }


   qosQNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This selects the next diffserv scheduler.  This must point
          to a qosSchedulerEntry.

          A value of zeroDotZero in this attribute indicates an
          incomplete qosQEntry instance.  In such a case, the entry
          has no operational effect, since it has no parameters to
          give it meaning."
       ::= { qosQEntry 2 }


   qosQMinRate OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This Prid indicates the entry in qosMinRateTable
          the scheduler, pointed to by qosQNext, should use to service
          this queue.
          If this value is zeroDotZero, then minimum rate and priority
       is unspecified.
          If this value is not zeroDotZero then the instance pointed to
must exist prior to installing
          this entry."
       ::= { qosQEntry 3 }

   qosQMaxRate OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This Prid indicates the entry in qosMaxRateTable
          the scheduler, pointed to by qosQNext, should use to service
          this queue.
          If this value is zeroDotZero, then the maximum rate is the
          line speed of the interface.
          If this value is not zeroDotZero
          then the instance pointed to must exist prior to installing
       this entry."
    ::= { qosQEntry 4 }
```

```
--
-- Scheduler Table
--
--
-- The Scheduler Table is used for representing packet schedulers:
-- it provides flexibility for multiple scheduling algorithms, each
-- servicing multiple queues, to be used on the same
-- logical/physical interface of a data path.
--
-- Notice the servicing parameters the scheduler uses is
-- specified by each of its upstream functional data path elements,
-- queues or schedulers of this PIB.
-- The coordination and coherency between the servicing parameters
-- of the scheduler's upstream functional data path elements must
-- be maintained for the scheduler to function correctly.
--
-- The qosSchedulerMinRate and qosSchedulerMaxRate attributes are
used for specifying
-- the servicing parameters for output of a scheduler when its
-- downstream functional data path element is another scheduler.
-- This is used for building hierarchical queue/scheduler.
--
-- More discussion of the scheduler functional data path element
-- is in [MODEL] section 7.1.2.
--


qosSchedulerTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF QosSchedulerEntry
    PIB-ACCESS   install
    STATUS       current
    DESCRIPTION
       "The Scheduler Table  enumerates  packet  schedulers.
       Multiple scheduling algorithms can be used on a given
       datapath,  with  each  algorithm  described  by  one
       qosSchedulerEntry."
    REFERENCE
        "[MODEL] section 7.1.2"
    ::= { qosPolicyClasses 13 }


qosSchedulerEntry OBJECT-TYPE
    SYNTAX       QosSchedulerEntry
    STATUS       current
    DESCRIPTION
       "An entry in the Scheduler Table describing a  single
       instance of a scheduling algorithm."
    PIB-INDEX { qosSchedulerPrid }
```

```
UNIQUENESS { qosSchedulerNext,
             qosSchedulerMethod,
             qosSchedulerMinRate,
             qosSchedulerMaxRate }
::= { qosSchedulerTable 1 }
```

```
   QosSchedulerEntry ::= SEQUENCE  {
       qosSchedulerPrid                 InstanceId,
       qosSchedulerNext                 Prid,
       qosSchedulerMethod               OBJECT IDENTIFIER,
       qosSchedulerMinRate              Prid,
       qosSchedulerMaxRate              Prid
   }


   qosSchedulerPrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
           "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosSchedulerEntry 1 }


   qosSchedulerNext OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
          "This selects the next diffserv  functional  datapath
          element  to  handle traffic for this data path.

          This attribute normally have a value of zeroDotZero       to
   indicate no further Diffserv treatment is performed on traffic of
          this datapath.  The use of zeroDotZero is the  normal
          usage for the last functional datapath element.  Any
          value other than zeroDotZero must point to a valid
          (pre-existing) instance of one of:
            qosSchedulerEntry
            qosQEntry,

          or:         qosClfrEntry
            qosMeterEntry
            qosActionEntry
            qosAlgDropEntry

          This points to another qosSchedulerEntry
          for implementation of multiple scheduler methods for
          the same  data path, and   for   implementation   of
          hierarchical schedulers."
       DEFVAL       { zeroDotZero }
       ::= { qosSchedulerEntry 2 }


   qosSchedulerMethod OBJECT-TYPE
```

```
SYNTAX       OBJECT IDENTIFIER
STATUS       current
DESCRIPTION
  "The scheduling algorithm used by this Scheduler.
```

```
      Standard values for generic algorithms:
           qosSchedulerPriority,
           qosSchedulerWRR,
           qosSchedulerWFQ
         are specified in this PIB.
         Additional values may be further specified in other PIBs.
         A value of zeroDotZero indicates this is unknown."
       REFERENCE
           "[MODEL] section 7.1.2"
       ::= { qosSchedulerEntry 3 }



   qosSchedulerMinRate OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
         "This Prid indicates the entry in qosMinRateTable
          which indicates the priority or minimum output rate from this
          scheduler.  This attribute is used only when there is more
          than one level of scheduler.

