Internet Engineering Task Force
Diffserv Working Group
INTERNET-DRAFT
Expires:September 2000

F. Baker
Cisco Systems
K. Chan
Nortel Networks
A. Smith
Extreme Networks

Management Information Base for the Differentiated Services Architecture

draft-ietf-diffserv-mib-02.txt

Abstract

This memo describes a proposed MIB for the Differentiated Services Architecture [Architecture] and described by the Differentiated Services Router Conceptual Model [Model].

Currently total agreement on content of this MIB has not been reached, especially in the dropping and queueing mechanism attributes. Further discussion on these topics are required for finalizing this memo.

1. **Status of this Memo**

This particular draft is being developed in the Differentiated Services Working Group. Discussion of it therefore belongs on that list. The charter for Differentiated Services may be

found at http://www.ietf.org/html.charters/diffserv-charter.html

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [1].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [2], RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in RFC 2578 [5], RFC 2579 [6] and RFC 2580 [7].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].

- A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [16].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through

the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine-readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3. Structure of this MIB

This MIB is designed according to the Differentiated Services implementation conceptual model documented in [Model].

3.1. Overview

In principle, if one were to construct a network entirely out of two-port routers (in appropriate places connected by LANs or similar media), then it would be necessary for each router to perform exactly four QoS control functions on traffic in each direction:

- Classify each message according to some set of rules

- In edge devices, determine whether the data stream the message is part of is within or outside its rate

- Perform some set of resulting actions, minimally including applying a drop policy appropriate to the classification and queue in question, and in edge devices perhaps additionally marking the traffic with a Differentiated Services Code Point (DSCP) as defined in [DSCP].

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

If we build the network out of N-port routers, we expect the behavior of the network to be identical. We are forced, therefore, to provide essentially the same set of functions on the ingress port of a router as on the egress port of a router. Some interfaces will be "edge" interfaces and some will be "interior" to the Differentiated Services domain. The one point of difference between an ingress and an egress interface is that all traffic on an egress interface is queued, while traffic on an ingress interface will typically be queued only for shaping purposes.

Hence, in this MIB, we model them identically, making the distinction between ingress and egress interfaces an index variable.

The MIB therefore contains the following elements:
- IP Six Tuple Multi-Field Classification Table
- Classifier Table

- Meter Table
- Action and Action Type Tables
- Queue Set, Queue, and Queue Measurement Tables

3.2. Classifier Table

The classifier table indicates how traffic is sorted out. It identifies separable classes of traffic, by reference to an appropriate classifier, which may be anything from an individual micro-flow to aggregates identified by DSCP. It then sends these classified streams to an appropriate meter or action. In a multi-stage meter, sub-classes of traffic may be sent to different stages. For example, in AF1, 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 failure action.

The structure of the classifier table is a sequence of unambiguous tests. Within each step in the sequence, it should not be important in which order - if order is present at all - the tests are made. This is to facilitate optimized implementations such as index trees. Sequence is present in order to resolve ambiguity.

For example, one might want first to disallow certain applications from using the network at all, or to classify some individual traffic streams that are not diff-serv marked. Traffic that fails those tests might then be inspected for a DSCP. "Then" implies sequence, and the sequence must be somehow specified.

An important form of classifier is "everything else". The final stage of the classifier should be configured to be complete, as the result of an incomplete classifier is not necessarily deterministic.

The actual classifier definition is referenced via a RowPointer, this enable the use of any sort of classification table that one might wish to design, public or proprietary. That classifier table need not be found in this MIB. When ambiguity is present, we disambiguate by explicitly ordering the application of classification rules.

The classifiers specified here are at the per interface level, they may be derived from some higher level policies, but such discussion is out- side the scope of this document.

3.2.1. IP Six Tuple Classifier Table

This MIB currently specifies the IP Six Tuple Classifier, used for IP traffic classification. Entry of this Classifier Table is referenced from the entries of the diffServClassifierTable via a RowPointer, namely diffServClassifierPattern attribute of diffServClassifierEntry object.

The Behavior Aggregate (BA) Classifier is a simple form of the IP Six Tuple Classifier. It is represented by having the diffServSixTupleClfrDscp attribute set to the desired DSCP, and all other classification attributes set to match-all, the default setting.

Each entry in the IP Six Tuple Classifier Table defines a single Classifier, with the use of InetAddress [INETADDR] for both IPv4 and IPv6 addressing. The use of IP Six Tuple Classifiers is discussed in [Architecture] and abstract examples of how they might be configured are provided in [Model].

3.3. Meter Table

A meter, according to the conceptual model, measures the rate at which a stream of traffic passes it, compares it to some set of thresholds, and produces some number (two or more) potential results. A given message is said to "conform" to the meter if at the time that the message is being looked at the stream appears to be within the meter's limit rate. In the MIB, the structure of SNMP makes it easiest to implement this as a set of one or more simple pass/fail tests, which are cascaded. It is to be understood that the meter in a Traffic Control Block is therefore implemented as a set of if-then-else constructs.

The concept of conformance to a meter bears a comment. The concept applied in several rate-control architectures, including ATM, Frame Relay, Integrated Services, and Differentiated Services, is variously described as a "leaky bucket" or a "token bucket".

A leaky bucket algorithm is primarily used for traffic shaping: traffic theoretically departs from the switch at a flat rate of one bit every so many time units, and in fact departs in packets at a rate approximating that. It is also possible to build multi-rate leaky buckets, in which traffic

departs from the switch at varying rates depending on recent activity or inactivity.

A token bucket is used to measure the behavior of a peer's leaky bucket, for verification purposes. It is, by definition, a relationship

```
interval = burst/rate, or
  rate = burst/interval
```

for some defined burst size, in bits, rate, in bits per second, and time interval. Multi-rate token buckets (token buckets with both a peak and a mean rate, and sometimes more rates) are commonly used. In this case, the burst size for the baseline traffic is conventionally referred to as the "committed burst", and the time interval is as specified by

```
interval = committed burst/mean rate
```

but additional burst sizes (each an increment over its predecessor) are defined, which are conventionally referred to as "excess" burst sizes. The peak rate therefore equals the sum of the burst sizes per interval.

A data stream is said to "conform" to a simple token bucket if the switch receives at most the burst size in a given time interval. In the multi-rate case, the traffic is said to conform to the token bucket at a given level if its rate does not exceed the sum of the relevant burst sizes in a given interval. Received traffic pre-classified at one of the "excess" rates (e.g., AF12 or AF13 traffic) is only compared to the relevant excess buckets.

The fact that data is organized into variable length packets introduces some uncertainty in this. For this reason, the token bucket accepts a packet if any of its bits would have been accepted, and "borrows" any excess capacity required from that allotted to equivalently classified traffic in a previous or subsequent interval. More information about this is available in [Model].

Multiple classes of traffic, as identified by the classifier table, may be presented to the same meter. Imagine, for example, that we desire to drop all traffic that uses any DSCP that has not been publicly defined. A classifier entry might exist for each such DSCP, shunting it to an "accepts everything" meter, and dropping all traffic that conforms to

only that meter.

Clearly, it is necessary to identify what is to be done with messages that conform to the meter, and with messages that do not. It is also necessary for the meter to be arbitrarily extensible, as some PHBs require the successive application of an arbitrary number of meters. The approach taken in this design is to have each meter indicate what action is to be taken for conforming traffic, and what meter is to be used for traffic which fails to conform. With the definition of a special type of meter to which all traffic conforms, we now have the necessary flexibility.

3.4. Action Table

Considerable discussion has taken place regarding the possible actions. Suggested actions include "no action", "mark the traffic", "drop the traffic, based on some algorithm", "shape the traffic", "count it". This MIB attempts to make the specification of the action flexible by using the Action Table to organize one Action's relationship with the Meter element before it, with the Queueing element following it, and with other Action elements to allow multiple Actions be applied to a single traffic stream. The parameters needed will depend on the type of Action to be taken. Hence there are Action Tables for the different Action Types. This MIB currently defines parameters for: 1. Mark Action, 2. Count Action, 3. Drop Action,

This flexibility allows additional Actions be specified in future revisions of this MIB, or in other MIBs. And possible usage of propietary Action without impact to the Actions provided here.

The Mark Action is relatively straight forward.

For Count Action, when it is specified, it will always be applied first, before any other type of Actions. For example, when both Count and a Drop Action is specified, the Count Action will always count the total counts of this traffic stream, before any traffic gets dropped, even if the Action entries are chained with the Drop Action first, before the Count Action. There are counters in the Drop Actions to indicate the ammount of traffic dropped, within the drop context.

