

Network Working Group
Request for Comments: 3289
Category: Standards Track

F. Baker
Cisco System
K. Chan
Nortel Networks
A. Smith
Harbour Networks
May 2002

Management Information Base for the
Differentiated Services Architecture

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This memo describes an SMIV2 (Structure of Management Information version 2) MIB for a device implementing the Differentiated Services Architecture. It may be used both for monitoring and configuration of a router or switch capable of Differentiated Services functionality.

Table of Contents

1 The SNMP Management Framework	3
2 Relationship to other working group documents	4
2.1 Relationship to the Informal Management Model for Differentiated Services Router	4
2.2 Relationship to other MIBs and Policy Management	5
3 MIB Overview	6
3.1 Processing Path	7
3.1.1 diffServDataPathTable - The Data Path Table	7
3.2 Classifier	7
3.2.1 diffServClfrElementTable - The Classifier Element Table ...	8
3.2.2 diffServMultiFieldClfrTable - The Multi-field Classifier Table	9
3.3 Metering Traffic	10
3.3.1 diffServMeterTable - The Meter Table	11

Baker, et. al. Standards Track [Page 1]

RFC 3289 Differentiated Services MIB May 2002

3.3.2 diffServTBParamTable - The Token Bucket Parameters Table...	11
3.4 Actions applied to packets	12
3.4.1 diffServActionTable - The Action Table	12
3.4.2 diffServCountActTable - The Count Action Table	12
3.4.3 diffServDscpMarkActTable - The Mark Action Table	13
3.4.4 diffServAlgDropTable - The Algorithmic Drop Table	13
3.4.5 diffServRandomDropTable - The Random Drop Parameters Table	14
3.5 Queuing and Scheduling of Packets	16
3.5.1 diffServQTable - The Class or Queue Table	16
3.5.2 diffServSchedulerTable - The Scheduler Table	16
3.5.3 diffServMinRateTable - The Minimum Rate Table	16
3.5.4 diffServMaxRateTable - The Maximum Rate Table	17
3.5.5 Using queues and schedulers together	17
3.6 Example configuration for AF and EF	20
3.6.1 AF and EF Ingress Interface Configuration	20
3.6.1.1 Classification In The Example	22
3.6.1.2 AF Implementation On an Ingress Edge Interface	22
3.6.1.2.1 AF Metering On an Ingress Edge Interface	22
3.6.1.2.2 AF Actions On an Ingress Edge Interface	23

of the MIB. The RowPointers, whether "next" or "specific", may point to Entries defined in other MIB modules. For example, the only type of meter defined in this MIB is a token bucket meter; if another type of meter is required, another MIB could be defined describing that type of meter, and diffServMeterSpecific could point to it. Similarly, if a new action is required, the "next" pointer of the previous functional datapath element could point to an Entry defined in another MIB, public or proprietary.

Baker, et. al. Standards Track [Page 6]
RFC 3289 Differentiated Services MIB May 2002

3.1. Processing Path

An interface has an ingress and an egress direction, and will generally have a different policy in each direction. As traffic enters an edge interface, it may be classified, metered, counted, and marked. Traffic leaving the same interface might be remarked according to the contract with the next network, queued to manage the bandwidth, and so on. As [MODEL] points out, the functional datapath elements used on ingress and egress are of the same type, but may be structured in very different ways to implement the relevant policies.

3.1.1. diffServDataPathTable - The Data Path Table

Therefore, when traffic arrives at an ingress or egress interface, the first step in applying the policy is determining what policy applies. This MIB does that by providing a table of pointers to the first functional data path element, indexed by interface and direction on that interface. The content of the diffServDataPathEntry is a single RowPointer, which points to that functional data path element.

When diffServDataPathStart in a direction on an interface is undefined or is set to zeroDotZero, the implication is that there is no specific policy to apply.

3.2. Classifier

Classifiers are used to differentiate among types of traffic. In the Differentiated Services architecture, one usually discusses a behavior aggregate identified by the application of one or more Differentiated Services Code Points (DSCPs). However, especially at network edges (which include hosts and first hop routers serving hosts), traffic may arrive unmarked or the marks may not be trusted. In these cases, one applies a Multi-Field Classifier, which may select an aggregate as coarse as "all traffic", as fine as a specific microflow identified by IP Addresses, IP Protocol, and TCP or UDP ports, or variety of slices in between.

Classifiers can be simple or complex. In a core interface, one would expect to find simple behavior aggregate classification to be used. However, in an edge interface, one might first ask what application is being used, meter the arriving traffic, and then apply various policies to the non-conforming traffic depending on the Autonomous System number advertising the destination address. To accomplish such a thing, traffic must be classified, metered, and then reclassified. To this end, the MIB defines separate classifiers, which may be applied at any point in processing, and may have different content as needed.

Baker, et. al. Standards Track [Page 7]
RFC 3289 Differentiated Services MIB May 2002

The MIB also allows for ambiguous classification in a structured fashion. In the end, traffic classification must be unambiguous; one must know for certain what policy to apply to any given packet.

Since port ranges, IP prefixes, or "any" are defined in each case, it is clear that a wide variety of filters can be constructed. The Differentiated Services Behavior Aggregate filter is a special case of this filter, in which only the DSCP is specified.

Other MIB modules may define similar filters in the same way. For example, a filter for Ethernet information might define source and destination MAC addresses of "any", Ethernet Packet Type, IEEE 802.2 SAPs, and IEEE 802.1 priorities. A filter related to policy routing might be structured like the diffServMultiFieldClfrTable, but contain the BGP Communities of the source and destination prefix rather than the prefix itself, meaning "any prefix in this community". For such a filter, a table similar to diffServMultiFieldClfrTable is constructed, and diffServClfrElementSpecific is configured to point to it.

When the MIB is used for configuration, diffServMultiFieldClfrNextFree always contains a legal value for diffServMultiFieldClfrId that is not currently used in the system's configuration.

3.3. Metering Traffic

As discussed in [MODEL], a meter and a shaper are functions that operate on opposing ends of a link. A shaper schedules traffic for transmission at specific times in order to approximate a particular line speed or combination of line speeds. In its simplest form, if the traffic stream contains constant sized packets, it might transmit one packet per unit time to build the equivalent of a CBR circuit. However, various factors intervene to make the approximation inexact; multiple classes of traffic may occasionally schedule their traffic at the same time, the variable length nature of IP traffic may introduce variation, and factors in the link or physical layer may change traffic timing. A meter integrates the arrival rate of traffic and determines whether the shaper at the far end was correctly applied, or whether the behavior of the application in question is naturally close enough to such behavior to be acceptable under a given policy.

A common type of meter is a Token Bucket meter, such as [srTCM] or [trTCM]. This type of meter assumes the use of a shaper at a previous node; applications which send at a constant rate when sending may conform if the token bucket is properly specified. It specifies the acceptable arrival rate and quantifies the acceptable variability, often by specifying a burst size or an interval; since rate = quantity/time, specifying any two of those parameters implies the third, and a large interval provides for a forgiving system. Multiple rates may be specified, as in AF, such that a subset of the traffic (up to one rate) is accepted with one set of guarantees, and traffic in excess of that but below another rate has a different set of guarantees. Other types of meters exist as well.

One use of a meter is when a service provider sells at most, a certain bit rate to one of its customers, and wants to drop the excess. In such a case, the fractal nature of normal Internet traffic must be reflected in large burst intervals, as TCP frequently sends packet pairs or larger bursts, and responds poorly when more than one packet in a round trip interval is dropped. Applications like FTP contain the effect by simply staying below the target bit rate; this type of configuration very adversely affects transaction applications like HTTP, however. Another use of a meter is in the AF specification, in which excess traffic is marked with a related DSCP and subjected to slightly more active queue depth management. The

application is not sharply limited to a contracted rate in such a case, but can be readily contained should its traffic create a burden.

3.3.1. diffServMeterTable - The Meter Table

The Meter Table is a structural table, specifying a specific functional data path element. Its entry consists essentially of three RowPointers - a "succeed" pointer, for traffic conforming to the meter, a "fail" pointer, for traffic not conforming to the meter, and a "specific" pointer, to identify the parameters in question. This structure is a bow to SNMP's limitations; it would be better to have a structure with N rates and N+1 "next" pointers, with a single algorithm specified. In this case, multiple meter entries connected by the "fail" link are understood to contain the parameters for a specified algorithm, and traffic conforming to a given rate follows their "succeed" paths. Within this MIB, only Token Bucket parameters are specified; other varieties of meters may be designed in other MIB modules.

When the MIB is used for configuration, diffServMeterNextFree always contains a legal value for diffServMeterId that is not currently used in the system's configuration.

3.3.2. diffServTBParamTable - The Token Bucket Parameters Table

The Token Bucket Parameters Table is a set of parameters that define a Token Bucket Meter. As a parameter, a token bucket may be specified once and applied to many interfaces, using diffServMeterSpecific. Specifically, several modes of [srTCM] and [trTCM] are addressed. Other varieties of meters may be specified in other MIB modules.

In general, if a Token Bucket has N rates, it has N+1 potential outcomes - the traffic stream is slower than and therefore conforms to all of the rates, it fails the first few but is slower than and therefore conforms to the higher rates, or it fails all of them. As such, multi-rate meters should specify those rates in monotonically increasing order, passing through the diffServMeterFailNext from more committed to more excess rates, and finally falling through diffServMeterFailNext to the set of actions that apply to traffic which conforms to none of the specified rates. diffServTBParamType in the first entry indicates the algorithm being used. At each rate, diffServTBParamRate is derivable from diffServTBParamBurstSize and diffServTBParamInterval; a superior implementation will allow the

configuration of any two of diffServTBParamRate, diffServTBParamBurstSize, and diffServTBParamInterval, and respond with the appropriate error code if all three are specified but are not mathematically related.

When the MIB is used for configuration, diffServTBParamNextFree always contains a legal value for diffServTBParamId that is not currently used in the system's configuration.

3.4. Actions applied to packets

"Actions" are the things a differentiated services interface PHB may do to a packet in transit. At a minimum, such a policy might calculate statistics on traffic in various configured classes, mark it with a DSCP, drop it, or enqueue it before passing it on for other processing.

Actions are composed of a structural element, the diffServActionTable, and various component action entries that may be

applied. In the case of the Algorithmic Dropper, an additional parameter table may be specified to control Active Queue Management, as defined in [RED93] and other AQM specifications.

3.4.1. diffServActionTable - The Action Table

The action table identifies sequences of actions to be applied to a packet. Successive actions are chained through diffServActionNext, ultimately resulting in zeroDotZero (indicating that the policy is complete), a pointer to a queue, or a pointer to some other functional data path element.

When the MIB is used for configuration, diffServActionNextFree always contains a legal value for diffServActionId that is not currently used in the system's configuration.

3.4.2. diffServCountActTable - The Count Action Table

The count action accumulates statistics pertaining to traffic passing through a given path through the policy. It is intended to be useful for usage-based billing, for statistical studies, or for analysis of the behavior of a policy in a given network. The objects in the Count Action are various counters and a discontinuity time. The counters display the number of packets and bytes encountered on the path since the discontinuity time. They share the same discontinuity time, which is the discontinuity time of the interface or agent.

Baker, et. al.

Standards Track

[Page 12]

RFC 3289

Differentiated Services MIB

May 2002

The designers of this MIB expect that every path through a policy should have a corresponding counter. In early versions, it was impossible to configure an action without implementing a counter, although the current design makes them in effect the network manager's option, as a result of making actions consistent in structure and extensibility. The assurance of proper debugging and accounting is therefore left with the policy designer.

When the MIB is used for configuration, diffServCountActNextFree always contains a legal value for diffServCountActId that is not currently used in the system's configuration.

3.4.3. diffServDscpMarkActTable - The Mark Action Table

The Mark Action table is an unusual table, both in SNMP and in this MIB. It might be viewed not so much as an array of single-object entries as an array of OBJECT-IDENTIFIER conventions, as the OID for a diffServDscpMarkActDscp instance conveys all of the necessary information: packets are to be marked with the requisite DSCP.

As such, contrary to common practice, the index for the table is read-only, and is both the Entry's index and its only value.

3.4.4. diffServAlgDropTable - The Algorithmic Drop Table

The Algorithmic Drop Table identifies a dropping algorithm, drops packets, and counts the drops. Classified as an action, it is in effect a method which applies a packet to a queue, and may modify either. When the algorithm is "always drop", this is simple; when the algorithm calls for head-drop, tail-drop, or a variety of Active Queue Management, the queue is inspected, and in the case of Active Queue Management, additional parameters are REQUIRED.

What may not be clear from the name is that an Algorithmic Drop action often does not drop traffic. Algorithms other than "always drop" normally drop a few percent of packets at most. The action inspects the diffServQEntry that diffServAlgDropQMeasure points to in order to determine whether the packet should be dropped.

When the MIB is used for configuration, diffServAlgDropNextFree always contains a legal value for diffServAlgDropId that is not currently used in the system's configuration.

3.4.5. diffServRandomDropTable - The Random Drop Parameters Table

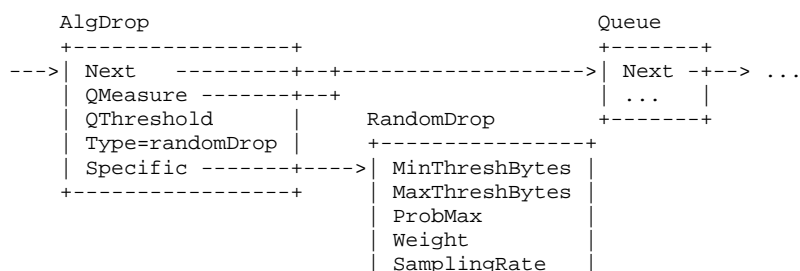
The Random Drop Table is an extension of the Algorithmic Drop Table intended for use on queues whose depth is actively managed. Active Queue Management algorithms are typified by [RED93], but the parameters they use vary. It was deemed for the purposes of this MIB that the proper values to represent include:

- o Target case mean queue depth, expressed in bytes or packets
- o Worst case mean queue depth, expressed in bytes or packets
- o Maximum drop rate expressed as drops per thousand
- o Coefficient of an exponentially weighted moving average, expressed as the numerator of a fraction whose denominator is 65536.
- o Sampling rate

An example of the representation chosen in this MIB for this element is shown in Figure 1.

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

In the MIB module, diffServRandomDropMinThreshBytes and diffServRandomDropMinThreshPkts represent Qmin. diffServRandomDropMaxThreshBytes and diffServRandomDropMaxThreshPkts represent Qmax. diffServAlgDropQThreshold represents Qclip. diffServRandomDropInvProbMax represents Pmax (inverse). This MIB does not represent Pmin (assumed to be zero unless otherwise represented). In addition, since message memory is finite, queues generally have some upper bound above which they are incapable of storing additional traffic. Normally this number is equal to Qclip, specified by diffServAlgDropQThreshold.



+-----+

Figure 1: Example Use of the RandomDropTable for Random Droppers

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

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

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

When the MIB is used for configuration, diffServRandomDropNextFree always contains a legal value for diffServRandomDropId that is not currently used in the system's configuration.

Baker, et. al. Standards Track [Page 15]

RFC 3289 Differentiated Services MIB May 2002

3.5. Queuing and Scheduling of Packets

These include Queues and Schedulers, which are inter-related in their use of queuing techniques. By doing so, it is possible to build multi-level schedulers, such as those which treat a set of queues as having priority among them, and at a specific priority find a secondary WFQ scheduler with some number of queues.

3.5.1. diffServQTable - The Class or Queue Table

The Queue Table models simple FIFO queues. The Scheduler Table allows flexibility in constructing both simple and somewhat more complex queuing hierarchies from those queues.

Queue Table entries are pointed at by the "next" attributes of the upstream elements, such as diffServMeterSucceedNext or diffServActionNext. Note that multiple upstream elements may direct their traffic to the same Queue Table entry. For example, the Assured Forwarding PHB suggests that all traffic marked AF11, AF12 or AF13 be placed in the same queue, after metering, without reordering. To accomplish that, the upstream diffServAlgDropNext pointers each point to the same diffServQEntry.

A common requirement of a queue is that its traffic enjoy a certain minimum or maximum rate, or that it be given a certain priority. Functionally, the selection of such is a function of a scheduler. The parameter is associated with the queue, however, using the Minimum or Maximum Rate Parameters Table.

When the MIB is used for configuration, diffServQNextFree always contains a legal value for diffServQId that is not currently used in the system's configuration.

3.5.2. diffServSchedulerTable - The Scheduler Table

The scheduler, and therefore the Scheduler Table, accepts inputs from either queues or a preceding scheduler. The Scheduler Table allows flexibility in constructing both simple and somewhat more complex queuing hierarchies from those queues.

When the MIB is used for configuration, diffServSchedulerNextFree always contains a legal value for diffServSchedulerId that is not currently used in the system's configuration.

3.5.3. diffServMinRateTable - The Minimum Rate Table

When the output rate of a queue or scheduler must be given a minimum rate or a priority, this is done using the diffServMinRateTable.

Baker, et. al. Standards Track [Page 16]
RFC 3289 Differentiated Services MIB May 2002

Rates may be expressed as absolute rates, or as a fraction of ifSpeed, and imply the use of a rate-based scheduler such as WFQ or WRR. The use of a priority implies the use of a Priority Scheduler. Only one of the Absolute or Relative rates needs to be set; the other takes the relevant value as a result. Excess capacity is distributed proportionally among the inputs to a scheduler using the assured rate. More complex functionality may be described by augmenting this MIB.

When a priority scheduler is used, its effect is to give the queue the entire capacity of the subject interface less the capacity used by higher priorities, if there is traffic present to use it. This is true regardless of the rate specifications applied to that queue or other queues on the interface. Policing excess traffic will mitigate this behavior.

When the MIB is used for configuration, diffServMinRateNextFree always contains a legal value for diffServMinRateId that is not currently used in the system's configuration.

3.5.4. diffServMaxRateTable - The Maximum Rate Table

When the output rate of a queue or scheduler must be limited to at most a specified maximum rate, this is done using the diffServMaxRateTable. Rates may be expressed as absolute rates, or as a fraction of ifSpeed. Only one of the Absolute or Relative rate needs to be set; the other takes the relevant value as a result.

