Network Working Group
Request for Comments: 3747
Category: Standards Track

H. Hazewinkel, Ed.
I.Net
D. Partain, Ed.
Ericsson

April 2004

The Differentiated Services Configuration MIB

Status of this Memo

Copyright Notice

Abstract

This memo describes a MIB module that provides a conceptual layer between high-level "network-wide" policy definitions that effect configuration of the Differentiated Services (diffserv) subsystem and the instance-specific information that would include such details as the parameters for all the queues associated with each interface in a system. This essentially provides an interface for configuring differentiated services at a conceptually higher layer than that of the Differentiated Services MIB.

Table of Contents

1. The Internet-Standard Management Framework 2
2. Introduction 2
3. Other Documents. 3
4. Relationship to other MIBs 3
4.1. The Policy-based Management MIB Module 3
4.2. The Differentiated Services MIB Module 4
5. The Differentiated Services Configuration MIB Module Design. . 5
6. Template Cloning 6
6.1. An Approach to Template Cloning. 6
6.2. Example. 7
6.2.1. The Initial Situation. 8
6.2.2. The Configuration Template 9
6.2.3. Applying the Template. 11
6.2.4. Applying the Template Using SNMP Messages. . . . 14
7. Managed Objects Definitions (MIB Module) 15
8. Security Considerations. 20
9. Acknowledgments. 22
10. References 22
10.1. Normative References 22
10.2. Informative References 23
11. Editors' Addresses 23
12. Full Copyright Statement 24

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

2. Introduction

This memo defines a MIB module that can be used to convey management information about desired network-wide Differentiated Services based policy behavior. This module is designed to integrate with the Differentiated Services MIB module [RFC3289] in order to provide template configurations for the Differentiated Services MIB module. The MIB module defined in this memo (the DIFFSERV-CONFIG-MIB) may be

used in combination with the Policy-based Management MIB module [PMMIBDR], but that is not a requirement. Without the Policy-based Management MIB module, a management application must emulate behavior provided by the Policy-based Management MIB using equivalent "low-level" SNMP operations in normal manager/agent communication.

Together, this memo, [RFC3289], and [PMMIBDR] represent an instance of an integrated architecture for both device-specific and network-wide policy (configuration) management, which is fully integrated with the Internet Standard Management Framework.

The Differentiated Services MIB module [RFC3289] operates on a device level. The MIB module in this memo, the DIFFSERV-CONFIG-MIB, creates a coherent configuration management view as an umbrella over [RFC3289]. That is, the DIFFSERV-CONFIG-MIB provides a conceptual Application Program Interface (API) for configuration of the Differentiated Services parameters. Since the Differentiated Services MIB module is able to maintain configuration information, the DIFFSERV-CONFIG-MIB configuration API consists only of configuration template information and the start of the so-called functional datapath.

3. Other Documents

It is assumed that the reader is familiar with Differentiated Services ([RFC2474] and [RFC2475]), the Policy-based Management MIB ([PMMIBDR]), and "Configuring Networks and Devices With SNMP" ([RFC3512]). These documents include all of the necessary terminology for understanding this memo. However, note that use of the MIB module in this memo does not require the use of [PMMIBDR]. [RFC3512] also provides an example MIB module which may help in understanding the relationship between DIFFSERV-CONFIG-MIB and the Differentiated Services MIB in [RFC3289].

4. Relationship to other MIBs

In this section, we describe the relationship of this MIB module to other MIB modules. The overall architecture used for policy configuration management is described in [PMMIBDR].

4.1. The Policy-based Management MIB Module

[PMMIBDR] defines a MIB module that enables policy-based configuration management of infrastructure using the Internet Standard Management Framework. The document includes a table for configuring policies to be implemented, tables for storing the roles of elements on a particular device, a table for representing the capabilities of a device with respect to policy management, a table

for referencing elements affected by a policy, as well as other infrastructure. There is no requirement that [PMMIBDR] be used in conjunction with the MIB module defined in this memo.

See [PMMIBDR] for a full description of the policy-based configuration framework it provides.

4.2. The Differentiated Services MIB Module

The Differentiated Services MIB module [RFC3289] provides a common set of managed objects useful for configuring Differentiated Services parameters on a Differentiated Services capable device. This is what is referred to as instance-level configuration. It is the alteration of the instance-level information in that MIB module which may be done using the objects in the MIB module defined in this memo.