The Drop Actions require close relationship with queueing,

with detail as follows:

The Tail Drop Action requires the specification of a maximum queue depth, at which point any traffic exceeding the maximum queue depth gets discarded.

Editor's Note Start
There is still debates on what
attributes are needed and how they may be related to queueing. The following is viewed as the complex description of how it may be done. The following set of attributes and its description may be simplified before this memo is finalized. There is a more detail discussion of why these attributes are required in [ActQMgmt].
Editor's Note End

The Random Drop Action requires the specification of its drop characteristic with the following parameters (drop characteristic described using a plot with drop probability, P, as Y axis, and average queue length, Q, as X axis):

1. **Pmin and Qmin defines the start of the characteristic plot. Normally Pmin=0, meaning with average queue length below** Qmin, there will be no drops.
2. **Pmax and Qmax defines a "knee" on the plot, after which point the drop probability become more progressive (greater** slope).
   Qclip defines the average queue length at which all packets will be dropped, probability = 1. Notice this is different from Tail Drop because this uses average queue length.
   It is possible for Qclip = Qmax, meaning when the average queue length exceeds Qmax, all packets will be dropped.
3. **The sampling interval and average weight parameters are used for calculation of average queue. These parameters are** important because they can affect the behavior and outcome of the drop process. They can also be very sensitive and may have a wide range of possible values due to wide range of link speeds, hence the use of real number format for average weight.

Deterministic Drop Action can be viewed as a special case of Random Drop with drop probability restricted to zero and one. Hence Deterministic Drop Action can be described as follows:
1. **Pmin = 0 and Pmax = 1.** 2. Qmin = Qmax = Qclip indicating the average queue length that drop occurs.

For the drop actions, each drop process specification is associated with a queue. This allows multiple drop processes (of same or different types) be associated with the same queue, as different PHB implementation may require. This setup allows the flexibility for Action specification, including multiple sequential drop processes if necessary.

When counters are specified, two sizes of objects are defined. These are defined in accordance with the method found in [IFMIB]; both 32 and 64 bit counters are defined, with the expectation that the 32 bit counter is simply the least significant bits of the 64 bit counter. For interfaces that operate at 20,000,000 (20 million) bits per second or less, 32-bit byte and packet counters MUST be used. For interfaces that operate faster than 20,000,000 bits/second, and slower than 650,000,000 bits/second, 32-bit packet counters MUST be used and 64-bit octet counters MUST be used. For interfaces that operate at 650,000,000 bits/second or faster, 64-bit packet counters AND 64-bit octet counters MUST be used.

Multiple Actions can be chained using the ActionNext attribute. The last Action's ActionNext attribute points to the next TCB, normally a Queue Entry for the Queue element.

3.5. Queueing Element

The Queueing element consists of Queue Table and Queue Set Table. With Queue Table containing relatively simple FIFO queues. Using the Queue Set Table to allow flexibility in constructing both simple and complex queueing hierarchies.

The queue entries in the Queue Table have simple attributes, it includes a reference to which queue set the queue belongs to, and a weight parameter. For Priority Queueing, the weight parameter indicates the priority of this queue with respect to all the other queues within the same queue set. A higher weight value queue will be service first over a lower weight value queue in the same queue set. For weighed queueing algorithms, the weight parameter is a percentage number. With the value of 1,000 meaning 1 percent, allowing fine control of bandwidth allocation when needed. A higher weight value queue will have higher probability of being service when compared to a lower weight value queue in the same queue set. The weight values for all the queues within a queue set must add up to less than or equal to 100,000 (100%). Each queue is capable of acting as a work-conserving queue, one which transmits as

rapidly as its weight allows, but guarantees to its class of traffic, as a side effect of its weight, a minimum rate. Or acting as a non-work-conserving "shaping" queue.

The entries in the Queue Set Table describes the attributes common to all queues within the queue set. This includes the dequeueing Method, or algorithm used amongst the queues in the queue set. Currently, priority queueing, Weighed Fair Queueing, Weighed Round Robin are listed as the possible chooses, other methods/algorithms, e.g. Class Base Queueing, can be added.

The rates, both minimum and maximum, are specified for the queue set instead of per queue. This allows

A hierarchical tree of queue sets can be constructed using the parent/child queue set concept. The attributes QSetParentId and QSetWeight is used for this purpose, with QSetParentId indicating the parent's QSetId, and QSetWeight used as the child queue set's total weight amongst the queues in the parent queue set. There can be multiple children queue set under one parent queue set, with each child queue set looks like a queue from the parent queue set's perspective. Hence queue sets can be recursively defined, inter mixing with queues at any level.

A mixed dequeue scheduling discipline can be built for an interface. For example, with the following queues and queue sets:

Q Parameters			Q Set Parameters			
QId	QSetId	QWeight	Method	MinRate	ParentId	QSetWeight
11	61	100	PQ	0	0	0
12	61	99	PQ	0	0	0
21	71	50,000	WFQ	10000	61	98
22	71	30,000	WFQ	10000	61	98
23	71	20,000	WFQ	10000	61	98
31	81	70,000	WRR	500	61	97
32	81	30,000	WRR	500	61	97

Notice in this example there are three queue sets: Queue Set **61 uses Priority Queueing, it have 2 child Queue Sets.** Queue Set 71 uses Weighed Fair Queueing with KBPS as RateUnit.

Queue Set 81 uses Weighed Round Robin with Packets/Sec as RateUnit.

Queues 11, 12, queue sets 71 and 81 belongs to Queue Set 61. Queues 21, 22, 23 belongs to Queue Set 71. Queues 31, 32 belongs to Queue Set 81.

All traffic in queue 11 will be serviced first, then all traffic in queue 12 will be serviced second. After traffic in queues 11 and 12 are serviced, queues 21, 22, 23 are serviced among themselves in a fair queueing fashion, based on their respective weight. After traffic in queues 21, 22, 23 are serviced, queues 31, 32 are serviced among themselves in a round robin fashion, based on their respective weight. Notice Queue Set 71 uses KBPS RateUnit, resulting in bit/byte fair queueing. Queue Set 81 uses Packet RateUnit, resulting in packet fair queueing.

The rates for each queue can be derived: Queue 21 have minimum rate of 50% of 10000 KBPS, 5000 KBPS. Queue 31 have minimum rate of 70% of 500 Pkt/Sec, 350 Pkt/Sec.

Other types of scheduling algorithms can be used in the parent or child queue sets, creating different queueing behaviors.

The queue set can also operate as a traffic shaper by using the maximum rate attribute.

Chains of Queues/Queue Sets can be built using the NextTCB attribute in Queue Set entry.

Multiple meters may direct their traffic to the same queue. For example, the Assured Forwarding PHB suggests that all traffic marked AF11, AF12, or AF13 be placed in the same queue without reordering.

3.6. The use of RowPointer

RowPointer is a textual convention used to identify a conceptual row in an SNMP Table by pointing to one of its objects. In this MIB, it is used in two ways: to indicate indirection, and to indicate succession.

When used for indirection, as in the Classifier table, the idea is to allow other MIBs, including proprietary ones, to identify new and arcane classifiers - MAC headers, IP4 and IP6

headers, BGP Communities, and all sorts of things.

When used for succession, it answers the question "what happens next?". Rather than presume that the next table must be as specified in the conceptual model and providing its index, the RowPointer takes you to the MIB row representing that thing. In the Meter Table, for example, the "FailNext" RowPointer might take you to another meter, while the "SucceedNext" RowPointer would take you to an action.