The definition of a multirate shaper requires multiple diffServMaxRateEntries. In this case, an algorithm such as [SHAPER] is used. In that algorithm, more than one rate is specified, and at any given time traffic is shaped to the lowest specified rate which exceeds the arrival rate of traffic.

When the MIB is used for configuration, diffServMaxRateNextFree always contains a legal value for diffServMaxRateId that is not currently used in the system's configuration.

3.5.5. Using queues and schedulers together

For representing a Strict Priority scheduler, each scheduler input is assigned a priority with respect to all the other inputs feeding the same scheduler, with default values for the other parameters. Higher-priority traffic that is not being delayed for shaping will be serviced before a lower-priority input. An example is found in Figure 2.

Baker, et. al. Standards Track [Page 17]
RFC 3289 Differentiated Services MIB May 2002

For weighted scheduling methods, such as WFQ or WRR, the "weight" of a given scheduler input is represented with a Minimum Service Rate leaky-bucket profile which provides a guaranteed minimum bandwidth to

that input, if required. This is represented by a rate `diffServMinRateAbsolute`; the classical weight is the ratio between that rate and the interface speed, or perhaps the ratio between that rate and the sum of the configured rates for classes. The rate may be represented by a relative value, as a fraction of the interface's current line rate, `diffServMinRateRelative`, to assist in cases where line rates are variable or where a higher-level policy might be expressed in terms of fractions of network resources. The two rate parameters are inter-related and changes in one may be reflected in the other. An example is found in figure 3.

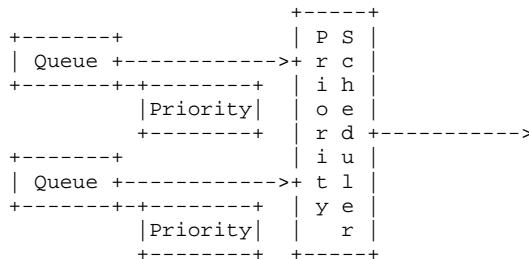


Figure 2: Priority Scheduler with two queues

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

A queue or scheduled set of queues (which is an input to a scheduler) may also be capable of acting as a non-work-conserving [MODEL] traffic shaper: this is done by defining a Maximum Service Rate leaky-bucket profile in order to limit the scheduler bandwidth available to that input. This is represented by a rate, in `diffServMaxRateAbsolute`; the classical weight is the ratio between that rate and the interface speed, or perhaps the ratio between that rate and the sum of the configured rates for classes. The rate may be represented by a relative value, as a fraction of the interface's current line rate, `diffServMaxRateRelative`. This MIB presumes that shaping is something a scheduler does to its inputs, which it models as a queue with a maximum rate or a scheduler whose output has a maximum rate.

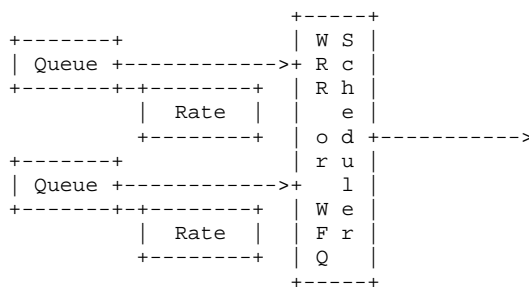
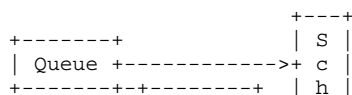


Figure 3: WRR or WFQ rate-based scheduler with two inputs

The same may be done on a queue, if a given class is to be shaped to a maximum rate without shaping other classes, as in Figure 5.

Other types of priority and weighted scheduling methods can be defined using existing parameters in `diffServMinRateEntry`. NOTE: `diffServSchedulerMethod` uses OBJECT IDENTIFIER syntax, with the different types of scheduling methods defined as OBJECT-IDENTITY.



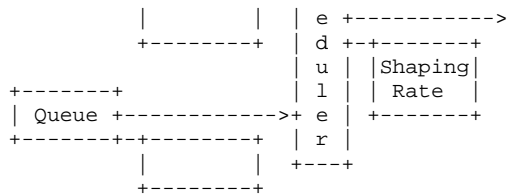


Figure 4: Shaping scheduled traffic to a known rate

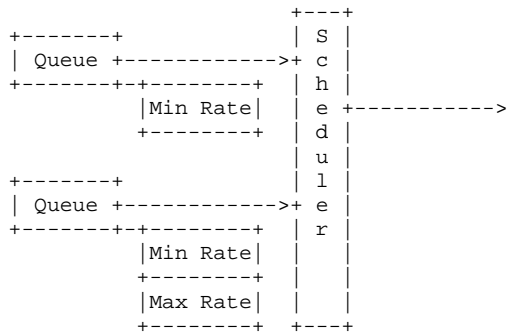


Figure 5: Shaping one input to a work-conserving scheduler

Future scheduling methods may be defined in other MIBs. This requires an OBJECT-IDENTITY definition, a description of how the existing objects are reused, if they are, and any new objects they require.

To implement an EF and two AF classes, one must use a combination of priority and WRR/WFQ scheduling. This requires us to cascade two schedulers. If one were to additionally shape the output of the system to a rate lower than the interface rate, one must place an upper bound rate on the output of the priority scheduler. See figure 6.

3.6. Example configuration for AF and EF

For the sake of argument, let us build an example with one EF class and four AF classes using the constructs in this MIB.

3.6.1. AF and EF Ingress Interface Configuration

The ingress edge interface identifies traffic into classes, meters it, and ensures that any excess is appropriately dealt with according to the PHB. For AF, this means marking excess; for EF, it means dropping excess or shaping it to a maximum rate.

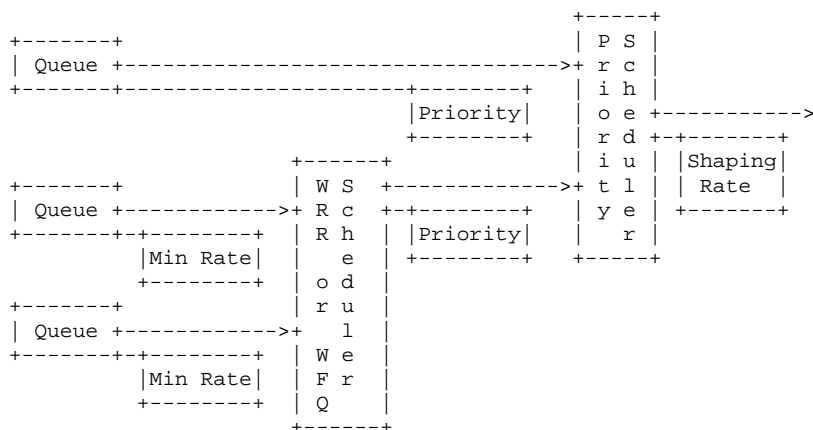


Figure 6: Combined EF and AF services using cascaded schedulers.

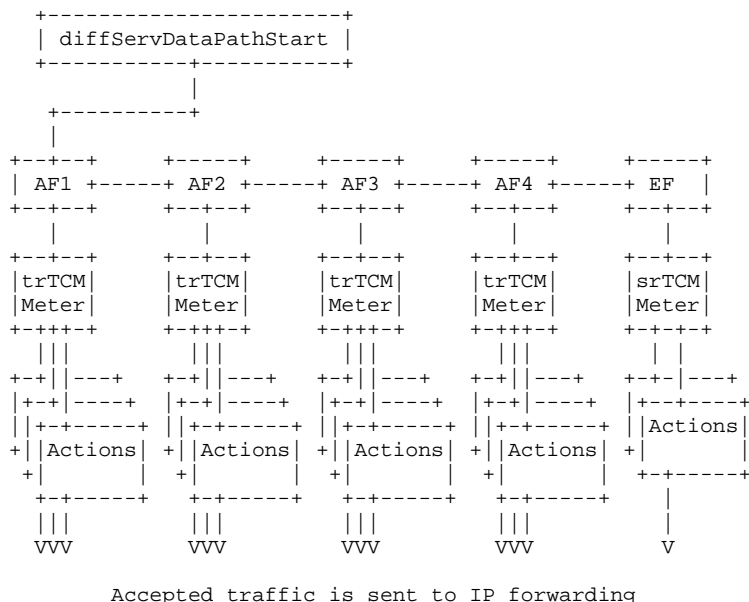


Figure 7: combined EF and AF implementation, ingress side

3.6.1.1. Classification In The Example

A packet arriving at an ingress interface picks up its policy from the diffServDataTable. This points to a classifier, which will select traffic according to some specification for each traffic class.

An example of a classifier for an AFm class would be a set of three classifier elements, each pointing to a Multi-field classification parameter block identifying one of the AFmn DSCPs. Alternatively, the filters might contain selectors for HTTP traffic or some other application.

An example of a classifier for EF traffic might be a classifier element pointing to a filter specifying the EF code point, a collection of classifiers with parameter blocks specifying individual telephone calls, or a variety of other approaches.

Typically, of course, a classifier identifies a variety of traffic

and breaks it up into separate classes. It might very well contain fourteen classifier elements indicating the twelve AFmn DSCP values, EF, and "everything else". These would presumably direct traffic down six functional data paths: one for each AF or EF class, and one for all other traffic.

3.6.1.2. AF Implementation On an Ingress Edge Interface

Each AFm class applies a Two Rate Three Color Meter, dividing traffic into three groups. These groups of traffic conform to both specified rates, only the higher one, or none. The intent, on the ingress interface at the edge of the network, is to measure and appropriately mark traffic.

3.6.1.2.1. AF Metering On an Ingress Edge Interface

Each AFm class applies a Two Rate Three Color Meter, dividing traffic into three groups. If two rates R and S, where $R < S$, are specified and traffic arrives at rate T, traffic comprising up to R bits per second is considered to conform to the "confirmed" rate, R. If $R < T$, traffic comprising up to $S-R$ bits per second is considered to conform to the "excess" rate, S. Any further excess is non-conformant.

Two meter entries are used to configure this, one for the conforming rate and one for the excess rate. The rate parameters are stored in associated Token Bucket Parameter Entries. The "FailNext" pointer of the lower rate Meter Entry points to the other Meter Entry; both "SucceedNext" pointers and the "FailNext" pointer of the higher Meter

Baker, et. al. Standards Track [Page 22]
RFC 3289 Differentiated Services MIB May 2002

Entry point to lists of actions. In the color-blind mode, all three classifier "next" entries point to the lower rate meter entry. In the color-aware mode, the AFm1 classifier points to the lower rate entry, the AFm2 classifier points to the higher rate entry (as it is only compared against that rate), and the AFm3 classifier points directly to the actions taken when both rates fail.

3.6.1.2.2. AF Actions On an Ingress Edge Interface

For network planning and perhaps for billing purposes, arriving traffic is normally counted. Therefore, a "count" action, consisting of an action table entry pointing to a count table entry, is configured.

Also, traffic is marked with the appropriate DSCP. The first R bits per second are marked AFm1, the next $S-R$ bits per second are marked AFm2, and the rest is marked AFm3. It may be that traffic is arriving marked with the same DSCP, but in general, the additional complexity of deciding that it is being remarked to the same value is not useful. Therefore, a "mark" action, consisting of an action table entry pointing to a mark table entry, is configured.

At this point, the usual case is that traffic is now forwarded in the usual manner. To indicate this, the "SucceedNext" pointer of the Mark Action is set to zeroDotZero.

3.6.1.3. EF Implementation On an Ingress Edge Interface

The EF class applies a Single Rate Two Color Meter, dividing traffic into "conforming" and "excess" groups. The intent, on the ingress interface at the edge of the network, is to measure and appropriately mark conforming traffic and drop the excess.

3.6.1.3.1. EF Metering On an Ingress Edge Interface

A single rate two color (srTCM) meter requires one token bucket. It is therefore configured using a single meter entry with a corresponding Token Bucket Parameter Entry. Arriving traffic either "succeeds" or "fails".

3.6.1.3.2. EF Actions On an Ingress Edge Interface

For network planning and perhaps for billing purposes, arriving

traffic that conforms to the meter is normally counted. Therefore, a "count" action, consisting of an action table entry pointing to a count table entry, is configured.

Also, traffic is (re)marked with the EF DSCP. Therefore, a "mark" action, consisting of an action table entry pointing to a mark table entry, is configured.

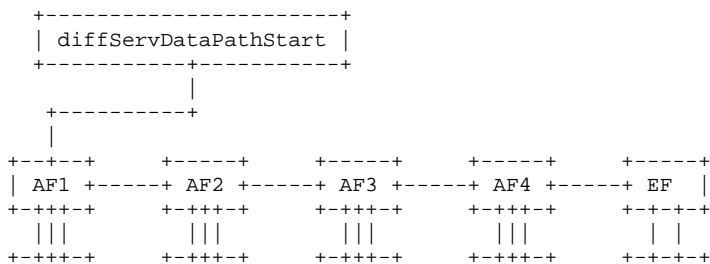
At this point, the successful traffic is now forwarded in the usual manner. To indicate this, the "SucceedNext" pointer of the Mark Action is set to zeroDotZero.

Traffic that exceeded the arrival policy, however, is to be dropped. One can use a count action on this traffic if the several counters are interesting. However, since the drop counter in the Algorithmic Drop Entry will count packets dropped, this is not clearly necessary. An Algorithmic Drop Entry of the type "alwaysDrop" with no successor is sufficient.

3.7. AF and EF Egress Edge Interface Configuration

3.7.1. Classification On an Egress Edge Interface

A packet arriving at an egress interface may have been classified on an ingress interface, and the egress interface may have access to that information. If it is relevant, there is no reason not to use that information. If it is not available, however, there may be a need to (re)classify on the egress interface. In any event, it picks up its "program" from the diffServDataPathTable. This points to a classifier, which will select traffic according to some specification for each traffic class.



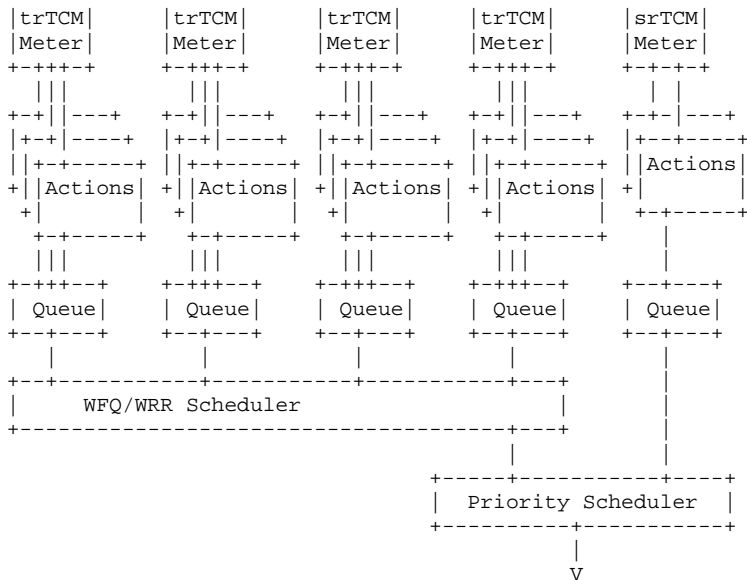


Figure 8: combined EF and AF implementation

An example of a classifier for an AFm class would be a succession of three classifier elements, each pointing to a Multi-field classification parameter block identifying one of the AFm DSCPs. Alternatively, the filter might contain selectors for HTTP traffic or some other application.

An example of a classifier for EF traffic might be either a classifier element pointing to a Multi-field parameter specifying the EF code point, or a collection of classifiers with parameter blocks specifying individual telephone calls, or a variety of other approaches.

Each classifier delivers traffic to appropriate functional data path elements.

3.7.2. AF Implementation On an Egress Edge Interface

Each AFm class applies a Two Rate Three Color Meter, dividing traffic into three groups. These groups of traffic conform to both specified rates, only the higher one, or none. The intent, on the ingress interface at the edge of the network, is to measure and appropriately mark traffic.

3.7.2.1. AF Metering On an Egress Edge Interface

Each AFm class applies a Two Rate Three Color Meter, dividing traffic into three groups. If two rates R and S, where R < S, are specified and traffic arrives at rate T, traffic comprising up to R bits per second is considered to conform to the "confirmed" rate, R. If R < T, traffic comprising up to S-R bits per second is considered to conform to the "excess" rate, S. Any further excess is non-conformant.