It is recognized that vendors may include additional managed objects in their devices (via vendor-specific MIB modules) for configuring Differentiated Services parameters. If a vendor chooses to use the objects defined in this memo for configuration, the vendor should provide additional managed objects in a similar approach as defined for the Differentiated Services MIB module.

Since the managed objects of the Differentiated Services MIB [RFC3289] are not directly associated with an instance (interface and interface direction), the same managed objects can be used for traffic treatment configuration templates in a Differentiated Services capable device and can then be applied on multiple instances. Therefore, the tables as defined in the Differentiated Services MIB can be used directly for template configuration purposes. Those tables are:

- diffServClfrTable
- diffServClfrElementTable
- diffServMultiFieldClfrTable
- diffServMeterTable
- diffServTBParamTable
- diffServActionTable
- diffServDscpMarkActTable
- diffServCountActTable
- diffServAlgDropTable
- diffServRandomDropTable
- diffServQTable
- diffServSchedulerTable
- diffServMinRateTable
- diffServMaxRateTable

Readers familiar with the Differentiated Services MIB will notice that these are all templates. Only the diffServDataPathTable defines a managed instance for Differentiated Services traffic treatment by its indexes of the interface and its direction. This also allows the tables mentioned above to be used as a configuration template without defining anything directly related to a managed instance.

5. The Differentiated Services Configuration MIB Module Design

The Differentiated Services Configuration MIB module (in this memo) of the SNMP-based configuration management framework is positioned between the Policy-based Management MIB module and the instance-specific Differentiated Services MIB module as described above.

The MIB module found in this memo is designed to maintain configuration templates for the Differentiated Services MIB [RFC3289] module. The module only has a template table that describes a Differentiated Services traffic treatment by providing the starting pointer of the functional datapath. The templates represent a specific configuration of traffic treatment in a functional datapath of a Differentiated Services capable device. To avoid duplication of managed objects, the actual templates defining the functional datapath are defined in the Differentiated Services MIB module. These are also used for the management of the instances. Therefore, the implementation of the DIFFSERV-CONFIG-MIB module uses the tables defined in the Differentiated Services MIB. As soon as a configuration is made active via the POLICY-MANAGEMENT-MIB or using normal SNMP operations, the configuration defined within this MIB module will be instantiated in the DIFFSERV-MIB.

Note that this is a conceptual process. That is, the configuration may not actually go through an API available in the subsystem which implements the DIFFSERV-MIB module. However, configuration via the DIFFSERV-CONFIG-MIB module will alter the same instrumentation as the DIFFSERV-MIB module whether it does it via the DIFFSERV-MIB module or not.

The Differentiated Services Configuration MIB module only needs to define a starting point of a traffic treatment configuration template. This table is similar to the diffServDataPathTable [RFC3289]. However, it has a semantic difference in that the diffServDataPathTable is associated with an instance (interface and interface direction), whereas the diffServConfigTable in this memo is not.

Unlike most MIB modules, changes to the managed objects in this MIB module do not cause a change in the external/traffic behavior of the device. This MIB module is used to set up per-hop-behavior configurations. As soon as configurations are made active via the POLICY-MANAGEMENT-MIB or SNMP operations, the configurations defined within this MIB module will be instantiated in the DIFFSERV-MIB.

The only table in this MIB module is the diffServConfigTable, which provides managed objects for registering traffic treatment configurations used in differentiated services. The sole purpose of this table is to provide the starting point for a traffic treatment configuration template. The traffic treatment itself is performed by functional datapath elements [RFC3289].

6. Template Cloning

The concept of the DIFFSERV-CONFIG-MIB is based on having traffic treatment configuration templates. The templates provide a set of configuration values that provide a particular behavior, such as Expedited Forwarding traffic treatment, in the functional datapath. The template (or functional datapath) is similar to a linked list from a starting point and each (functional datapath) element is connected to the next element via the so-called next RowPointer.

The moment a template is activated (instantiated) on an interface and its interface direction, the template needs to be copied/cloned, so that the template remains as a template. Note that the template is logically "locked" through the cloning process. That is, the template cannot be changed part way through the cloning process. With the exception of the indices, the cloned template will be identical to the source template.

