Network Working Group
Request for Comments: 3317
Category: Informational

K. Chan
Nortel Networks
R. Sahita
S. Hahn
Intel
K. McCloghrie
Cisco Systems

March 2003

Differentiated Services Quality of Service Policy Information Base

Status of this Memo

Copyright Notice

Abstract

This document describes a Policy Information Base (PIB) for a device implementing the Differentiated Services Architecture. The provisioning classes defined here provide policy control over resources implementing the Differentiated Services Architecture. These provisioning classes can be used with other none Differentiated Services provisioning classes (defined in other PIBs) to provide for a comprehensive policy controlled mapping of service requirement to device resource capability and usage.

Table of Contents

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1. Glossary

PRC	Provisioning Class. A type of policy data. See [POLTERM].
PRI	Provisioning Instance. An instance of a PRC. See [POLTERM].
PIB	Policy Information Base. The database of policy information. See [POLTERM].
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 provisioning classes and instances of these classes residing in a virtual information store called the Policy Information Base (PIB).

This document specifies a set of provisioning 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 diffServ client type. Furthermore, these PRCs are in addition to any other PIBs that may be defined for the diffServ 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, and are 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 is 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, Capability Set, 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.

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

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

- Enqueue the traffic for output in the appropriate queue, whose scheduler 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 class 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 example of parameterization tables for Mark action. 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 PRCs 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 [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 instance (interface type, role combination, interface capability), one for ingress and one for egress. Each instance provides the first DiffServ Functional Element that each packet, at a specific interface (identified by the roles assigned to the interface) traveling in a specific relative direction, should experience. Notice this class is interface specific, with the use of interface type capability set and RoleCombination. To indicate explicitly that there are no DiffServ treatments for a particular interface type capability set, role combination and direction, an instance of the Data Path PRC can be created with zeroDotZero in the dsDataPathStart attribute. This situation can also be indicated implicitly by not supplying an instance of a Data Path PRC for that particular interface type capability set, 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 dsDataPathStart 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 has 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. A 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 an 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 has 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 some 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 composing 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 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 class are referenced from the dsMeterSpecific attributes of meter PRC instances. The parameters are represented by a rate dsTBParamRate, a burst size dsTBParamBurstSize, and an interval dsTBparamInterval. The type of meter being parameterized is indicated by the dsTBParamType 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 dsActionTable to organize one Action’s relationship with the element(s) before and after it. It allows Actions to be cascaded to enable that multiple Actions be applied to a single traffic stream by using each entry’s dsActionNext attribute. The dsActionNext 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 to 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 Algorithmic 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 dsDscpMarkActTable.

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 dsAlgDropType attribute.

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

Algorithmic Droppers may also contain a pointer to a specific detail of the drop algorithm, dsAlgDropSpecific. 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, dsAlgDropQThreshold, of the Algorithmic Dropper entry. The attribute, dsAlgDropQMeasure, specifies which queue’s depth dsAlgDropQThreshold is to be compared against.

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

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

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

- o Random Droppers are recommended as a way to control congestion, in [QUEUEMGMT] and called for in the [AF-PHB]. Various implementations exist, that 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 dsAlgDropEntry; these are discussed below.

A Dropper Type of other is provided for the implementation of dropper types not defined here. When the Dropper Type is other, its full specification will need to be provided by another PRC referenced by dsAlgDropSpecific. A Dropper Type of Multiple Queue Random Dropper is also provided; please reference section 5.5.3 of this document for more details.

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 that 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, dsRandomDropMinThreshBytes and dsRandomDropMinThreshPkts represent Qmin. dsRandomDropMaxThreshBytes and dsRandomDropMaxThreshPkts represent Qmax. dsAlgDropQThreshold represents Qclip. dsRandomDropProbMax 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 dsAlgDropQThreshold.

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      |                    +-------+
      | Type=randomDrop |   +----------------+
      | Specific -------+-->|RandomDrop      |
      +-----------------+   | MinThreshBytes |
                            | MaxThreshBytes |
                            | ProbMax        |
                            | Weight         |
                            | 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 dsRandomDropWeight. The availability of dsRandomDropSamplingRate as readable is important; the information provided by the Sampling Rate is essential to the configuration of dsRandomDropWeight. Having the Sampling Rate be configurable is also helpful, because 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 class, 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., dsMeterSucceedNext. 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 dsMeterSucceedNext 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.

A Queue Table entry specifies 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. [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 class 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 that provides a guaranteed minimum bandwidth to that input, if required. This is represented by a rate dsMinRateAbsolute; 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. Alternatively, the rate may be represented by a relative value, as a fraction of the interface’s current line rate, dsMinRateRelative 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 dsMaxRateAbsolute; 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. Alternatively, the rate may, be represented by a relative value, as a fraction of the interface’s current line rate, dsMaxRateRelative. 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. Additionally, multi-rate shaper [SHAPER] can be represented by the use of multiple dsMaxRateTable entries.

Other types of priority and weighted scheduling methods can be defined using existing parameters in dsMinRateEntry. NOTE: dsSchedulerMethod uses AutonomousType 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 frwkCapabilitySetTable and frwkIfRoleComboTable defined in [FR-PIB] to specify the device's capability sets, interface types, and role combinations. Each instance of the capability PRC frwkCapabilitySetTable 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, that 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 dsIfClassificationCaps PRC. Metering capabilities, which indicate what the device can do with out-of-profile packets, are specified using the dsIfMeteringCaps PRC. Scheduling capabilities, such as the number of inputs supported, are reported using the dsIfSchedulingCaps PRC. Algorithmic drop capabilities, such as the types of algorithms supported, are reported using the dsIfAlgDropCaps PRC. Queue capabilities, such as the maximum number of queues, are reported using the dsIfQueueCaps PRC. Maximum Rate capabilities, such as the maximum number of max rate Levels, are reported using the dsIfMaxRateCaps PRC.

Two PRC's are defined to allow specification of the element linkage capabilities of the PEP. The dsIfElmDepthCaps PRC indicates the maximum number of functional datapath elements that can be linked consecutively in a datapath. The dsIfElmLinkCaps 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. Currently [COPS-PR] error handling mechanism as specified in [COPS-PR] sections 4.4, 4.5, and 4.6 completely handles all known error cases of this PIB; hence no additional methods or PRCs need to be specified here.

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 "ds" 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 diffServSchedulerX".

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 such entry.

```
+---------------------+
|DataPath             |
| CapSetName ="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 on the filters because they are not defined by this PIB. Continuing in the Data Path example from the previous section, lets 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 the following PIB table entries, 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 assume the next differentiated services functional datapath element is Meter and leads 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|
            |                                   |        +-------+
            |  +------------+                   |  +------------+
            +->|TBParam     |                   +->|TBParam     |
               | Type=TRTCM |                      | Type=TRTCM |
               | Rate       |                      | Rate       |
               | BurstSize  |                      | BurstSize  |
               | Interval   |                      | Interval   |
               +------------+                      +------------+
```

Figure 4: Meter Usage Example

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

The use of Random Dropper will introduce the usage of dsRandomDropEntry 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, dsAlgDropQThreshold contains the maximum average queue length, Qclip, for the queue being measured as indicated by dsAlgDropQMeasure, the rest of the Random Dropper parameters are specified by dsRandomDropEntry as referenced by dsAlgDropSpecific. In this example, both dsAlgDropNext and dsAlgDropQMeasure references the same queue. This is the simple case but dsAlgDropQMeasure 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 in 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 dsAlgDropType of dsAlgDropEntry to indicate Multiple Queue Dropping Algorithm.

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

- dsAlgDropType indicates the algorithm used, mQDrop.
- dsAlgDropNext is used to indicate the next functional data path element to handle the flow when no drop occurs.
- dsAlgDropQMeasure is used as the anchor for the list of dsMQAlgDropEntry, one for each queue being measured.

- dsAlgDropQThreshold is used to indicate the size of the shared buffer pool.
- dsAlgDropSpecific can be used to reference instances 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 dsMQAlgDropEntrys, one for each queue being measured, with its attributes used as follows:

- dsMQAlgDropType indicates the algorithm used, for this example, both dsMQAlgDropType uses randomDrop.
- dsMQAlgDropQMeasure indicates the queue being measured.
- dsMQAlgDropNext indicates the next functional data path element to handle the flow when no drop occurs.
- dsMQAlgDropExceedNext is used to indicate the next queue's dsMQAlgDropEntry. With the use of zeroDotZero to indicate the last queue.
- dsMQAlgDropQMeasure is used to indicate the queue being measured. For this example, Q_AF1 and Q_AF2 are the two queues used.
- dsAlgDropQThreshold is used as in single queue Random Dropper.
- dsAlgDropSpecific is used to reference the PRID that describes the dropper parameters as in its normal usage. For this example both dsAlgDropSpecifics reference dsRandomDropEntrys.