Two meter entries are used to configure this, one for the conforming rate and one for the excess rate. The rate parameters are stored in associated Token Bucket Parameter Entries. The "FailNext" pointer of the lower rate Meter Entry points to the other Meter Entry; both "SucceedNext" pointers and the "FailNext" pointer of the higher Meter Entry point to lists of actions. In the color-blind mode, all three classifier "next" entries point to the lower rate meter entry. In the color-aware mode, the AFm classifier points to the lower rate

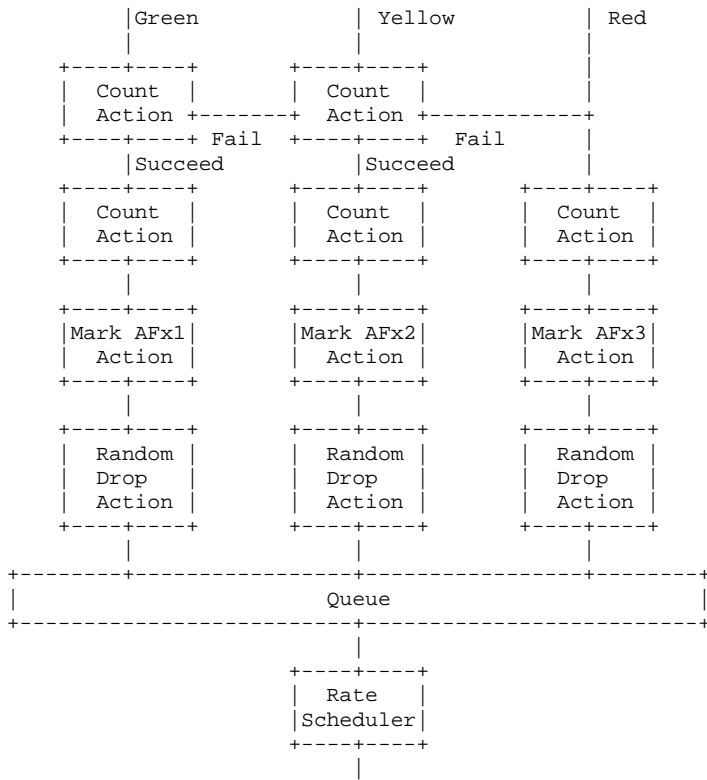


Figure 9b: Typical AF Edge egress interface configuration, using color-aware meters

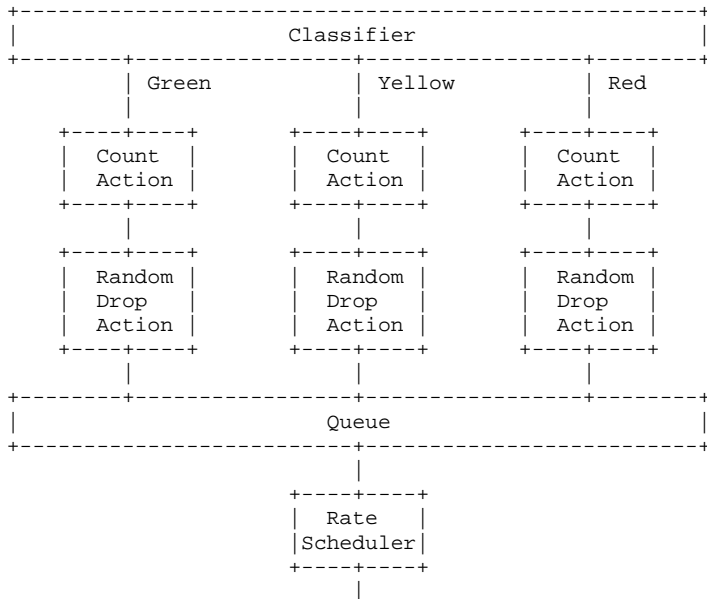


Figure 10: Typical AF Edge core interface configuration

3.7.2.2. AF Actions On an Egress Edge Interface

For network planning and perhaps for billing purposes, departing traffic is normally counted. Therefore, a "count" action, consisting of an action table entry pointing to a count table entry, is configured.

Also, traffic may be marked with an appropriate DSCP. The first R bits per second are marked AFm1, the next S-R bits per second are marked AFm2, and the rest is marked AFm3. It may be that traffic is arriving marked with the same DSCP, but in general, the additional complexity of deciding that it is being remarked to the same value is not useful. Therefore, a "mark" action, consisting of an action table entry pointing to a mark table entry, is configured.

At this point, the usual case is that traffic is now queued for transmission. The queue uses Active Queue Management, using an algorithm such as RED. Therefore, an Algorithmic Dropper is

configured for each AFmn traffic stream, with a slightly lower min-threshold (and possibly lower max-threshold) for the excess traffic than for the committed traffic.

3.7.2.3. AF Rate-based Queuing On an Egress Edge Interface

The queue expected by AF is normally a work-conserving queue. It usually has a specified minimum rate, and may have a maximum rate below the bandwidth of the interface. In concept, it will use as much bandwidth as is available to it, but assure the lower bound.

Common ways to implement this include various forms of Weighted Fair Queuing (WFQ) or Weighted Round Robin (WRR). Integrated over a longer interval, these give each class a predictable throughput rate. They differ in that over short intervals they will order traffic differently. In general, traffic classes that keep traffic in queue will tend to absorb latency from queues with lower mean occupancy, in exchange for which they make use of any available capacity.

3.7.3. EF Implementation On an Egress Edge Interface

The EF class applies a Single Rate Two Color Meter, dividing traffic into "conforming" and "excess" groups. The intent, on the egress interface at the edge of the network, is to measure and appropriately mark conforming traffic and drop the excess.

3.7.3.1. EF Metering On an Egress Edge Interface

A single rate two color (srTCM) meter requires one token bucket. It is therefore configured using a single meter entry with a corresponding Token Bucket Parameter Entry. Arriving traffic either "succeeds" or "fails".

3.7.3.2. EF Actions On an Egress Edge Interface

For network planning and perhaps for billing purposes, departing traffic that conforms to the meter is normally counted. Therefore, a "count" action, consisting of an action table entry pointing to a count table entry, is configured.

Also, traffic is (re)marked with the EF DSCP. Therefore, a "mark" action, consisting of an action table entry pointing to a mark table entry, is configured.

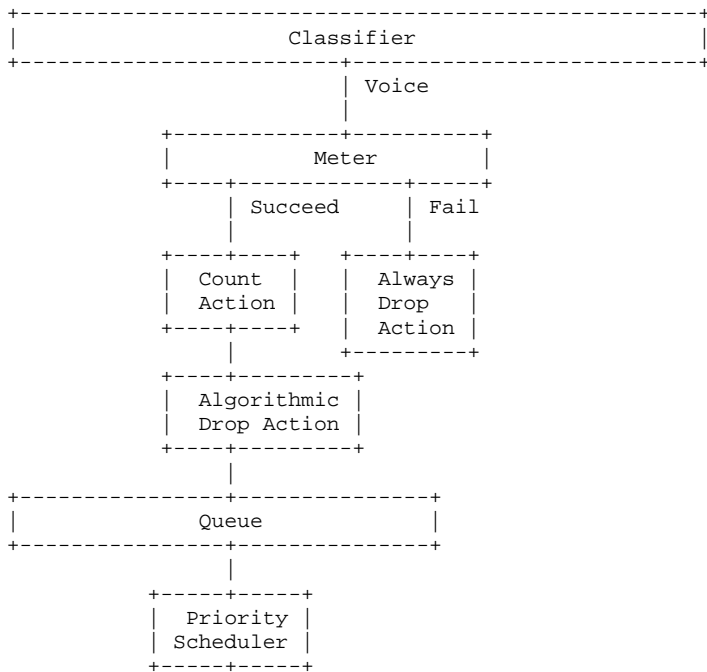
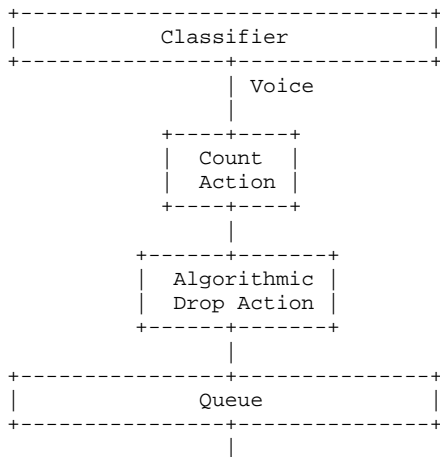


Figure 11: Typical EF Edge (Policing) Configuration



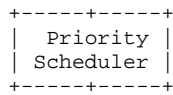


Figure 12: Typical EF Core interface Configuration

At this point, the successful traffic is now queued for transmission, using a priority queue or perhaps a rate-based queue with significant over-provision. Since the amount of traffic present is known, one might not drop from this queue at all.

Traffic that exceeded the policy, however, is dropped. A count action can be used on this traffic if the several counters are interesting. However, since the drop counter in the Algorithmic Drop Entry will count packets dropped, this is not clearly necessary. An Algorithmic Drop Entry of the type "alwaysDrop" with no successor is sufficient.

3.7.3.3. EF Priority Queuing On an Egress Edge Interface

The normal implementation is a priority queue, to minimize induced jitter. A separate queue is used for each EF class, with a strict ordering.

4. Conventions used in this MIB

4.1. The use of RowPointer to indicate data path linkage

RowPointer is a textual convention used to identify a conceptual row in a MIB Table by pointing to one of its objects. One of the ways this MIB uses it is to indicate succession, pointing to data path linkage table entries.

For succession, it answers the question "what happens next?". Rather than presume that the next table must be as specified in the conceptual model [MODEL] and providing its index, the RowPointer takes you to the MIB row representing that thing. In the diffServMeterTable, for example, the diffServMeterFailNext RowPointer might take you to another meter, while the diffServMeterSucceedNext RowPointer would take you to an action.

Since a RowPointer is not tied to any specific object except by the value it contains, it is possible and acceptable to use RowPointers to merge data paths. An obvious example of such a use is in the classifier: traffic matching the DSCPs AF11, AF12, and AF13 might be presented to the same meter in order to perform the processing described in the Assured Forwarding PHB. Another use would be to merge data paths from several interfaces; if they represent a single service contract, having them share a common set of counters and common policy may be a desirable configuration. Note well, however, that such configurations may have related implementation issues - if Differentiated Services processing for the interfaces is implemented in multiple forwarding engines, the engines will need to communicate if they are to implement such a feature. An implementation that fails to provide this capability is not considered to have failed the intention of this MIB or of the [MODEL]; an implementation that does provide it is not considered superior from a standards perspective.

NOTE -- the RowPointer construct is used to connect the functional data paths. The [MODEL] describes these as TCBS, as an aid to understanding. This MIB, however, does not model TCBS directly. It operates at a lower level of abstraction using only individual elements, connected in succession by RowPointers. Therefore, the concept of TCBS enclosing individual Functional Data Path elements is not directly applicable to this MIB, although management tools that use this MIB may employ such a concept.

The use of RowStatus is covered in more detail in [RFC 2579].

5. Extending this MIB

With the structures of this MIB divided into data path linkage tables and parameterization tables, and with the use of RowPointer, new data path linkage and parameterization tables can be defined in other MIB modules, and used with tables defined in this MIB. This MIB does not limit the type of entries its RowPointer attributes can point to, hence new functional data path elements can be defined in other MIBs and integrated with functional data path elements of this MIB. For example, new Action functional data path element can be defined for Traffic Engineering and be integrated with Differentiated Services functional data path elements, possibly within the same data path sharing the same classifiers and meters.

It is more likely that new parameterization tables will be created in other MIBs as new methods or proprietary methods get deployed for existing Differentiated Services Functional Data Path Elements. For example, different kinds of filters can be defined by using new filter parameterization tables. New scheduling methods can be deployed by defining new scheduling method OIDs and new scheduling parameter tables.

Notice both new data path linkage tables and parameterization tables can be added without needing to change this MIB document or affect existing tables and their usage.

6. MIB Definition

```
DIFFSERV-DSCP-TC DEFINITIONS ::= BEGIN
```

```
IMPORTS
Integer32, MODULE-IDENTITY, mib-2
FROM SNMPv2-SMI
TEXTUAL-CONVENTION
FROM SNMPv2-TC;
```

Baker, et. al. Standards Track [Page 35]

RFC 3289 Differentiated Services MIB May 2002

```
diffServDSCP TC MODULE-IDENTITY
LAST-UPDATED "200205090000Z"
ORGANIZATION "IETF Differentiated Services WG"
CONTACT-INFO
"
Fred Baker
Cisco Systems
1121 Via Del Rey
Santa Barbara, CA 93117, USA
E-mail: fred@cisco.com

Kwok Ho Chan
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821, USA
E-mail: khchan@nortelnetworks.com

Andrew Smith
Harbour Networks
Jiuling Building
21 North Xisanhuan Ave.
Beijing, 100089, PRC
E-mail: ah_smith@acm.org

Differentiated Services Working Group:
diffserv@ietf.org"
DESCRIPTION
"The Textual Conventions defined in this module should be used
whenever a Differentiated Services Code Point is used in a MIB."
REVISION "200205090000Z"
DESCRIPTION
"Initial version, published as RFC 3289."
::= { mib-2 96 }

Dscp ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
```

STATUS current
DESCRIPTION
"A Differentiated Services Code-Point that may be used for
marking a traffic stream."
REFERENCE
"RFC 2474, RFC 2780"
SYNTAX Integer32 (0..63)

DscpOrAny ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS current
DESCRIPTION
"The IP header Differentiated Services Code-Point that may be

Baker, et. al. Standards Track [Page 36]
RFC 3289 Differentiated Services MIB May 2002

used for discriminating among traffic streams. The value -1 is
used to indicate a wild card i.e. any value."
REFERENCE
"RFC 2474, RFC 2780"
SYNTAX Integer32 (-1 | 0..63)

END

DIFFSERV-MIB DEFINITIONS ::= BEGIN

IMPORTS
Unsigned32, Counter64, MODULE-IDENTITY, OBJECT-TYPE,
OBJECT-IDENTITY, zeroDotZero, mib-2
FROM SNMPv2-SMI
TEXTUAL-CONVENTION, RowStatus, RowPointer,
StorageType, AutonomousType
FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF
ifIndex, InterfaceIndexOrZero
FROM IF-MIB
InetAddressType, InetAddress, InetAddressPrefixLength,
InetPortNumber
FROM INET-ADDRESS-MIB
BurstSize
FROM INTEGRATED-SERVICES-MIB
Dscp, DscpOrAny
FROM DIFFSERV-DSCP-TC;

diffServMib MODULE-IDENTITY
LAST-UPDATED "200202070000Z"
ORGANIZATION "IETF Differentiated Services WG"
CONTACT-INFO
" Fred Baker
Cisco Systems
1121 Via Del Rey
Santa Barbara, CA 93117, USA
E-mail: fred@cisco.com

Kwok Ho Chan
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821, USA
E-mail: khchan@nortelnetworks.com

Andrew Smith
Harbour Networks
Jiuling Building

Baker, et. al. Standards Track [Page 37]
RFC 3289 Differentiated Services MIB May 2002

21 North Xisanhuan Ave.
Beijing, 100089, PRC
E-mail: ah_smith@acm.org

Differentiated Services Working Group:
diffserv@ietf.org"

DESCRIPTION

"This MIB defines the objects necessary to manage a device that uses the Differentiated Services Architecture described in RFC 2475. The Conceptual Model of a Differentiated Services Router provides supporting information on how such a router is modeled."

REVISION "200202070000Z"

DESCRIPTION

"Initial version, published as RFC 3289."

::= { mib-2 97 }

diffServMIBObjects OBJECT IDENTIFIER ::= { diffServMib 1 }
diffServMIBConformance OBJECT IDENTIFIER ::= { diffServMib 2 }
diffServMIBAdmin OBJECT IDENTIFIER ::= { diffServMib 3 }

IndexInteger ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"An integer which may be used as a table index."

SYNTAX Unsigned32 (1..4294967295)

IndexIntegerNextFree ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"An integer which may be used as a new Index in a table."

The special value of 0 indicates that no more new entries can be created in the relevant table.

When a MIB is used for configuration, an object with this SYNTAX always contains a legal value (if non-zero) for an index that is not currently used in the relevant table. The Command Generator (Network Management Application) reads this variable and uses the (non-zero) value read when creating a new row with an SNMP SET. When the SET is performed, the Command Responder (agent) must determine whether the value is indeed still unused; Two Network Management Applications may attempt to create a row (configuration entry) simultaneously and use the same value. If it is currently unused, the SET succeeds and the Command Responder (agent) changes the value of this object, according to an implementation-specific algorithm. If the value is in use,

Baker, et. al. Standards Track [Page 38]

RFC 3289 Differentiated Services MIB May 2002

however, the SET fails. The Network Management Application must then re-read this variable to obtain a new usable value.

An OBJECT-TYPE definition using this SYNTAX MUST specify the relevant table for which the object is providing this functionality."

SYNTAX Unsigned32 (0..4294967295)

IfDirection ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"IfDirection specifies a direction of data travel on an interface. 'inbound' traffic is operated on during reception from the interface, while 'outbound' traffic is operated on prior to transmission on the interface."

SYNTAX INTEGER {
 inbound(1), -- ingress interface
 outbound(2) -- egress interface

}

--
-- Data Path
--

diffServDataPath OBJECT IDENTIFIER ::= { diffServMIBObjects 1 }

--

```

-- Data Path Table
--
-- The Data Path Table enumerates the Differentiated Services
-- Functional Data Paths within this device. Each entry in this table
-- is indexed by ifIndex and ifDirection. Each entry provides the
-- first Differentiated Services Functional Data Path Element to
-- process data flowing along specific data path. This table should
-- have at most two entries for each interface capable of
-- Differentiated Services processing on this device: ingress and
-- egress.

-- Note that Differentiated Services Functional Data Path Elements
-- linked together using their individual next pointers and anchored by
-- an entry of the diffServDataPathTable constitute a functional data
-- path.
--

```

```

diffServDataPathTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiffServDataPathEntry
    MAX-ACCESS  not-accessible
    STATUS      current

```

```

Baker, et. al.           Standards Track           [Page 39]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

DESCRIPTION
    "The data path table contains RowPointers indicating the start of
    the functional data path for each interface and traffic direction
    in this device. These may merge, or be separated into parallel
    data paths."
 ::= { diffServDataPath 1 }

```

```

diffServDataPathEntry OBJECT-TYPE
    SYNTAX      DiffServDataPathEntry
    MAX-ACCESS  not-accessible
    STATUS      current

```

```

DESCRIPTION
    "An entry in the data path table indicates the start of a single
    Differentiated Services Functional Data Path in this device.

    These are associated with individual interfaces, logical or
    physical, and therefore are instantiated by ifIndex. Therefore,
    the interface index must have been assigned, according to the
    procedures applicable to that, before it can be meaningfully
    used. Generally, this means that the interface must exist.

    When diffServDataPathStorage is of type nonVolatile, however,
    this may reflect the configuration for an interface whose ifIndex
    has been assigned but for which the supporting implementation is
    not currently present."
    INDEX { ifIndex, diffServDataPathIfDirection }
 ::= { diffServDataPathTable 1 }

```

```

DiffServDataPathEntry ::= SEQUENCE {
    diffServDataPathIfDirection  IfDirection,
    diffServDataPathStart       RowPointer,
    diffServDataPathStorage     StorageType,
    diffServDataPathStatus      RowStatus
}

```

```

diffServDataPathIfDirection OBJECT-TYPE
    SYNTAX      IfDirection
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "IfDirection specifies whether the reception or transmission path
        for this interface is in view."
 ::= { diffServDataPathEntry 1 }

```

```

diffServDataPathStart OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-create
    STATUS      current

```

DESCRIPTION

"This selects the first Differentiated Services Functional Data Path Element to handle traffic for this data path. This RowPointer should point to an instance of one of:

```
diffServClfrEntry
diffServMeterEntry
diffServActionEntry
diffServAlgDropEntry
diffServQEntry
```

A value of zeroDotZero in this attribute indicates that no Differentiated Services treatment is performed on traffic of this data path. A pointer with the value zeroDotZero normally terminates a functional data path.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

```
::= { diffServDataPathEntry 2 }
```

diffServDataPathStorage OBJECT-TYPE

```
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

```
DEFVAL { nonVolatile }
```

```
::= { diffServDataPathEntry 3 }
```

diffServDataPathStatus OBJECT-TYPE

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"The status of this conceptual row. All writable objects in this row may be modified at any time."