A literal copy/clone of the template is not possible, since the same indices inside the element tables cannot be re-used. The instantiation process must therefore generate a new index for each element. As a result of this, the 'NEXT' pointers also need to be updated. Otherwise, those will point to the template.

6.1. An Approach to Template Cloning

What should a system containing Differentiated Services capabilities and Differentiated Services configuration capabilities do conceptually at the moment a template is activated on an interface? The following approach should not be considered implementation guidelines, but rather a conceptual explanation of what should be done.

1) Get the index of the template to be activated
2) Get RowPointer (current) from
   diffServConfigStart.index
   of the diffServConfigTable
3) Check if RowPointer (current) exists
4) Logically "lock" the entry (current) pointed to by
   RowPointer so that its values are not changed part way
   through the cloning process.
5) Copy/Clone the entry (current) pointed to by RowPointer
   a) Get a new index for the entry
   b) Configure the new entry with the values
      of the entry to be cloned
   c) Update the NEXT pointer with a new RowPointer
      that pointed to the previous entry that was copied
      part of this template
6) Store RowPointer of cloned entry as (previous) in order to
   update the NEXT pointer with the next cloned entry.
7) Get the RowPointer of the next element in the template
   as (current)
8) If (current) RowPointer does not equal zeroDotZero go to 4
9) Logically "unlock" all the locked entries done by step 4).

If a configuration/template is activated via a means other than a direct SNMP SET request, such as via the Policy-based Management MIB, the handling of the activation and potential error response code must be provided via that mechanism. If a configuration/template is activated using SNMP SET requests, an accurate error response value must be returned. For example, if a configuration/template has inconsistent values, the SNMP SET should return an error. Whether the configuration is already finished is not of direct importance, since the SNMP SET response must be accurate. On systems where the activation may take a long time, a response may be given prior to completion, but extra mechanisms must be provided to detect any errors.

6.2. Example

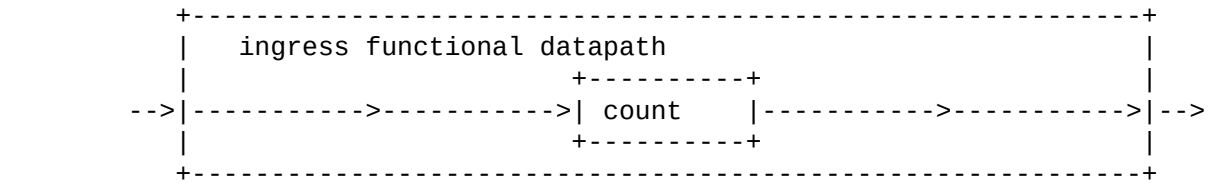
This section provides an example of the process described in the previous section. This example will show a Differentiated Services capable incoming (ingress) interface that only counts the traffic stream. Then, with the policy-based configuration concept as defined in this document and in [PMMIBDR], a traffic marking configuration will be applied. The example will walk the reader through all of the steps involved in this process. Again, the use of [PMMIBDR] is simply an example and is not required.

NOTE WELL: For brevity and clarity, the example does not always show the complete entry (row) of a table. The only objects shown are those needed for creating the row pointers to the next functional datapath element or needed to provide information about the specific parameters of the functional datapath elements. The column named 'INDEX' always defines the complete index as defined for the associated entry. In some cases, this is a combined index of multiple components. Therefore, the names of the columns are omitted.

Also note that the values Assured Forwarding and Expedited Forwarding are abstracted as DSCP(AF) and DSCP(EF) (respectively) or simply as AF and EF. For the actual values refer to [RFC3289].

6.2.1. The Initial Situation

The initial configuration is the existing configuration of an ingress interface.



This figure depicts a simple traffic treatment functional datapath for an ingress interface. The functional datapath only consists of a count action.