Notice the anchoring dsAlgDropEntry and the two dsMQAlgDropEntrys 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 dsRandomDropEntrys 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 the dsAlgDropSpecific of the anchoring dsAlgDropEntry. A 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 of dsAlgDropSpecific. 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 the 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 that 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. The 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 that are summarized in this section.

DiffServ Capabilities Group
   This group consists of PRCs to indicate to the PDP the types of interface supported on the PEP in terms of their DiffServ 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.

DiffServ Policy Group

This group contains configurations of the functional elements that comprise the DiffServ 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 operational overview of configuring DiffServ QoS policy.

After the 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 capabilities of interface types the PEP supports are described by rows in the capability set table, frwkCapabilitySetTable. Each row, or instance of this class contains a pointer to an instance of a PRC that describes the capabilities of the interface type. The capability objects may reside in the dsIfClassifierCapsTable, the dsIfMeteringCapsTable, the dsIfSchedulerCapsTable, the dsIfElmDepthCapsTable, the dsIfElmLinkCapsTable, or in a table defined in another PIB.

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

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

8. PIB Definition

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

IMPORTS
    Unsigned32, MODULE-IDENTITY, MODULE-COMPLIANCE,
    OBJECT-TYPE, OBJECT-GROUP, pib
            FROM COPS-PR-SPPI
    InstanceId, Prid, TagId, TagReferenceId
            FROM COPS-PR-SPPI-TC
    zeroDotZero
         FROM SNMPv2-SMI
    AutonomousType
            FROM SNMPv2-TC
    SnmpAdminString
            FROM SNMP-FRAMEWORK-MIB
    RoleCombination, PrcIdentifierOid, PrcIdentifierOidOrZero,
    AttrIdentifier
            FROM FRAMEWORK-TC-PIB
    Dscp
            FROM DIFFSERV-DSCP-TC
    IfDirection
            FROM DIFFSERV-MIB
    BurstSize
            FROM INTEGRATED-SERVICES-MIB;


dsPolicyPib  MODULE-IDENTITY
    SUBJECT-CATEGORIES { diffServ (2) } -- DiffServ QoS COPS Client Type
    LAST-UPDATED "200302180000Z"        -- 18 Feb 2003
    ORGANIZATION "IETF DIFFSERV WG"
    CONTACT-INFO "
                  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

                  Differentiated Services Working Group:
                  diffserv@ietf.org"
    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.

         Copyright (C) The Internet Society (2003). This version of
         this PIB module is part of RFC 3317; see the RFC itself for
         full legal notices."

    REVISION "200302180000Z"        -- 18 Feb 2003
    DESCRIPTION
         "Initial version, published as RFC 3317."
    ::= { pib 4 }

dsCapabilityClasses    OBJECT IDENTIFIER ::= { dsPolicyPib 1 }
dsPolicyClasses        OBJECT IDENTIFIER ::= { dsPolicyPib 2 }
dsPolicyPibConformance OBJECT IDENTIFIER ::= { dsPolicyPib 3 }

--
-- Interface Type 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 capability tables for DiffServ Functionalities.
--

--
-- The Base Capability Table
--

dsBaseIfCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsBaseIfCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
```

```
      "The Base Interface Type Capability class.  This class
       represents a generic capability supported by a device in the
       ingress, egress, or both directions."
    ::= { dsCapabilityClasses 1 }

dsBaseIfCapsEntry OBJECT-TYPE
    SYNTAX         DsBaseIfCapsEntry
    STATUS         current
    DESCRIPTION
      "An instance of this class describes the dsBaseIfCaps class."

    PIB-INDEX { dsBaseIfCapsPrid }
::= { dsBaseIfCapsTable 1 }

DsBaseIfCapsEntry ::= SEQUENCE {
        dsBaseIfCapsPrid           InstanceId,
        dsBaseIfCapsDirection      INTEGER
}

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


dsBaseIfCapsDirection OBJECT-TYPE
    SYNTAX         INTEGER {
                        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."
    ::= { dsBaseIfCapsEntry 2 }

--
-- The Classification Capability Table
--
```

```
dsIfClassificationCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfClassificationCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
        "This class specifies the classification capabilities of
        a Capability Set."
    ::= { dsCapabilityClasses 2 }


dsIfClassificationCapsEntry OBJECT-TYPE
    SYNTAX         DsIfClassificationCapsEntry
    STATUS         current
    DESCRIPTION
        "An instance of this class describes the classification
        capabilities of a Capability Set."


    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfClassificationCapsSpec }
    ::= { dsIfClassificationCapsTable 1 }

DsIfClassificationCapsEntry ::= SEQUENCE {
        dsIfClassificationCapsSpec BITS
}

dsIfClassificationCapsSpec OBJECT-TYPE
    SYNTAX       BITS {
                       ipSrcAddrClassification(0),
                       -- indicates the ability to classify based on
                       -- IP source addresses
                       ipDstAddrClassification(1),
                       -- indicates the ability to classify based on
                       -- IP destination addresses
                       ipProtoClassification(2),
                       -- indicates the ability to classify based on
                       -- IP protocol numbers
                       ipDscpClassification(3),
                       -- indicates the ability to classify based on
                       -- IP DSCP
                       ipL4Classification(4),
                       -- indicates the ability to classify based on
                       -- IP layer 4 port numbers for UDP and TCP
                       ipV6FlowID(5)
                       -- 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)."
    ::= { dsIfClassificationCapsEntry 1 }

--
-- Metering Capabilities
--

dsIfMeteringCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfMeteringCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
        "This class specifies the metering capabilities of a
        Capability Set."
    ::= { dsCapabilityClasses 3 }

dsIfMeteringCapsEntry OBJECT-TYPE
    SYNTAX         DsIfMeteringCapsEntry
    STATUS         current
    DESCRIPTION
      "An instance of this class describes the metering
      capabilities of a Capability Set."

    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfMeteringCapsSpec }
    ::= { dsIfMeteringCapsTable 1 }

DsIfMeteringCapsEntry ::= SEQUENCE {
        dsIfMeteringCapsSpec       BITS
}

dsIfMeteringCapsSpec OBJECT-TYPE
    SYNTAX  BITS {
                  zeroNotUsed(0),
                  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)."
    ::= { dsIfMeteringCapsEntry 1 }

--
-- Algorithmic Dropper Capabilities
--

dsIfAlgDropCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfAlgDropCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
        "This class specifies the algorithmic dropper
        capabilities of a Capability Set.

        This capability table indicates the types of algorithmic
        drop supported by a Capability Set 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 class."
    ::= { dsCapabilityClasses 4 }

dsIfAlgDropCapsEntry OBJECT-TYPE
    SYNTAX         DsIfAlgDropCapsEntry
    STATUS         current
    DESCRIPTION
        "An instance of this class describes the algorithmic dropper
        capabilities of a Capability Set."
    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfAlgDropCapsType,
                 dsIfAlgDropCapsMQCount }
    ::= { dsIfAlgDropCapsTable 1 }

DsIfAlgDropCapsEntry ::= SEQUENCE {
        dsIfAlgDropCapsType                BITS,
        dsIfAlgDropCapsMQCount             Unsigned32
}

dsIfAlgDropCapsType OBJECT-TYPE
```

```
    SYNTAX      BITS {
                     zeroNotUsed(0),
                     oneNotUsed(1),
                     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
      dsRandomDropTable.  The alwaysDrop(5) will drop every packet
      presented to it.  The mQDrop(6) algorithm will drop packets
      based on measurement from multiple queues."
    ::= { dsIfAlgDropCapsEntry 1 }

dsIfAlgDropCapsMQCount OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    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 dsMQAlgDropEntry
      that can be chained together."
    ::= { dsIfAlgDropCapsEntry 2 }

--
-- Queue Capabilities
--

dsIfQueueCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfQueueCapsEntry
    PIB-ACCESS     notify
    STATUS         current
```

```
    DESCRIPTION
        "This class specifies the queueing capabilities of a
        Capability Set."
    ::= { dsCapabilityClasses 5 }

dsIfQueueCapsEntry OBJECT-TYPE
    SYNTAX         DsIfQueueCapsEntry
    STATUS         current
    DESCRIPTION
        "An instance of this class describes the queue
        capabilities of a Capability Set."
    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfQueueCapsMinQueueSize,
                 dsIfQueueCapsMaxQueueSize,
                 dsIfQueueCapsTotalQueueSize }
    ::= { dsIfQueueCapsTable 1 }