4. MIB Definition

```
DIFF-SERV-MIB DEFINITIONS ::= BEGIN

    IMPORTS
    Unsigned32, Counter32, Counter64, OBJECT-TYPE,
    MODULE-IDENTITY, zeroDotZero, mib-2          FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, RowPointer, TestAndIncr
                                                 FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP              FROM SNMPv2-CONF
    ifIndex                                      FROM IF-MIB
    DisplayString                                FROM RFC1213-MIB
    InetAddressType, InetAddress                 FROM INET-ADDRESS-
MIB;

diffServMib MODULE-IDENTITY
    LAST-UPDATED "9907190100Z" -- Mon Jul 19 01:00:00 PDT 1999
    ORGANIZATION "Cisco Systems"
    CONTACT-INFO
       "       Fred Baker
       Postal: 519 Lado Drive
               Santa Barbara, California 93111
       Tel: +1 (408) 526-4257
       FAX: +1 (805) 681-0115
       E-mail: fred@cisco.com

               Kwok Ho Chan
       Postal: 600 Technology Park Drive
               Billerica, Massachusetts 01821, USA
       Tel: +1 (978) 288-8175
       FAX: +1 (978) 288-4690
       E-mail: khchan@nortelnetworks.com

               Andrew Smith
       Postal: 3585 Monroe St.
               Santa Clara, California 95051
       Tel: +1 (408) 579 2821
       FAX: +1 (408) 579 3000
       E-mail: andrew@extremenetworks.com"
    DESCRIPTION
       "This MIB defines the objects necessary to manage a
       device that uses the Differentiated Services
       Architecture described in RFC 2475 and the Conceptual
       Model for DiffServ Routers in draft-ietf-diffserv-
       model-01.txt."
    REVISION "9907190100Z" -- Mon Jul 19 01:00:00 PDT 1999
    DESCRIPTION
       "Initial version, published as RFC xxxx."
```

```
    ::= { mib-2 12345 }  -- anybody who uses this unassigned
                         -- number deserves the wrath of IANA

diffServObjects        OBJECT IDENTIFIER ::= { diffServMib 1 }
diffServTables         OBJECT IDENTIFIER ::= { diffServMib 2 }
diffServAugments       OBJECT IDENTIFIER ::= { diffServMib 3 }
diffServMIBConformance OBJECT IDENTIFIER ::= { diffServMib 4 }

--
-- These textual conventions has no effect on either the syntax
-- nor the semantics of any managed object.  Objects defined
-- using this convention are always encoded by means of the
-- rules that define their primitive type.
--
Dscp ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS   current
    DESCRIPTION
       "The code point used for discriminating a traffic
       stream."
    SYNTAX   INTEGER (-1 | 0..63)

SixTupleClfrL4Port ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS   current
    DESCRIPTION
       "A value indicating a Layer-4 protocol port number."
    SYNTAX   INTEGER (0..65535)
```

```
--
-- Classifiers
--
-- The tools for IP Six Tuple Classification.

-- This object allows a configuring system to obtain a
-- unique value for diffServSixTupleClfrId for purposes
-- of configuration.

diffServSixTupleClfrUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServSixTupleClfrUnique object yields a unique new
       value for diffServSixTupleClfrId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 1 }

diffServSixTupleClfrTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServSixTupleClfrEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "A table of IP Six Tuple Classifier entries that a
       system may use to identify traffic."
    ::= { diffServTables 1 }

diffServSixTupleClfrEntry OBJECT-TYPE
    SYNTAX       DiffServSixTupleClfrEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An IP Six Tuple Classifier entry describes a single
       classifier."
    INDEX { diffServSixTupleClfrId }
    ::= { diffServSixTupleClfrTable 1 }

DiffServSixTupleClfrEntry ::= SEQUENCE {
    diffServSixTupleClfrId           Unsigned32,
    diffServSixTupleClfrAddrType     InetAddressType,
    diffServSixTupleClfrDstAddr      InetAddress,
    diffServSixTupleClfrDstAddrMask  InetAddress,
    diffServSixTupleClfrSrcAddr      InetAddress,
    diffServSixTupleClfrSrcAddrMask  InetAddress,
    diffServSixTupleClfrDscp         Dscp,
    diffServSixTupleClfrProtocol     INTEGER,
```

```
    diffServSixTupleClfrDstL4PortMin SixTupleClfrL4Port,
    diffServSixTupleClfrDstL4PortMax SixTupleClfrL4Port,
    diffServSixTupleClfrSrcL4PortMin SixTupleClfrL4Port,
    diffServSixTupleClfrSrcL4PortMax SixTupleClfrL4Port,
    diffServSixTupleClfrStatus       RowStatus
}

diffServSixTupleClfrId OBJECT-TYPE
    SYNTAX         Unsigned32 (1..2147483647)
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
       "A unique id for the classifier. This object is meant
       to be pointed to by a RowPointer from other tables,
       such as the diffServClassifierPattern."
    ::= { diffServSixTupleClfrEntry 1 }

diffServSixTupleClfrAddrType OBJECT-TYPE
    SYNTAX         InetAddressType
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The type of IP address used by this classifier entry."
    ::= { diffServSixTupleClfrEntry 2 }

diffServSixTupleClfrDstAddr OBJECT-TYPE
    SYNTAX         InetAddress
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The IP address to match against the packet's
       destination IP address."
    ::= { diffServSixTupleClfrEntry 3 }

diffServSixTupleClfrDstAddrMask OBJECT-TYPE
    SYNTAX         InetAddress
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "A mask for the matching of the destination IP address.
       A zero bit in the mask means that the corresponding bit
       in the address always matches."
    DEFVAL         {0}
    ::= { diffServSixTupleClfrEntry 4 }

diffServSixTupleClfrSrcAddr OBJECT-TYPE
    SYNTAX         InetAddress
```

```
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The IP address to match against the source IP address
       of each packet."
    ::= { diffServSixTupleClfrEntry 5 }

diffServSixTupleClfrSrcAddrMask OBJECT-TYPE
    SYNTAX         InetAddress
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "A mask for the matching of the source IP address.  A
       zero bit in the mask means that the corresponding bit
       in the address always matches."
    DEFVAL         {0}
    ::= { diffServSixTupleClfrEntry 6 }

diffServSixTupleClfrDscp OBJECT-TYPE
    SYNTAX         Dscp
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The value that the DSCP in the packet must have to
       match this entry. A value of -1 indicates that a
       specific DSCP value has not been defined and thus all
       DSCP values are considered a match."
    DEFVAL         {-1}
    ::= { diffServSixTupleClfrEntry 7 }

diffServSixTupleClfrProtocol OBJECT-TYPE
    SYNTAX         INTEGER (0..255)
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The IP protocol to match against the IPv4 protocol
       number in the packet. A value of zero means match all."
    DEFVAL         {0}
    ::= { diffServSixTupleClfrEntry 8 }

diffServSixTupleClfrDstL4PortMin OBJECT-TYPE
    SYNTAX         SixTupleClfrL4Port
    MAX-ACCESS     read-create
    STATUS         current
    DESCRIPTION
       "The minimum value that the layer-4 destination port
       number in the packet must have in order to match this
```

```
       classifier entry."
    DEFVAL         {0}
    ::= { diffServSixTupleClfrEntry 9 }

diffServSixTupleClfrDstL4PortMax OBJECT-TYPE
    SYNTAX         SixTupleClfrL4Port
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The maximum value that the layer-4 destination port
       number in the packet must have in order to match this
       classifier entry. This value must be equal to or
       greater that the value specified for this entry in
       diffServSixTupleClfrDstL4PortMin."
    DEFVAL         {65535}
    ::= { diffServSixTupleClfrEntry 10 }

diffServSixTupleClfrSrcL4PortMin OBJECT-TYPE
    SYNTAX         SixTupleClfrL4Port
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The minimum value that the layer-4 source port number
       in the packet must have in order to match this
       classifier entry."
    DEFVAL         {0}
    ::= { diffServSixTupleClfrEntry 11 }

diffServSixTupleClfrSrcL4PortMax OBJECT-TYPE
    SYNTAX         SixTupleClfrL4Port
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
       "The maximum value that the layer-4 source port number
       in the packet must have in oder to match this
       classifier entry. This value must be equal to or
       greater that the value specified for this entry in
       dsSixTupleIpSrcL4PortMin."
    DEFVAL         {65535}
    ::= { diffServSixTupleClfrEntry 12 }

diffServSixTupleClfrStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The RowStatus variable controls the activation,
```

```
           deactivation, or deletion of a classifier. Any writable
           variable may be modified whether the row is active or
           notInService."
        ::= { diffServSixTupleClfrEntry 13 }


-- Classifier Table
-- This object allows a configuring system to obtain a
-- unique value for diffServClassifierNumber for purposes of
-- configuration

diffServClassifierUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServClassifierUnique object yields a unique
       new value for diffServClassifierId when read and
       subsequently set. This value must be tested for
       uniqueness."
    ::= { diffServObjects 2 }


-- The Classifier Table allows us to enumerate the relationship
-- between arbitrary classifiers and the meters which apply
-- to classified streams.

diffServClassifierTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServClassifierEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The classifier table defines the classifiers that a
       system applies to incoming traffic.  Specific
       classifiers are defined by RowPointers in this table
       which identify entries in classifier tables of specific
       type, e.g.  Multi-field classifiers for IP are defined
       in diffServSixTupleClfrTable.  Other classifier types
       may be defined elsewhere."
    ::= { diffServTables 2 }

diffServClassifierEntry OBJECT-TYPE
    SYNTAX       DiffServClassifierEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the classifier table describes a single
```

```
       classifier."
    INDEX { ifIndex, diffServInterfaceDirection,
            diffServClassifierId }
    ::= { diffServClassifierTable 1 }

DiffServClassifierEntry ::= SEQUENCE  {
    diffServInterfaceDirection     INTEGER,
    diffServClassifierId           Unsigned32,
    diffServClassifierPattern      RowPointer,
    diffServClassifierNext         RowPointer,
    diffServClassifierPrecedence   Unsigned32,
    diffServClassifierStatus       RowStatus
}

diffServInterfaceDirection OBJECT-TYPE
    SYNTAX  INTEGER {
                inbound(1),     -- ingress interface
                outbound(2)     -- egress interface
            }
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Specifies the direction for this entry on the
       interface. 'inbound' traffic is operated on during
       receipt, while 'outbound' traffic is operated on prior
       to transmission."
    ::= { diffServClassifierEntry 1 }

diffServClassifierId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Classifier Id enumerates the classifier entry."
    ::= { diffServClassifierEntry 2 }

diffServClassifierPattern OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "A pointer to a valid entry in another table that
       describes the applicable classification pattern, e.g.
       an entry in diffServSixTupleClfrTable.  If the row
       pointed to does not exist, the classifier is ignored.

       The value zeroDotZero is interpreted to match anything
```

```
       not matched by another classifier - only one such entry
       may exist in this table."
    DEFVAL { zeroDotZero }
    ::= { diffServClassifierEntry 3 }