```
::= { diffServDataPathEntry 4 }
```

```
--
```

```
-- Classifiers
```

```
--
```

```
diffServClassifier      OBJECT IDENTIFIER ::= { diffServMIBObjects 2 }
```

```
--
```

```
-- Classifier Table
```

```
--
```

```
-- The Classifier Table allows multiple classifier elements, of same or
-- different types, to be used together. A classifier must completely
-- classify all packets presented to it. This means that all traffic
-- presented to a classifier must match at least one classifier element
-- within the classifier, with the classifier element parameters
-- specified by a filter.
```

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

```
-- Entries in the classifier element table serves as the anchor for
-- each classification pattern, defined in filter table entries. Each
-- classifier element table entry also specifies the subsequent
-- downstream Differentiated Services Functional Data Path Element when
-- the classification pattern is satisfied. Each entry in the
-- classifier element table describes one branch of the fan-out
```

-- characteristic of a classifier indicated in the Informal
-- Differentiated Services Model section 4.1. A classifier is composed
-- of one or more classifier elements.

```
diffServClfrNextFree OBJECT-TYPE
    SYNTAX      IndexIntegerNextFree
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains an unused value for diffServClfrId, or a
        zero to indicate that none exist."
    ::= { diffServClassifier 1 }
```

```
diffServClfrTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiffServClfrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table enumerates all the diffserv classifier functional
        data path elements of this device. The actual classification
        definitions are defined in diffServClfrElementTable entries
        belonging to each classifier.
```

An entry in this table, pointed to by a RowPointer specifying an instance of diffServClfrStatus, is frequently used as the name for a set of classifier elements, which all use the index diffServClfrId. Per the semantics of the classifier element table, these entries constitute one or more unordered sets of tests which may be simultaneously applied to a message to

Baker, et. al. Standards Track [Page 42]

RFC 3289 Differentiated Services MIB May 2002

classify it.

The primary function of this table is to ensure that the value of diffServClfrId is unique before attempting to use it in creating a diffServClfrElementEntry. Therefore, the diffServClfrEntry must be created on the same SET as the diffServClfrElementEntry, or before the diffServClfrElementEntry is created."

```
::= { diffServClassifier 2 }
```

```
diffServClfrEntry OBJECT-TYPE
    SYNTAX      DiffServClfrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the classifier table describes a single classifier.
        All classifier elements belonging to the same classifier use the
        classifier's diffServClfrId as part of their index."
    INDEX { diffServClfrId }
    ::= { diffServClfrTable 1 }
```

```
DiffServClfrEntry ::= SEQUENCE {
    diffServClfrId      IndexInteger,
    diffServClfrStorage StorageType,
    diffServClfrStatus RowStatus
}
```

```
diffServClfrId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the classifier entries. Managers
        should obtain new values for row creation in this table by
        reading diffServClfrNextFree."
    ::= { diffServClfrEntry 1 }
```

```
diffServClfrStorage OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The storage type for this conceptual row. Conceptual rows
        having the value 'permanent' need not allow write-access to any
```

```
    columnar objects in the row."
DEFVAL { nonVolatile }
 ::= { diffServClfrEntry 2 }
```

```
diffServClfrStatus OBJECT-TYPE
```

```
Baker, et. al.                Standards Track                [Page 43]
```

```
RFC 3289                Differentiated Services MIB                May 2002
```

```
SYNTAX          RowStatus
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The status of this conceptual row. All writable objects in this
    row may be modified at any time. Setting this variable to
    'destroy' when the MIB contains one or more RowPointers pointing
    to it results in destruction being delayed until the row is no
    longer used."
 ::= { diffServClfrEntry 3 }
```

```
--
-- Classifier Element Table
--
```

```
diffServClfrElementNextFree OBJECT-TYPE
```

```
SYNTAX          IndexIntegerNextFree
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object contains an unused value for diffServClfrElementId,
    or a zero to indicate that none exist."
 ::= { diffServClassifier 3 }
```

```
diffServClfrElementTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF DiffServClfrElementEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "The classifier element table enumerates the relationship between
    classification patterns and subsequent downstream Differentiated
    Services Functional Data Path elements.
    diffServClfrElementSpecific points to a filter that specifies the
    classification parameters. A classifier may use filter tables of
    different types together.
```

```
One example of a filter table defined in this MIB is
diffServMultiFieldClfrTable, for IP Multi-Field Classifiers
(MFCs). Such an entry might identify anything from a single
micro-flow (an identifiable sub-session packet stream directed
from one sending transport to the receiving transport or
transports), or aggregates of those such as the traffic from a
host, traffic for an application, or traffic between two hosts
using an application and a given DSCP. The standard Behavior
Aggregate used in the Differentiated Services Architecture is
encoded as a degenerate case of such an aggregate - the traffic
using a particular DSCP value.
```

```
Filter tables for other filter types may be defined elsewhere."
```

```
Baker, et. al.                Standards Track                [Page 44]
```

```
RFC 3289                Differentiated Services MIB                May 2002
```

```
 ::= { diffServClassifier 4 }
```

```
diffServClfrElementEntry OBJECT-TYPE
```

```
SYNTAX          DiffServClfrElementEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An entry in the classifier element table describes a single
    element of the classifier."
INDEX { diffServClfrID, diffServClfrElementId }
 ::= { diffServClfrElementTable 1 }
```

```

DiffServClfrElementEntry ::= SEQUENCE {
    diffServClfrElementId      IndexInteger,
    diffServClfrElementPrecedence Unsigned32,
    diffServClfrElementNext    RowPointer,
    diffServClfrElementSpecific RowPointer,
    diffServClfrElementStorage  StorageType,
    diffServClfrElementStatus   RowStatus
}

diffServClfrElementId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the Classifier Element entries.
        Managers obtain new values for row creation in this table by
        reading diffServClfrElementNextFree."
    ::= { diffServClfrElementEntry 1 }

diffServClfrElementPrecedence OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The relative order in which classifier elements are applied:
        higher numbers represent classifier element with higher
        precedence. Classifier elements with the same order must be
        unambiguous i.e. they must define non-overlapping patterns, and
        are considered to be applied simultaneously to the traffic
        stream. Classifier elements with different order may overlap in
        their filters: the classifier element with the highest order
        that matches is taken.

        On a given interface, there must be a complete classifier in
        place at all times in the ingress direction. This means one or
        more filters must match any possible pattern. There is no such

Baker, et. al.                Standards Track                [Page 45]
RFC 3289                      Differentiated Services MIB                May 2002

        requirement in the egress direction."
    ::= { diffServClfrElementEntry 2 }

diffServClfrElementNext OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This attribute provides one branch of the fan-out functionality
        of a classifier described in the Informal Differentiated Services
        Model section 4.1.

        This selects the next Differentiated Services Functional Data
        Path Element to handle traffic for this data path. This
        RowPointer should point to an instance of one of:
            diffServClfrEntry
            diffServMeterEntry
            diffServActionEntry
            diffServAlgDropEntry
            diffServQEntry

        A value of zeroDotZero in this attribute indicates no further
        Differentiated Services treatment is performed on traffic of this
        data path. The use of zeroDotZero is the normal usage for the
        last functional data path element of the current data path.

        Setting this to point to a target that does not exist results in
        an inconsistentValue error. If the row pointed to is removed or
        becomes inactive by other means, the treatment is as if this
        attribute contains a value of zeroDotZero."

    ::= { diffServClfrElementEntry 3 }

diffServClfrElementSpecific OBJECT-TYPE
    SYNTAX      RowPointer

```


MAX-ACCESS read-create
STATUS current
DESCRIPTION
"A pointer to a valid entry in another table, filter table, that describes the applicable classification parameters, e.g. an entry in diffServMultiFieldClfrTable.

The value zeroDotZero is interpreted to match anything not matched by another classifier element - only one such entry may exist for each classifier.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or

Baker, et. al. Standards Track [Page 46]
RFC 3289 Differentiated Services MIB May 2002

becomes inactive by other means, the element is ignored."
 ::= { diffServClfrElementEntry 4 }

diffServClfrElementStorage OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."
DEFVAL { nonVolatile }
 ::= { diffServClfrElementEntry 5 }

diffServClfrElementStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."
 ::= { diffServClfrElementEntry 6 }

--
-- IP Multi-field Classification Table
--
-- Classification based on six different fields in the IP header.
-- Functional Data Paths may share definitions by using the same entry.
--

diffServMultiFieldClfrNextFree OBJECT-TYPE
SYNTAX IndexIntegerNextFree
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object contains an unused value for diffServMultiFieldClfrId, or a zero to indicate that none exist."
 ::= { diffServClassifier 5 }

diffServMultiFieldClfrTable OBJECT-TYPE
SYNTAX SEQUENCE OF DiffServMultiFieldClfrEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A table of IP Multi-field Classifier filter entries that a

Baker, et. al. Standards Track [Page 47]
RFC 3289 Differentiated Services MIB May 2002

system may use to identify IP traffic."
 ::= { diffServClassifier 6 }

```

diffServMultiFieldClfrEntry OBJECT-TYPE
    SYNTAX          DiffServMultiFieldClfrEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An IP Multi-field Classifier entry describes a single filter."
    INDEX { diffServMultiFieldClfrId }
    ::= { diffServMultiFieldClfrTable 1 }

DiffServMultiFieldClfrEntry ::= SEQUENCE {
    diffServMultiFieldClfrId          IndexInteger,
    diffServMultiFieldClfrAddrType    InetAddressType,
    diffServMultiFieldClfrDstAddr     InetAddress,
    diffServMultiFieldClfrDstPrefixLength InetAddressPrefixLength,
    diffServMultiFieldClfrSrcAddr     InetAddress,
    diffServMultiFieldClfrSrcPrefixLength InetAddressPrefixLength,
    diffServMultiFieldClfrDscp        DscpOrAny,
    diffServMultiFieldClfrFlowId      Unsigned32,
    diffServMultiFieldClfrProtocol    Unsigned32,
    diffServMultiFieldClfrDstL4PortMin InetPortNumber,
    diffServMultiFieldClfrDstL4PortMax InetPortNumber,
    diffServMultiFieldClfrSrcL4PortMin InetPortNumber,
    diffServMultiFieldClfrSrcL4PortMax InetPortNumber,
    diffServMultiFieldClfrStorage     StorageType,
    diffServMultiFieldClfrStatus      RowStatus
}

diffServMultiFieldClfrId OBJECT-TYPE
    SYNTAX          IndexInteger
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An index that enumerates the MultiField Classifier filter
        entries. Managers obtain new values for row creation in this
        table by reading diffServMultiFieldClfrNextFree."

    ::= { diffServMultiFieldClfrEntry 1 }

diffServMultiFieldClfrAddrType OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The type of IP address used by this classifier entry. While
        other types of addresses are defined in the InetAddressType

Baker, et. al.          Standards Track          [Page 48]
RFC 3289                Differentiated Services MIB          May 2002

        textual convention, and DNS names, a classifier can only look at
        packets on the wire. Therefore, this object is limited to IPv4
        and IPv6 addresses."
    ::= { diffServMultiFieldClfrEntry 2 }

diffServMultiFieldClfrDstAddr OBJECT-TYPE
    SYNTAX          InetAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The IP address to match against the packet's destination IP
        address. This may not be a DNS name, but may be an IPv4 or IPv6
        prefix. diffServMultiFieldClfrDstPrefixLength indicates the
        number of bits that are relevant."
    ::= { diffServMultiFieldClfrEntry 3 }

diffServMultiFieldClfrDstPrefixLength OBJECT-TYPE
    SYNTAX          InetAddressPrefixLength
    UNITS           "bits"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The length of the CIDR Prefix carried in
        diffServMultiFieldClfrDstAddr. In IPv4 addresses, a length of 0
        indicates a match of any address; a length of 32 indicates a
        match of a single host address, and a length between 0 and 32
        indicates the use of a CIDR Prefix. IPv6 is similar, except that

```



```
::= { diffServMultiFieldClfrEntry 10 }
```

```
diffServMultiFieldClfrDstL4PortMax OBJECT-TYPE
```

```
SYNTAX      InetPortNumber
```

```
MAX-ACCESS  read-create
```

```
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 than the value specified
for this entry in diffServMultiFieldClfrDstL4PortMin."
```

```
DEFVAL      { 65535 }
```

```
::= { diffServMultiFieldClfrEntry 11 }
```

```
diffServMultiFieldClfrSrcL4PortMin OBJECT-TYPE
```

```
SYNTAX      InetPortNumber
```

```
MAX-ACCESS  read-create
```

```
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 }
```

```
::= { diffServMultiFieldClfrEntry 12 }
```

```
diffServMultiFieldClfrSrcL4PortMax OBJECT-TYPE
```

```
SYNTAX      InetPortNumber
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The maximum value that the layer-4 source port number in the
packet must have in order to match this classifier entry. This
value must be equal to or greater than the value specified for
this entry in diffServMultiFieldClfrSrcL4PortMin."
```

```
DEFVAL      { 65535 }
```

```
::= { diffServMultiFieldClfrEntry 13 }
```

```
diffServMultiFieldClfrStorage OBJECT-TYPE
```

```
SYNTAX      StorageType
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The storage type for this conceptual row. Conceptual rows
having the value 'permanent' need not allow write-access to any
columnar objects in the row."
```

```
DEFVAL { nonVolatile }
```

```
::= { diffServMultiFieldClfrEntry 14 }
```

```
diffServMultiFieldClfrStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The status of this conceptual row. All writable objects in this
row may be modified at any time. Setting this variable to
'destroy' when the MIB contains one or more RowPointers pointing
to it results in destruction being delayed until the row is no
longer used."
```

```
::= { diffServMultiFieldClfrEntry 15 }
```

```
--
```

```
-- Meters
```

```
--
```

```
diffServMeter          OBJECT IDENTIFIER ::= { diffServMIBObjects 3 }
```

```
--
```

```
-- This MIB supports a variety of Meters. It includes a specific
```

```

-- definition for Token Bucket Meter, which are but one type of
-- specification. Other metering parameter sets can be defined in other
-- MIBs.

-- Multiple meter elements may be logically cascaded using their
-- diffServMeterSucceedNext and diffServMeterFailNext pointers if
-- required. One example of this might be for an AF PHB implementation
-- that uses multiple level conformance meters.

-- Cascading of individual meter elements in the MIB is intended to be
-- functionally equivalent to multiple level conformance determination
-- of a packet. The sequential nature of the representation is merely
-- a notational convenience for this MIB.

-- srTCM meters (RFC 2697) can be specified using two sets of
-- diffServMeterEntry and diffServTBParamEntry. The first set specifies
-- the Committed Information Rate and Committed Burst Size
-- token-bucket. The second set specifies the Excess Burst Size
-- token-bucket.

-- trTCM meters (RFC 2698) can be specified using two sets of
-- diffServMeterEntry and diffServTBParamEntry. The first set specifies
-- the Committed Information Rate and Committed Burst Size
-- token-bucket. The second set specifies the Peak Information Rate
-- and Peak Burst Size token-bucket.

-- tswTCM meters (RFC 2859) can be specified using two sets of
-- diffServMeterEntry and diffServTBParamEntry. The first set specifies
-- the Committed Target Rate token-bucket. The second set specifies

```

```

Baker, et. al.           Standards Track           [Page 52]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

-- the Peak Target Rate token-bucket. diffServTBParamInterval in each
-- token bucket reflects the Average Interval.
--

```

```
diffServMeterNextFree OBJECT-TYPE
```

```

SYNTAX      IndexIntegerNextFree
MAX-ACCESS  read-only
STATUS      current

```

```
DESCRIPTION
```

```

"This object contains an unused value for diffServMeterId, or a
zero to indicate that none exist."

```

```
::= { diffServMeter 1 }
```

```
diffServMeterTable OBJECT-TYPE
```

```

SYNTAX      SEQUENCE OF DiffServMeterEntry
MAX-ACCESS  not-accessible
STATUS      current

```

```
DESCRIPTION
```

```

"This table enumerates specific meters that a system may use to
police a stream of traffic. The traffic stream to be metered is
determined by the Differentiated Services Functional Data Path
Element(s) upstream of the meter i.e. by the object(s) that point
to each entry in this table. This may include all traffic on an
interface.

```

```

Specific meter details are to be found in table entry referenced
by diffServMeterSpecific."

```

```
::= { diffServMeter 2 }
```

```
diffServMeterEntry OBJECT-TYPE
```

```

SYNTAX      DiffServMeterEntry
MAX-ACCESS  not-accessible
STATUS      current

```

```
DESCRIPTION
```

```

"An entry in the meter table describes a single conformance level
of a meter."

```

```

INDEX { diffServMeterId }
::= { diffServMeterTable 1 }

```

```
DiffServMeterEntry ::= SEQUENCE {
```

```

diffServMeterId      IndexInteger,
diffServMeterSucceedNext  RowPointer,
diffServMeterFailNext  RowPointer,

```

```

diffServMeterSpecific      RowPointer,
diffServMeterStorage      StorageType,
diffServMeterStatus       RowStatus
}

```

Baker, et. al. Standards Track [Page 53]
RFC 3289 Differentiated Services MIB May 2002

```

diffServMeterId OBJECT-TYPE
SYNTAX      IndexInteger
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An index that enumerates the Meter entries. Managers obtain new
    values for row creation in this table by reading
    diffServMeterNextFree."
 ::= { diffServMeterEntry 1 }

```

```

diffServMeterSucceedNext OBJECT-TYPE
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "If the traffic does conform, this selects the next
    Differentiated Services Functional Data Path element to handle
    traffic for this data path. This RowPointer should point to an
    instance of one of:
        diffServClfrEntry
        diffServMeterEntry
        diffServActionEntry
        diffServAlgDropEntry
        diffServQEntry

    A value of zeroDotZero in this attribute indicates that no
    further Differentiated Services treatment is performed on traffic
    of this data path. The use of zeroDotZero is the normal usage for
    the last functional data path element of the current data path.