DsIfQueueCapsEntry ::= SEQUENCE {
        dsIfQueueCapsMinQueueSize          Unsigned32,
        dsIfQueueCapsMaxQueueSize          Unsigned32,
        dsIfQueueCapsTotalQueueSize        Unsigned32
}

dsIfQueueCapsMinQueueSize OBJECT-TYPE
    SYNTAX      Unsigned32  (0..4294967295)
    UNITS       "Bytes"
    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.
        dsIfQueueCapsMinQueueSize must be less than or equals to
        dsIfQueueCapsMaxQueueSize when both are specified.
        A zero value indicates not specified."
    ::= { dsIfQueueCapsEntry 1 }

dsIfQueueCapsMaxQueueSize OBJECT-TYPE
    SYNTAX      Unsigned32  (0..4294967295)
    UNITS       "Bytes"
    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.
        dsIfQueueCapsMinQueueSize must be less than or equals to
        dsIfQueueCapsMaxQueueSize when both are specified.
        A zero value indicates not specified."
```

```
    ::= { dsIfQueueCapsEntry 2 }

dsIfQueueCapsTotalQueueSize OBJECT-TYPE
    SYNTAX      Unsigned32  (0..4294967295)
    UNITS       "Bytes"
    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.
        A zero value indicates not specified."
    ::= { dsIfQueueCapsEntry 3 }

--
-- Scheduler Capabilities
--

dsIfSchedulerCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfSchedulerCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
      "This class specifies the scheduler capabilities of a
      Capability Set."
    ::= { dsCapabilityClasses 6 }

dsIfSchedulerCapsEntry OBJECT-TYPE
    SYNTAX         DsIfSchedulerCapsEntry
    STATUS         current
    DESCRIPTION
      "An instance of this class describes the scheduler
      capabilities of a Capability Set."
    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfSchedulerCapsServiceDisc,
                 dsIfSchedulerCapsMaxInputs }
    ::= { dsIfSchedulerCapsTable 1 }

DsIfSchedulerCapsEntry ::= SEQUENCE {
        dsIfSchedulerCapsServiceDisc      AutonomousType,
        dsIfSchedulerCapsMaxInputs        Unsigned32,
        dsIfSchedulerCapsMinMaxRate       INTEGER
}
```

```
dsIfSchedulerCapsServiceDisc OBJECT-TYPE
    SYNTAX      AutonomousType
    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
      shared with and defined in the DiffServ MIB.

      These include diffServSchedulerPriority,
      diffServSchedulerWRR, diffServSchedulerWFQ."
    ::= { dsIfSchedulerCapsEntry 1 }

dsIfSchedulerCapsMaxInputs OBJECT-TYPE
    SYNTAX      Unsigned32  (0..4294967295)
    STATUS      current
    DESCRIPTION
      "The maximum number of queues and/or schedulers that can
      feed into a scheduler indicated by this capability entry.
      A value of zero means there is no maximum."
    ::= { dsIfSchedulerCapsEntry 2 }

dsIfSchedulerCapsMinMaxRate OBJECT-TYPE
    SYNTAX      INTEGER {
                      minRate(1),
                      maxRate(2),
                      minAndMaxRates(3)
                }
    STATUS      current
    DESCRIPTION
      "Scheduler capability indicating ability to handle inputs
      with minimum rate, maximum rate, or both."
    ::= { dsIfSchedulerCapsEntry 3 }

--
-- Maximum Rate Capabilities
--

dsIfMaxRateCapsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF DsIfMaxRateCapsEntry
    PIB-ACCESS     notify
    STATUS         current
    DESCRIPTION
        "This class specifies the maximum rate capabilities of a
        Capability Set."
    ::= { dsCapabilityClasses 7 }

dsIfMaxRateCapsEntry OBJECT-TYPE
```

```
    SYNTAX          DsIfMaxRateCapsEntry
    STATUS          current
    DESCRIPTION
        "An instance of this class describes the maximum rate
        capabilities of a Capability Set."
    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfMaxRateCapsMaxLevels }
    ::= { dsIfMaxRateCapsTable 1 }

DsIfMaxRateCapsEntry ::= SEQUENCE {
        dsIfMaxRateCapsMaxLevels           Unsigned32
}

dsIfMaxRateCapsMaxLevels OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    STATUS       current
    DESCRIPTION
        "The maximum number of levels a maximum rate specification
        may have for this Capability Set and flow direction."
    ::= { dsIfMaxRateCapsEntry 1 }

--
-- DataPath Element Linkage Capabilities
--

--
-- DataPath Element Cascade Depth
--

dsIfElmDepthCapsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DsIfElmDepthCapsEntry
    PIB-ACCESS      notify
    STATUS          current
    DESCRIPTION
        "This class specifies the number of elements of the same
        type that can be cascaded together in a datapath."
    ::= { dsCapabilityClasses 8 }

dsIfElmDepthCapsEntry OBJECT-TYPE
    SYNTAX          DsIfElmDepthCapsEntry
    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
        class can be assumed to have no specific maximum
        depth."
```

```
    EXTENDS { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfElmDepthCapsPrc }
    ::= { dsIfElmDepthCapsTable 1 }

DsIfElmDepthCapsEntry ::= SEQUENCE {
        dsIfElmDepthCapsPrc                PrcIdentifierOid,
        dsIfElmDepthCapsCascadeMax         Unsigned32
}

dsIfElmDepthCapsPrc OBJECT-TYPE
    SYNTAX         PrcIdentifierOid
    STATUS         current
    DESCRIPTION
      "The object identifier of a PRC that represents a functional
      datapath element.  This may be one of:  dsClfrElementEntry,
      dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry, or
      dsSchedulerEntry.
      There may not be more than one instance of this class with
      the same value of dsIfElmDepthCapsPrc and same value of
      dsBaseIfCapsDirection.  Must not contain the value of
      zeroDotZero."
    ::= { dsIfElmDepthCapsEntry 1 }

dsIfElmDepthCapsCascadeMax OBJECT-TYPE
    SYNTAX         Unsigned32  (0..4294967295)
    STATUS         current
    DESCRIPTION
      "The maximum number of elements of type dsIfElmDepthCapsPrc
      that can be linked consecutively in a data path.  A value of
      zero indicates there is no specific maximum."
    ::= { dsIfElmDepthCapsEntry 2 }

--
-- DataPath Element Linkage Types
--

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

dsIfElmLinkCapsEntry OBJECT-TYPE
```

```
    SYNTAX         DsIfElmLinkCapsEntry
    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 { dsBaseIfCapsEntry }
    UNIQUENESS { dsBaseIfCapsDirection,
                 dsIfElmLinkCapsPrc,
                 dsIfElmLinkCapsAttr,
                 dsIfElmLinkCapsNextPrc }
    ::= { dsIfElmLinkCapsTable 1 }

DsIfElmLinkCapsEntry ::= SEQUENCE {
        dsIfElmLinkCapsPrc              PrcIdentifierOid,
        dsIfElmLinkCapsAttr             AttrIdentifier,
        dsIfElmLinkCapsNextPrc          PrcIdentifierOidOrZero
}

dsIfElmLinkCapsPrc OBJECT-TYPE
    SYNTAX         PrcIdentifierOid
    STATUS         current
    DESCRIPTION
      " The object identifier of a PRC that represents a functional
      datapath element.  This may be one of:  dsClfrElementEntry,
      dsMeterEntry, dsActionEntry, dsAlgDropEntry, dsQEntry, or
      dsSchedulerEntry.
      This must not have the value zeroDotZero."
    ::= { dsIfElmLinkCapsEntry 1 }

dsIfElmLinkCapsAttr OBJECT-TYPE
    SYNTAX         AttrIdentifier
    STATUS         current
    DESCRIPTION
      "The value represents the attribute in the PRC
      indicated by dsIfElmLinkCapsPrc that is used to
      specify the next functional element in the datapath."
    ::= { dsIfElmLinkCapsEntry 2 }

dsIfElmLinkCapsNextPrc OBJECT-TYPE
    SYNTAX         PrcIdentifierOidOrZero
    STATUS         current
    DESCRIPTION
      "The value is the OID of a PRC table entry from which
      instances can be referenced by the attribute indicated
      by dsIfElmLinkCapsPrc and dsIfElmLinkAttr.

      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    dsMeterEntry   dsMeterSucceedNext   dsActionEntry
      2    dsMeterEntry   dsMeterSucceedNext   dsMeterEntry
      3    dsMeterEntry   dsMeterFailNext      dsActionEntry.