diffServClassifierNext OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The 'next' variable selects the next datapath element
       to handle the classified flow.  For example, this can
       points to an entry in the meter or action table."
    ::= { diffServClassifierEntry 4 }

diffServClassifierPrecedence OBJECT-TYPE
    SYNTAX       Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The relative precedence in which classifiers are
       applied, higer numbers represent classifiers with
       higher precedence.  Classifiers with the same
       precedence must be unambiguous, i.e. they must define
       non-overlapping patterns.  Classifiers with different
       precedence may overlap in their patterns:  the
       classifier with the highest precedence that matches is
       taken."
    DEFVAL { 0 }
    ::= { diffServClassifierEntry 5 }

diffServClassifierStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of a classifier. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServClassifierEntry 6 }
```

```
--
-- Meters
--

-- This MIB includes definitions for Token-Bucket
-- Meters as one example of possible meters.

-- This object allows a configuring system to obtain a
-- unique value for diffServTBMeterId for purposes of
-- configuration

diffServTBMeterUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServTBMeterUnique object yields a unique new
       value for diffServTBMeterId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 3 }

diffServTBMeterTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServTBMeterEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Meter Table enumerates specific token bucket
       meters that a system may use to police a stream of
       classified traffic. The traffic stream is defined by
       the classifier.  It may include all traffic."
    ::= { diffServTables 3 }

diffServTBMeterEntry OBJECT-TYPE
    SYNTAX       DiffServTBMeterEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the meter table describes a single token
       bucket meter. Note that a meter has exactly one rate,
       defined as the burst size each time interval. Multiple
       meters may be cascaded should a multi-rate token bucket
       be needed in a given Per-Hop Behavior. An example of
       such a PHB is AF."
    INDEX { ifIndex, diffServInterfaceDirection,
            diffServTBMeterId  }
    ::= { diffServTBMeterTable 1 }
```

```
DiffServTBMeterEntry ::= SEQUENCE  {
    diffServTBMeterId                Unsigned32,
    diffServTBMeterRate              Unsigned32,
    diffServTBMeterBurstSize         Unsigned32,
    diffServTBMeterFailNext          RowPointer,
    diffServTBMeterSucceedNext       RowPointer,
    diffServTBMeterStatus            RowStatus
}

diffServTBMeterId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "MeterId enumerates the meter entry."
    ::= { diffServTBMeterEntry 1 }

diffServTBMeterRate OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "KBPS"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The token bucket rate, in kilo-bits per second (KBPS).
       Note that if multiple meters are cascaded onto one PHB,
       the peak rate of the data stream is the sum of their
       rates."
    ::= { diffServTBMeterEntry 2 }

diffServTBMeterBurstSize OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "bytes"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The number of bytes in a single transmission burst.
       The interval can be derived with (BurstSizex8)/Rate."
    ::= { diffServTBMeterEntry 3 }

diffServTBMeterFailNext OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "If the traffic does NOT conform to the meter, FailNext
       indicates the next datapath element to handle the
       traffic.  For example, an Action or Meter datapath
```

```
          element.  The value zeroDotZero in this variable
          indicates no further DiffServ treatment is performed on
          this flow by the current interface for this interface
          direction."
    DEFVAL      { zeroDotZero }
    ::= { diffServTBMeterEntry 4 }

diffServTBMeterSucceedNext OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
          "If the traffic does conform to the meter, SucceedNext
          indicates the next datapath element to handle the
          traffic.  For example, an Action or Meter datapath
          element.  The value zeroDotZero in this variable
          indicates no further DiffServ treatment is performed on
          this flow by the current interface for this interface
          direction."
    DEFVAL      { zeroDotZero }
    ::= { diffServTBMeterEntry 5 }

diffServTBMeterStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
          "The RowStatus variable controls the activation,
          deactivation, or deletion of a meter. Any writable
          variable may be modified whether the row is active or
          notInService."
    ::= { diffServTBMeterEntry 6 }
```

```
--
-- Actions
--
-- Notice the Drop Action attributes are referenced by the
-- action table rather than by the queue table because
-- Differentiated Services PHBs, such as the Assured Service,
-- permit differently classified traffic to have different
-- drop parameters even though they occupy the same queue."
--

-- Mark Action Table
-- Rows of this table is pointed to by diffServAction to
-- provide detail parameters specific to an Action Type.

-- This object allows a configuring system to obtain a
-- unique value for diffServMarkActId for purposes of
-- configuration.

diffServMarkActUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServMarkActUnique object yields a unique new
       value for diffServMarkActId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 4 }

diffServMarkActTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServMarkActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The mark action table enumerates specific DSCPs used
       for marking or remarking the DSCP field.  The entries
       of this table is meant to be referenced by the
       diffServAction attribute of entries in
       diffServActionTable for diffServActionType = mark."
    ::= { diffServTables 4 }

diffServMarkActEntry OBJECT-TYPE
    SYNTAX       DiffServMarkActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the mark action table describes a single
       DSCP used for marking."
```

```
    INDEX { diffServMarkActId }
    ::= { diffServMarkActTable 1 }

DiffServMarkActEntry ::= SEQUENCE  {
    diffServMarkActId            Unsigned32,
    diffServMarkActDscp          Dscp,
    diffServMarkActStatus        RowStatus
}

diffServMarkActId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Mark Action Id enumerates the Mark Action entry."
    ::= { diffServMarkActEntry 1 }

diffServMarkActDSCP OBJECT-TYPE
    SYNTAX       Dscp
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The DSCP this Action TCB uses for marking/remarking
       traffic with."  Note that if the classifier is working
       from the same DSCP value, no effective change in the
       DSCP results.

       Differentiated Services may result in packet remarking
       both on ingress to a network and on egress, and it is
       quite possible that ingress and egress would occur in
       the same router."
    ::= { diffServMarkActEntry 2 }

diffServMarkActStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of this entry. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServMarkActEntry 3 }


-- Count Action Table
-- Rows of this table is pointed to by diffServAction to
```

```
-- provide detail parameters specific to Count Action.

-- This object allows a configuring system to obtain a
-- unique value for diffServCountActId for purposes of
-- configuration.

diffServCountActUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServCountActUnique object yields a unique new
       value for diffServCountActId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 5 }

diffServCountActTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServCountActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The count action table contains flow specific
       counters.  The entries of this table is meant to be
       referenced by the diffServAction attribute of entries
       in diffServActionTable for diffServActionType = count."
    ::= { diffServTables 5 }

diffServCountActEntry OBJECT-TYPE
    SYNTAX       DiffServCountActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the count action table describes a single
       set of flow specific counters.  This counter entry is
       associated with a traffic flow via the
       diffServActionEntry pointing to it."
    INDEX { diffServActionId, diffServCountActId }
    ::= { diffServCountActTable 1 }

DiffServCountActEntry ::= SEQUENCE  {
    diffServCountActId           Unsigned32,
    diffServCountActOctetsCnt    Counter32,
    diffServCountActPktsCnt      Counter32,
    diffServCountActStatus       RowStatus
}

diffServCountActId OBJECT-TYPE
```

```
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Count Action Id enumerates the Count Action entry."
    ::= { diffServCountActEntry 1 }

diffServCountActOctetsCnt OBJECT-TYPE
    SYNTAX       Counter32
    UNITS        "octets"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of Octets at the Action datapath element.
       Meaning the octets has been classified and possibly
       metered, and prior to any dropping process. This object
       may be used on low speed interfaces, and represents the
       least significant 32 bits of diffServCountActOctetsHCnt
       in the augmented extension.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServCountActEntry 2 }

diffServCountActPktsCnt OBJECT-TYPE
    SYNTAX       Counter32
    UNITS        "packets"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of Packets at the Action datapath element.
       Meaning the packets has been classified and possibly
       metered, and prior to any dropping process. This object
       may be used on low speed interfaces, and represents the
       least significant 32 bits of diffServCountActPktsHCount
       in the augmented extension.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServCountActEntry 3 }

diffServCountActStatus OBJECT-TYPE
    SYNTAX       RowStatus
```

```
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of this entry. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServCountActEntry 4 }

--
-- High Capacity Counter Extension for Count Action Table
--

diffServCountActXTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServCountActXEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains the high capacity counters for the
       counters in the Count Action Table.  These objects are
       all 64-bit versions of the basic counters, having the
       same basic semantics as their 32-bit counterparts, with
       syntax extended to 64 bits."
    AUGMENTS { diffServCountActEntry }
    ::= { diffServAugments 1 }

diffServCountActXEntry OBJECT-TYPE
    SYNTAX       DiffServCountActXEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry containing the 64 bit counters applicable to
       a specific drop action entry."
    ::= { diffServActionXTable 1 }

DiffServCountActXEntry ::= SEQUENCE  {
    diffServCountActOctetsHCnt     Counter64,
    diffServCountActPktsHCnt       Counter64
}

diffServCountActOctetsHCnt OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of Octets at the Action datapath element.
       Meaning the packets has been classified and possibly
```

```
       metered, and prior to any dropping process.  This
       object should be used on high speed interfaces.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServCountActXEntry 1 }

diffServCountActPktsHCnt OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of Packets at the Action datapath element.
       Meaning the packets has been classified and possibly