          When it has the value zeroDotZero, it indicates that no
          Minimum rate or priority is imposed."
       DEFVAL      { zeroDotZero }
       ::= { qosSchedulerEntry 4 }


   qosSchedulerMaxRate OBJECT-TYPE
       SYNTAX       Prid
       STATUS       current
       DESCRIPTION
         "This Prid indicates the entry in qosMaxRateTable
          which indicates the maximum output rate from this scheduler.
          When more than one maximum rate applies (e.g. a multi-rate
          shaper is used), it points to the first of the rate entries.
          This attribute is only used when there is more than one level
          of scheduler.

          When it has the value zeroDotZero, it indicates that no
          Maximum rate is imposed."
        DEFVAL      { zeroDotZero }
       ::= { qosSchedulerEntry 5 }
```

```
   --
   -- Minimum Rate Parameters Table
   --
   -- The parameters used by a scheduler for its inputs or outputs are
   -- maintained separately from the Queue or Scheduler table entries
   -- for reusability reasons and so that they may be used by both
   -- queues and schedulers.  This follows the approach for separation
   -- of data path elements from parameterization that is used
   -- throughout this MIB.
   -- Use of these Minimum Rate Parameter Table entries by Queues and
   -- Schedulers allows the modeling of hierarchical scheduling
   -- systems.
   --
   -- Specifically, a Scheduler has one or more inputs and one output.
   -- Any queue feeding a scheduler, or any scheduler which feeds a
   -- second scheduler, might specify a minimum transfer rate by
   -- pointing to a Minimum Rate Parameter Table entry.
   --
   -- The qosMinRatePriority/Abs/Rel attributes are used as
   -- parameters to the work-conserving portion of a scheduler:
   -- "work-conserving" implies that the scheduler can continue to emit
   -- data as long as there is data available at its input(s).  This
   -- has the effect of guaranteeing a certain priority relative to
   -- other scheduler inputs and/or a certain minimum proportion of the
   -- available output bandwidth. Properly configured, this means a
   -- certain minimum rate, which may be exceeded should traffic be
   -- available should there be spare bandwidth after all other classes
   -- have had opportunities to consume their own minimum rates.
   --

   qosMinRateTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosMinRateEntry
       PIB-ACCESS   install
       STATUS       current
       DESCRIPTION
          "The Minimum Rate Table enumerates individual
          sets  of scheduling parameter that can be used/reused
          by Queues and Schedulers."
       ::= { qosPolicyClasses 14 }


   qosMinRateEntry OBJECT-TYPE
       SYNTAX       QosMinRateEntry
       STATUS       current
       DESCRIPTION
          "An entry in the Minimum Rate Table describes
          a  single  set  of  scheduling  parameter  for use by
          queues and schedulers."
```

```
PIB-INDEX { qosMinRatePrid }
UNIQUENESS { qosMinRatePriority,
             qosMinRateAbsolute,
             qosMinRateRelative }
::= { qosMinRateTable 1 }
```

```
   QosMinRateEntry ::= SEQUENCE  {
       qosMinRatePrid            InstanceId,
       qosMinRatePriority        Unsigned32,
       qosMinRateAbsolute        Unsigned32,
       qosMinRateRelative        Unsigned32
   }


   qosMinRatePrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
           "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosMinRateEntry 1 }


   qosMinRatePriority OBJECT-TYPE
       SYNTAX       Unsigned32
        STATUS       current
       DESCRIPTION
         "The priority of this input to the associated scheduler,
         relative to the scheduler's other inputs. Higher Priority
         value indicates the associated queue/scheduler will get
         service first before others with lower Priority values."
       ::= { qosMinRateEntry 2 }


   qosMinRateAbsolute OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "kilobits per second"
       STATUS       current
       DESCRIPTION
         "The minimum absolute rate, in kilobits/sec, that a downstream
         scheduler element should allocate to this queue. If the value
         is zero, then there is effectively no minimum rate guarantee.
         If the value is non-zero, the scheduler will assure the
         servicing of this queue to at least this rate.

         Note that this attribute's value is coupled  to  that
         of  qosMinRateRelative:  changes to one will affect the value
         of the other.