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

--
-- Policy Classes
--

--
-- Data Path Table
--

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

       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 set's capability
       set name, role combination, and direction. This class can
       therefore have up to two entries for each interface set,
       ingress and egress."
    ::= { dsPolicyClasses 1 }

dsDataPathEntry OBJECT-TYPE
    SYNTAX       DsDataPathEntry
    STATUS       current
    DESCRIPTION
       "Each entry in this class indicates the start of a single
       functional data path, defined by its capability set name,
       role combination and traffic direction.  The first
       functional datapath element to handle traffic for each
       data path is defined by the dsDataPathStart attribute
```

```
       of each table entry.
       Notice for each entry:
       1. dsDataPathCapSetName must reference an existing capability
          set name in frwkCapabilitySetTable [FR-PIB].
       2. dsDataPathRoles must reference existing Role Combination
          in frwkIfRoleComboTable [FR-PIB].
       3. dsDataPathStart 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 dsDataPathEntry will not be installed."
    PIB-INDEX { dsDataPathPrid }
    UNIQUENESS { dsDataPathCapSetName,
                 dsDataPathRoles,
                 dsDataPathIfDirection }
    ::= { dsDataPathTable 1 }

DsDataPathEntry ::= SEQUENCE  {
    dsDataPathPrid            InstanceId,
    dsDataPathCapSetName      SnmpAdminString,
    dsDataPathRoles           RoleCombination,
    dsDataPathIfDirection     IfDirection,
    dsDataPathStart           Prid
}

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

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

dsDataPathRoles 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 capability
         set specified by dsDataPathCapSetName, prior to
         association with an instance of this class."
     ::= { dsDataPathEntry 3 }

dsDataPathIfDirection OBJECT-TYPE
    SYNTAX       IfDirection
    STATUS       current
    DESCRIPTION
       "Specifies the direction for which this data path
       entry applies."
     ::= { dsDataPathEntry 4 }

dsDataPathStart 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:
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry
         dsQEntry

       The PRI pointed to must exist prior to the installation of
       this datapath start element."
     ::= { dsDataPathEntry 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
--

dsClfrTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsClfrEntry
    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 dsClfrElementTable entries
       belonging to each classifier.  Each classifier is referenced
       by its classifier elements using its classifier ID.

       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 dsClfrId of each entry is used to organize all
       classifier elements belonging to the same classifier."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 4.1"
    ::= { dsPolicyClasses 2 }

dsClfrEntry OBJECT-TYPE
    SYNTAX       DsClfrEntry
    STATUS       current
    DESCRIPTION
       "An entry in the classifier table describes a single
       classifier. Each classifier element belonging to this
       classifier must have its dsClfrElementClfrId attribute equal
       to dsClfrId."
    PIB-INDEX { dsClfrPrid }
    UNIQUENESS { dsClfrId }
    ::= { dsClfrTable 1 }