       metered, and prior to any dropping process.  This
       object should be used on high speed interfaces.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServCountActXEntry 2 }


-- Drop Action Table
-- Rows of this table is pointed to by diffServAction to
-- provide detail parameters specific to an Action Type.

-- Notice the use of diffServQueueId as part of this
-- table's index.  Hence each entry is queue specific.

-- This object allows a configuring system to obtain a
-- unique value for diffServDropActId for purposes of
-- configuration.

diffServDropActUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServDropActUnique object yields a unique new
       value for diffServDropActId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 6 }
```

```
diffServDropActTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServDropActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The drop action table enumerates sets of attributes
       used to represent a drop process.  Each set is normally
       associated with a queue.  The entries of this table is
       meant to be referenced by the diffServAction attribute
       of entries in diffServActionTable.  The entries of this
       table is used for:  Tail Drop Action when
       diffServActionType = tailDrop Random Drop Action when
       diffServActionType = randomDrop Deterministic Drop
       Action when diffServActionType = deterDrop"
    ::= { diffServTables 6 }

diffServDropActEntry OBJECT-TYPE
    SYNTAX       DiffServDropActEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the drop action table describes a single
       drop process's configuration.  For Tail Drop Process:

       For Random Drop Process:  (QMin,PMin) and (QMax,PMax)
       defines the drop probability used for the random drop
       process.  Normally PMin have a value of zero.  QClip
       defines the guaranteed average queue depth, after which
       the drop probability reaches 100%.  For Deterministic
       Drop Process:  QMeasure points to an entry providing
       Queue Measurement needed for the drop process."
    INDEX { diffServQueueId, diffServDropActId }
    ::= { diffServTable 1 }

DiffServDropActEntry ::= SEQUENCE  {
    diffServDropActId                Unsigned32,
    diffServDropActQMin              Unsigned32,
    diffServDropActQMax              Unsigned32,
    diffServDropActQClip             Unsigned32,
    diffServDropActPMin              Unsigned32,
    diffServDropActPMax              Unsigned32,
    diffServDropActPCur              Unsigned32,
    diffServDropActQMeasure          RowPointer,
    diffServDropActOctetsCnt         Counter32,
    diffServDropActPktsCnt           Counter32,
    diffServDropActStatus            RowStatus
}
```

```
diffServDropActId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Drop Action Id enumerates the Drop Action entry."
    ::= { diffServDropActEntry 1 }

diffServDropActQMin OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "Kilo Bits"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "QMin, with PMin, defines the lowerest drop probability
       point for this random drop process.  With PMin=0, the
       queue may drop if a packet is presented to it and the
       average queue depth exceeds QMin."
    ::= { diffServDropActEntry 2 }

diffServDropActQMax OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "Kilo Bits"
    MAX-ACCESS   read-create
    STATUS       current
       "QMax, with PMax, defines the higher point of drop
       probability for random drop process.  For Tail Drop
       Process:  This represents the measure by which the
       queue will drop if a packet is presented to it."
    ::= { diffServDropActEntry 3 }

diffServDropActQClip OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "Kilo Bits"
    MAX-ACCESS   read-create
    STATUS       current
       "The average queue length at which point the drop
       probability reaches 100%."
    ::= { diffServDropActEntry 4 }

diffServDropActPMin OBJECT-TYPE
    SYNTAX       Unsigned32 (0..1000000)
    UNITS        "per-micro-age"
    MAX-ACCESS   read-write
    STATUS       current
       "QMin, with PMin, defines the lowerest drop probability
       point for this random drop process.  With PMin=0, the
```

```
       queue may drop if a packet is presented to it and the
       average queue depth exceeds QMin.  This drop
       probability is expressed in per-micro-age, the value in
       this attribute needs to be divided by 1,000,000 to
       obtain the drop probability between 0 and 1."
    ::= { diffServDropActEntry 5 }

diffServDropActPMax OBJECT-TYPE
    SYNTAX       Unsigned32 (0..1000000)
    UNITS        "per-micro-age"
    MAX-ACCESS   read-write
    STATUS       current
       "QMax and PMax are part of the drop probability
       configuration.  This drop probability is expressed in
       per-micro-age, the value in this attribute needs to be
       divided by 1,000,000 to obtain the drop probability
       between 0 and 1."
    ::= { diffServDropActEntry 6 }

diffServDropActPCur OBJECT-TYPE
    SYNTAX       Unsigned32 (0..1000000)
    UNITS        "per-micro-age"
    MAX-ACCESS   read-only
    STATUS       current
       "The current drop probability.  This drop probability
       is expressed in per-micro-age, the value in this
       attribute needs to be divided by 1,000,000 to obtain
       the drop probability between 0 and 1."
    ::= { diffServDropActEntry 7 }

diffServDropActQMeasure OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
       "Points to an entry in the diffServQMeasureTable for
       queue information required by the drop process."
    ::= { diffServDropActEntry 8 }

diffServDropActOctetsCnt OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of octets that have been dropped by a drop
       process.  On high speed devices, this object implements
       the least significant 32 bits of
       diffServDropActOctetsHCnt.
```

```
       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServDropActEntry 9 }

diffServDropActPktsCnt OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of packets that have been dropped by a drop
       process.  On high speed devices, this object implements
       the least significant 32 bits of
       diffServDropActPktsHCnt.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServDropActEntry 10 }

diffServDropActStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of this entry. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServDropActEntry 11 }

--
-- High Capacity Counter Extension for Drop Action Table
--

diffServDropActXTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServDropActXEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains the high capacity counters for the
       counters in the Drop Action Table.  These objects are
       all 64 bit versions of the basic counters, having the
       same basic semantics as their 32-bit counterparts, with
       syntax extended to 64 bits."
```

```
    AUGMENTS { diffServDropActEntry }
    ::= { diffServAugments 2 }

diffServDropActXEntry OBJECT-TYPE
    SYNTAX       DiffServDropActXEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry containing the 64 bit counters applicable to
       a specific drop action entry."
    ::= { diffServDropActXTable 1 }

DiffServDropActXEntry ::= SEQUENCE  {
    diffServDropActOctetsHCnt       Counter64,
    diffServDropActPktsHCnt         Counter64
}

diffServDropActOctetsHCnt OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of octets that have been dropped by a drop
       process.  This object should be used on high speed
       interfaces.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServDropActXEntry 1 }

diffServDropActPktsHCnt OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of packets that have been dropped by a drop
       process.  This object should be used on high speed
       interfaces.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       ifCounterDiscontinuityTime."
    ::= { diffServDropActXEntry 2 }
```

```
-- This object allows a configuring system to obtain a
-- unique value for diffServActionId for purposes of
-- configuration

diffServActionUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServActionUnique object yields a unique new
       value for diffServActionId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 7 }

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

diffServActionTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServActionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Action Table enumerates actions that can be
       performed to a stream of traffic.  Multiple Actions can
       be concatenated.  For example, marking of a stream of
       traffic exiting a meter (conforming or non-conforming),
       then perform a drop process with dropped traffic counts
       maintained.  Notice counting is considered as a type of
       action.  The set of flow specific counters in the Count
       Action Table maintains statistics for a flow that
       arrives to this Action datapath element.  This count is
       always taken before any drop processing is performed."
    ::= { diffServTables 7 }

diffServActionEntry OBJECT-TYPE
    SYNTAX       DiffServActionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the action table describes the actions
       applied to traffic exiting a given meter."
    INDEX { ifIndex, diffServInterfaceDirection,
            diffServActionId }
    ::= { diffServActionTable 1 }

DiffServActionEntry ::= SEQUENCE  {
    diffServActionId                Unsigned32,
```

```
    diffServActionNext              RowPointer,
    diffServActionType              INTEGER,
    diffServAction                  RowPointer,
    diffServActionStatus            RowStatus
}

diffServActionId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Action Id enumerates the Action entry."
    ::= { diffServActionEntry 1 }

diffServActionNext OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The Next pointer indicates the next datapath element
       to handle the traffic.  For example, a queue datapath
       element.  The value zeroDotZero in this variable
       indicates no further DiffServ treatment is performed on
       this flow by the current interface for this interface
       direction."
    DEFVAL      { zeroDotZero }
    ::= { diffServActionEntry 2 }

diffServActionType OBJECT-TYPE
    SYNTAX       INTEGER {
                     other(1),      -- types not specified here.
                     mark(2),       -- mark or remark
                     count(3),      -- count
                     alwaysDrop(4), -- disallow traffic
                     tailDrop(5),   -- fix queue size Drop
                     randomDrop(6), -- Random Drop
                     deterDrop(7)   -- Deterministic Drop
                 }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "Indicates the type of action diffServAction points
       to."
    ::= { diffServActionEntry 3 }

diffServAction OBJECT-TYPE
    SYNTAX       RowPointer
```

```
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "Points to a row in a Action Type Table that provides
       all the parameters for the type of action indicated in
       diffServActionType.  Can also points to a row in some
       other MIB to provide some proprietary action type."
    ::= { diffServActionEntry 4 }

diffServActionStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of a meter. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServActionEntry 5 }
```

```
--
-- Queue Set Table
-- The Queue Set Table is used for organizing queues
-- defined in the Queue Table into Queue Sets, with
-- queue scheduling defined in the queue set entry.
-- Queue Set Table provides flexibility in queue
-- organization and allows more complex hierarchical
-- scheduling algorithms be defined.  For example,