Within the DIFFSERV-MIB, this would be instantiated as follows. Note that RowPointer objects must point to the first accessible columnar object in the conceptual row. Thus, while perhaps more instructive to use the index value for the RowPointer object's value (e.g., diffServCountActId.1) in the example, it would nonetheless be incorrect, and the first accessible columnar object has been used as should be done (e.g., diffServCountActOctets.1).

diffServDataPathTable

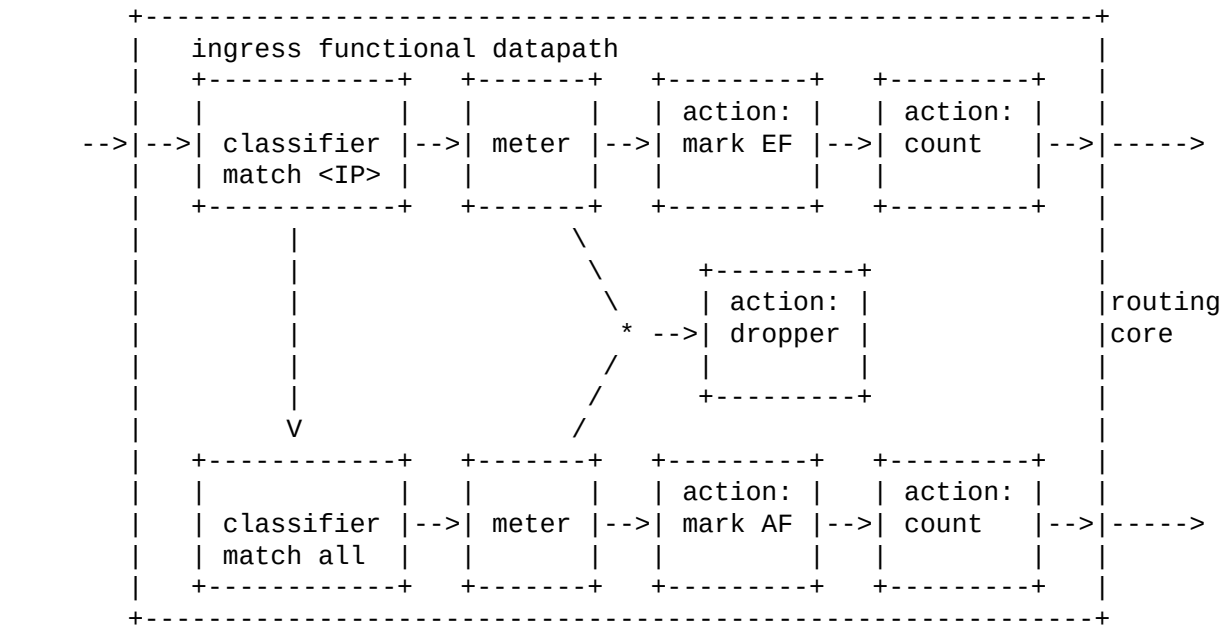
INDEX	diffServDataPathStart
ifIndex.ingress	diffServActionNext.1

```
diffServActionTable
+-------+--------------------+-------------------------+--
| INDEX | diffServActionNext |diffServActionSpecific   |
+-------+--------------------+-------------------------+--
| 1     | 0.0                |diffServCountActOctets.1 |
+-------+--------------------+-------------------------+--

diffServCountActTable
+-------+------------------------+--
| INDEX | diffServCountActOctets |
+-------+------------------------+--
| 1     |                        |
+-------+------------------------+--
```

6.2.2. The Configuration Template

The following provides an example of a policy configuration in which traffic is classified by a specific IP filter, that results in two classifiers (one for the IP filter and one for match all). Both streams are then metered, marked, and counted. This is an example of usage on the edge (an ingress interface) of a Differentiated Services domain that wants to have Expedited Forwarding and Assured Forwarding marked traffic within the Differentiated Services domain.



This figure depicts a policy configuration for ingress traffic treatment in a Differentiated Services capable device. The

configuration is represented as follows in the DIFFSERV-CONFIG-MIB module and the DIFFSERV-MIB module.

Note that the original (existing) traffic treatment described in 6.2.1 is also in the tables.

Note also that in the diffServDscpMarkActTable, DSCP(EF) represents the DSCP value for Expedited Forwarding and DSCP(AF) represents the DSCP value for Assured Forwarding.

diffServConfigTable (in the MIB module in this memo)

INDEX	diffServConfigStart	diffServConfigDescr
"foo"	diffServClfrStorage.1	Example traffic treatment

diffServClfrTable

INDEX	diffServClfrStorage	diffServClfrStatus
1		

diffServClfrElementTable (shares index with diffServClfrTable)