DsClfrEntry ::= SEQUENCE  {
    dsClfrPrid            InstanceId,
    dsClfrId              TagReferenceId
}
```

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

dsClfrId OBJECT-TYPE
    SYNTAX       TagReferenceId
    PIB-TAG      { dsClfrElementClfrId }
    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."
    ::= { dsClfrEntry 2 }

--
-- Classifier Element Table
--

dsClfrElementTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsClfrElementEntry
    PIB-ACCESS   install
    STATUS       current
    DESCRIPTION
       "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.  Hence
       the classifier element table enumerates the relationship
       between classification patterns and subsequent downstream
       diffserv functional data path elements, describing one
       branch of the fan-out characteristic of a classifier
       indicated in [Model].

       Classification parameters are defined by entries of filter
       tables pointed to by dsClfrElementSpecific.  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 [FR-PIB], for IP Multi-Field
       Classifiers (MFCs).
```

```
       If there is ambiguity between classifier elements of the same
       classifier, then dsClfrElementPrecedence needs to be used."
    ::= { dsPolicyClasses 3 }

dsClfrElementEntry OBJECT-TYPE
    SYNTAX       DsClfrElementEntry
    STATUS       current
    DESCRIPTION
       "An entry in the classifier element table describes a
       single element of the classifier."
    PIB-INDEX { dsClfrElementPrid }
    UNIQUENESS { dsClfrElementClfrId,
                 dsClfrElementPrecedence,
                 dsClfrElementSpecific }
    ::= { dsClfrElementTable 1 }

DsClfrElementEntry ::= SEQUENCE  {
    dsClfrElementPrid        InstanceId,
    dsClfrElementClfrId      TagId,
    dsClfrElementPrecedence  Unsigned32,
    dsClfrElementNext        Prid,
    dsClfrElementSpecific    Prid
}

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

dsClfrElementClfrId 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 must be
        the value of dsClfrId attribute for an existing
        instance of dsClfrEntry."
    ::= { dsClfrElementEntry 2 }

dsClfrElementPrecedence OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
```

```
    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.
       Classifier 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
       classifier in place at all times in the ingress
       direction.  This means that there will always be one
       or more filters that match every possible pattern
       that could be presented in an incoming packet.
       There is no such requirement in the egress direction."
    ::= { dsClfrElementEntry 3 }

dsClfrElementNext OBJECT-TYPE
    SYNTAX       Prid
    STATUS       current
    DESCRIPTION
       "This attribute provides one branch  of  the  fan-out
       functionality  of  a  classifier described in Diffserv
       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:
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry
         dsQEntry."
    DEFVAL      { zeroDotZero }
    ::= { dsClfrElementEntry 4 }

dsClfrElementSpecific 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 (Framework 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."
    ::= { dsClfrElementEntry 5 }

--
-- Meters
--
-- This PIB supports a variety of Meters.  It includes a
-- specific definition for Meters whose parameter set can
-- be modeled using Token Bucket parameters.
-- Other metering parameter sets can be defined by other PIBs.
--
-- Multiple meter elements may be logically cascaded
-- using their dsMeterSucceedNext and dsMeterFailNext 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
-- dsMeterEntry and dsTBParamEntry. 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
-- dsMeterEntry and dsTBParamEntry. 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
-- dsMeterEntry and dsTBParamEntry. First set specifies the
-- Committed Target Rate token-bucket. Second set specifies the
-- Peak Target Rate token-bucket. dsTBParamInterval in each
-- token bucket reflects the Average Interval.

dsMeterTable OBJECT-TYPE
```

```
    SYNTAX       SEQUENCE OF DsMeterEntry
    PIB-ACCESS   install
    STATUS       current
    DESCRIPTION
       "This class 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 class. This may include
       all traffic on an interface.

       Specific meter details are to be found in table entry
       referenced by dsMeterSpecific."
   REFERENCE
       "An Informal Management Model for Diffserv Routers,
       RFC 3290, section 5"
    ::= { dsPolicyClasses 4 }

dsMeterEntry OBJECT-TYPE
    SYNTAX       DsMeterEntry
    STATUS       current
    DESCRIPTION
       "An entry in the meter table describes a single
       conformance level of a meter."
    PIB-INDEX { dsMeterPrid }
    UNIQUENESS { dsMeterSucceedNext,
                 dsMeterFailNext,
                 dsMeterSpecific }
    ::= { dsMeterTable 1 }

DsMeterEntry ::= SEQUENCE  {
    dsMeterPrid              InstanceId,
    dsMeterSucceedNext       Prid,
    dsMeterFailNext          Prid,
    dsMeterSpecific          Prid
}

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

dsMeterSucceedNext 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:
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry
         dsQEntry."
    DEFVAL      { zeroDotZero }
    ::= { dsMeterEntry 2 }

dsMeterFailNext 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:
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry
         dsQEntry."
    DEFVAL      { zeroDotZero }
    ::= { dsMeterEntry 3 }

dsMeterSpecific 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, dsMeterSpecific may point to an
       entry in dsTBMeterTable, 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."
    ::= { dsMeterEntry 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 dsMeterTable entries.
--

dsTBParamTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsTBParamEntry
    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
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 5.1"
    ::= { dsPolicyClasses 5 }

dsTBParamEntry OBJECT-TYPE
    SYNTAX       DsTBParamEntry
    STATUS       current
    DESCRIPTION
       "An entry that describes a single token-bucket
       parameter set."
    PIB-INDEX { dsTBParamPrid }
    UNIQUENESS { dsTBParamType,
                 dsTBParamRate,
                 dsTBParamBurstSize,
                 dsTBParamInterval }
    ::= { dsTBParamTable 1 }

DsTBParamEntry ::= SEQUENCE  {
    dsTBParamPrid            InstanceId,
```

```
    dsTBParamType            AutonomousType,
    dsTBParamRate            Unsigned32,
    dsTBParamBurstSize       BurstSize,
    dsTBParamInterval        Unsigned32
}

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

dsTBParamType OBJECT-TYPE
    SYNTAX       AutonomousType
    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:

      diffServTBParamSimpleTokenBucket, diffServTBParamAvgRate,
      diffServTBParamSrTCMBlind, diffServTBParamSrTCMAware,
      diffServTBParamTrTCMBlind, diffServTBParamTrTCMAware,
      diffServTBParamTswTCM

      These are specified in the DiffServ MIB."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 5.1"
    ::= { dsTBParamEntry 2 }

dsTBParamRate OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    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 RFC 3290, section 5.1.1"
    ::= { dsTBParamEntry 3 }
```

```
dsTBParamBurstSize 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 RFC 2698 for trTCM
       3. Burst Size in RFC 3290, section 5."
    ::= { dsTBParamEntry 4 }

dsTBParamInterval OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    UNITS        "microseconds"
    STATUS       current
    DESCRIPTION
       "The time interval used with the token bucket.  For:
       1. Average Rate Meter, RFC 3290, section 5.1.1,
         -Delta.
       2. Simple Token Bucket Meter, RFC 3290, section
          5.1.3, - time interval t.
       3. RFC 2859  TSWTCM, -  AVG_INTERVAL.
       4. RFC 2697 srTCM, RFC 2698 trTCM, - token
          bucket update time interval."
    ::= { dsTBParamEntry 5 }

--
-- Actions
--

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

dsActionTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsActionEntry
    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.

       Specific actions are indicated by dsAction-
       Specific which points to an entry of a specific
       action type parameterizing the action in detail."
```

```
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 6."
    ::= { dsPolicyClasses 6 }

dsActionEntry OBJECT-TYPE
    SYNTAX       DsActionEntry
    STATUS       current
    DESCRIPTION
       "Each entry in the action table allows description of
       one specific action to be applied to traffic."
    PIB-INDEX { dsActionPrid }
    UNIQUENESS { dsActionNext,
                 dsActionSpecific }
    ::= { dsActionTable 1 }

DsActionEntry ::= SEQUENCE  {
    dsActionPrid                InstanceId,
    dsActionNext                Prid,
    dsActionSpecific            Prid
}

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

dsActionNext 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:
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry
         dsQEntry."
    DEFVAL      { zeroDotZero }
    ::= { dsActionEntry 2 }
```

```
dsActionSpecific 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 dsDscpMarkActEntry.
       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."
    ::= { dsActionEntry 3 }

-- DSCP Mark Action Table
--
-- Rows of this class are pointed to by dsActionSpecific
-- to provide detailed parameters specific to the DSCP
-- Mark action.
-- This class should at most contain one entry for each supported
-- DSCP value.  These entries should be reused by different
-- dsActionEntry in same or different data paths.
--

dsDscpMarkActTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsDscpMarkActEntry
    PIB-ACCESS   install
    STATUS       current
    DESCRIPTION
       "This class enumerates specific DSCPs used for marking or
       remarking the DSCP field of IP packets. The entries of this
       table may be referenced by a dsActionSpecific attribute."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 6.1"
    ::= { dsPolicyClasses 7 }

dsDscpMarkActEntry OBJECT-TYPE
    SYNTAX       DsDscpMarkActEntry
    STATUS       current
    DESCRIPTION
      "An entry in the DSCP mark action table that describes a
      single DSCP used for marking."
    PIB-INDEX { dsDscpMarkActPrid }
```

```
    UNIQUENESS { dsDscpMarkActDscp }
    ::= { dsDscpMarkActTable 1 }

DsDscpMarkActEntry ::= SEQUENCE  {
    dsDscpMarkActPrid          InstanceId,
    dsDscpMarkActDscp          Dscp
}

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

dsDscpMarkActDscp 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 class.  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."
    ::= { dsDscpMarkActEntry 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 dsAlgDropSpecific
-- attribute can be used to reference an entry in a parameter table,
-- e.g., dsRandomDropTable for random dropper.

-- For yet more complex dropper, for example, dropper that measures
-- multiple queues, each queue with its own algorithm, can use a
-- dsAlgDropTable entry as the entry point for Algorithmic Dropper
```

```
-- functional data path element, leaving the dropper parameters
-- for each queue be specified by entries of dsMQAlgDropTable.
-- In such usage, the anchoring dsAlgDropEntry's dsAlgDropType
-- should be mQDrop, and its dsAlgDropQMeasure should reference
-- the subsequent dsMQAlgDropEntry's, its dsAlgDropSpecific
-- should be used to reference parameters applicable to all the
-- queues being measured.
-- The subsequent dsMQAlgDropEntry's will provide the parameters,
-- one for each queue being measured.  The dsMQAlgDropEntry's are
-- chained using their dsMQAlgDropNext attributes.
--

dsAlgDropTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsAlgDropEntry
    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
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 7.1.3"
    ::= { dsPolicyClasses 8 }

dsAlgDropEntry OBJECT-TYPE
    SYNTAX       DsAlgDropEntry
    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 dsAlgDropType
       and with more detail parameter entry pointed to by
       dsAlgDropSpecific when necessary."
    PIB-INDEX { dsAlgDropPrid }
    UNIQUENESS { dsAlgDropType,
                 dsAlgDropNext,
                 dsAlgDropQMeasure,
                 dsAlgDropQThreshold,
                 dsAlgDropSpecific }
    ::= { dsAlgDropTable 1 }

DsAlgDropEntry ::= SEQUENCE  {
    dsAlgDropPrid            InstanceId,
    dsAlgDropType            INTEGER,
    dsAlgDropNext            Prid,
    dsAlgDropQMeasure        Prid,
    dsAlgDropQThreshold      Unsigned32,
```

```
    dsAlgDropSpecific         Prid
}

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

dsAlgDropType 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 dsAlgDropSpecific 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:
       dsAlgDropQThreshold represents the depth of the
       queue, pointed to by dsAlgDropQMeasure, 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 dsAlgDropQMeasure, is at
       dsAlgDropQThreshold, 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,
           dsAlgDropSpecific points to a dsRandomDropEntry
           that describes the algorithm.  For this
           algorithm, dsAlgQThreshold is understood to be
           the absolute maximum size of the queue and additional
           parameters are described in dsRandomDropTable.

           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,
           dsAlgDropNext, dsAlgDropQMeasure, and