-- multiple scheduling algorithms, each with multiple
-- queues, used on the same logical/physical interface.
--

-- This object allows a configuring system to obtain a
-- unique value for diffServQSetId for purposes of
-- configuration

diffServQSetUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServQSetUnique object yields a unique new
       value for diffServQSetId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 8 }


diffServQSetTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServQSetEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Set Table enumerates the queue sets.  Queue
       Sets are used to organize queues based on their
       scheduling algorithms.  Multiple sheduling algorithms
       can be used, with each algorithm described by one Queue
       Set Entry.  Multiple instances of a single sheduling
       algorithm, each with different scheduling parameters
       can also be expressed, each described by its own Queue
       Set Entry.  Relationships between Queue Sets are used
       to build scheduling algorithm hierarchies.  For
       example, a weighed fair queueing queue set can be a
       part of a priority queueing queue set, having the
       weighed fair queueing queue set be a branch of the
       priority queueing queue set.  More complex hierarchies
       can also be expressed using this mechanism."
    ::= { diffServTables 8 }
```

```
diffServQSetEntry OBJECT-TYPE
    SYNTAX       DiffServQSetEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the Queue Set Table describes a single
       queue set."
    INDEX { ifIndex, diffServInterfaceDirection,
            diffServQSetId }
    ::= { diffServQSetTable 1 }

DiffServQSetEntry ::= SEQUENCE  {
    diffServQSetId                   Unsigned32,
    diffServQSetParentId             Unsigned32,
    diffServQSetWeight               Unsigned32,
    diffServQSetMethod               INTEGER,
    diffServQSetRateUnit             INTEGER,
    diffServQSetMinRate              Unsigned32,
    diffServQSetMaxRate              Unsigned32,
    diffServQSetNext                 RowPointer,
    diffServQSetStatus               RowStatus
}

diffServQSetId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Set Id enumerates the Queue Set entry."
    ::= { diffServQSetEntry 1 }

diffServQSetParentId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Set Parent Id allows the formation of
       hierarchical relationships between scheduling
       algorithms."
    ::= { diffServQSetEntry 2 }

diffServQSetWeight OBJECT-TYPE
    SYNTAX       Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "Used with QSetParentId in hierarchical scheduling
```

```
       setup.  QSetWeight represent the weight of all queues
       within this queue set, with respect to queues in other
       queue sets in hierarchical scheduling.  For example,
       this queue set represents the weighed fair queueing
       scheduling amongst all the queues in this queue set.
       This set of weighted fair queueing queues as a whole
       belongs to a priority queueing queue set.  QSetWeight
       determines this queue set's priority/weight in the
       parent queue set's priority queueing scheduling
       algorithm.  There can be more than one weighed fair
       queueing queue sets belonging to the same priority
       queueing parent queue set."
    ::= { diffServQSetEntry 3 }

diffServQSetMethod OBJECT-TYPE
    SYNTAX       INTEGER {
                     other(1), -- not listed here
                     pq(2),    -- Priority Queueing
                     wfq(3),   -- Weighed Fair Queueing
                     wrr(4)    -- Weighed Round Robin
                 }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The scheduling algorithm used by queues in this queue
       set."
    ::= { diffServQSetEntry 4 }

diffServQSetRateUnit OBJECT-TYPE
    SYNTAX       INTEGER {
                     kbps(1),    -- kilo bits per second
                     packets(2)  -- packets per second
                 }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The unit of measure for the MinRate and MaxRate
       attributes.  The packet unit allows packet fair
       algorithms in addition to bit fair algorithms."
    ::= { diffServQSetEntry 5 }

diffServQSetMinRate OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        "KBPS"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
          "The minimum rate for the whole queue set.  If the
          value is zero, then there is effectively no minimum
          rate.  If the value is non-zero, the queue set will
          seek to assure this class of traffic at least this
          rate."
       ::= { diffServQSetEntry 6 }

   diffServQSetMaxRate OBJECT-TYPE
       SYNTAX       Unsigned32
       UNITS        "KBPS"
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "The maximum rate for the whole queue set.  If the
          value is zero, then there is effectively no maximum
          rate.  If the value is non-zero, the queue set will
          seek to assure this class of traffic at most this
          rate."
       ::= { diffServQSetEntry 7 }

   diffServQSetNext OBJECT-TYPE
       SYNTAX       RowPointer
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "Selects the next data path component, which can be
          another Queue Set.  One usage of multiple serial Queue
          Sets is for Class Base Queueing (CBQ).  The value
          zeroDotZero in this variable indicates no further
          DiffServ treatment is performed on this flow by the
          current interface for this interface direction.  For
          example, for an inbound interface the value zeroDotZero
          indicates that the packet flow has now completed
          inbound DiffServ treatment and should be forwarded on
          to the appropriate outbound interface."
       DEFVAL       { zeroDotZero }
       ::= { diffServQSetEntry 8 }

   diffServQSetStatus OBJECT-TYPE
       SYNTAX       RowStatus
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "The RowStatus variable controls the activation,
          deactivation, or deletion of a queue. Any writable
          variable may be modified whether the row is active or
          notInService."
```

```
    ::= { diffServQSetEntry 9 }
```

```
--
-- Queue Table
--

-- This object allows a configuring system to obtain a
-- unique value for diffServQId for purposes of
-- configuration.

diffServQUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServQUnique object yields a unique new value
       for diffServQId when read and subsequently set. This
       value must be tested for uniqueness."
    ::= { diffServObjects 9 }

-- The Queue Table allows us to describe individual queues

diffServQTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServQEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Table enumerates the queues on an
       interface."
    ::= { diffServTables 9 }

diffServQEntry OBJECT-TYPE
    SYNTAX       DiffServQEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the Queue Table describes a single FIFO
       queue."
    INDEX { diffServQQSetId, diffServQId }
    ::= { diffServQTable 1 }

DiffServQEntry ::= SEQUENCE  {
    diffServQId                         Unsigned32,
    diffServQQSetId                     Unsigned32,
    diffServQSchedulerWeight            Unsigned32,
    diffServQStatus                     RowStatus
}

diffServQId OBJECT-TYPE
```

```
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Id enumerates the Queue entry."
    ::= { diffServQEntry 1 }

diffServQQSetId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Indicates the Queue Set this queue is part of."
    ::= { diffServQEntry 2 }

diffServQSchedulerWeight OBJECT-TYPE
    SYNTAX       Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The weight or priority of the queue, depending on the
       scheduling method used.  Notice only the weight of the
       queue is used, instead of the rate.  The rate can be
       derived based on the rate of the queue set.  This is to
       facilitate changing link speed and/or changing
       scheduling method without reconfiguring the queues."
    ::= { diffServQEntry 3 }

diffServQStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of a queue. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServQEntry 4 }

-- Queue Measurement Table

-- This object allows a configuring system to obtain a
-- unique value for diffServQMeasureId for purposes of
-- configuration.

diffServQMeasureUnique OBJECT-TYPE
    SYNTAX       TestAndIncr
```

```
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The diffServQMeasureUnique object yields a unique new
       value for diffServQMeasureId when read and subsequently
       set. This value must be tested for uniqueness."
    ::= { diffServObjects 10 }

diffServQMeasureTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServQMeasureEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Measurement Table contains entries
       describing the state of queues, this include states for
       implementing traffic treatment algorithms.  Notice
       multiple queue measurement entries for the same queue
       is allowed."
    ::= { diffServTables 10 }

diffServQMeasureEntry OBJECT-TYPE
    SYNTAX       DiffServQMeasureEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the Queue Measure Table describes a single
       set of measurement for a specific queue."
    INDEX { diffServQMeasureQId, diffServQMeasureId }
    ::= { diffServQMeasureTable 1 }

DiffServQMeasureEntry ::= SEQUENCE  {
    diffServQMeasureId              Unsigned32,
    diffServQMeasureQId             Unsigned32,
    diffServQMeasureAvgSampleInt    Unsigned32,
    diffServQMeasureAvgWeightExp    Unsigned32,
    diffServQMeasureAvgWeightMan    Unsigned32,
    diffServQMeasureQAverage        Unsigned32,
    diffServQMeasureStatus          RowStatus
}

diffServQMeasureId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The Queue Measure Id enumerates the Queue Measure
       entry."
```

```
    ::= { diffServQMeasureEntry 1 }

diffServQMeasureQId OBJECT-TYPE
    SYNTAX       Unsigned32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Indicates the queue this measurement is associated
       with."
    ::= { diffServQMeasureEntry 2 }

diffServQMeasureAvgSampleInt OBJECT-TYPE
    SYNTAX       Unsigned32
    UNIT         millisecond
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The sampling interval for queue average calculation,
       in milliseconds.  For queue sampling based on packet
       enqueueing or dequeueing intervals, this attribute
       should contain the value of zero."
    ::= { diffServQMeasureEntry 3 }

diffServQMeasureAvgWeightExp OBJECT-TYPE
    SYNTAX       Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
       "The exponent part of weight (in real number format)
       for queue average calculation.  This is a base 10
       exponent, with the attribute representing a negative
       value.  For example, with 8 in this attribute meaning
       10 to the power of -8.  An 8 bit value here will be
       sufficient."
    ::= { diffServQMeasureEntry 4 }

diffServQMeasureAvgWeightMan OBJECT-TYPE
    SYNTAX       Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
       "The mantissa part of weight (in real number format)
       for queue average calculation.  Always a positive
       number.  Need 16 bits of accuracy."
    ::= { diffServQMeasureEntry 5 }

diffServQMeasureQAverage OBJECT-TYPE
    SYNTAX       Unsigned32
    UNITS        kilo-bits
```

```
    MAX-ACCESS   read-only
    STATUS       current
       "The current queue average in kilo bits."
    ::= { diffServQMeasureEntry 6 }

diffServQMeasureStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The RowStatus variable controls the activation,
       deactivation, or deletion of a queue. Any writable