    Setting this to point to a target that does not exist results in
    an inconsistentValue error. If the row pointed to is removed or
    becomes inactive by other means, the treatment is as if this
    attribute contains a value of zeroDotZero."
DEFVAL      { zeroDotZero }
 ::= { diffServMeterEntry 2 }

```

```

diffServMeterFailNext OBJECT-TYPE
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "If the traffic does not conform, this selects the next
    Differentiated Services Functional Data Path element to handle
    traffic for this data path. This RowPointer should point to an
    instance of one of:
        diffServClfrEntry
        diffServMeterEntry

```

Baker, et. al. Standards Track [Page 54]
RFC 3289 Differentiated Services MIB May 2002

```

diffServActionEntry
diffServAlgDropEntry
diffServQEntry

```

A value of zeroDotZero in this attribute indicates no further Differentiated Services treatment is performed on traffic of this data path. The use of zeroDotZero is the normal usage for the last functional data path element of the current data path.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or

becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."
DEFVAL { zeroDotZero }
::= { diffServMeterEntry 3 }

diffServMeterSpecific OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This indicates the behavior of the meter by pointing to an entry containing detailed parameters. Note that entries in that specific table must be managed explicitly.

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

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the meter always succeeds."

::= { diffServMeterEntry 4 }

diffServMeterStorage OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

DEFVAL { nonVolatile }

::= { diffServMeterEntry 5 }

diffServMeterStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create

Baker, et. al. Standards Track [Page 55]

RFC 3289 Differentiated Services MIB May 2002

STATUS current

DESCRIPTION

"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."

::= { diffServMeterEntry 6 }

--
-- Token Bucket Parameter Table
--

diffServTBParam OBJECT IDENTIFIER ::= { diffServMIBObjects 4 }

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

-- Note that an entry in the Token Bucket Parameter Table can be shared by multiple diffServMeterTable entries.
--

diffServTBParamNextFree OBJECT-TYPE

SYNTAX IndexIntegerNextFree
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"This object contains an unused value for diffServTBParamId, or a zero to indicate that none exist."

::= { diffServTBParam 1 }

diffServTBParamTable OBJECT-TYPE

SYNTAX SEQUENCE OF DiffServTBParamEntry
MAX-ACCESS not-accessible

```

STATUS      current
DESCRIPTION
  "This table enumerates a single set of token bucket meter
  parameters that a system may use to police a stream of traffic.
  Such meters are modeled here as having a single rate and a single
  burst size. Multiple entries are used when multiple rates/burst
  sizes are needed."
 ::= { diffServTBParam 2 }

```

```

diffServTBParamEntry OBJECT-TYPE
  SYNTAX      DiffServTBParamEntry
  MAX-ACCESS  not-accessible
  STATUS      current

```

```

Baker, et. al.           Standards Track           [Page 56]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

DESCRIPTION
  "An entry that describes a single set of token bucket
  parameters."
INDEX { diffServTBParamId }
 ::= { diffServTBParamTable 1 }

```

```

DiffServTBParamEntry ::= SEQUENCE {
  diffServTBParamId      IndexInteger,
  diffServTBParamType    AutonomousType,
  diffServTBParamRate    Unsigned32,
  diffServTBParamBurstSize  BurstSize,
  diffServTBParamInterval Unsigned32,
  diffServTBParamStorage  StorageType,
  diffServTBParamStatus  RowStatus
}

```

```

diffServTBParamId OBJECT-TYPE
  SYNTAX      IndexInteger
  MAX-ACCESS  not-accessible
  STATUS      current
DESCRIPTION
  "An index that enumerates the Token Bucket Parameter entries.
  Managers obtain new values for row creation in this table by
  reading diffServTBParamNextFree."
 ::= { diffServTBParamEntry 1 }

```

```

diffServTBParamType OBJECT-TYPE
  SYNTAX      AutonomousType
  MAX-ACCESS  read-create
  STATUS      current
DESCRIPTION
  "The Metering algorithm associated with the Token Bucket
  parameters. zeroDotZero indicates this is unknown.

  Standard values for generic algorithms:
  diffServTBParamSimpleTokenBucket, diffServTBParamAvgRate,
  diffServTBParamSrTCMblind, diffServTBParamSrTCMAware,
  diffServTBParamTrTCMblind, diffServTBParamTrTCMAware, and
  diffServTBParamTswTCM are specified in this MIB as OBJECT-
  IDENTITIES; additional values may be further specified in other
  MIBs."
 ::= { diffServTBParamEntry 2 }

```

```

diffServTBParamRate OBJECT-TYPE
  SYNTAX      Unsigned32 (1..4294967295)
  UNITS       "kilobits per second"
  MAX-ACCESS  read-create
  STATUS      current

```

```

Baker, et. al.           Standards Track           [Page 57]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

DESCRIPTION
  "The token-bucket rate, in kilobits per second (kbps). This
  attribute is used for:

```



```
1. CIR in RFC 2697 for srTCM
2. CIR and PIR in RFC 2698 for trTCM
3. CTR and PTR in RFC 2859 for TSWTCM
4. AverageRate in RFC 3290."
 ::= { diffServTBParamEntry 3 }
```

```
diffServTBParamBurstSize OBJECT-TYPE
```

```
SYNTAX      BurstSize
UNITS       "Bytes"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The maximum number of bytes in a single transmission burst. This
    attribute is used for:
    1. CBS and EBS in RFC 2697 for srTCM
    2. CBS and PBS in RFC 2698 for trTCM
    3. Burst Size in RFC 3290."
 ::= { diffServTBParamEntry 4 }
```

```
diffServTBParamInterval OBJECT-TYPE
```

```
SYNTAX      Unsigned32 (1..4294967295)
UNITS       "microseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The time interval used with the token bucket. For:
    1. Average Rate Meter, the Informal Differentiated Services Model
       section 5.2.1, - Delta.
    2. Simple Token Bucket Meter, the Informal Differentiated
       Services Model section 5.1, - time interval t.
    3. RFC 2859 TSWTCM, - AVG_INTERVAL.
    4. RFC 2697 srTCM, RFC 2698 trTCM, - token bucket update time
       interval."
 ::= { diffServTBParamEntry 5 }
```

```
diffServTBParamStorage OBJECT-TYPE
```

```
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The storage type for this conceptual row. Conceptual rows
    having the value 'permanent' need not allow write-access to any
    columnar objects in the row."
DEFVAL { nonVolatile }
 ::= { diffServTBParamEntry 6 }
```

```
Baker, et. al.           Standards Track           [Page 58]
```

```
RFC 3289                 Differentiated Services MIB           May 2002
```

```
diffServTBParamStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The status of this conceptual row. All writable objects in this
    row may be modified at any time. Setting this variable to
    'destroy' when the MIB contains one or more RowPointers pointing
    to it results in destruction being delayed until the row is no
    longer used."
 ::= { diffServTBParamEntry 7 }
```

```
--
-- OIDs for diffServTBParamType definitions.
--
```

```
diffServTBMeters OBJECT IDENTIFIER ::= { diffServMIBAdmin 1 }
```

```
diffServTBParamSimpleTokenBucket OBJECT-IDENTITY
```

```
STATUS      current
DESCRIPTION
    "Two Parameter Token Bucket Meter as described in the Informal
    Differentiated Services Model section 5.2.3."
 ::= { diffServTBMeters 1 }
```

```
diffServTBParamAvgRate OBJECT-IDENTITY
```

```
STATUS      current
```

DESCRIPTION
"Average Rate Meter as described in the Informal Differentiated
Services Model section 5.2.1."
 ::= { diffServTBMeters 2 }

diffServTBParamSrTCMBlind OBJECT-IDENTITY
STATUS current
DESCRIPTION
"Single Rate Three Color Marker Metering as defined by RFC 2697,
in the `Color Blind' mode as described by the RFC."
REFERENCE
"RFC 2697"
 ::= { diffServTBMeters 3 }

diffServTBParamSrTCMAware OBJECT-IDENTITY
STATUS current
DESCRIPTION
"Single Rate Three Color Marker Metering as defined by RFC 2697,
in the `Color Aware' mode as described by the RFC."
REFERENCE
"RFC 2697"

Baker, et. al. Standards Track [Page 59]
RFC 3289 Differentiated Services MIB May 2002

::= { diffServTBMeters 4 }

diffServTBParamTrTCMBlind OBJECT-IDENTITY
STATUS current
DESCRIPTION
"Two Rate Three Color Marker Metering as defined by RFC 2698, in
the `Color Blind' mode as described by the RFC."
REFERENCE
"RFC 2698"
 ::= { diffServTBMeters 5 }

diffServTBParamTrTCMAware OBJECT-IDENTITY
STATUS current
DESCRIPTION
"Two Rate Three Color Marker Metering as defined by RFC 2698, in
the `Color Aware' mode as described by the RFC."
REFERENCE
"RFC 2698"
 ::= { diffServTBMeters 6 }

diffServTBParamTswTCM OBJECT-IDENTITY
STATUS current
DESCRIPTION
"Time Sliding Window Three Color Marker Metering as defined by
RFC 2859."
REFERENCE
"RFC 2859"
 ::= { diffServTBMeters 7 }

--
-- Actions
--

diffServAction OBJECT IDENTIFIER ::= { diffServMIBObjects 5 }

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

diffServActionNextFree OBJECT-TYPE
SYNTAX IndexIntegerNextFree
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object contains an unused value for diffServActionId, or a
zero to indicate that none exist."
 ::= { diffServAction 1 }

```
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, traffic exiting from a meter may be counted, marked, and
        potentially dropped before entering a queue.

        Specific actions are indicated by diffServActionSpecific which
        points to an entry of a specific action type parameterizing the
        action in detail."
    ::= { diffServAction 2 }

diffServActionEntry OBJECT-TYPE
    SYNTAX      DiffServActionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in the action table allows description of one
        specific action to be applied to traffic."
    INDEX { diffServActionId }
    ::= { diffServActionTable 1 }

DiffServActionEntry ::= SEQUENCE {
    diffServActionId      IndexInteger,
    diffServActionInterface  InterfaceIndexOrZero,
    diffServActionNext    RowPointer,
    diffServActionSpecific  RowPointer,
    diffServActionStorage  StorageType,
    diffServActionStatus   RowStatus
}

diffServActionId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the Action entries. Managers obtain
        new values for row creation in this table by reading
        diffServActionNextFree."
    ::= { diffServActionEntry 1 }

diffServActionInterface OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-create
    STATUS      current
```

DESCRIPTION

"The interface index (value of ifIndex) that this action occurs on. This may be derived from the diffServDataPathStartEntry's index by extension through the various RowPointers. However, as this may be difficult for a network management station, it is placed here as well. If this is indeterminate, the value is zero.

This is of especial relevance when reporting the counters which may apply to traffic crossing an interface:

```
diffServCountActOctets,
diffServCountActPkts,
diffServAlgDropOctets,
diffServAlgDropPkts,
diffServAlgRandomDropOctets, and
diffServAlgRandomDropPkts.
```

It is also especially relevant to the queue and scheduler which may be subsequently applied."

```
::= { diffServActionEntry 2 }
```

```
diffServActionNext OBJECT-TYPE
```

```
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

"This selects the next Differentiated Services Functional Data Path Element to handle traffic for this data path. This RowPointer should point to an instance of one of:

```
diffServClfrEntry
diffServMeterEntry
diffServActionEntry
diffServAlgDropEntry
diffServQEntry
```

A value of zeroDotZero in this attribute indicates no further Differentiated Services treatment is performed on traffic of this data path. The use of zeroDotZero is the normal usage for the last functional data path element of the current data path.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

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

```
diffServActionSpecific OBJECT-TYPE
```

Baker, et. al.

Standards Track

[Page 62]

RFC 3289

Differentiated Services MIB

May 2002

```
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

"A pointer to an object instance providing additional information for the type of action indicated by this action table entry.

For the standard actions defined by this MIB module, this should point to either a diffServDscpMarkActEntry or a diffServCountActEntry. For other actions, it may point to an object instance defined in some other MIB.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the Meter should be treated as if it were not present. This may lead to incorrect policy behavior."

```
::= { diffServActionEntry 4 }
```

```
diffServActionStorage OBJECT-TYPE
```

```
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

```
DEFVAL { nonVolatile }
::= { diffServActionEntry 5 }
```

```
diffServActionStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."

```
::= { diffServActionEntry 6 }
```

```
-- DSCP Mark Action Table
```

```
--
-- Rows of this table are pointed to by diffServActionSpecific to
-- provide detailed parameters specific to the DSCP Mark action.
--
-- A single entry in this table can be shared by multiple
```

```
Baker, et. al.           Standards Track           [Page 63]
RFC 3289                 Differentiated Services MIB       May 2002
```

```
-- diffServActionTable entries.
--
```

```
diffServDscpMarkActTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiffServDscpMarkActEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table enumerates specific DSCPs used for marking or
        remarking the DSCP field of IP packets. The entries of this table
        may be referenced by a diffServActionSpecific attribute."
    ::= { diffServAction 3 }
```

```
diffServDscpMarkActEntry OBJECT-TYPE
    SYNTAX      DiffServDscpMarkActEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the DSCP mark action table that describes a single
        DSCP used for marking."
    INDEX { diffServDscpMarkActDscp }
    ::= { diffServDscpMarkActTable 1 }
```

```
DiffServDscpMarkActEntry ::= SEQUENCE {
    diffServDscpMarkActDscp      Dscp
}
```

```
diffServDscpMarkActDscp OBJECT-TYPE
    SYNTAX      Dscp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The DSCP that this Action will store into the DSCP field of the
        subject. It is quite possible that the only packets subject to
        this Action are already marked with this DSCP. Note also that
        Differentiated Services processing may result in packet being
        marked on both ingress to a network and on egress from it, and
        that ingress and egress can occur in the same router."
    ::= { diffServDscpMarkActEntry 1 }
```

```
--
-- Count Action Table
--
-- Because the MIB structure allows multiple cascading
-- diffServActionEntry be used to describe multiple actions for a data
-- path, the counter became an optional action type. In normal
-- implementation, either a data path has counters or it does not, as
-- opposed to being configurable. The management entity may choose to
```

```
Baker, et. al.           Standards Track           [Page 64]
RFC 3289                 Differentiated Services MIB       May 2002
```

```
-- read the counter or not. Hence it is recommended for implementation
-- that have counters to always configure the count action as the first
-- of multiple actions.
--
```

```
diffServCountActNextFree OBJECT-TYPE
    SYNTAX      IndexIntegerNextFree
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains an unused value for
```

```
diffServCountActId, or a zero to indicate that none exist."  
 ::= { diffServAction 4 }
```

```
diffServCountActTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF DiffServCountActEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
 "This table contains counters for all the traffic passing through  
 an action element."  
 ::= { diffServAction 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  
 traffic counters."  
 INDEX { diffServCountActId }  
 ::= { diffServCountActTable 1 }
```

```
DiffServCountActEntry ::= SEQUENCE {  
 diffServCountActId IndexInteger,  
 diffServCountActOctets Counter64,  
 diffServCountActPkts Counter64,  
 diffServCountActStorage StorageType,  
 diffServCountActStatus RowStatus  
 }
```

```
diffServCountActId OBJECT-TYPE  
 SYNTAX IndexInteger  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
 "An index that enumerates the Count Action entries. Managers  
 obtain new values for row creation in this table by reading
```

Baker, et. al. Standards Track [Page 65]

RFC 3289 Differentiated Services MIB May 2002

```
diffServCountActNextFree."  
 ::= { diffServCountActEntry 1 }
```

```
diffServCountActOctets OBJECT-TYPE  
 SYNTAX Counter64  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The number of octets at the Action data path element.  
  
 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 on the  
 relevant interface."  
 ::= { diffServCountActEntry 2 }
```

```
diffServCountActPkts OBJECT-TYPE  
 SYNTAX Counter64  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The number of packets at the Action data path element.  
  
 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 on the  
 relevant interface."  
 ::= { diffServCountActEntry 3 }
```

```
diffServCountActStorage OBJECT-TYPE  
 SYNTAX StorageType  
 MAX-ACCESS read-create  
 STATUS current  
 DESCRIPTION  
 "The storage type for this conceptual row. Conceptual rows
```

having the value 'permanent' need not allow write-access to any
columnar objects in the row."
DEFVAL { nonVolatile }
::= { diffServCountActEntry 4 }

diffServCountActStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The status of this conceptual row. All writable objects in this
row may be modified at any time. Setting this variable to
'destroy' when the MIB contains one or more RowPointers pointing

Baker, et. al. Standards Track [Page 66]

RFC 3289 Differentiated Services MIB May 2002

to it results in destruction being delayed until the row is no
longer used."
::= { diffServCountActEntry 5 }

--
-- Algorithmic Drop Table
--

diffServAlgDrop OBJECT IDENTIFIER ::= { diffServMIBObjects 6 }

diffServAlgDropNextFree OBJECT-TYPE
SYNTAX IndexIntegerNextFree
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object contains an unused value for diffServAlgDropId, or a
zero to indicate that none exist."
::= { diffServAlgDrop 1 }

diffServAlgDropTable OBJECT-TYPE
SYNTAX SEQUENCE OF DiffServAlgDropEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The algorithmic drop table contains entries describing an
element that drops packets according to some algorithm."
::= { diffServAlgDrop 2 }

diffServAlgDropEntry OBJECT-TYPE
SYNTAX DiffServAlgDropEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"An entry describes a process that drops packets according to
some algorithm. Further details of the algorithm type are to be
found in diffServAlgDropType and with more detail parameter entry
pointed to by diffServAlgDropSpecific when necessary."
INDEX { diffServAlgDropId }
::= { diffServAlgDropTable 1 }

DiffServAlgDropEntry ::= SEQUENCE {
diffServAlgDropId IndexInteger,
diffServAlgDropType INTEGER,
diffServAlgDropNext RowPointer,
diffServAlgDropQMeasure RowPointer,
diffServAlgDropQThreshold Unsigned32,
diffServAlgDropSpecific RowPointer,
diffServAlgDropOctets Counter64,

Baker, et. al. Standards Track [Page 67]

RFC 3289 Differentiated Services MIB May 2002

diffServAlgDropPkts Counter64,
diffServAlgRandomDropOctets Counter64,
diffServAlgRandomDropPkts Counter64,

```

diffServAlgDropStorage      StorageType,
diffServAlgDropStatus      RowStatus
}

diffServAlgDropId OBJECT-TYPE
SYNTAX      IndexInteger
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An index that enumerates the Algorithmic Dropper entries.
    Managers obtain new values for row creation in this table by
    reading diffServAlgDropNextFree."
 ::= { diffServAlgDropEntry 1 }

diffServAlgDropType OBJECT-TYPE
SYNTAX      INTEGER {
                other(1),
                tailDrop(2),
                headDrop(3),
                randomDrop(4),
                alwaysDrop(5)
            }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The type of algorithm used by this dropper. The value other(1)
    requires further specification in some other MIB module.