           dsAlgDropSpecific should be all set to zeroDotZero.

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

        ::= { dsAlgDropEntry 2 }

dsAlgDropNext 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:
          dsClfrEntry
          dsMeterEntry
          dsActionEntry
          dsAlgDropEntry
          dsQEntry.

        When dsAlgDropType is alwaysDrop(5), this attribute is
        Ignored."
    DEFVAL      { zeroDotZero }
    ::= { dsAlgDropEntry 3 }

dsAlgDropQMeasure 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 dsQTable.
       For mQDrop(6), this should point to a dsMQAlgDropEntry that
       Describe one of the queues being measured for multiple
       queue dropper.

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

dsAlgDropQThreshold OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    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 dsAlgDropType 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
       dsAlgDropQMeasure, at which the drop action
       will take place. Other algorithms will need to define
       their own semantics for this threshold."
    ::= { dsAlgDropEntry 5 }

dsAlgDropSpecific 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 dsAlgDropType equal to other(1) must
       have this point to an instance of a PRC defined
       in another PIB module.

       Entries with dsAlgDropType equal to random-
       Drop(4) must have this point to an entry in
       dsRandomDropTable.

       Entries with dsAlgDropType 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."
    ::= { dsAlgDropEntry 6 }

--
-- Multiple Queue Algorithmic Drop Table
--
-- Entries of this table should be referenced by dsAlgDropQMeasure
-- when dsAlgDropType 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, dsMQAlgDropEntry, is extended from
-- dsAlgDropEntry, with usage of corresponding parameters the same
-- except:
--   dsAlgDropNext is used to point to the next diffserv
--     functional data path element when the packet is not dropped.
--   dsMQAlgDropExceedNext is used to point to the next
--     dsMQAlgDropEntry for chaining together the multiple
--     dsMQAlgDropEntry's for the multiple queues being measured.
--

dsMQAlgDropTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsMQAlgDropEntry
    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."
    ::= { dsPolicyClasses 9 }

dsMQAlgDropEntry OBJECT-TYPE
    SYNTAX       DsMQAlgDropEntry
    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 dsAlgDropEntry with adding of a
       dsMQAlgDropExceedNext attribute.

       Further details of the algorithm type are to be found in
       dsAlgDropType and with more detail parameter entry pointed
```

```
       to by dsMQAlgDropSpecific when necessary."
    EXTENDS { dsAlgDropEntry }
    UNIQUENESS { dsMQAlgDropExceedNext }
    ::= { dsMQAlgDropTable 1 }

DsMQAlgDropEntry ::= SEQUENCE  {
    dsMQAlgDropExceedNext     Prid
}

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

--
-- Random Drop Table
--

dsRandomDropTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsRandomDropEntry
    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 dsAlgDropSpecific
       when dsAlgDropType is randomDrop(4)."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 7.1.3"
    ::= { dsPolicyClasses 10 }

dsRandomDropEntry OBJECT-TYPE
    SYNTAX       DsRandomDropEntry
    STATUS       current
    DESCRIPTION
       "An entry describes a process that drops packets
       according to a random algorithm."
    PIB-INDEX { dsRandomDropPrid }
    UNIQUENESS { dsRandomDropMinThreshBytes,
                 dsRandomDropMinThreshPkts,
                 dsRandomDropMaxThreshBytes,
                 dsRandomDropMaxThreshPkts,
```

```
                 dsRandomDropProbMax,
                 dsRandomDropWeight,
                 dsRandomDropSamplingRate
               }
    ::= { dsRandomDropTable 1 }

DsRandomDropEntry ::= SEQUENCE  {
    dsRandomDropPrid             InstanceId,
    dsRandomDropMinThreshBytes   Unsigned32,
    dsRandomDropMinThreshPkts    Unsigned32,
    dsRandomDropMaxThreshBytes   Unsigned32,
    dsRandomDropMaxThreshPkts    Unsigned32,
    dsRandomDropProbMax          Unsigned32,
    dsRandomDropWeight           Unsigned32,
    dsRandomDropSamplingRate     Unsigned32
}

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

dsRandomDropMinThreshBytes OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    UNITS        "bytes"
    STATUS       current
    DESCRIPTION
       "The average queue depth in bytes, beyond which traffic has a
       non-zero probability of being dropped."
     ::= { dsRandomDropEntry 2 }

dsRandomDropMinThreshPkts OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    UNITS        "packets"
    STATUS       current
    DESCRIPTION
      "The average queue depth in packets, beyond which traffic has
      a non-zero probability of being dropped."
    ::= { dsRandomDropEntry 3 }

dsRandomDropMaxThreshBytes OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    UNITS        "bytes"
    STATUS       current
    DESCRIPTION
```

```
      "The average queue depth beyond which traffic has a
      probability indicated by dsRandomDropProbMax of being dropped
      or marked.  Note that this differs from the physical queue
      limit, which is stored in dsAlgDropQThreshold."
    ::= { dsRandomDropEntry 4 }

dsRandomDropMaxThreshPkts OBJECT-TYPE
    SYNTAX       Unsigned32  (1..4294967295)
    UNITS        "packets"
    STATUS       current
    DESCRIPTION
      "The average queue depth beyond which traffic has a
      probability indicated by dsRandomDropProbMax of being dropped
      or marked.  Note that this differs from the physical queue
      limit, which is stored in dsAlgDropQThreshold."
    ::= { dsRandomDropEntry 5 }

dsRandomDropProbMax OBJECT-TYPE
    SYNTAX       Unsigned32  (0..1000)
    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."
    ::= { dsRandomDropEntry 6 }

dsRandomDropWeight OBJECT-TYPE
    SYNTAX       Unsigned32  (0..4294967295)
    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
      dsRandomDropWeight/MaxValue as the coefficient for the new
      sample in the equation, and
      (MaxValue - dsRandomDropWeight)/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
      dsRandomDropWeight via the capability tables."
    ::= { dsRandomDropEntry 7 }

dsRandomDropSamplingRate OBJECT-TYPE
```

```
    SYNTAX       Unsigned32  (0..1000000)
    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)."
    ::= { dsRandomDropEntry 8 }

--
-- Queue Table
--

--
-- An entry of dsQTable 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, dsQNext.
-- 2. How the scheduler will service this queue, with respect
--    to all the other queues the same scheduler needs to service,
--    dsQMinRate and dsQMaxRate.
--
-- Notice one or more upstream diffserv functional data path element
-- may share, point to, a dsQTable entry as described in [MODEL]
-- section 7.1.1.
--

dsQTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsQEntry
    PIB-ACCESS   install
    STATUS       current
    DESCRIPTION
    "The Queue Table enumerates the queues."
    ::= { dsPolicyClasses 11 }

dsQEntry OBJECT-TYPE
    SYNTAX       DsQEntry
    STATUS       current
    DESCRIPTION
       "An entry in the Queue Table describes a single queue
       as a functional data path element."
    PIB-INDEX { dsQPrid }
    UNIQUENESS { dsQNext,
```

```
                 dsQMinRate,
                 dsQMaxRate }
    ::= { dsQTable 1 }

DsQEntry ::= SEQUENCE  {
    dsQPrid                     InstanceId,
    dsQNext                     Prid,
    dsQMinRate                  Prid,
    dsQMaxRate                  Prid
}

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

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

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

dsQMinRate OBJECT-TYPE
    SYNTAX       Prid
    STATUS       current
    DESCRIPTION
       "This Prid indicates the entry in dsMinRateTable
       the scheduler, pointed to by dsQNext, 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."
    ::= { dsQEntry 3 }

dsQMaxRate OBJECT-TYPE
    SYNTAX       Prid
    STATUS       current
```

```
    DESCRIPTION
       "This Prid indicates the entry in dsMaxRateTable
       the scheduler, pointed to by dsQNext, 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."
    ::= { dsQEntry 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 dsSchedulerMinRate and dsSchedulerMaxRate 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.
--

dsSchedulerTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsSchedulerEntry
    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
       dsSchedulerEntry."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 7.1.2"
```

```
    ::= { dsPolicyClasses 12 }

dsSchedulerEntry OBJECT-TYPE
    SYNTAX       DsSchedulerEntry
    STATUS       current
    DESCRIPTION
       "An entry in the Scheduler Table describing a single
       instance of a scheduling algorithm."
    PIB-INDEX { dsSchedulerPrid }
    UNIQUENESS { dsSchedulerNext,
                 dsSchedulerMethod,
                 dsSchedulerMinRate,
                 dsSchedulerMaxRate }
    ::= { dsSchedulerTable 1 }

DsSchedulerEntry ::= SEQUENCE  {
    dsSchedulerPrid                  InstanceId,
    dsSchedulerNext                  Prid,
    dsSchedulerMethod                AutonomousType,
    dsSchedulerMinRate               Prid,
    dsSchedulerMaxRate               Prid
}

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

dsSchedulerNext 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:
         dsSchedulerEntry
         dsQEntry,

       or:
```

```
         dsClfrEntry
         dsMeterEntry
         dsActionEntry
         dsAlgDropEntry

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

dsSchedulerMethod OBJECT-TYPE
    SYNTAX       AutonomousType
    STATUS       current
    DESCRIPTION
      "The scheduling algorithm used by this Scheduler.
      Standard values for generic algorithms:
        diffServSchedulerPriority,
        diffServSchedulerWRR,
        diffServSchedulerWFQ
      are specified in the DiffServ MIB.
      Additional values may be further specified in other PIBs.
      A value of zeroDotZero indicates this is unknown."
    REFERENCE
        "An Informal Management Model for Diffserv Routers,
        RFC 3290, section 7.1.2"
    ::= { dsSchedulerEntry 3 }

dsSchedulerMinRate OBJECT-TYPE
    SYNTAX       Prid
    STATUS       current
    DESCRIPTION
      "This Prid indicates the entry in dsMinRateTable
       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 }
    ::= { dsSchedulerEntry 4 }

dsSchedulerMaxRate OBJECT-TYPE
    SYNTAX       Prid
    STATUS       current
    DESCRIPTION
      "This Prid indicates the entry in dsMaxRateTable
```

```
           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 }
        ::= { dsSchedulerEntry 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 PIB.
   -- 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 dsMinRatePriority/Absolute/Relative 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.
   --

   dsMinRateTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF DsMinRateEntry
       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."
        ::= { dsPolicyClasses 13 }

    dsMinRateEntry OBJECT-TYPE
        SYNTAX       DsMinRateEntry
        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 { dsMinRatePrid }
        UNIQUENESS { dsMinRatePriority,
                     dsMinRateAbsolute,
                     dsMinRateRelative }
        ::= { dsMinRateTable 1 }

    DsMinRateEntry ::= SEQUENCE  {
        dsMinRatePrid            InstanceId,
        dsMinRatePriority        Unsigned32,
        dsMinRateAbsolute        Unsigned32,
        dsMinRateRelative        Unsigned32
    }

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

    dsMinRatePriority OBJECT-TYPE
        SYNTAX       Unsigned32 (1..4294967295)
        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."
        ::= { dsMinRateEntry 2 }

    dsMinRateAbsolute OBJECT-TYPE
        SYNTAX       Unsigned32 (1..4294967295)
        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 dsMinRateRelative:  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 the IF-MIB, RFC 2863."
    ::= { dsMinRateEntry 3 }

dsMinRateRelative OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4294967295)
    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 dsMinRateAbsolute:  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 the IF-MIB, RFC 2863."
    ::= { dsMinRateEntry 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 PIB.
--
-- 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
-- dsMaxRateId but different dsMaxRateLevels.
--
-- The dsMaxRateLevel/Absolute/Relative 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.
--

dsMaxRateTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DsMaxRateEntry
    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."
    ::= { dsPolicyClasses 14 }

dsMaxRateEntry OBJECT-TYPE
    SYNTAX       DsMaxRateEntry
    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 { dsMaxRatePrid }
    UNIQUENESS { dsMaxRateId,
                 dsMaxRateLevel,
                 dsMaxRateAbsolute,
                 dsMaxRateRelative,
                 dsMaxRateThreshold }
    ::= { dsMaxRateTable 1 }

DsMaxRateEntry ::= SEQUENCE  {
    dsMaxRatePrid            InstanceId,
    dsMaxRateId              Unsigned32,
    dsMaxRateLevel           Unsigned32,
    dsMaxRateAbsolute        Unsigned32,
    dsMaxRateRelative        Unsigned32,
    dsMaxRateThreshold       BurstSize
}

dsMaxRatePrid OBJECT-TYPE
    SYNTAX       InstanceId
    STATUS       current
    DESCRIPTION
```