       variable may be modified whether the row is active or
       notInService."
    ::= { diffServQMeasureEntry 7 }
```

```
          -- MIB Compliance statements. Three variations of
          -- compliance are described, for optical, LAN, and low speed
          -- interfaces.  The difference is the implementation of
          -- diffServActionOctetsHCnt, diffServActionPktsHCnt
          -- and diffServCountActOctetsHCnt, diffServCountActPktsHCnt.

          diffServMIBCompliances OBJECT IDENTIFIER ::= { diffServMIBConformance
1 }
          diffServMIBGroups      OBJECT IDENTIFIER ::= { diffServMIBConformance
2 }

          diffServMIBCompliance MODULE-COMPLIANCE
              STATUS current
              DESCRIPTION
                 "This MIB may be implemented as a read-only or as a
                 read-create MIB. As a result, it may be used for
                 monitoring or for configuration.

                 Standard compliance implies that the implementation
                 complies for interfaces for which an interface's octet
                 counter might wrap at most once an hour, which by the
                 IFMIB's convention applies to interfaces under 20 MBPS.
                 It thus applies to any device which might implement a
                 low speed serial line, Ethernet, Token Ring."
              MODULE -- This Module
              MANDATORY-GROUPS {
                  diffServMIBClassifierGroup, diffServMIBMeterGroup,
                  diffServMIBQueueGroup, diffServMIBActionGroup

                  -- note that diffServMIBHCCounterGroup is
                  -- mandatory for medium and high speed interfaces

                  -- note that diffServMIBVHCCounterGroup is
                  -- mandatory for high speed interfaces

                  -- note that the diffServMIBStaticGroup is
                  -- mandatory for implementations that implement a
                  -- read-write or read-create mode.
              }

              GROUP diffServMIBHCCounterGroup
              DESCRIPTION
                 "This group is mandatory for those network interfaces
                 for which the value of the corresponding instance of
                 ifSpeed is greater than 20,000,000 bits/second."

              GROUP diffServMIBVHCCounterGroup
              DESCRIPTION
                 "This group is mandatory for those network interfaces
```

```
       for which the value of the corresponding instance of
       ifSpeed is greater than 650,000,000 bits/second."

    OBJECT diffServClassifierMatchObject
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierSequence
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterInterval
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterBurstSize
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterFailNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterSucceedNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."
```

```
    OBJECT diffServActionNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionDSCP
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionMinThreshold
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionMaxThreshold
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionDropPolicy
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMinimumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMaximumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueuePriority
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueNextTCB
    MIN-ACCESS read-only
```

```
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."
    ::= { diffServMIBCompliances 1 }
```

```
diffServMIBVHCCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
       "This MIB may be implemented as a read-only or as a
       read-create MIB. As a result, it may be used for
       monitoring or for configuration.

       Very High Speed compliance implies that the
       implementation complies for interfaces for which an
       interface's packet or octet counters might wrap more
       than once an hour, which by the IFMIB's convention
       applies to interfaces over 650 MBPS, or OC-12."
    MODULE -- This Module
    MANDATORY-GROUPS {
        diffServMIBClassifierGroup, diffServMIBMeterGroup,
        diffServMIBQueueGroup, diffServMIBHCCounterGroup,
        diffServMIBVHCCounterGroup, diffServMIBActionGroup

        -- note that the diffServMIBStaticGroup is
        -- mandatory for implementations that implement a
        -- read-write or read-create mode.
    }


    OBJECT diffServClassifierMatchObject
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierSequence
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServTBMeterInterval
    MIN-ACCESS read-only
    DESCRIPTION
```

```
           "Write access is not required."

        OBJECT diffServTBMeterBurstSize
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServTBMeterFailNext
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServTBMeterSucceedNext
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServTBMeterStatus
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServActionNext
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServActionDSCP
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServActionMinThreshold
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServActionMaxThreshold
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."

        OBJECT diffServActionDropPolicy
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required."
```

```
    OBJECT diffServActionStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMinimumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMaximumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueuePriority
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueNextTCB
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."
    ::= { diffServMIBCompliances 2 }
```

```
diffServMIBHCCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
       "This MIB may be implemented as a read-only or as a
       read-create MIB. As a result, it may be used for
       monitoring or for configuration.

       High Speed compliance implies that the implementation
       complies for interfaces for which an interface's octet
       counters might wrap more than once an hour, which by
       the IFMIB's convention applies to interfaces over 20
       MBPS, but under 650 MBPS.  It thus applies to devices
       which implement a 100 MBPS Ethernet, FDDI, E3, DS3, or
       SONET/SDH interface up to OC-12."
    MODULE -- This Module
    MANDATORY-GROUPS {
        diffServMIBClassifierGroup, diffServMIBMeterGroup,
        diffServMIBQueueGroup, diffServMIBHCCounterGroup,
        diffServMIBActionGroup

        -- note that diffServMIBVHCCounterGroup is
        -- mandatory for high speed interfaces

        -- note that the diffServMIBStaticGroup is
        -- mandatory for implementations that implement a
        -- read-write or read-create mode.
    }

    GROUP diffServMIBVHCCounterGroup
    DESCRIPTION
       "This group is mandatory for those network interfaces
       for which the value of the corresponding instance of
       ifSpeed is greater than 650,000,000 bits/second."

    OBJECT diffServClassifierMatchObject
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierNext
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServClassifierSequence
    MIN-ACCESS read-only
    DESCRIPTION
```

```
           "Write access is not required."

       OBJECT diffServClassifierStatus
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServTBMeterInterval
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServTBMeterBurstSize
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServTBMeterFailNext
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServTBMeterSucceedNext
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServTBMeterStatus
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServActionNext
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServActionDSCP
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."

       OBJECT diffServActionMinThreshold
       MIN-ACCESS read-only
       DESCRIPTION
          "Write access is not required."
```

```
    OBJECT diffServActionMaxThreshold
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionDropPolicy
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServActionStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMinimumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueMaximumRate
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueuePriority
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueNextTCB
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."

    OBJECT diffServQueueStatus
    MIN-ACCESS read-only
    DESCRIPTION
       "Write access is not required."
    ::= { diffServMIBCompliances 3 }
```

```
diffServMIBClassifierGroup OBJECT-GROUP
    OBJECTS {
        diffServAggregateDSCP,
        diffServClassifierMatchObject,
        diffServClassifierNext,
        diffServClassifierSequence,
        diffServClassifierStatus
    }
    STATUS current
    DESCRIPTION
       "The Classifier Group defines the MIB Objects that
       describe a classifier."
    ::= { diffServMIBGroups 1 }

diffServMIBMeterGroup OBJECT-GROUP
    OBJECTS {
        diffServTBMeterInterval, diffServTBMeterBurstSize,
        diffServTBMeterSucceedNext, diffServTBMeterFailNext,
        diffServTBMeterStatus
    }
    STATUS current
    DESCRIPTION
       "The Meter Group defines the objects used in describing
       a meter."
    ::= { diffServMIBGroups 2 }

diffServMIBActionGroup OBJECT-GROUP
    OBJECTS {
        diffServActionDropPolicy,
        diffServActionRandomDrops,
        diffServActionTailDrops,
        diffServActionMinThreshold,
        diffServActionMaxThreshold, diffServActionDSCP,
        diffServActionNext,
        diffServActionConformingPackets,
        diffServActionConformingOctets,
        diffServActionStatus
    }
    STATUS current
    DESCRIPTION
       "The Action Group defines the objects used in
       describing an action."
    ::= { diffServMIBGroups 3 }

diffServMIBHCCounterGroup OBJECT-GROUP
    OBJECTS {
        diffServActionHCConformingOctets
```

```
    }
    STATUS current
    DESCRIPTION
       "At 20,000,000 bits per second or greater, the number
       of octets a given class may count can overflow a 32 bit
       counter in under an hour.  Therefore, by convention
       established in the IFMIB, the 64 bit counter must be
       implemented as well."
    ::= { diffServMIBGroups 4 }

diffServMIBVHCCounterGroup OBJECT-GROUP
    OBJECTS {
        diffServActionHCConformingPackets,
        diffServActionHCRandomDrops,
        diffServActionHCTailDrops
    }
    STATUS current
    DESCRIPTION
       "At 650,000,000 bits per second or greater, the number
       of packets a given class may count can overflow a 32
       bit counter in under an hour.  Therefore, by convention
       established in the IFMIB, the 64 bit counter must be
       implemented as well."
    ::= { diffServMIBGroups 5 }

diffServMIBQueueGroup OBJECT-GROUP
    OBJECTS {
        diffServQueueMinimumRate,
        diffServQueueMaximumRate,
        diffServQueuePriority, diffServQueueStatus,
        diffServQueueNextTCB
    }
    STATUS current
    DESCRIPTION
       "The Queue Group contains the objects that describe an
       interface's queues."
    ::= { diffServMIBGroups 6 }

diffServMIBStaticGroup OBJECT-GROUP
    OBJECTS {
        diffServClassifierUnique, diffServTBMeterUnique,
        diffServQueueUnique, diffServActionUnique
    }
    STATUS current
    DESCRIPTION
       "The Static Group contains scalar objects used in
       creating unique enumerations for classifiers, meters,
```

```
           and queues."
       ::= { diffServMIBGroups 7 }
   END
```