    In the tailDrop(2) algorithm, diffServAlgDropQThreshold
    represents the maximum depth of the queue, pointed to by
    diffServAlgDropQMeasure, beyond which all newly arriving packets
    will be dropped.

    In the headDrop(3) algorithm, if a packet arrives when the
    current depth of the queue, pointed to by
    diffServAlgDropQMeasure, is at diffServAlgDropQThreshold, packets
    currently at the head of the queue are dropped to make room for
    the new packet to be enqueued at the tail of the queue.

    In the randomDrop(4) algorithm, on packet arrival, an Active
    Queue Management algorithm is executed which may randomly drop a
    packet. This algorithm may be proprietary, and it may drop either
    the arriving packet or another packet in the queue.
    diffServAlgDropSpecific points to a diffServRandomDropEntry that
    describes the algorithm. For this algorithm,

```

Baker, et. al. Standards Track [Page 68]
RFC 3289 Differentiated Services MIB May 2002

diffServAlgDropQThreshold is understood to be the absolute maximum size of the queue and additional parameters are described in diffServRandomDropTable.

The alwaysDrop(5) algorithm is as its name specifies; always drop. In this case, the other configuration values in this Entry are not meaningful; There is no useful 'next' processing step, there is no queue, and parameters describing the queue are not useful. Therefore, diffServAlgDropNext, diffServAlgDropMeasure, and diffServAlgDropSpecific are all zeroDotZero."
 ::= { diffServAlgDropEntry 2 }

```

diffServAlgDropNext OBJECT-TYPE
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This selects the next Differentiated Services Functional Data
    Path Element to handle traffic for this data path. This
    RowPointer should point to an instance of one of:
        diffServClfrEntry
        diffServMeterEntry
        diffServActionEntry
        diffServQEntry

    A value of zeroDotZero in this attribute indicates no further
    Differentiated Services treatment is performed on traffic of this

```


data path. The use of zeroDotZero is the normal usage for the last functional data path element of the current data path.

When diffServAlgDropType is alwaysDrop(5), this object is ignored.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

```
::= { diffServAlgDropEntry 3 }
```

diffServAlgDropQMeasure OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Points to an entry in the diffServQTable to indicate the queue that a drop algorithm is to monitor when deciding whether to drop a packet. If the row pointed to does not exist, the algorithmic dropper element is considered inactive.

Baker, et. al.

Standards Track

[Page 69]

RFC 3289

Differentiated Services MIB

May 2002

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

```
::= { diffServAlgDropEntry 4 }
```

diffServAlgDropQThreshold OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)
UNITS "Bytes"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"A threshold on the depth in bytes of the queue being measured at which a trigger is generated to the dropping algorithm, unless diffServAlgDropType is alwaysDrop(5) where this object is ignored.

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

```
::= { diffServAlgDropEntry 5 }
```

diffServAlgDropSpecific OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Points to a table entry that provides further detail regarding a drop algorithm.

Entries with diffServAlgDropType equal to other(1) may have this point to a table defined in another MIB module.

Entries with diffServAlgDropType equal to randomDrop(4) must have this point to an entry in diffServRandomDropTable.

For all other algorithms specified in this MIB, this should take the value zeroDotZero.

The diffServAlgDropType is authoritative for the type of the drop algorithm and the specific parameters for the drop algorithm needs to be evaluated based on the diffServAlgDropType.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

```
::= { diffServAlgDropEntry 6 }
```

```
diffServAlgDropOctets OBJECT-TYPE
```

```
SYNTAX Counter64
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of octets that have been deterministically dropped by
this drop process.
```

```
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 on the
relevant interface."
```

```
::= { diffServAlgDropEntry 7 }
```

```
diffServAlgDropPkts OBJECT-TYPE
```

```
SYNTAX Counter64
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of packets that have been deterministically dropped
by this drop process.
```

```
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 on the
relevant interface."
```

```
::= { diffServAlgDropEntry 8 }
```

```
diffServAlgRandomDropOctets OBJECT-TYPE
```

```
SYNTAX Counter64
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of octets that have been randomly dropped by this
drop process. This counter applies, therefore, only to random
droppers.
```

```
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 on the
relevant interface."
```

```
::= { diffServAlgDropEntry 9 }
```

```
diffServAlgRandomDropPkts OBJECT-TYPE
```

```
SYNTAX Counter64
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of packets that have been randomly dropped by this
drop process. This counter applies, therefore, only to random
droppers.
```

```
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 on the
relevant interface."
```

```
::= { diffServAlgDropEntry 10 }
```

```
diffServAlgDropStorage OBJECT-TYPE
```

```
SYNTAX StorageType
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The storage type for this conceptual row. Conceptual rows
having the value 'permanent' need not allow write-access to any
```

```
    columnar objects in the row."
DEFVAL { nonVolatile }
 ::= { diffServAlgDropEntry 11 }
```

```
diffServAlgDropStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

```
"The status of this conceptual row. All writable objects in this
row may be modified at any time. Setting this variable to
'destroy' when the MIB contains one or more RowPointers pointing
to it results in destruction being delayed until the row is no
longer used."
```

```
 ::= { diffServAlgDropEntry 12 }
```

```
--
```

```
-- Random Drop Table
```

```
--
```

```
diffServRandomDropNextFree OBJECT-TYPE
```

```
SYNTAX      IndexIntegerNextFree
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
```

```
"This object contains an unused value for diffServRandomDropId,
or a zero to indicate that none exist."
```

```
 ::= { diffServAlgDrop 3 }
```

```
Baker, et. al.
```

```
Standards Track
```

```
[Page 72]
```

```
RFC 3289
```

```
Differentiated Services MIB
```

```
May 2002
```

```
diffServRandomDropTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF DiffServRandomDropEntry
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
```

```
"The random drop table contains entries describing a process that
drops packets randomly. Entries in this table are pointed to by
diffServAlgDropSpecific."
```

```
 ::= { diffServAlgDrop 4 }
```

```
diffServRandomDropEntry OBJECT-TYPE
```

```
SYNTAX      DiffServRandomDropEntry
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
```

```
"An entry describes a process that drops packets according to a
random algorithm."
```

```
INDEX { diffServRandomDropId }
```

```
 ::= { diffServRandomDropTable 1 }
```

```
DiffServRandomDropEntry ::= SEQUENCE {
```

```
diffServRandomDropId      IndexInteger,
diffServRandomDropMinThreshBytes  Unsigned32,
diffServRandomDropMinThreshPkts  Unsigned32,
diffServRandomDropMaxThreshBytes  Unsigned32,
diffServRandomDropMaxThreshPkts  Unsigned32,
diffServRandomDropProbMax      Unsigned32,
diffServRandomDropWeight      Unsigned32,
diffServRandomDropSamplingRate  Unsigned32,
diffServRandomDropStorage      StorageType,
diffServRandomDropStatus      RowStatus
```

```
}
```

```
diffServRandomDropId OBJECT-TYPE
```

```
SYNTAX      IndexInteger
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
```

```
"An index that enumerates the Random Drop entries. Managers
obtain new values for row creation in this table by reading
diffServRandomDropNextFree."
```

```
 ::= { diffServRandomDropEntry 1 }
```

diffServRandomDropMinThreshBytes OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
UNITS "bytes"
MAX-ACCESS read-create
STATUS current

Baker, et. al. Standards Track [Page 73]
RFC 3289 Differentiated Services MIB May 2002

DESCRIPTION

"The average queue depth in bytes, beyond which traffic has a non-zero probability of being dropped. Changes in this variable may or may not be reflected in the reported value of diffServRandomDropMinThreshPkts."

::= { diffServRandomDropEntry 2 }

diffServRandomDropMinThreshPkts OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
UNITS "packets"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The average queue depth in packets, beyond which traffic has a non-zero probability of being dropped. Changes in this variable may or may not be reflected in the reported value of diffServRandomDropMinThreshBytes."

::= { diffServRandomDropEntry 3 }

diffServRandomDropMaxThreshBytes OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
UNITS "bytes"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The average queue depth beyond which traffic has a probability indicated by diffServRandomDropProbMax of being dropped or marked. Note that this differs from the physical queue limit, which is stored in diffServAlgDropQThreshold. Changes in this variable may or may not be reflected in the reported value of diffServRandomDropMaxThreshPkts."

::= { diffServRandomDropEntry 4 }

diffServRandomDropMaxThreshPkts OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
UNITS "packets"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The average queue depth beyond which traffic has a probability indicated by diffServRandomDropProbMax of being dropped or marked. Note that this differs from the physical queue limit, which is stored in diffServAlgDropQThreshold. Changes in this variable may or may not be reflected in the reported value of diffServRandomDropMaxThreshBytes."

::= { diffServRandomDropEntry 5 }

diffServRandomDropProbMax OBJECT-TYPE

Baker, et. al. Standards Track [Page 74]
RFC 3289 Differentiated Services MIB May 2002

SYNTAX Unsigned32 (0..1000)
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The worst case random drop probability, expressed in drops per thousand packets.

For example, if in the worst case every arriving packet may be dropped (100%) for a period, this has the value 1000. Alternatively, if in the worst case only one percent (1%) of traffic may be dropped, it has the value 10."

```

 ::= { diffServRandomDropEntry 6 }

diffServRandomDropWeight OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65536)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The weighting of past history in affecting the Exponentially
        Weighted Moving Average function that calculates the current
        average queue depth. The equation uses
        diffServRandomDropWeight/65536 as the coefficient for the new
        sample in the equation, and (65536 -
        diffServRandomDropWeight)/65536 as the coefficient of the old
        value.

        Implementations may limit the values of diffServRandomDropWeight
        to a subset of the possible range of values, such as powers of
        two. Doing this would facilitate implementation of the
        Exponentially Weighted Moving Average using shift instructions or
        registers."
    ::= { diffServRandomDropEntry 7 }

diffServRandomDropSamplingRate OBJECT-TYPE
    SYNTAX      Unsigned32 (0..1000000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of times per second the queue is sampled for queue
        average calculation. A value of zero is used to mean that the
        queue is sampled approximately each time a packet is enqueued (or
        dequeued)."
    ::= { diffServRandomDropEntry 8 }

diffServRandomDropStorage OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current

```

Baker, et. al. Standards Track [Page 75]

RFC 3289 Differentiated Services MIB May 2002

```

DESCRIPTION
    "The storage type for this conceptual row. Conceptual rows
    having the value 'permanent' need not allow write-access to any
    columnar objects in the row."
DEFVAL { nonVolatile }
 ::= { diffServRandomDropEntry 9 }

```

```

diffServRandomDropStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this conceptual row. All writable objects in this
        row may be modified at any time. Setting this variable to
        'destroy' when the MIB contains one or more RowPointers pointing
        to it results in destruction being delayed until the row is no
        longer used."
    ::= { diffServRandomDropEntry 10 }

```

```

--
-- Queue Table
--

```

```

diffServQueue OBJECT IDENTIFIER ::= { diffServMIBObjects 7 }

```

```

--
-- An entry of diffServQueueTable represents a FIFO queue Differentiated
-- Services Functional Data Path element as described in the Informal
-- Differentiated Services Model section 7.1.1. Note that the
-- specification of scheduling parameters for a queue as part of the
-- input to a scheduler functional data path element as described in
-- the Informal Differentiated Services Model section 7.1.2. This
-- allows building of hierarchical queuing/scheduling. A queue
-- therefore has these attributes:
--

```

```

-- 1. Which scheduler will service this queue, diffServQNext.
-- 2. How the scheduler will service this queue, with respect
--    to all the other queues the same scheduler needs to service,
--    diffServQMinRate.
--
-- Note that upstream Differentiated Services Functional Data Path
-- elements may point to a shared diffServQTable entry as described
-- in the Informal Differentiated Services Model section 7.1.1.
--

```

```

diffServQNextFree OBJECT-TYPE
    SYNTAX      IndexIntegerNextFree
    MAX-ACCESS  read-only

```

```

Baker, et. al.           Standards Track           [Page 76]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

STATUS      current
DESCRIPTION
    "This object contains an unused value for diffServQId, or a zero
    to indicate that none exist."
 ::= { diffServQueue 1 }

```

```

diffServQTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiffServQEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Queue Table enumerates the individual queues. Note that the
        MIB models queuing systems as composed of individual queues, one
        per class of traffic, even though they may in fact be structured
        as classes of traffic scheduled using a common calendar queue, or
        in other ways."
    ::= { diffServQueue 2 }

```

```

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

```

```

DiffServQEntry ::= SEQUENCE {
    diffServQId      IndexInteger,
    diffServQNext    RowPointer,
    diffServQMinRate RowPointer,
    diffServQMaxRate RowPointer,
    diffServQStorage StorageType,
    diffServQStatus  RowStatus
}

```

```

diffServQId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the Queue entries. Managers obtain new
        values for row creation in this table by reading
        diffServQNextFree."
    ::= { diffServQEntry 1 }

```

```

diffServQNext OBJECT-TYPE

```

```

Baker, et. al.           Standards Track           [Page 77]
RFC 3289                 Differentiated Services MIB       May 2002

```

```

SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current

```

DESCRIPTION

"This selects the next Differentiated Services Scheduler. The RowPointer must point to a diffServSchedulerEntry.

A value of zeroDotZero in this attribute indicates an incomplete diffServQEntry instance. In such a case, the entry has no operational effect, since it has no parameters to give it meaning.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

::= { diffServQEntry 2 }

diffServQMinRate OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This RowPointer indicates the diffServMinRateEntry that the scheduler, pointed to by diffServQNext, should use to service this queue.

If the row pointed to is zeroDotZero, the minimum rate and priority is unspecified.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

::= { diffServQEntry 3 }

diffServQMaxRate OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This RowPointer indicates the diffServMaxRateEntry that the scheduler, pointed to by diffServQNext, should use to service this queue.

If the row pointed to is zeroDotZero, the maximum rate is the line speed of the interface.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

::= { diffServQEntry 4 }

diffServQStorage OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

DEFVAL { nonVolatile }

::= { diffServQEntry 5 }

diffServQStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."

```

 ::= { diffServQEntry 6 }

--
-- Scheduler Table
--

diffServScheduler OBJECT IDENTIFIER ::= { diffServMIBObjects 8 }

--
-- A Scheduler Entry represents a packet scheduler, such as a priority
-- scheduler or a WFQ scheduler. It provides flexibility for multiple
-- scheduling algorithms, each servicing multiple queues, to be used on
-- the same logical/physical interface.
--
-- Note that upstream queues or schedulers specify several of the
-- scheduler's parameters. These must be properly specified if the
-- scheduler is to behave as expected.
--
-- The diffServSchedulerMaxRate attribute specifies the parameters when
-- a scheduler's output is sent to another scheduler. This is used in
-- building hierarchical queues or schedulers.

Baker, et. al. Standards Track [Page 79]
RFC 3289 Differentiated Services MIB May 2002

--
-- More discussion of the scheduler functional data path element is in
-- the Informal Differentiated Services Model section 7.1.2.
--

diffServSchedulerNextFree OBJECT-TYPE
    SYNTAX IndexIntegerNextFree
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object contains an unused value for diffServSchedulerId, or
        a zero to indicate that none exist."
    ::= { diffServScheduler 1 }

diffServSchedulerTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DiffServSchedulerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The Scheduler Table enumerates packet schedulers. Multiple
        scheduling algorithms can be used on a given data path, with each
        algorithm described by one diffServSchedulerEntry."
    ::= { diffServScheduler 2 }

diffServSchedulerEntry OBJECT-TYPE
    SYNTAX DiffServSchedulerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the Scheduler Table describing a single instance of
        a scheduling algorithm."
    INDEX { diffServSchedulerId }
    ::= { diffServSchedulerTable 1 }

DiffServSchedulerEntry ::= SEQUENCE {
    diffServSchedulerId IndexInteger,
    diffServSchedulerNext RowPointer,
    diffServSchedulerMethod AutonomousType,
    diffServSchedulerMinRate RowPointer,
    diffServSchedulerMaxRate RowPointer,
    diffServSchedulerStorage StorageType,
    diffServSchedulerStatus RowStatus
}

diffServSchedulerId OBJECT-TYPE
    SYNTAX IndexInteger
    MAX-ACCESS not-accessible
    STATUS current

```


DESCRIPTION

"An index that enumerates the Scheduler entries. Managers obtain new values for row creation in this table by reading diffServSchedulerNextFree."

::= { diffServSchedulerEntry 1 }

diffServSchedulerNext OBJECT-TYPE

SYNTAX RowPointer
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"This selects the next Differentiated Services Functional Data Path Element to handle traffic for this data path. This normally is null (zeroDotZero), or points to a diffServSchedulerEntry or a diffServQEntry.

However, this RowPointer may also point to an instance of:

diffServClfrEntry,
 diffServMeterEntry,
 diffServActionEntry,
 diffServAlgDropEntry.

It would point another diffServSchedulerEntry when implementing multiple scheduler methods for the same data path, such as having one set of queues scheduled by WRR and that group participating in a priority scheduling system in which other queues compete with it in that way. It might also point to a second scheduler in a hierarchical scheduling system.

If the row pointed to is zeroDotZero, no further Differentiated Services treatment is performed on traffic of this data path.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this attribute contains a value of zeroDotZero."