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

    dsMaxRateId OBJECT-TYPE
        SYNTAX       Unsigned32  (0..4294967295)
        STATUS       current
        DESCRIPTION
          "An identifier used together with dsMaxRateLevel for
          representing a multi-rate shaper.  This attribute is used for
          associating all the rate attributes of a multi-rate shaper.
          Each dsMaxRateEntry of a multi-rate shaper must have the same
          value in this attribute.  The different rates of a multi-rate
          shaper is identified using dsMaxRateLevel.
          This attribute uses the value of zero to indicate this
          attribute is not used, for single rate shaper."
        DEFVAL { 0 }
        ::= { dsMaxRateEntry 2 }

    dsMaxRateLevel OBJECT-TYPE
        SYNTAX       Unsigned32 (1..32)
        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 one."
        DEFVAL { 1 }
        ::= { dsMaxRateEntry 3 }

    dsMaxRateAbsolute OBJECT-TYPE
        SYNTAX       Unsigned32 (1..4294967295)
        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 dsMaxRateRelative:  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."
    ::= { dsMaxRateEntry 4 }

dsMaxRateRelative OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4294967295)
    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 dsMaxRateAbsolute:  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 the IF-MIB, RFC 2863."
    ::= { dsMaxRateEntry 5 }

dsMaxRateThreshold 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"
 ::= { dsMaxRateEntry 6 }

--
-- Conformance Section
--

dsPolicyPibCompliances
                OBJECT IDENTIFIER ::= { dsPolicyPibConformance 1 }
dsPolicyPibGroups
                OBJECT IDENTIFIER ::= { dsPolicyPibConformance 2 }

dsPolicyPibCompliance MODULE-COMPLIANCE
```

```
    STATUS  current
    DESCRIPTION
            "Describes the requirements for conformance to the
            QoS Policy PIB."

    MODULE FRAMEWORK-PIB
        MANDATORY-GROUPS {
            frwkPrcSupportGroup,
            frwkPibIncarnationGroup,
            frwkDeviceIdGroup,
            frwkCompLimitsGroup,
            frwkCapabilitySetGroup,
            frwkRoleComboGroup,
            frwkIfRoleComboGroup,
            frwkBaseFilterGroup,
            frwkIpFilterGroup }

    OBJECT frwkPibIncarnationLongevity
    PIB-MIN-ACCESS  notify
    DESCRIPTION
       "Install support is required if policy expiration is to
       be supported."

    OBJECT frwkPibIncarnationTtl
    PIB-MIN-ACCESS  notify
    DESCRIPTION
       "Install support is required if policy expiration is to
       be supported."

    MODULE DIFFSERV-PIB -- this module
        MANDATORY-GROUPS {
            dsPibBaseIfCapsGroup,
            dsPibIfClassificationCapsGroup,
            dsPibIfAlgDropCapsGroup,
            dsPibIfQueueCapsGroup,
            dsPibIfSchedulerCapsGroup,
            dsPibIfMaxRateCapsGroup,
            dsPibIfElmDepthCapsGroup,
            dsPibIfElmLinkCapsGroup,
            dsPibDataPathGroup,
            dsPibClfrGroup,
            dsPibClfrElementGroup,
            dsPibActionGroup,
            dsPibAlgDropGroup,
            dsPibQGroup,
            dsPibSchedulerGroup,
            dsPibMinRateGroup,
            dsPibMaxRateGroup }
```

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

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

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

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

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

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

    OBJECT dsClfrId
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsClfrElementClfrId
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsClfrElementPrecedence
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsClfrElementNext
    PIB-MIN-ACCESS not-accessible
```

```
    DESCRIPTION
       "Install support is not required."

    OBJECT dsClfrElementSpecific
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMeterSucceedNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMeterFailNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMeterSpecific
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsTBParamType
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsTBParamRate
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsTBParamBurstSize
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsTBParamInterval
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsActionNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."
```

```
    OBJECT dsActionSpecific
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsAlgDropType
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsAlgDropNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsAlgDropQMeasure
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsAlgDropQThreshold
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsAlgDropSpecific
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropMinThreshBytes
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropMinThreshPkts
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropMaxThreshBytes
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropMaxThreshPkts
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
```

```
       "Install support is not required."

    OBJECT dsRandomDropProbMax
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropWeight
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsRandomDropSamplingRate
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsQNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsQMinRate
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsQMaxRate
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsSchedulerNext
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsSchedulerMethod
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsSchedulerMinRate
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsSchedulerMaxRate
```

```
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMinRatePriority
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMinRateAbsolute
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMinRateRelative
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMaxRateId
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMaxRateLevel
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMaxRateAbsolute
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMaxRateRelative
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    OBJECT dsMaxRateThreshold
    PIB-MIN-ACCESS not-accessible
    DESCRIPTION
       "Install support is not required."

    ::= { dsPolicyPibCompliances 1 }

dsPibBaseIfCapsGroup OBJECT-GROUP
    OBJECTS {
```

```
        dsBaseIfCapsPrid, dsBaseIfCapsDirection
    }
    STATUS current
    DESCRIPTION
       "The Base Interface Capability Group defines the PIB
       Objects that describe the base for interface capabilities."
    ::= { dsPolicyPibGroups 1 }

dsPibIfClassificationCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfClassificationCapsSpec
    }
    STATUS current
    DESCRIPTION
       "The Classification Capability Group defines the PIB
       Objects that describe the classification capabilities."
    ::= { dsPolicyPibGroups 2 }

dsPibIfMeteringCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfMeteringCapsSpec
    }
    STATUS current
    DESCRIPTION
       "The Metering Capability Group defines the PIB
       Objects that describe the metering capabilities."
    ::= { dsPolicyPibGroups 3 }

dsPibIfAlgDropCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfAlgDropCapsType, dsIfAlgDropCapsMQCount
    }
    STATUS current
    DESCRIPTION
       "The Algorithmic Dropper Capability Group defines the
       PIB Objects that describe the algorithmic dropper
       capabilities."
    ::= { dsPolicyPibGroups 4 }

dsPibIfQueueCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfQueueCapsMinQueueSize, dsIfQueueCapsMaxQueueSize,
        dsIfQueueCapsTotalQueueSize
    }
    STATUS current
    DESCRIPTION
       "The Queueing Capability Group defines the PIB
       Objects that describe the queueing capabilities."
```

```
    ::= { dsPolicyPibGroups 5 }

dsPibIfSchedulerCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfSchedulerCapsServiceDisc, dsIfSchedulerCapsMaxInputs,
        dsIfSchedulerCapsMinMaxRate
    }
    STATUS current
    DESCRIPTION
       "The Scheduler Capability Group defines the PIB
       Objects that describe the scheduler capabilities."
    ::= { dsPolicyPibGroups 6 }

dsPibIfMaxRateCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfMaxRateCapsMaxLevels
    }
    STATUS current
    DESCRIPTION
       "The Max Rate Capability Group defines the PIB
       Objects that describe the max rate capabilities."
    ::= { dsPolicyPibGroups 7 }

dsPibIfElmDepthCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfElmDepthCapsPrc, dsIfElmDepthCapsCascadeMax
    }
    STATUS current
    DESCRIPTION
       "The DataPath Element Depth Capability Group defines the PIB
       Objects that describe the datapath element depth
       capabilities."
    ::= { dsPolicyPibGroups 8 }

dsPibIfElmLinkCapsGroup OBJECT-GROUP
    OBJECTS {
        dsIfElmLinkCapsPrc, dsIfElmLinkCapsAttr,
        dsIfElmLinkCapsNextPrc
    }
    STATUS current
    DESCRIPTION
       "The DataPath Element Linkage Capability Group defines the
       PIB Objects that describe the datapath element linkage
       capabilities."
    ::= { dsPolicyPibGroups 9 }

dsPibDataPathGroup OBJECT-GROUP
    OBJECTS {
```

```
        dsDataPathPrid, dsDataPathCapSetName,
        dsDataPathRoles, dsDataPathIfDirection,
        dsDataPathStart
    }
    STATUS current
    DESCRIPTION
       "The Data Path Group defines the PIB Objects that
       describe a data path."
    ::= { dsPolicyPibGroups 10 }

dsPibClfrGroup OBJECT-GROUP
    OBJECTS {
        dsClfrPrid, dsClfrId
    }
    STATUS current
    DESCRIPTION
       "The Classifier Group defines the PIB Objects that
       describe a generic classifier."
    ::= { dsPolicyPibGroups 11 }

dsPibClfrElementGroup OBJECT-GROUP
    OBJECTS {
        dsClfrElementPrid, dsClfrElementClfrId,
        dsClfrElementPrecedence, dsClfrElementNext,
        dsClfrElementSpecific
    }
    STATUS current
    DESCRIPTION
       "The Classifier Group defines the PIB Objects that
       describe a generic classifier."
    ::= { dsPolicyPibGroups 12 }

dsPibMeterGroup OBJECT-GROUP
    OBJECTS {
        dsMeterPrid, dsMeterSucceedNext,
        dsMeterFailNext, dsMeterSpecific
    }
    STATUS current
    DESCRIPTION
       "The Meter Group defines the objects used in describ-
       ing a generic meter element."
    ::= { dsPolicyPibGroups 13 }

dsPibTBParamGroup OBJECT-GROUP
    OBJECTS {
        dsTBParamPrid, dsTBParamType, dsTBParamRate,
        dsTBParamBurstSize, dsTBParamInterval
    }
```

```
    STATUS current
    DESCRIPTION
       "The Token-Bucket Parameter Group defines the objects
       used in describing a single-rate token bucket meter
       element."
    ::= { dsPolicyPibGroups 14 }

dsPibActionGroup OBJECT-GROUP
    OBJECTS {
        dsActionPrid, dsActionNext, dsActionSpecific
    }
    STATUS current
    DESCRIPTION
       "The  Action Group defines the objects used in
       describing a generic action element."
    ::= { dsPolicyPibGroups 15 }

dsPibDscpMarkActGroup OBJECT-GROUP
    OBJECTS {
        dsDscpMarkActPrid, dsDscpMarkActDscp
    }
    STATUS current
    DESCRIPTION
       "The DSCP Mark Action Group defines the objects used
       in describing a DSCP Marking Action element."
    ::= { dsPolicyPibGroups 16 }

dsPibAlgDropGroup OBJECT-GROUP
    OBJECTS {
        dsAlgDropPrid, dsAlgDropType, dsAlgDropNext,
        dsAlgDropQMeasure, dsAlgDropQThreshold,
        dsAlgDropSpecific
    }
    STATUS current
    DESCRIPTION
       "The Algorithmic Drop Group contains the objects that
       describe algorithmic dropper operation and configura-
       tion."
    ::= { dsPolicyPibGroups 17 }

dsPibMQAlgDropGroup OBJECT-GROUP
    OBJECTS {
        dsMQAlgDropExceedNext
    }
    STATUS current
    DESCRIPTION
       "The Multiple Queue Measured Algorithmic Drop Group
       contains the objects that describe multiple queue
```

```
      measured algorithmic dropper operation and configuration."
    ::= { dsPolicyPibGroups 18 }

dsPibRandomDropGroup OBJECT-GROUP
    OBJECTS {
        dsRandomDropPrid,
        dsRandomDropMinThreshBytes,
        dsRandomDropMinThreshPkts,
        dsRandomDropMaxThreshBytes,
        dsRandomDropMaxThreshPkts,
        dsRandomDropProbMax,
        dsRandomDropWeight,
        dsRandomDropSamplingRate
    }
    STATUS current
    DESCRIPTION
       "The Random Drop Group augments the Algorithmic Drop Group
       for random dropper operation and configuration."
    ::= { dsPolicyPibGroups 19 }

dsPibQGroup OBJECT-GROUP
    OBJECTS {
        dsQPrid, dsQNext, dsQMinRate, dsQMaxRate
    }
    STATUS current
    DESCRIPTION
       "The Queue Group contains the objects that describe
       an interface type’s queues."
    ::= { dsPolicyPibGroups 20 }

dsPibSchedulerGroup OBJECT-GROUP
    OBJECTS {
        dsSchedulerPrid, dsSchedulerNext, dsSchedulerMethod,
        dsSchedulerMinRate, dsSchedulerMaxRate
    }
    STATUS current
    DESCRIPTION
       "The Scheduler Group contains the objects that
       describe packet schedulers on interface types."
    ::= { dsPolicyPibGroups 21 }

dsPibMinRateGroup OBJECT-GROUP
    OBJECTS {
        dsMinRatePrid, dsMinRatePriority,
        dsMinRateAbsolute, dsMinRateRelative
    }
    STATUS current
    DESCRIPTION
```

```
       "The Minimum Rate Group contains the objects
       that describe packet schedulers' parameters on interface
       types."
    ::= { dsPolicyPibGroups 22 }

dsPibMaxRateGroup OBJECT-GROUP
    OBJECTS {
        dsMaxRatePrid, dsMaxRateId, dsMaxRateLevel,
        dsMaxRateAbsolute, dsMaxRateRelative,
        dsMaxRateThreshold
    }
    STATUS current
    DESCRIPTION
       "The Maximum Rate Group contains the objects
       that describe packet schedulers' parameters on interface
       types."
    ::= { dsPolicyPibGroups 23 }

END
```