         [IFMIB] defines ifSpeed as Gauge32 in units of bits per
         second, and ifHighSpeed as Gauge32 in units of 1,000,000 bits
         per second.
         This yields the following equations:
```

```
      RateRelative  = [ (RateAbsolute * 1000) / ifSpeed ] * 1,000
```

Where, 1000 is for converting kbps used by RateAbsolute to bps used

```
         by ifSpeed, 1,000 is for 'in units of 1/1,000 of 1' for
         RateRelative.

         or, if appropriate:

         RateRelative  =
            { [ (RateAbsolute * 1000) / 1,000,000 ] / ifHIghSpeed } *
   1,000

         Where, 1000 and 1,000,000 is for converting kbps used by
         RateAbsolute to 1 million bps used by ifHighSpeed, 1,000 is
   for
         'in units of 1/1,000 of 1' for RateRelative."
       REFERENCE
           "ifSpeed, ifHighSpeed from [IFMIB]"
       ::= { qosMinRateEntry 3 }


   qosMinRateRelative OBJECT-TYPE
       SYNTAX       Unsigned32
        STATUS       current
       DESCRIPTION
         "The minimum rate that a downstream scheduler element
         should  allocate  to this queue, relative to the max-
         imum rate of the interface as reported by ifSpeed  or
         ifHighSpeed, in units of 1/1,000 of 1.  If the value
         is zero, then there is effectively  no  minimum  rate
         guarantee.   If  the value is non-zero, the scheduler
         will assure the servicing of this queue to  at  least
         this rate.

         Note that this attribute's value is coupled  to  that
         of  qosMinRateAbsolute:  changes to one will
         affect the value of the other.

         [IFMIB] defines ifSpeed as Gauge32 in units of bits per
         second, and ifHighSpeed as Gauge32 in units of 1,000,000 bits
         per second.
         This yields the following equations:

         RateRelative  = [ (RateAbsolute * 1000) / ifSpeed ] * 1,000




         Where, 1000 is for converting kbps used by RateAbsolute to bps
         used by ifSpeed, 1,000 is for 'in units of 1/1,000 of 1' for
         RateRelative.
```

or, if appropriate:

RateRelative =

```
            { [ (RateAbsolute * 1000) / 1,000,000 ] / ifHIghSpeed } *
   1,000

         Where, 1000 and 1,000,000 is for converting kbps used by
         RateAbsolute to 1 million bps used by ifHighSpeed, 1,000 is
   for
         'in units of 1/1,000 of 1' for RateRelative."
       REFERENCE
           "ifSpeed, ifHighSpeed from [IFMIB]"
       ::= { qosMinRateEntry 4 }


   --
   -- Maximum Rate Parameters Table
   --
   -- The parameters used by a scheduler for its inputs or outputs are
   -- maintained separately from the Queue or Scheduler table entries
   -- for reusability reasons and so that they may be used by both
   -- queues and schedulers.  This follows the approach for separation
   -- of data path elements from parameterization that is used
   -- throughout this MIB.
   -- Use of these Maximum Rate Parameter Table entries by Queues and
   -- Schedulers allows the modeling of hierarchical scheduling
   -- systems.
   --
   -- Specifically, a Scheduler has one or more inputs and one output.
   -- Any queue feeding a scheduler, or any scheduler which feeds a
   -- second scheduler, might specify a maximum transfer rate by
   -- pointing to a Maximum Rate Parameter Table entry. Multi-rate
   -- shapers, such as a Dual Leaky Bucket algorithm, specify their
   -- rates using multiple Maximum Rate Parameter Entries with the same
   -- qosMaxRateId but different qosMaxRateLevels.
   --
   -- The qosMaxRateLevel/Abs/Rel attributes are used as
   -- parameters to the non-work-conserving portion of a scheduler:
   -- non-work-conserving implies that the scheduler may sometimes not
   -- emit a packet, even if there is data available at its input(s).
   -- This has the effect of limiting the servicing of the
   -- queue/scheduler input or output, in effect performing shaping of
   -- the packet stream passing through the queue/scheduler, as
   -- described in the Informal Differentiated Services Model
   -- section 7.2.
   --


   qosMaxRateTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF QosMaxRateEntry
       PIB-ACCESS   install
       STATUS       current
```

```
DESCRIPTION
   "The Maximum Rate Table enumerates individual
   sets  of scheduling parameter that can be used/reused
   by Queues and Schedulers."
::= { qosPolicyClasses 15 }
```

```
   qosMaxRateEntry OBJECT-TYPE
       SYNTAX       QosMaxRateEntry
       STATUS       current
       DESCRIPTION
          "An entry in the Maximum Rate Table describes
          a single  set  of  scheduling  parameter  for use by
          queues and schedulers."
       PIB-INDEX { qosMaxRatePrid }
       UNIQUENESS { qosMaxRateId,
                    qosMaxRateLevel,
                    qosMaxRateAbsolute,
                    qosMaxRateRelative,
                    qosMaxRateThreshold }
       ::= { qosMaxRateTable 1 }