DEFVAL { zeroDotZero }
 ::= { diffServSchedulerEntry 2 }

diffServSchedulerMethod OBJECT-TYPE

SYNTAX AutonomousType
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The scheduling algorithm used by this Scheduler. zeroDotZero indicates that this is unknown. Standard values for generic algorithms: diffServSchedulerPriority, diffServSchedulerWRR, and diffServSchedulerWFQ are specified in this MIB; additional values

may be further specified in other MIBs."

::= { diffServSchedulerEntry 3 }

diffServSchedulerMinRate OBJECT-TYPE

SYNTAX RowPointer
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"This RowPointer indicates the entry in diffServMinRateTable which indicates the priority or minimum output rate from this scheduler. This attribute is used only when there is more than one level of scheduler.

When it has the value zeroDotZero, it indicates that no minimum rate or priority is imposed.

Setting this to point to a target that does not exist results in an inconsistentValue error. If the row pointed to is removed or becomes inactive by other means, the treatment is as if this

```
    attribute contains a value of zeroDotZero."
DEFVAL      { zeroDotZero }
 ::= { diffServSchedulerEntry 4 }
```

```
diffServSchedulerMaxRate OBJECT-TYPE
```

```
SYNTAX      RowPointer
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

```
"This RowPointer indicates the entry in diffServMaxRateTable
which indicates the maximum output rate from this scheduler.
When more than one maximum rate applies (eg, when a multi-rate
shaper is in view), it points to the first of those rate entries.
This attribute is used only when there is more than one level of
scheduler.
```

```
When it has the value zeroDotZero, it indicates that no maximum
rate is imposed.
```

```
Setting this to point to a target that does not exist results in
an inconsistentValue error. If the row pointed to is removed or
becomes inactive by other means, the treatment is as if this
attribute contains a value of zeroDotZero."
```

```
DEFVAL      { zeroDotZero }
 ::= { diffServSchedulerEntry 5 }
```

```
diffServSchedulerStorage OBJECT-TYPE
```

```
SYNTAX      StorageType
MAX-ACCESS  read-create
```

```
Baker, et. al.
```

```
Standards Track
```

```
[Page 82]
```

```
RFC 3289
```

```
Differentiated Services MIB
```

```
May 2002
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The storage type for this conceptual row. Conceptual rows
having the value 'permanent' need not allow write-access to any
columnar objects in the row."
```

```
DEFVAL { nonVolatile }
 ::= { diffServSchedulerEntry 6 }
```

```
diffServSchedulerStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

```
"The status of this conceptual row. All writable objects in this
row may be modified at any time. Setting this variable to
'destroy' when the MIB contains one or more RowPointers pointing
to it results in destruction being delayed until the row is no
longer used."
```

```
 ::= { diffServSchedulerEntry 7 }
```

```
--
```

```
-- OIDs for diffServTBParamType definitions.
```

```
--
```

```
diffServSchedulers OBJECT IDENTIFIER ::= { diffServMIBAdmin 2 }
```

```
diffServSchedulerPriority OBJECT-IDENTITY
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"For use with diffServSchedulerMethod to indicate the Priority
scheduling method. This is defined as an algorithm in which the
presence of data in a queue or set of queues absolutely precludes
dequeue from another queue or set of queues of lower priority.
Note that attributes from diffServMinRateEntry of the
queues/schedulers feeding this scheduler are used when
determining the next packet to schedule."
```

```
 ::= { diffServSchedulers 1 }
```

```
diffServSchedulerWRR OBJECT-IDENTITY
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"For use with diffServSchedulerMethod to indicate the Weighted
Round Robin scheduling method, defined as any algorithm in which
```

a set of queues are visited in a fixed order, and varying amounts of traffic are removed from each queue in turn to implement an average output rate by class. Notice attributes from diffServMinRateEntry of the queues/schedulers feeding this scheduler are used when determining the next packet to schedule."

Baker, et. al. Standards Track [Page 83]
RFC 3289 Differentiated Services MIB May 2002

```
::= { diffServSchedulers 2 }
```

diffServSchedulerWFQ OBJECT-IDENTITY

STATUS current

DESCRIPTION

"For use with diffServSchedulerMethod to indicate the Weighted Fair Queuing scheduling method, defined as any algorithm in which a set of queues are conceptually visited in some order, to implement an average output rate by class. Notice attributes from diffServMinRateEntry of the queues/schedulers feeding this scheduler are used when determining the next packet to schedule."

```
::= { diffServSchedulers 3 }
```

--

-- Minimum Rate Parameters Table

--

-- The parameters used by a scheduler for its inputs or outputs are maintained separately from the Queue or Scheduler table entries for reusability reasons and so that they may be used by both queues and schedulers. This follows the approach for separation of data path elements from parameterization that is used throughout this MIB. Use of these Minimum Rate Parameter Table entries by Queues and Schedulers allows the modeling of hierarchical scheduling systems.

--

-- Specifically, a Scheduler has one or more inputs and one output. Any queue feeding a scheduler, or any scheduler which feeds a second scheduler, might specify a minimum transfer rate by pointing to an Minimum Rate Parameter Table entry.

--

-- The diffServMinRatePriority/Abs/Rel attributes are used as parameters to the work-conserving portion of a scheduler: "work-conserving" implies that the scheduler can continue to emit data as long as there is data available at its input(s). This has the effect of guaranteeing a certain priority relative to other scheduler inputs and/or a certain minimum proportion of the available output bandwidth. Properly configured, this means a certain minimum rate, which may be exceeded should traffic be available should there be spare bandwidth after all other classes have had opportunities to consume their own minimum rates.

--

diffServMinRateNextFree OBJECT-TYPE

SYNTAX IndexIntegerNextFree

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an unused value for diffServMinRateId, or a zero to indicate that none exist."

Baker, et. al. Standards Track [Page 84]
RFC 3289 Differentiated Services MIB May 2002

```
::= { diffServScheduler 3 }
```

diffServMinRateTable OBJECT-TYPE

SYNTAX SEQUENCE OF DiffServMinRateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Minimum Rate Parameters Table enumerates individual sets of scheduling parameter that can be used/reused by Queues and Schedulers."

```
::= { diffServScheduler 4 }
```

```

diffServMinRateEntry OBJECT-TYPE
    SYNTAX      DiffServMinRateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the Minimum Rate Parameters Table describes a single
        set of scheduling parameters for use by one or more queues or
        schedulers."
    INDEX { diffServMinRateId }
    ::= { diffServMinRateTable 1 }

DiffServMinRateEntry ::= SEQUENCE {
    diffServMinRateId      IndexInteger,
    diffServMinRatePriority Unsigned32,
    diffServMinRateAbsolute Unsigned32,
    diffServMinRateRelative Unsigned32,
    diffServMinRateStorage StorageType,
    diffServMinRateStatus RowStatus
}

diffServMinRateId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the Scheduler Parameter entries.
        Managers obtain new values for row creation in this table by
        reading diffServMinRateNextFree."
    ::= { diffServMinRateEntry 1 }

diffServMinRatePriority OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The priority of this input to the associated scheduler, relative

```

Baker, et. al. Standards Track [Page 85]
RFC 3289 Differentiated Services MIB May 2002

```

    to the scheduler's other inputs. A queue or scheduler with a
    larger numeric value will be served before another with a smaller
    numeric value."
    ::= { diffServMinRateEntry 2 }

```

```

diffServMinRateAbsolute OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    UNITS       "kilobits per second"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The minimum absolute rate, in kilobits/sec, that a downstream
        scheduler element should allocate to this queue. If the value is
        zero, then there is effectively no minimum rate guarantee. If the
        value is non-zero, the scheduler will assure the servicing of
        this queue to at least this rate.

        Note that this attribute value and that of
        diffServMinRateRelative are coupled: changes to one will affect
        the value of the other. They are linked by the following
        equation, in that setting one will change the other:

            diffServMinRateRelative =
                (diffServMinRateAbsolute*1000000)/ifSpeed

        or, if appropriate:

            diffServMinRateRelative = diffServMinRateAbsolute/ifHighSpeed"
    REFERENCE
        "ifSpeed, ifHighSpeed, Interface MIB, RFC 2863"
    ::= { diffServMinRateEntry 3 }

```

```

diffServMinRateRelative OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  read-create

```

STATUS current

DESCRIPTION

"The minimum rate that a downstream scheduler element should allocate to this queue, relative to the maximum rate of the interface as reported by ifSpeed or ifHighSpeed, in units of 1/1000 of 1. If the value is zero, then there is effectively no minimum rate guarantee. If the value is non-zero, the scheduler will assure the servicing of this queue to at least this rate.

Note that this attribute value and that of diffServMinRateAbsolute are coupled: changes to one will affect the value of the other. They are linked by the following equation, in that setting one will change the other:

Baker, et. al. Standards Track [Page 86]

RFC 3289 Differentiated Services MIB May 2002

```
diffServMinRateRelative =
    (diffServMinRateAbsolute*1000000)/ifSpeed
```

or, if appropriate:

```
diffServMinRateRelative = diffServMinRateAbsolute/ifHighSpeed"
```

REFERENCE

```
"ifSpeed, ifHighSpeed, Interface MIB, RFC 2863"
::= { diffServMinRateEntry 4 }
```

diffServMinRateStorage OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

DEFVAL { nonVolatile }

```
::= { diffServMinRateEntry 5 }
```

diffServMinRateStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."

```
::= { diffServMinRateEntry 6 }
```

--

-- Maximum Rate Parameter Table

--

-- The parameters used by a scheduler for its inputs or outputs are maintained separately from the Queue or Scheduler table entries for reusability reasons and so that they may be used by both queues and schedulers. This follows the approach for separation of data path elements from parameterization that is used throughout this MIB. Use of these Maximum Rate Parameter Table entries by Queues and Schedulers allows the modeling of hierarchical scheduling systems.

--

-- Specifically, a Scheduler has one or more inputs and one output. Any queue feeding a scheduler, or any scheduler which feeds a second scheduler, might specify a maximum transfer rate by pointing to a Maximum Rate Parameter Table entry. Multi-rate shapers, such as a

Baker, et. al. Standards Track [Page 87]

RFC 3289 Differentiated Services MIB May 2002

-- Dual Leaky Bucket algorithm, specify their rates using multiple
-- Maximum Rate Parameter Entries with the same diffServMaxRateId but
-- different diffServMaxRateLevels.

```
--
-- The diffServMaxRateLevel/Abs/Rel attributes are used as
-- parameters to the non-work-conserving portion of a scheduler:
-- non-work-conserving implies that the scheduler may sometimes not
-- emit a packet, even if there is data available at its input(s).
-- This has the effect of limiting the servicing of the queue/scheduler
-- input or output, in effect performing shaping of the packet stream
-- passing through the queue/scheduler, as described in the Informal
-- Differentiated Services Model section 7.2.
--
```

```
diffServMaxRateNextFree OBJECT-TYPE
    SYNTAX      IndexIntegerNextFree
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains an unused value for diffServMaxRateId, or a
        zero to indicate that none exist."
    ::= { diffServScheduler 5 }
```

```
diffServMaxRateTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiffServMaxRateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Maximum Rate Parameter Table enumerates individual sets of
        scheduling parameter that can be used/reused by Queues and
        Schedulers."
    ::= { diffServScheduler 6 }
```

```
diffServMaxRateEntry OBJECT-TYPE
    SYNTAX      DiffServMaxRateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the Maximum Rate Parameter Table describes a single
        set of scheduling parameters for use by one or more queues or
        schedulers."
    INDEX { diffServMaxRateId, diffServMaxRateLevel }
    ::= { diffServMaxRateTable 1 }
```

```
DiffServMaxRateEntry ::= SEQUENCE {
    diffServMaxRateId      IndexInteger,
    diffServMaxRateLevel   Unsigned32,
    diffServMaxRateAbsolute Unsigned32,
```

```
Baker, et. al.           Standards Track           [Page 88]
RFC 3289                 Differentiated Services MIB       May 2002
```

```
diffServMaxRateRelative   Unsigned32,
diffServMaxRateThreshold  BurstSize,
diffServMaxRateStorage    StorageType,
diffServMaxRateStatus     RowStatus
}
```

```
diffServMaxRateId OBJECT-TYPE
    SYNTAX      IndexInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that enumerates the Maximum Rate Parameter entries.
        Managers obtain new values for row creation in this table by
        reading diffServMaxRateNextFree."
    ::= { diffServMaxRateEntry 1 }
```

```
diffServMaxRateLevel OBJECT-TYPE
    SYNTAX      Unsigned32 (1..32)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that indicates which level of a multi-rate shaper is
        being given its parameters. A multi-rate shaper has some number
        of rate levels. Frame Relay's dual rate specification refers to a
        'committed' and an 'excess' rate; ATM's dual rate specification
        refers to a 'mean' and a 'peak' rate. This table is generalized
        to support an arbitrary number of rates. The committed or mean
```

rate is level 1, the peak rate (if any) is the highest level rate configured, and if there are other rates they are distributed in monotonically increasing order between them."
 ::= { diffServMaxRateEntry 2 }

diffServMaxRateAbsolute OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)
UNITS "kilobits per second"
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The maximum rate in kilobits/sec that a downstream scheduler element should allocate to this queue. If the value is zero, then there is effectively no maximum rate limit and that the scheduler should attempt to be work conserving for this queue. If the value is non-zero, the scheduler will limit the servicing of this queue to, at most, this rate in a non-work-conserving manner.

Note that this attribute value and that of diffServMaxRateRelative are coupled: changes to one will affect the value of the other. They are linked by the following

Baker, et. al. Standards Track [Page 89]
RFC 3289 Differentiated Services MIB May 2002

equation, in that setting one will change the other:

diffServMaxRateRelative =
(diffServMaxRateAbsolute*1000000)/ifSpeed

or, if appropriate:

diffServMaxRateRelative = diffServMaxRateAbsolute/ifHighSpeed"

REFERENCE

"ifSpeed, ifHighSpeed, Interface MIB, RFC 2863"
 ::= { diffServMaxRateEntry 3 }

diffServMaxRateRelative OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The maximum rate that a downstream scheduler element should allocate to this queue, relative to the maximum rate of the interface as reported by ifSpeed or ifHighSpeed, in units of 1/1000 of 1. If the value is zero, then there is effectively no maximum rate limit and the scheduler should attempt to be work conserving for this queue. If the value is non-zero, the scheduler will limit the servicing of this queue to, at most, this rate in a non-work-conserving manner.

Note that this attribute value and that of diffServMaxRateAbsolute are coupled: changes to one will affect the value of the other. They are linked by the following equation, in that setting one will change the other:

diffServMaxRateRelative =
(diffServMaxRateAbsolute*1000000)/ifSpeed

or, if appropriate:

diffServMaxRateRelative = diffServMaxRateAbsolute/ifHighSpeed"

REFERENCE

"ifSpeed, ifHighSpeed, Interface MIB, RFC 2863"
 ::= { diffServMaxRateEntry 4 }

diffServMaxRateThreshold OBJECT-TYPE

SYNTAX BurstSize
UNITS "Bytes"
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The number of bytes of queue depth at which the rate of a

multi-rate scheduler will increase to the next output rate. In the last conceptual row for such a shaper, this threshold is ignored and by convention is zero."

REFERENCE

"Adaptive rate Shaper, RFC 2963"

```
::= { diffServMaxRateEntry 5 }
```

diffServMaxRateStorage OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

DEFVAL { nonVolatile }

```
::= { diffServMaxRateEntry 6 }
```

diffServMaxRateStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."

```
::= { diffServMaxRateEntry 7 }
```

--

-- MIB Compliance statements.

--

diffServMIBCompliances OBJECT IDENTIFIER ::=

```
{ diffServMIBConformance 1 }
```

diffServMIBGroups OBJECT IDENTIFIER ::=

```
{ diffServMIBConformance 2 }
```

diffServMIBFullCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"When this MIB is implemented with support for read-create, then such an implementation can claim full compliance. Such devices can then be both monitored and configured with this MIB."

MODULE IF-MIB -- The interfaces MIB, RFC2863

```
MANDATORY-GROUPS {
```

```
    ifCounterDiscontinuityGroup
  }
```

MODULE -- This Module

MANDATORY-GROUPS {

```
    diffServMIBDataPathGroup, diffServMIBClfrGroup,
    diffServMIBClfrElementGroup, diffServMIBMultiFieldClfrGroup,
    diffServMIBActionGroup, diffServMIBAlgDropGroup,
    diffServMIBQGroup, diffServMIBSchedulerGroup,
    diffServMIBMaxRateGroup, diffServMIBMinRateGroup,
    diffServMIBCounterGroup
  }
```

GROUP diffServMIBMeterGroup

DESCRIPTION

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

GROUP diffServMIBTBParamGroup

"Support for createAndWait and notInService is not required."

```
OBJECT diffServMaxRateStatus
SYNTAX RowStatus { active(1) }
WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
```

Baker, et. al. Standards Track [Page 93]
RFC 3289 Differentiated Services MIB May 2002

DESCRIPTION
"Support for createAndWait and notInService is not required."

::= { diffServMIBCompliances 1 }

--
-- Read-Only Compliance
--

```
diffServMIBReadOnlyCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
    "When this MIB is implemented without support for read-create
    (i.e. in read-only mode), then such an implementation can claim
    read-only compliance. Such a device can then be monitored but can
    not be configured with this MIB."

MODULE IF-MIB -- The interfaces MIB, RFC2863
MANDATORY-GROUPS {
    ifCounterDiscontinuityGroup
}

MODULE -- This Module
MANDATORY-GROUPS {
    diffServMIBDataPathGroup, diffServMIBClfrGroup,
    diffServMIBClfrElementGroup, diffServMIBMultiFieldClfrGroup,
    diffServMIBActionGroup, diffServMIBAlgDropGroup,
    diffServMIBQGroup, diffServMIBSchedulerGroup,
    diffServMIBMaxRateGroup, diffServMIBMinRateGroup,
    diffServMIBCounterGroup
}

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

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

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

GROUP diffServMIBRandomDropGroup
```

Baker, et. al. Standards Track [Page 94]
RFC 3289 Differentiated Services MIB May 2002

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

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

```
OBJECT diffServDataPathStorage
MIN-ACCESS read-only
```

DESCRIPTION
 "Write access is not required."

OBJECT diffServDataPathStatus
 SYNTAX RowStatus { active(1) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required, and active is the only status that
 needs to be supported."