9. Acknowledgments

Early versions of this specification were also co-authored by Michael Fine, John Seligson, Carol Bell, Andrew Smith, and Francis Reichmeyer.

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 Diana Rawlins, Martin Bokaemper, Walter Weiss, and Bert Wijnen.

10. 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.

In this PIB, there are no PRCs which are sensitive in their own right, such as passwords or monetary amounts. But there are a number of PRCs in this PIB that may contain information that may be sensitive from a business perspective, in that they may represent a customer's service contract or the filters that the service provider chooses to apply to a customer's traffic. These PRCs have a PIB-ACCESS clause of install:

dsDataPathTable, dsClfrTable, dsClfrElementTable, dsMeterTable, dsTBParamTable, dsActionTable, dsDscpMarkActTable, dsAlgDropTable, dsMQAlgDropTable, dsRandomDropTable, dsQTable, dsSchedulerTable, dsMinRateTable, dsMaxRateTable

Malicious altering of the above PRCs may affect the DiffServ behavior of the device being provisioned.

Malicious access of the above PRCs exposes policy information concerning how the device is provisioned.

This PIB also contain PRCs with PIB-ACCESS clause of notify:

dsBaseIfCapsTAble, dsIfClassificationCapsTable, dsIfMeteringCapsTable, dsIfAlgDropCapsTable, dsIfQueueCapsTable, dsIfSchedulerCapsTable, dsIfMaxRateCapsTable, dsIfElmDepthCapsTable, dsIfElmLinkCapsTable

Malicious access of the above PRCs exposes information concerning the device being provisioned.

The use of IPSEC between PDP and PEP, as described in [COPS], provides the necessary protection.

11. Intellectual Property Considerations

The IETF has been 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.

12. IANA Considerations

This document describes the dsPolicyPib Policy Information Base (PIB) modules for standardization under the "pib" branch registered with IANA. The IANA has assigned a PIB number (4) under the "pib" branch.

[SPPI] PIB SUBJECT-CATEGORIES are mapped to COPS Client Types. IANA Considerations for SUBJECT-CATEGORIES follow the same requirements as specified in [COPS] IANA Considerations for COPS Client Types. The DiffServ QoS PIB defines a new COPS Client Type in the Standards space. The IANA has assigned a COPS client type diffServ (2) as described in [COPS] IANA Considerations. IANA has updated the registry (http://www.iana.org/assignments/cops-parameters) for COPS Client Types as a result.

13. Normative 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] Chan, K., Durham, D., Gai, S., Herzog, S., McCloghrie, K., Reichmeyer, F., Seligson, J., Smith, A. and R. Yavatkar, "COPS Usage for Policy Provisioning", RFC 3084, March 2001.

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

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

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

[FR-PIB] Fine, M., McCloghrie, K., Seligson, J., Chan, K., Hahn, S., Sahita, R., Smith, A. and F. Reichmeyer, "Framework Policy Information Base", RFC 3318, March 2003.

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

[SNMP-SMI] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[MODEL] Bernet, Y., Blake, S., Grossman, D. and A. Smith "An Informal Management Model for Diffserv Routers", RFC 3290, May 2002.

[IFMIB] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

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

[ACTQMGMT] Firoiu, V. and 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] Misra, V., Gong, W. and 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] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[EF-PHB] Jacobson, V., Nichols, K. and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.

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

[QUEUEMGMT] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J. and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

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

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

[TSWTCM] Fang, W., Seddigh, N. and 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.

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

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[SHAPER] Bonaventure, O. and S. De Cnodder, "A Rate Adaptive Shaper for Differentiated Services", RFC 2963, October 2000.

[POLTERM] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

14. Authors' Addresses

Kwok Ho Chan
Nortel Networks, Inc.
600 Technology Park Drive
Billerica, MA 01821 USA

Phone: +1 978 288 8175
EMail: khchan@nortelnetworks.com

Ravi Sahita
Intel Labs.
2111 NE 25th Avenue
Hillsboro, OR 97124 USA

Phone: +1 503 712 1554
EMail: ravi.sahita@intel.com

Scott Hahn
Intel
2111 NE 25th Avenue
Hillsboro, OR 97124 USA

Phone: +1 503 264 8231
EMail: scott.hahn@intel.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

15. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.