   QosMaxRateEntry ::= SEQUENCE  {
       qosMaxRatePrid            InstanceId,
       qosMaxRateId                     Unsigned32,
       qosMaxRateLevel           Unsigned32,
       qosMaxRateAbsolute        Unsigned32,
       qosMaxRateRelative        Unsigned32,
       qosMaxRateThreshold       BurstSize
   }


   qosMaxRatePrid OBJECT-TYPE
       SYNTAX       InstanceId
       STATUS       current
       DESCRIPTION
           "An arbitrary integer index that uniquely identifies an
           instance of the class."
       ::= { qosMaxRateEntry 1 }


   qosMaxRateId OBJECT-TYPE
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
         "An index used together with qosMaxRateId for representing
         a multi-rate shaper.  This attribute is used for associating
         all the rate attributes of a multi-rate shaper.  Each
         qosMaxRateEntry of a multi-rate shaper must have the same
         value in this attribute.  The different rates of a multi-rate
         shaper is identified using qosMaxRateLevel.
         This attribute uses the value of zero to indicate this
         attribute is not used, for single rate shaper."
```

```
        DEFVAL { 0 }
        ::= { qosMaxRateEntry 2 }


    qosMaxRateLevel OBJECT-TYPE
```

```
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
         "An index that indicates which level of a multi-rate shaper is
         being given its parameters. A multi-rate shaper has some
         number of rate levels. Frame Relay's dual rate specification
         refers to a 'committed' and an 'excess' rate; ATM's dual rate
         specification refers to a 'mean' and a 'peak' rate. This table
         is generalized to support an arbitrary number of rates. The
         committed or mean rate is level 1, the peak rate (if any) is
         the highest level rate configured, and if there are other
         rates they are distributed in monotonically increasing order
         between them.
         When the entry is used for a single rate shaper, this
         attribute contains a value of zero."
       DEFVAL { 0 }
       ::= { qosMaxRateEntry 3 }


   qosMaxRateAbsolute OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "kilobits per second"
       STATUS       current
       DESCRIPTION
         "The maximum rate in kilobits/sec that  a  downstream
         scheduler  element  should allocate to this queue. If
         the value is zero, then there is effectively no  max-
         imum rate limit and that the scheduler should attempt
         to be work-conserving for this queue.  If  the  value
         is  non-zero,  the scheduler will limit the servicing
         of this queue to, at most, this rate in  a  non-work-
         conserving manner.

         Note that this attribute's value is coupled  to  that
         of  qosMaxRateRelative:  changes to one will
         affect the value of the other.

         [IFMIB] defines ifSpeed as Gauge32 in units of bits per
         second, and ifHighSpeed as Gauge32 in units of 1,000,000 bits
         per second.
         This yields the following equations:

         RateRelative  = [ (RateAbsolute * 1000) / ifSpeed ] * 1,000

         Where, 1000 is for converting kbps used by RateAbsolute to bps
         used by ifSpeed, 1,000 is for 'in units of 1/1,000 of 1'
         for RateRelative.

         or, if appropriate:
```

```
      RateRelative  =
         { [ (RateAbsolute * 1000) / 1,000,000 ] / ifHIghSpeed } *
1,000
```

```
         Where, 1000 and 1,000,000 is for converting kbps used by
         RateAbsolute to 1 million bps used by ifHighSpeed, 1,000 is
   for
         'in units of 1/1,000 of 1' for RateRelative."
       ::= { qosMaxRateEntry 4 }


   qosMaxRateRelative OBJECT-TYPE
       SYNTAX       Unsigned32
       STATUS       current
       DESCRIPTION
         "The maximum rate that a downstream scheduler element
         should  allocate  to this queue, relative to the max-
         imum rate of the interface as reported by ifSpeed  or
         ifHighSpeed, in units of 1/1,000 of 1.  If the value
         is zero, then there is effectively  no  maximum  rate
         limit  and  the  scheduler should attempt to be work-
         conserving for this queue.  If the value is non-zero,
         the  scheduler will limit the servicing of this queue
         to, at  most,  this  rate  in  a  non-work-conserving
         manner.

         Note that this attribute's value is coupled  to  that
         of  qosMaxRateAbsolute:  changes to one will
         affect the value of the other.

         [IFMIB] defines ifSpeed as Gauge32 in units of bits per
         second, and ifHighSpeed as Gauge32 in units of 1,000,000 bits
         per second.
         This yields the following equations:

         RateRelative  = [ (RateAbsolute * 1000) / ifSpeed ] * 1,000

         Where, 1000 is for converting kbps used by RateAbsolute to bps
   used
         by ifSpeed, 1,000 is for 'in units of 1/1,000 of 1' for
         RateRelative.

         or, if appropriate:

         RateRelative  =
            { [ (RateAbsolute * 1000) / 1,000,000 ] / ifHIghSpeed } *
   1,000

         Where, 1000 and 1,000,000 is for converting kbps used by
         RateAbsolute to 1 million bps used by ifHighSpeed, 1,000 is
   for
         'in units of 1/1,000 of 1' for RateRelative."
       REFERENCE
           "ifSpeed, ifHighSpeed from [IFMIB]"
```