INDEX	diffServClfrElementNext	diffServClfrElementPrecedence
1.1	diffServMeterSucceedNext.1	1
1.2	diffServMeterSucceedNext.2	2

diffServMeterTable

INDEX	diffServMeterSucceedNext	diffServMeterFailNext
1	diffServActionNext.2	diffServAlgDropType.1
2	diffServActionNext.3	diffServAlgDropType.1

```
diffServActionTable
+-------+----------------------+----------------------------+--
| INDEX | diffServActionNext   | diffServActionSpecific     |
+-------+----------------------+----------------------------+--
| 1     | 0.0                  | diffServCountActOctets.1   |
| 2     | diffServActionNext.4 | diffServDscpMarkActDscp.EF |
| 3     | diffServActionNext.5 | diffServDscpMarkActDscp.AF |
| 4     | 0.0                  | diffServCountActOctets.2   |
| 5     | 0.0                  | diffServCountActOctets.3   |
+-------+----------------------+----------------------------+--

diffServCountActTable
+-------+-------------------------+--
| INDEX | diffServCountActOctets  |
+-------+-------------------------+--
| 1     |                         |
| 2     |                         |
| 3     |                         |
+-------+-------------------------+--

diffServAlgDropTable
+-------+----------------------+--------------------------+--
| INDEX | diffServAlgDropType  | diffServAlgDropSpecific  |
+-------+----------------------+--------------------------+--
| 1     | alwaysDrop(5)        | 0.0                      |
+-------+----------------------+--------------------------+--

diffServDscpMarkActTable
+--------------------------+
| diffServDscpMarkActDscp  |
+--------------------------+
| DSCP(EF)                 |
| DSCP(AF)                 |
+--------------------------+
```

6.2.3. Applying the Template

Now we have the original ingress interface configuration and the policy configuration we want to apply to the actual interface.

The example policy must provide the required Differentiated Services traffic treatment to all interfaces used by system administrators. The traffic treatment required is described in 6.2.2 above.

Therefore, we have the following example policy which is configured via the POLICY-BASED-MANAGEMENT-MIB module (see [PMMIBDR]):

```
if ( roleMatch("Administrator") )
```

```
   then
      /*
       * The $0 gets the "element" returned from the previous
       * statement.  the .1 at the end is the ingress interface
       * This sets, for example, diffServDataPathStart.3.1 to be
       * "diffServConfigStart.3.f.o.o" if interface 3 has the role
       * "Administrator".
       */
      setVar("diffServDataPathStart.$0.1",
             "diffServConfigStart.3.f.o.o", Oid)
```

For our purposes, we only apply this on the inbound (ingress) direction of the interface.

Note that although object descriptors are used in this PolicyScript example, the object identifiers must be used in the running script. For more information on policies and their syntax refer to [PMMIBDR].

The following tables in this section provide the cloned entries in the tables of the DIFFSERV-MIB module. All tables may have columns that contain contents or administrative objects that are not shown. These columns do not determine a function in the datapath and they are not shown for clarity of the cloning mechanism.

Note that the original (existing) traffic treatment of 6.2.1 and 6.2.2 are also in the tables.

diffServConfigTable

INDEX	diffServConfigStart	diffServConfigDescr
"foo"	diffServClfrStorage.1	Example traffic treatment

diffServDataPathTable

INDEX	diffServDataPathStart
ifIndex.ingress	diffServActionNext.2

diffServClfrTable

INDEX	diffServClfrStorage	diffServClfrStatus
1		
2		

diffServClfrElementTable

INDEX	diffServClfrElementNext	diffServClfrElementPrecedence
1.1	diffServMeterSucceedNext.1	1
1.2	diffServMeterSucceedNext.2	2
2.3	diffServMeterSucceedNext.3	1
2.4	diffServMeterSucceedNext.4	2

diffServMeterTable

INDEX	diffServMeterSucceedNext	diffServMeterFailNext
1	diffServActionNext.2	diffServAlgDropType.1
2	diffServActionNext.3	diffServAlgDropType.1
3	diffServActionNext.6	diffServAlgDropType.2
4	diffServActionNext.7	diffServAlgDropType.2