OBJECT diffServClfrNextFree
 MIN-ACCESS not-accessible
 DESCRIPTION
 "Object not needed when diffServClfrTable is implemented read-
 only"

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

OBJECT diffServClfrStatus
 SYNTAX RowStatus { active(1) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required, and active is the only status that
 needs to be supported."

OBJECT diffServClfrElementNextFree
 MIN-ACCESS not-accessible
 DESCRIPTION
 "Object not needed when diffServClfrelementTable is implemented
 read-only"

OBJECT diffServClfrElementPrecedence
 MIN-ACCESS read-only
 DESCRIPTION

Baker, et. al. Standards Track [Page 95]
 RFC 3289 Differentiated Services MIB May 2002

"Write access is not required."

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

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

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

OBJECT diffServClfrElementStatus
 SYNTAX RowStatus { active(1) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required, and active is the only status that
 needs to be supported."

OBJECT diffServMultiFieldClfrNextFree
 MIN-ACCESS not-accessible
 DESCRIPTION
 "Object is not needed when diffServMultiFieldClfrTable is
 implemented in read-only mode."

OBJECT diffServMultiFieldClfrAddrType
 SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required. An implementation is only required

to support IPv4 and IPv6 addresses."

OBJECT diffServMultiFieldClfrDstAddr
SYNTAX InetAddress (SIZE(0|4|16))
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required. An implementation is only required to support IPv4 and globally unique IPv6 addresses."

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

Baker, et. al. Standards Track [Page 96]
RFC 3289 Differentiated Services MIB May 2002

OBJECT diffServMultiFieldClfrSrcAddr
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required. An implementation is only required to support IPv4 and globally unique IPv6 addresses."

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

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

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

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

OBJECT diffServMultiFieldClfrDstL4PortMin
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT diffServMultiFieldClfrDstL4PortMax
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT diffServMultiFieldClfrSrcL4PortMin
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT diffServMultiFieldClfrSrcL4PortMax
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT diffServMultiFieldClfrStorage
MIN-ACCESS read-only

Baker, et. al. Standards Track [Page 97]
RFC 3289 Differentiated Services MIB May 2002

DESCRIPTION
"Write access is not required."

OBJECT diffServMultiFieldClfrStatus
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required, createAndWait and notInService support is not required."

OBJECT diffServMeterNextFree
 MIN-ACCESS not-accessible
 DESCRIPTION
 "Object is not needed when diffServMultiFieldClfrTable is implemented in read-only mode."

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

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

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

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

OBJECT diffServMeterStatus
 SYNTAX RowStatus { active(1) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServTBParamNextFree
 MIN-ACCESS not-accessible
 DESCRIPTION
 "Object is not needed when diffServTBParamTable is implemented in read-only mode."

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

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

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

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

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

OBJECT diffServTBParamStatus
 SYNTAX RowStatus { active(1) }

MIN-ACCESS read-only
DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServActionNextFree
MIN-ACCESS not-accessible
DESCRIPTION
"Object is not needed when diffServActionTable is implemented in read-only mode."

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

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

Baker, et. al. Standards Track [Page 99]
RFC 3289 Differentiated Services MIB May 2002

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

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

OBJECT diffServActionStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServCountActNextFree
MIN-ACCESS not-accessible
DESCRIPTION
"Object is not needed when diffServCountActTable is implemented in read-only mode."

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

OBJECT diffServCountActStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServAlgDropNextFree
MIN-ACCESS not-accessible
DESCRIPTION
"Object is not needed when diffServAlgDropTable is implemented in read-only mode."

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

OBJECT diffServAlgDropNext
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServAlgDropQMeasure
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServAlgDropQThreshold
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServAlgDropSpecific
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServAlgDropStorage
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServAlgDropStatus
SYNTAX RowStatus { active(1) }

DESCRIPTION

"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServRandomDropNextFree
MIN-ACCESS not-accessible

DESCRIPTION

"Object is not needed when diffServRandomDropTable is implemented in read-only mode."

OBJECT diffServRandomDropMinThreshBytes
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropMinThreshPkts
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropMaxThreshBytes
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropMaxThreshPkts
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropProbMax
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropWeight
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT diffServRandomDropSamplingRate

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

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

OBJECT diffServRandomDropStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServQNextFree
MIN-ACCESS not-accessible
DESCRIPTION
"Object is not needed when diffServQTable is implemented in read-only mode."

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

OBJECT diffServQMinRate
MIN-ACCESS read-only

Baker, et. al. Standards Track [Page 102]
RFC 3289 Differentiated Services MIB May 2002

DESCRIPTION
"Write access is not required."

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

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

OBJECT diffServQStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServSchedulerNextFree
MIN-ACCESS not-accessible
DESCRIPTION
"Object is not needed when diffServSchedulerTable is implemented in read-only mode."

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

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

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

OBJECT diffServSchedulerMaxRate
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServSchedulerStorage
MIN-ACCESS read-only

Baker, et. al. Standards Track [Page 103]
RFC 3289 Differentiated Services MIB May 2002

DESCRIPTION
"Write access is not required."

OBJECT diffServSchedulerStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServMinRateNextFree
MIN-ACCESS not-accessible

DESCRIPTION
"Object is not needed when diffServMinRateTable is implemented in read-only mode."

OBJECT diffServMinRatePriority
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMinRateAbsolute
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMinRateRelative
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMinRateStorage
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMinRateStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required, and active is the only status that needs to be supported."

OBJECT diffServMaxRateNextFree
MIN-ACCESS not-accessible

DESCRIPTION
"Object is not needed when diffServMaxrateTable is implemented in read-only mode."

Baker, et. al. Standards Track [Page 104]
RFC 3289 Differentiated Services MIB May 2002

OBJECT diffServMaxRateAbsolute
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMaxRateRelative
MIN-ACCESS read-only

DESCRIPTION
"Write access is not required."

OBJECT diffServMaxRateThreshold

```

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

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

OBJECT        diffServMaxRateStatus
SYNTAX        RowStatus { active(1) }
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required, and active is the only status that
    needs to be supported."

 ::= { diffServMIBCompliances 2 }

```

```

diffServMIBDataPathGroup OBJECT-GROUP
  OBJECTS {
      diffServDataPathStart, diffServDataPathStorage,
      diffServDataPathStatus
  }
  STATUS        current
  DESCRIPTION
      "The Data Path Group defines the MIB Objects that describe a
      functional data path."
  ::= { diffServMIBGroups 1 }

```

```

diffServMIBClfrGroup OBJECT-GROUP
  OBJECTS {
      diffServClfrNextFree, diffServClfrStorage,
      diffServClfrStatus
  }
  STATUS        current
  DESCRIPTION
      "The Classifier Group defines the MIB Objects that describe the

```

Baker, et. al. Standards Track [Page 105]

RFC 3289 Differentiated Services MIB May 2002

```

    list the starts of individual classifiers."
 ::= { diffServMIBGroups 2 }

```

```

diffServMIBClfrElementGroup OBJECT-GROUP
  OBJECTS {
      diffServClfrElementNextFree,
      diffServClfrElementPrecedence, diffServClfrElementNext,
      diffServClfrElementSpecific, diffServClfrElementStorage,
      diffServClfrElementStatus
  }
  STATUS        current
  DESCRIPTION
      "The Classifier Element Group defines the MIB Objects that
      describe the classifier elements that make up a generic
      classifier."
  ::= { diffServMIBGroups 3 }

```

```

diffServMIBMultiFieldClfrGroup OBJECT-GROUP
  OBJECTS {
      diffServMultiFieldClfrNextFree,
      diffServMultiFieldClfrAddrType,
      diffServMultiFieldClfrDstAddr,
      diffServMultiFieldClfrDstPrefixLength,
      diffServMultiFieldClfrFlowId,
      diffServMultiFieldClfrSrcAddr,
      diffServMultiFieldClfrSrcPrefixLength,
      diffServMultiFieldClfrDscp,
      diffServMultiFieldClfrProtocol,
      diffServMultiFieldClfrDstL4PortMin,
      diffServMultiFieldClfrDstL4PortMax,
      diffServMultiFieldClfrSrcL4PortMin,
      diffServMultiFieldClfrSrcL4PortMax,
      diffServMultiFieldClfrStorage,
      diffServMultiFieldClfrStatus
  }

```

```
STATUS      current
DESCRIPTION
"The Multi-field Classifier Group defines the MIB Objects that
describe a classifier element for matching on various fields of
an IP and upper-layer protocol header."
 ::= { diffServMIBGroups 4 }
```

```
diffServMIBMeterGroup OBJECT-GROUP
OBJECTS {
    diffServMeterNextFree, diffServMeterSucceedNext,
    diffServMeterFailNext, diffServMeterSpecific,
    diffServMeterStorage, diffServMeterStatus
}
```

Baker, et. al. Standards Track [Page 106]
RFC 3289 Differentiated Services MIB May 2002

```
STATUS      current
DESCRIPTION
"The Meter Group defines the objects used in describing a generic
meter element."
 ::= { diffServMIBGroups 5 }
```

```
diffServMIBTParamGroup OBJECT-GROUP
OBJECTS {
    diffServTParamNextFree, diffServTParamType,
    diffServTParamRate, diffServTParamBurstSize,
    diffServTParamInterval, diffServTParamStorage,
    diffServTParamStatus
}
```

```
STATUS      current
DESCRIPTION
"The Token-Bucket Meter Group defines the objects used in
describing a token bucket meter element."
 ::= { diffServMIBGroups 6 }
```

```
diffServMIBActionGroup OBJECT-GROUP
OBJECTS {
    diffServActionNextFree, diffServActionNext,
    diffServActionSpecific, diffServActionStorage,
    diffServActionInterface, diffServActionStatus
}
STATUS      current
DESCRIPTION
"The Action Group defines the objects used in describing a
generic action element."
 ::= { diffServMIBGroups 7 }
```

```
diffServMIBDscpMarkActGroup OBJECT-GROUP
OBJECTS {
    diffServDscpMarkActDscp
}
STATUS      current
DESCRIPTION
"The DSCP Mark Action Group defines the objects used in
describing a DSCP Marking Action element."
 ::= { diffServMIBGroups 8 }
```

```
diffServMIBCounterGroup OBJECT-GROUP
OBJECTS {
    diffServCountActOctets, diffServCountActPkts,
    diffServAlgDropOctets, diffServAlgDropPkts,
    diffServAlgRandomDropOctets, diffServAlgRandomDropPkts,
    diffServCountActStorage, diffServCountActStatus,
    diffServCountActNextFree
}
```

Baker, et. al. Standards Track [Page 107]
RFC 3289 Differentiated Services MIB May 2002

```
STATUS      current
DESCRIPTION
```

"A collection of objects providing information specific to packet-oriented network interfaces."
 ::= { diffServMIBGroups 9 }

diffServMIBAlgDropGroup OBJECT-GROUP
OBJECTS {
 diffServAlgDropNextFree, diffServAlgDropType,
 diffServAlgDropNext, diffServAlgDropQMeasure,
 diffServAlgDropQThreshold, diffServAlgDropSpecific,
 diffServAlgDropStorage, diffServAlgDropStatus
}
STATUS current
DESCRIPTION
 "The Algorithmic Drop Group contains the objects that describe algorithmic dropper operation and configuration."
 ::= { diffServMIBGroups 10 }

diffServMIBRandomDropGroup OBJECT-GROUP
OBJECTS {
 diffServRandomDropNextFree,
 diffServRandomDropMinThreshBytes,
 diffServRandomDropMinThreshPkts,
 diffServRandomDropMaxThreshBytes,
 diffServRandomDropMaxThreshPkts,
 diffServRandomDropProbMax,
 diffServRandomDropWeight,
 diffServRandomDropSamplingRate,
 diffServRandomDropStorage,
 diffServRandomDropStatus
}
STATUS current
DESCRIPTION
 "The Random Drop Group augments the Algorithmic Drop Group for random dropper operation and configuration."
 ::= { diffServMIBGroups 11 }

diffServMIBQGroup OBJECT-GROUP

OBJECTS {
 diffServQNextFree, diffServQNext, diffServQMinRate,
 diffServQMaxRate, diffServQStorage, diffServQStatus
}
STATUS current
DESCRIPTION
 "The Queue Group contains the objects that describe an

Baker, et. al. Standards Track [Page 108]

RFC 3289 Differentiated Services MIB May 2002

interface's queues."
 ::= { diffServMIBGroups 12 }

diffServMIBSchedulerGroup OBJECT-GROUP
OBJECTS {
 diffServSchedulerNextFree, diffServSchedulerNext,
 diffServSchedulerMethod, diffServSchedulerMinRate,
 diffServSchedulerMaxRate, diffServSchedulerStorage,
 diffServSchedulerStatus
}
STATUS current
DESCRIPTION
 "The Scheduler Group contains the objects that describe packet schedulers on interfaces."
 ::= { diffServMIBGroups 13 }

diffServMIBMinRateGroup OBJECT-GROUP
OBJECTS {
 diffServMinRateNextFree, diffServMinRatePriority,
 diffServMinRateAbsolute, diffServMinRateRelative,
 diffServMinRateStorage, diffServMinRateStatus
}
STATUS current
DESCRIPTION
 "The Minimum Rate Parameter Group contains the objects that describe packet schedulers' minimum rate or priority guarantees."
 ::= { diffServMIBGroups 14 }

```
diffServMIBMaxRateGroup OBJECT-GROUP
  OBJECTS {
    diffServMaxRateNextFree, diffServMaxRateAbsolute,
    diffServMaxRateRelative, diffServMaxRateThreshold,
    diffServMaxRateStorage, diffServMaxRateStatus
  }
  STATUS      current
  DESCRIPTION
    "The Maximum Rate Parameter Group contains the objects that
    describe packet schedulers' maximum rate guarantees."
 ::= { diffServMIBGroups 15 }

END
```

Baker, et. al. Standards Track [Page 109]
RFC 3289 Differentiated Services MIB May 2002

7. Acknowledgments

This MIB builds on all the work that has gone into the Informal Management Model for Differentiated Services Routers, Differentiated Services PIB, and Differentiated Services Policy MIB (SNMPCONF WG).

It has been developed with the active involvement of many people, but most notably Yoram Bernet, Steve Blake, Brian Carpenter, Dave Durham, Michael Fine, Victor Firoiu, Jeremy Greene, Dan Grossman, Roch Guerin, Scott Hahn, Joel Halpern, Van Jacobsen, Keith McCloghrie, Bob Moore, Kathleen Nichols, Ping Pan, Nabil Seddigh, John Seligson, and Walter Weiss.

Juergen Schoenwaelder, Dave Perkins, Frank Strauss, Harrie Hazewinkel, and Bert Wijnen are especially to be noted for review comments on the structure and usage of the MIB for network management purposes, and its compliance with SMIV2.

8. Security Considerations

It is clear that this MIB is potentially useful for configuration. Anything that can be configured can be misconfigured, with potentially disastrous effects.

At this writing, no security holes have been identified beyond those that SNMP Security is itself intended to address. These relate primarily to 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 many read-write and read-create management objects defined in this MIB. Such objects are often 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 even control GET access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 2574] and the View-based Access Control Model [RFC 2575] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. Intellectual Property Rights

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. References

- [RFC 2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [RFC 1155] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [RFC 1212] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [RFC 1215] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [RFC 2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC

2578, April 1999.

- [RFC 2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [RFC 2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [RFC 1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [RFC 1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [RFC 1906] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [RFC 2572] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

Baker, et. al. Standards Track [Page 112]
RFC 3289 Differentiated Services MIB May 2002

- [RFC 2574] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [RFC 1905] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [RFC 2573] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [RFC 2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [RFC 2570] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.
- [RFC 2119] Bradner, S., "Key words to use in the RFCs", BCP 14, RFC 2119, March 1997.
- [ACTQMGMT] V. Firoiu, M. Borden, "A Study of Active Queue Management for Congestion Control", March 2000, In IEEE Infocom 2000, <http://www.ieee-infocom.org/2000/papers/405.pdf>
- [AQMROUTER] V. Misra, W. Gong, D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED", In SIGCOMM 2000, <http://www.acm.org/sigcomm/sigcomm2000/conf/paper/sigcomm2000-4-3.ps.gz>
- [AF-PHB] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [DSARCH] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Service", RFC 2475, December 1998.
- [DSFIELD] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS

Field) in the IPv4 and IPv6 Headers", RFC 2474,
December 1998.

Baker, et. al.	Standards Track	[Page 113]
RFC 3289	Differentiated Services MIB	May 2002
[DSPIB]	Fine, M., McCloghrie, K., Seligson, J., Chan, K., Hahn, S. and A. Smith, "Differentiated Services Quality of Service Policy Information Base", Work in Progress.	
[DSTERMS]	Grossman, D., "New Terminology for Differentiated Services", RFC 3260, April 2002.	
[EF-PHB]	Jacobson, V., Nichols, K. and K. Poduri, "An Expedited Forwarding PHB", RFC 3246, March 2002.	
[IF-MIB]	McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMiv2", RFC 2863, June 2000.	
[INETADDRESS]	Daniele, M., Haberman, B., Routhier, S. and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses.", RFC 3291, May 2002.	
[INTSERVMIB]	Baker, F., Krawczyk, J. and A. Sastry, "Integrated Services Management Information Base using SMiv2", RFC 2213, September 1997.	
[MODEL]	Bernet, Y., Blake, S., Smith, A. and D. Grossman, "An Informal Management Model for Differentiated Services Routers", Work in Progress.	
[RED93]	"Random Early Detection", 1993.	
[srTCM]	Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, September 1999.	
[trTCM]	Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999.	
[TSWTCM]	Fang, W., Seddigh, N. and B. Nandy, "A Time Sliding Window Three Color Marker (TSWTCM)", RFC 2859, June 2000.	
[SHAPER]	Bonaventure, O. and S. De Cnodder, "A Rate Adaptive Shaper for Differentiated Services", RFC 2963, October 2000.	

Baker, et. al.	Standards Track	[Page 114]
RFC 3289	Differentiated Services MIB	May 2002

11. Authors' Addresses

Fred Baker
Cisco Systems
1121 Via Del Rey
Santa Barbara, California 93117

EMail: fred@cisco.com

Kwok Ho Chan