```
        ::= { qosMaxRateEntry 5 }


qosMaxRateThreshold OBJECT-TYPE
```

```
        SYNTAX       BurstSize
        UNITS        "Bytes"
        STATUS       current
        DESCRIPTION
          "The number of bytes of queue depth at which the rate of a
          multi-rate scheduler will increase to the next output rate. In
          the last PRI for such a shaper, this threshold is
          ignored and by convention is zero."
        REFERENCE
            "Adaptive Rate Shaper, RFC 2963"
        ::= { qosMaxRateEntry 6 }
```

```
   --
   -- Parameters Section
   --

   -- The Parameters Section defines parameter objects that can be used
   -- for specific attributes defined in the PIB PRCs.

   qosTBParameters OBJECT IDENTIFIER ::= { qosPolicyParameters 1 }
   qosSchedulerParameters OBJECT IDENTIFIER
                                       ::= { qosPolicyParameters 2 }

   --
   -- Token Bucket Type Parameters
   --

   qosTBParamSimpleTokenBucket OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
          "This value indicates the use of a Two Parameter Token Bucket
          as described in [MODEL] section 5.2.3."
       REFERENCE
           "[MODEL] sections 5 and 7.1.2"
       ::= { qosTBParameters 1 }


   qosTBParamAvgRate OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
          "This value indicates the use of an Average Rate Meter as
          described in [MODEL] section 5.2.1."
       REFERENCE
           "[MODEL] sections 5 and 7.1.2"
       ::= { qosTBParameters 2 }


   qosTBParamSrTCMBlind OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
         "This value indicates the use of Single Rate Three Color
         Marker Metering as defined by RFC 2697, with `Color Blind'
         mode as described by the RFC."
       REFERENCE
           "[MODEL] sections 5 and 7.1.2"
       ::= { qosTBParameters 3 }


   qosTBParamSrTCMAware OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
```

```
  "This value indicates the use of Single Rate Three Color
  Marker Metering as defined by RFC 2697, with `Color Aware'
  mode as described by the RFC."
REFERENCE
```

```
             "[MODEL] sections 5 and 7.1.2"
        ::= { qosTBParameters 4 }


    qosTBParamTrTCMBlind OBJECT-IDENTITY
        STATUS       current
        DESCRIPTION
           "This value indicates the use of Two Rate Three Color Marker
           Metering as defined by RFC 2698, with `Color Blind' mode as
           described by the RFC."
        REFERENCE
             "[MODEL] sections 5 and 7.1.2"
        ::= { qosTBParameters 5 }


    qosTBParamTrTCMAware OBJECT-IDENTITY
        STATUS       current
        DESCRIPTION
           "This value indicates the use of Two Rate Three Color Marker
           Metering as defined by RFC 2698, with `Color Aware' mode as
           described by the RFC."
        REFERENCE
             "[MODEL] sections 5 and 7.1.2"
        ::= { qosTBParameters 6 }


    qosTBParamTswTCM OBJECT-IDENTITY
        STATUS       current
        DESCRIPTION
           "This value indicates the use of Time Sliding Window
           Three Color Marker Metering as defined by RFC 2859."
        REFERENCE
             "[MODEL] sections 5 and 7.1.2"
        ::= { qosTBParameters 7 }


    --
    -- Scheduler Method Parameters
    --

    qosSchedulerPriority OBJECT-IDENTITY
        STATUS       current
        DESCRIPTION
          "For use with qosSchedulerMethod and
           qosIfSchedulingCapsServiceDisc to indicate Priority
           scheduling method, defined as an algorithm in which the
           presence of data in a queue or set of queues absolutely
           precludes dequeue from another queue or set of queues.
           Notice attributes from qosMinRateEntry of the
           queues/schedulers feeding this scheduler are used when
```

```
   determining the next packet to schedule."
REFERENCE
    "[MODEL] section 7.1.2"
::= { qosSchedulerParameters 1 }
```

```
   qosSchedulerWRR OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
          "For use with qosSchedulerMethod and
          qosIfSchedulingCapsServiceDisc to indicate Weighted Round
          Robin scheduling method, defined as any algorithm in which
          a set of
          queues are visited in a fixed order, and varying amounts of
          traffic are removed from each queue in turn to implement an
          average output rate by class.  Notice attributes from
          qosMinRateEntry of the queues/schedulers feeding this
          scheduler are used when determining the next packet to
          schedule."
       REFERENCE
           "[MODEL] section 7.1.2"
       ::= { qosSchedulerParameters 2 }


   qosSchedulerWFQ OBJECT-IDENTITY
       STATUS       current
       DESCRIPTION
          "For use with qosSchedulerMethod and
   qosIfSchedulingCapsServiceDisc
          to indicate Weighted Fair Queueing scheduling method, defined
   as any
          algorithm in which a set of queues are conceptually visited
   in some
          order, to implement an average output rate by class.  Notice
          attributes from qosMinRateEntry of the queues/schedulers
   feeding
          this scheduler are used when determining the next packet to
   schedule."
       REFERENCE
           "[MODEL] section 7.1.2"
       ::= { qosSchedulerParameters 3 }