diffServActionTable

INDEX	diffServActionNext	diffServActionSpecific
1	0.0	diffServCountActOctets.1
2	diffServActionNext.4	diffServDscpMarkActDscp.EF
3	diffServActionNext.5	diffServDscpMarkActDscp.AF
4	0.0	diffServCountActOctets.2
5	0.0	diffServCountActOctets.3
6	diffServActionNext.8	diffServDscpMarkActDscp.EF
7	diffServActionNext.9	diffServDscpMarkActDscp.AF
8	0.0	diffServCountActOctets.4
9	0.0	diffServCountActOctets.5

```
diffServCountActTable
+-------+------------------------+--
| INDEX | diffServActCountOctets |
+-------+------------------------+--
| 1     |                        |
| 2     |                        |
| 3     |                        |
| 4     |                        |
| 5     |                        |
+-------+------------------------+--

diffServAlgDropTable
+-------+---------------------+-------------------------+--
| INDEX | diffServAlgDropType | diffServAlgDropSpecific |
+-------+---------------------+-------------------------+--
| 1     | alwaysDrop(5)       | 0.0                     |
+-------+---------------------+-------------------------+--

diffServDscpMarkActTable
+-------------------------+
| diffServDscpMarkActDscp |
+-------------------------+
| DSCP(EF)                |
| DSCP(AF)                |
+-------------------------+
```

As one can see in the example, the main elements from which a functional datapath is constructed are duplicated/copied/cloned. That process is needed in order to preserve the policy configuration for reuse at a later time.

It is up to the SNMP agent to keep track of which network interfaces are under policy control and which policy rules are being used. This avoids duplication of policy enforcement. How the agent does this is an implementation issue.

One can see that the old functional datapath configurations stay in the MIB module tables. It is up to the SNMP agent implementation to decide whether to delete stale entries or keep them. Garbage collection of stale entries is an implementation issue.

6.2.4. Applying the Template Using SNMP Messages

In this section, the above example is explained by using SNMP communication between the SNMP "manager" and the SNMP "agent".

In order to apply the template to all interfaces that have a role match of "Administrator," the SNMP manager must have a list of the roles of the interface. This means the SNMP manager must do an SNMP-SET for all those interfaces. This is expressed in the following pseudo code function.

```
set_template_if_administrator_interface(
         <interface_list>, <template_name>
) {
   template_oid = SNMP-GET("diffServConfigStart.<template_name>");
   foreach interface (<ifRole_list>) {
      if (interface.role == "Administrator") {
         SNMP-SET("diffServDataPathStart.$interface.1",
               Oid, template_oid);
      }
   }
}
```

For example, on a system with 3 interfaces, the following list would be known to the manager. The first value indicates the interface number (ifIndex) and the second value is its role.

```
interface_list IF_LIST = {
    { 1, ... , "Administrator", ... },
    { 2, ... , "User", ... },
    { 3, ... , "Administrator", ... } }
```

This will result in the communication between a manager and agent of 1 SNMP-GET and 2 SNMP-SETs:

- SNMP-GET("diffServConfigStart.3.f.o.o")
- SNMP-SET("diffServDataPathStart.1.1", Oid, "diffServActionNext.1")
- SNMP-SET("diffServDataPathStart.3.1", Oid, "diffServActionNext.1")