5. Acknowledgments

This MIB has been developed with active involvement from a number of sources, but most notably Yoram Bernet, Steve Blake, Brian Carpenter, Kwok Chan, Dave Durham, Jeremy Greene, Roch Guerin, Scott Hahn, Keith McCloghrie, Kathleen Nichols, Ping Pan, Andrew Smith, and Bert Wijnen.

6. Security Considerations

It is clear that this MIB is potentially useful for configuration, and anything that can be configured can be misconfigured, with potentially disastrous effect.

At this writing, no security holes have been identified beyond those that SNMP Security is itself intended to address. These relate to primarily controlled access to sensitive information and the ability to configure a device - or which might result from operator error, which is beyond the scope of any security architecture.

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The use of SNMP Version 3 is recommended over prior versions, for configuration control, as its security model is improved.

There are a number of managed objects in this MIB 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 ingress or egress traffic. There are no objects which are sensitive in their own right, such as passwords or monetary amounts.

It may be important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

7. References

[1] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April 1999

[2] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, STD 16, Performance Systems International, Hughes LAN Systems, May 1990

[3] Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, STD 16, Performance Systems International, Hughes LAN Systems, March 1991

[4] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991

[5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", RFC 2578, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999

[6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", RFC 2579, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999

[7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", RFC 2580, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999

[8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, STD 15, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, SNMP

Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, SNMP Research, Inc., Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April 1999

[12] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, IBM T. J. Watson Research, April 1999

[13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[14] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, SNMP Research, Inc., Secure Computing Corporation, Cisco Systems, April 1999

[15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, IBM T. J. Watson Research, BMC Software, Inc., Cisco Systems, Inc., April 1999

[16] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, SNMP Research, Inc., TIS Labs at Network Associates, Inc., Ericsson, Cisco Systems, April 1999

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

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

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

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

[Model]
Bernet et al, "A Conceptual Model for Diffserv Routers", March 2000, draft-ietf-diffserv-model-02.txt

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

[DSPIB]
M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn, A. Smith "Differentiated Services Policy Information Base", March 2000, draft-ietf-diffserv-pib-00.txt

[INETADDRESS]
Daniele, M., Haberman, B., Routhier, S., Schoenwaelder, J. "Textual Conventions for Internet Network Addresses.", February 17, 2000, draft-ops-endpoint-mib-07.txt

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

8. **Authors'** Addresses:

Fred Baker
519 Lado Drive
Santa Barbara, California 93111
fred@cisco.com

Kwok Ho Chan
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821
khchan@nortelnetworks.com

Andrew Smith
Extreme Networks
3585 Monroe Street
Santa Clara, CA 95051
USA
andrew@extremenetworks.com