   --
   -- Conformance Section
   --


   qosPolicyPibCompliances
                   OBJECT IDENTIFIER ::= { qosPolicyPibConformance 1 }
   qosPolicyPibGroups
                   OBJECT IDENTIFIER ::= { qosPolicyPibConformance 2 }
```

```
qosPolicyPibCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "Describes the requirements for conformance to the
```

```
                QoS Policy PIB."


        MODULE  -- this module
            MANDATORY-GROUPS {
                qosPibDataPathGroup,
                qosPibClfrGroup,
                qosPibClfrElementGroup,
                qosPibActionGroup,
                qosPibAlgDropGroup,
                qosPibQGroup,
                qosPibSchedulerGroup,
                qosPibMinRateGroup,
                qosPibMaxRateGroup }


        GROUP qosPibMeterGroup
        DESCRIPTION
           "This group is mandatory for devices  that  implement
           metering functions."


        GROUP qosPibTBParamGroup
        DESCRIPTION
           "This group is mandatory for devices  that  implement
           token-bucket metering functions."


        GROUP qosPibDscpMarkActGroup
        DESCRIPTION
           "This group is mandatory for devices  that  implement
           DSCP-Marking functions."

        GROUP qosPibMQAlgDropGroup
        DESCRIPTION
           "This group is mandatory for devices  that  implement
           Multiple Queue Measured Algorithmic Drop functions."


        GROUP qosPibRandomDropGroup
        DESCRIPTION
           "This group is mandatory for devices  that  implement
           Random Drop functions."


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

```
OBJECT qosClfrElementClfrId
MIN-ACCESS notify
DESCRIPTION
```

```
              "Install support is not required."


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


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


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


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


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


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


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


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

```
OBJECT qosTBParamBurstSize
MIN-ACCESS notify
DESCRIPTION
```

```
          "Install support is not required."


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


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


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


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


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


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


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


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

```
OBJECT qosRandomDropMinThreshBytes
MIN-ACCESS notify
DESCRIPTION
```

```
             "Install support is not required."

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

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

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

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

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

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

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

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

```
OBJECT qosQMaxRate
MIN-ACCESS notify
DESCRIPTION
```

```
          "Install support is not required."