7. Managed Objects Definitions (MIB Module)

```
DIFFSERV-CONFIG-MIB DEFINITIONS ::= BEGIN
    IMPORTS

    OBJECT-TYPE, MODULE-IDENTITY,
    zeroDotZero, mib-2            FROM SNMPv2-SMI          -- [RFC2578]

    RowStatus, StorageType,
    RowPointer, DateAndTime       FROM SNMPv2-TC           -- [RFC2579]

    MODULE-COMPLIANCE,
    OBJECT-GROUP                  FROM SNMPv2-CONF         -- [RFC2580]
```

```
    SnmpAdminString              FROM SNMP-FRAMEWORK-MIB; -- [RFC3411]

diffServConfigMib MODULE-IDENTITY
    LAST-UPDATED "200401220000Z" -- 22 January 2004
    ORGANIZATION "SNMPCONF WG"
    CONTACT-INFO
       "SNMPCONF Working Group
        http://www.ietf.org/html.charters/snmpconf-charter.html
        WG mailing list: snmpconf@snmp.com

        Editors:
        Harrie Hazewinkel
        I.Net
        via Darwin 85
        20019 - Settimo Milanese (MI)
        Italy
        EMail: harrie@inet.it

        David Partain
        Ericsson AB
        P.O. Box 1248
        SE-581 12 Linkoping
        Sweden
        E-mail: David.Partain@ericsson.com"
    DESCRIPTION
        "This MIB module contains differentiated services
        specific managed objects to perform higher-level
        configuration management.  This MIB allows policies
        to use 'templates' to instantiate Differentiated
        Services functional datapath configurations to
        be assigned (associated with an interface and
        direction) when a policy is activated.

        Copyright (C) The Internet Society (2004).  This version
        of this MIB module is part of RFC 3747;  see the RFC
        itself for full legal notices."
    REVISION "200401220000Z" -- 22 January 2004
    DESCRIPTION
        "Initial version published as RFC 3747"
    ::= { mib-2 108 }

diffServConfigMIBObjects OBJECT IDENTIFIER ::= { diffServConfigMib 1 }
diffServConfigMIBConformance OBJECT IDENTIFIER ::=
                                              { diffServConfigMib 2 }

--
-- The Differentiated Services configuration objects
--
```

```
diffServConfigTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF DiffServConfigEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "A table which defines the various per-hop-behaviors
       for which the system has default 'templates'."
    ::= { diffServConfigMIBObjects 2 }

diffServConfigEntry OBJECT-TYPE
    SYNTAX       DiffServConfigEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry defining a per-hop-behavior.  Each entry in
       this table combines the various parameters (entries)
       into a specific per-hop-behavior.  Entries in this
       table might be defined by a vendor (pre-configured)
       or defined by a management application."
    INDEX { diffServConfigId }
    ::= { diffServConfigTable 1 }

DiffServConfigEntry ::= SEQUENCE {
    diffServConfigId           SnmpAdminString,
    diffServConfigDescr        SnmpAdminString,
    diffServConfigOwner        SnmpAdminString,
    diffServConfigLastChange   DateAndTime,
    diffServConfigStart        RowPointer,
    diffServConfigStorage      StorageType,
    diffServConfigStatus       RowStatus
}

diffServConfigId OBJECT-TYPE
    SYNTAX         SnmpAdminString (SIZE(1..116))
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
      "A unique id for the per-hop-behavior policy for at
       least the SNMP agent.  For ease of administration the
       value may be unique within an administrative domain,
       but this is not required.

       The range of up to 116 octets is chosen to stay within
       the SMI limit of 128 sub-identifiers in an object
       identifier."
    ::= { diffServConfigEntry 1 }

diffServConfigDescr OBJECT-TYPE
```

```
    SYNTAX          SnmpAdminString
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
       "A human-readable description to identify this defined
       per-hop-behavior.  Note that this is an SnmpAdminString,
       which permits UTF-8 strings.  An administratively assigned
       identifier for a template that would be unique within
       an administrative domain.  It is up to the management
       applications to agree how these are assigned within the
       administrative domain.  Once a description, such as
       'EF' is assigned, that has a certain set of parameters
       that achieve 'EF' from box to box. Management
       application code or script code can then scan
       the table to find the proper template and then
       assign it."
    ::= { diffServConfigEntry 2 }

diffServConfigOwner OBJECT-TYPE
    SYNTAX          SnmpAdminString
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
       "The owner who created this entry."
    ::= { diffServConfigEntry 3 }

diffServConfigLastChange OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
       "The date and time when this entry was last changed."
    ::= { diffServConfigEntry 4 }

diffServConfigStart OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
       "The pointer to a functional datapath configuration template as
       set up in the DIFFSERV-MIB.  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 Diffserv treatment is performed on traffic of
          this functional datapath.  This also means that the
          template described by this row is not defined.

          If the row pointed to does not exist, the treatment
          is as if this attribute contains a value of zeroDotZero."
      REFERENCE
          "Differentiated Services MIB module"
      DEFVAL { zeroDotZero }
      ::= { diffServConfigEntry 5 }

diffServConfigStorage OBJECT-TYPE
      SYNTAX          StorageType
      MAX-ACCESS      read-create
      STATUS          current
      DESCRIPTION
          "The type of storage used for this row.

          Since an entry in this table serves as a starting
          point for a configuration, it is recommended that
          all entries comprising the configuration started by
          diffServConfigStart follow the storage type of this
          entry.  Otherwise, after agent reboots a configuration
          may differ.  It may very well be that the agent is
          not capable of detecting such changes and therefore,
          the management application should verify the correct
          configuration after a reboot.  Rows with a StorageType
          of 'permanent' do not need to allow write access to
          any of the columnar objects in that row."
      DEFVAL { nonVolatile }
      ::= { diffServConfigEntry 6 }

diffServConfigStatus OBJECT-TYPE
      SYNTAX          RowStatus
      MAX-ACCESS      read-create
      STATUS          current
      DESCRIPTION
          "RowStatus object used for creation and deletion of
          rows in this table.  All writable objects in this row
          may be modified at any time."
      DEFVAL { notInService }
      ::= { diffServConfigEntry 7 }

--
-- MIB Compliance statements.
--
```

```
diffServConfigMIBCompliances
   OBJECT IDENTIFIER ::= { diffServConfigMIBConformance 1 }
diffServConfigMIBGroups
   OBJECT IDENTIFIER ::= { diffServConfigMIBConformance 2 }

diffServConfigMIBFullCompliance MODULE-COMPLIANCE
    STATUS       current
    DESCRIPTION
       "The full compliance for this MIB module.

       For this compliance level the 'diffServMIBFullCompliance'
       must be met, since this MIB module depends on it in order
       to provide the configuration entries.
       "
    MODULE  -- This module
    MANDATORY-GROUPS {  diffServConfigMIBConfigGroup }

    OBJECT diffServConfigStatus
    SYNTAX RowStatus { active(1) }
    WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
    DESCRIPTION
       "Support for createAndWait and notInService is not required."

    ::= { diffServConfigMIBCompliances 1 }

diffServConfigMIBConfigGroup OBJECT-GROUP
    OBJECTS {  diffServConfigDescr,
               diffServConfigOwner,
               diffServConfigLastChange,
               diffServConfigStart,
               diffServConfigStorage,
               diffServConfigStatus
    }
    STATUS current
    DESCRIPTION
       "The per-hop-behavior Group defines the MIB objects that
       describe the configuration template for the per-hop-behavior."
    ::= { diffServConfigMIBGroups 1 }
END
```

8. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These managed objects are:

- The diffServConfigDescr, diffServConfigOwner, and diffServConfigStatus are not security sensitive since these three objects do not affect any direct operational behavior of a diffserv capable device.

- Unauthorized change of the diffServConfigStart could lead to a different configuration, and the 'changed' configuration could lead to different traffic treatment for the diffserv capable device than desired.

- Unauthorized change of the diffServConfigStorage could lead to unknown behavior of the diffserv capable device after a reboot of the SNMP agent. This may be caused by 'not having saved changes of the configuration' or unavailable configurations.

In addition, the managed objects of the DIFFSERV-MIB are also security sensitive, since unauthorized changes may cause configuration changes. For more detail, refer to [RFC3289].

Allowing read access to objects in this MIB module is generally not considered sensitive, as read access only provides information that a template exists. This is due to the fact that the managed objects that actually instantiate the template are in the DIFFSERV-MIB [RFC3289]. However, in environments where the template description (diffServConfigDescr) or owner (diffServConfigOwner) is considered sensitive information, appropriate access control should be exercised for these objects.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), 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 module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, deployment of SNMPv3 with cryptographic security enabled is RECOMMENDED. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to GET or SET (change/create/delete) them.

9. Acknowledgments

The editors gratefully acknowledge the significant contributions to this work made by several members of both the SNMPCONF and DiffServ working groups.

10. References

10.1. Normative References

[RFC2578] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

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

[RFC2580] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3289] Baker, F., Chan, K. and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

10.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC2474] 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.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC3512] MacFaden, M., Partain, D., Saperia, J. and W. Tackabury, "Configuring Networks and Devices with Simple Network Management Protocol (SNMP)", RFC 3512, April 2003.

[PMMIBDR] Waldbusser, S., Saperia, J. and T. Hongal, "Policy-based Management MIB", Work in Progress.

11. Editors' Addresses

Harrie Hazewinkel
I.Net
via Darwin 85
20019 - Settimo Milanese (MI)
Italy

EMail: harrie@inet.it

David Partain
Ericsson AB
P.O. Box 1248
SE-581 12 Linkoping
Sweden

EMail: David.Partain@ericsson.com

12. Full Copyright Statement

Intellectual Property

Acknowledgement

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