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


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


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


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


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


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


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


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

```
OBJECT qosMaxRateAbsolute
MIN-ACCESS notify
DESCRIPTION
```

```
           "Install support is not required."


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


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


        ::= { qosPibCompliances 1 }


    qosPibDataPathGroup OBJECT-GROUP
        OBJECTS {
            qosDataPathIfName, qosDataPathRoles,
            qosDataPathDirection, qosDataPathStart
        }
        STATUS current
        DESCRIPTION
           "The Data Path Group defines  the  PIB  Objects  that
           describe a data path."
        ::= { qosPolicyPibGroups 1 }


    qosPibClfrGroup OBJECT-GROUP
        OBJECTS {
            qosClfrId
        }
        STATUS current
        DESCRIPTION
           "The Classifier Group defines the  PIB  Objects  that
           describe a generic classifier."
        ::= { qosPolicyPibGroups 2 }


    qosPibClfrElementGroup OBJECT-GROUP
        OBJECTS {
            qosClfrElementClfrId,  qosClfrElementPrecedence,
            qosClfrElementNext, qosClfrElementSpecific
        }

        STATUS current
        DESCRIPTION
           "The Classifier Group defines the  PIB  Objects  that
```

```
   describe a generic classifier."
::= { qosPolicyPibGroups 3 }
```

```
   qosPibMeterGroup OBJECT-GROUP
       OBJECTS {
           qosMeterSucceedNext, qosMeterFailNext,
           qosMeterSpecific
       }
       STATUS current
       DESCRIPTION
          "The Meter Group defines the objects used in describ-
          ing a generic meter element."
       ::= { qosPolicyPibGroups 4 }


   qosPibTBParamGroup OBJECT-GROUP
       OBJECTS {
           qosTBParamType, qosTBParamRate,
           qosTBParamBurstSize, qosTBParamInterval
       }
       STATUS current
       DESCRIPTION
          "The Token-Bucket Parameter Group  defines  the  objects
          used  in  describing a single-rate token bucket meter
          element."
       ::= { qosPolicyPibGroups 5 }


   qosPibActionGroup OBJECT-GROUP
       OBJECTS {
           qosActionNext, qosActionSpecific
       }
       STATUS current
       DESCRIPTION
          "The  Action  Group  defines  the  objects  used   in
          describing a generic action element."
       ::= { qosPolicyPibGroups 6 }


   qosPibDscpMarkActGroup OBJECT-GROUP
       OBJECTS {
           qosDscpMarkActDscp
       }
       STATUS current
       DESCRIPTION
          "The DSCP Mark Action Group defines the objects  used
          in describing a DSCP Marking Action element."
       ::= { qosPolicyPibGroups 7 }


   qosPibAlgDropGroup OBJECT-GROUP
       OBJECTS {
```

```
    qosAlgDropType, qosAlgDropNext,
    qosAlgDropQMeasure, qosAlgDropQThreshold,
    qosAlgDropSpecific
}
STATUS current
```

```
       DESCRIPTION
          "The Algorithmic Drop Group contains the objects that
          describe algorithmic dropper operation and configura-
          tion."
       ::= { qosPolicyPibGroups 8 }


   qosPibMQAlgDropGroup OBJECT-GROUP
       OBJECTS {
           qosMQAlgDropExceedNext
       }
       STATUS current
       DESCRIPTION
          "The Multiple Queue Measured Algorithmic Drop Group
          contains the objects that describe multiple queue
          measured algorithmic dropper operation and configuration."
       ::= { qosPolicyPibGroups 9 }


   qosPibRandomDropGroup OBJECT-GROUP
       OBJECTS {
           qosRandomDropMinThreshBytes,
           qosRandomDropMinThreshPkts,
           qosRandomDropMaxThreshBytes,
           qosRandomDropMaxThreshPkts,
           qosRandomDropProbMax,
           qosRandomDropWeight,
           qosRandomDropSamplingRate
       }
       STATUS current
       DESCRIPTION
          "The Random Drop Group augments the Algorithmic Drop Group
   for
          random dropper operation and configuration."
       ::= { qosPolicyPibGroups 10 }


   qosPibQGroup OBJECT-GROUP
       OBJECTS {
           qosQNext, qosQMinRate, qosQMaxRate
       }
       STATUS current
       DESCRIPTION
          "The Queue Group contains the objects that describe
          an interface type's queues."
       ::= { qosPolicyPibGroups 11 }


   qosPibSchedulerGroup OBJECT-GROUP
```

```
OBJECTS {
    qosSchedulerNext, qosSchedulerMethod,
    qosSchedulerMinRate, qosSchedulerMaxRate
}
STATUS current
```

```
       DESCRIPTION
          "The  Scheduler  Group  contains  the  objects   that
          describe packet schedulers on interface types."
       ::= { qosPolicyPibGroups 12 }


   qosPibMinRateGroup OBJECT-GROUP
       OBJECTS {
           qosMinRatePriority,
           qosMinRateAbsolute, qosMinRateRelative
       }
       STATUS current
       DESCRIPTION
          "The Minimum Rate Group contains the objects
          that describe packet schedulers' parameters on interface
          types."
       ::= { qosPolicyPibGroups 13 }


   qosPibMaxRateGroup OBJECT-GROUP
       OBJECTS {
           qosMaxRateLevel, qosMaxRateAbsolute, qosMaxRateRelative,
           qosMaxRateThreshold
       }
       STATUS current
       DESCRIPTION
          "The Maximum Rate Group contains the objects
          that describe packet schedulers' parameters on interface
          types."
       ::= { qosPolicyPibGroups 14 }


   END
```

9. Acknowledgments

This PIB builds on all the work that has gone into the Informal Management Model for Diffserv Routers and Management Information Base for the Differentiated Services Architecture.

It has been developed with the active involvement of many people, but most notably Ravi Sahita and Walter Weiss.

10. Subject Category Considerations

The numbering space used for the DiffServ PIB, as indicated by the SUBJECT-CATEGORIES clause, will be assigned by the Internet Assigned Numbers Authority (IANA). Notice the numbering space used by SUBJECT-CATEGORIES maps to the Client Type numbering space in [COPS-PR]. This relationship is detailed in section 7.1 of [SPPI]. Due to the fact that Client Type value of 1 has already been used by [COPS-RSVP], the numbering space for SUBJECT-CATEGORIES will need to start with the value of 2.

Other PIB Modules may use the same SUBJECT-CATEGORIES as this DiffServ PIB Module. In such situations, PRC numbering space under a specific SUBJECT-CATEGORIES should be coordinated with existing PIB Modules using the same SUBJECT-CATEGORIES.

11. 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.

12. 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.

13. RFC Editor Considerations

Some IETF documents this document references are in the IESG last call stage. This document references them as internet drafts. Please use their corresponding RFC numbers prior to publishing of this document as a RFC. The referenced IETF documents are [FR-PIB], [MODEL], and [DS-MIB].

14. IANA Considerations

This document standardizes a Policy Information Base (PIB) module, requesting an IANA assigned PIB number.

15. 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

Carol Bell
Intel
2111 NE 25th Avenue
Hillsboro, OR 97124 USA

Phone: +1 503 264 8491
Email: carol.a.bell@intel.com

Andrew Smith
Allegro Networks
6399 San Ignacio Ave
San Jose, CA 95119
andrew@allegronetworks.com

Francis Reichmeyer
PFN, Inc.
University Park at MIT
26 Landsdowne Street
Cambridge, MA 02139
Phone: +1 617 494 9980
Email: franr@pfn.com

16. 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,", RFC 3084, March 2001

[SPPI]
K. McCloghrie, M. Fine, J. Seligson, K. Chan, S. Hahn, R. Sahita, A. Smith, F. Reichmeyer, "Structure of Policy Provisioning Information", RFC 3159,August 2001.

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

[DSFIELD]

K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[FR-PIB]
M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn,
R. Sahita, A. Smith, F. Reichmeyer, "Framework Policy
Information Base",
Internet Draft <draft-ietf-rap-frameworkpib-06.txt>,
November 2001.

[RAP-FRAMEWORK]
R. Yavatkar, D. Pendarakis, "A Framework for
Policy-based Admission Control", RFC 2753, January 2000.

[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, S. Blake, D. Grossman, A. Smith "An Informal
Management Model for Diffser Routers",
Internet Draft <draft-ietf
diffserv - -model-06.txt>,
February 2001.

[IFMIB]
K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB using
SMIv2", RFC 2233, November 1997.

[DS-MIB]
F. Baker, K. Chan, A. Smith, "Management Information Base for
the Differentiated Services Architecture",
draft-ietf-diffserv-mib-14.txt, October 2001

[ACTQMGMT]
V. Firoiu, M. Borden "A Study of Active Queue Management for
Congestion Control", March 2000, In IEEE Infocom 2000,
http://www.ieee-infocom.org/2000/papers/405.pdf

[AQMROUTER]
V.Misra, W.Gong, D.Towsley "Fluid-based analysis of a network
of AQM routers supporting TCP flows with an application to
RED", In SIGCOMM 2000,
http://www.acm.org/sigcomm/sigcomm2000/conf/paper/
sigcomm2000-4-3.ps.gz

[AF-PHB]
J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured
Forwarding PHB Group.", RFC 2597, June 1999.

[EF-PHB]

V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB." RFC 2598, June 1999.

[INETADDRESS]
Daniele, M., Haberman, B., Routhier, S., Schoenwaelder, J.,

"Textual Conventions for Internet Network Addresses.", RFC 2851, June 2000.

[INTSERVMIB]
F. Baker, J. Krawczyk, A. Sastry, "Integrated Services Management Information Base using SMIv2", RFC 2213, September 1997.

[QUEUEMGMT]
B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RED93]
"Random Early Detection", 1993.

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

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

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

[RFC2026]
Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[SHAPER]
"A Rate Adaptive Shaper for Differentiated Services", RFC 2963, October 2000.

Table of Contents

1. Glossary...2
2. Introduction...2
3. Relationship to the Diffserv Informal Management Model............2
3.1. PIB Overview...3

4. Structure of the PIB...4
4.1. General Conventions..5
4.2. DiffServ Data Paths..5
4.2.1. Data Path PRC..5
4.3. Classifiers..6
4.3.1. Classifier PRC...7
4.3.2. Classifier Element PRC..................................7

4.4. Meters..7

4.4.1. Meter PRC...8
4.4.2. Token-Bucket Parameter PRC....................................8

4.5. Actions...8
4.5.1. DSCP Mark Action PRC......................................9
4.6. Queueing Elements...9
4.6.1. Algorithmic Dropper PRC...................................9

4.6.2. Random Dropper PRC.......................................10
4.6.3. Queues and Schedulers....................................12
4.7. Specifying Device Capabilities.............................13
5. PIB Usage Example..14
5.1. Data Path Example..15
5.2. Classifier and Classifier Element Example..................15
5.3. Meter Example..17
5.4. Action Example...18

5.5. Dropper Examples...18
5.5.1. Tail Dropper Example.....................................19
5.5.2. Single Queue Random Dropper Example......................19
5.5.3. Multiple Queue Random Dropper Example....................20
5.6. Queue and Scheduler Example................................21
6. Summary of the DiffServ PIB..................................24
7. PIB Operational Overview.....................................25

8. PIB Definitions..26
8.1. The DiffServ Base PIB......................................26
9.   Acknowledgments...90
10. Subject Category Considerations.............................90
11. Security Considerations.....................................90
12. Intellectual Property Considerations........................90
13. RFC Editor Considerations...................................90
14. IANA Considerations...91

15. Authors' Addresses..91
16. References..92